



Michel Voss

**Wykorzystanie głębkich sieci neuronowych
na potrzeby statystyki energii odnawialnej**

**Deep neural networks for renewable energy
statistics**

Praca magisterska

Promotor: dr Maciej Beręsewicz

Prace przyjęto dnia:

Podpis promotora

Kierunek: Informatyka i ekonometria

Poznań 2021

Spis treści

Wstęp	1
1 Odnawialne źródła energii w Polsce	5
1.1 Energetyka konwencjonalna i odnawialna	5
1.2 Ogniwa i kolektory słoneczne	10
1.2.1 Ogniwa słoneczne	10
1.2.2 Kolektory słoneczne	12
1.3 Prawne aspekty rozwoju energetyki odnawialnej w Unii Europejskiej i w Polsce	14
1.3.1 Regulacje w Unii Europejskiej	14
1.3.2 Regulacje w Polsce	15
1.3.3 Regulacje prawne dotyczące przyłączeń elektrowni	16
1.4 Źródła danych o odnawialnych źródłach energii w Polsce	17
1.4.1 Statystyka publiczna	17
1.4.2 Rejestry administracyjne	19
1.4.3 Big data	25
1.5 Podsumowanie	27
2 Detekcja obiektów z wykorzystaniem głębokich sieci neuronowych	28
2.1 Sieci neuronowe – ujęcie historyczne	28
2.2 Teoretyczne podstawy głębokich sieci neuronowych	31
2.2.1 Budowa sztucznego neuronu	32
2.2.2 Wielowarstwowe sieci neuronowe	35
2.2.3 Wsteczna propagacja	37
2.3 Darknet i algorytm YOLO	38
2.3.1 YOLO v4	46

2.3.2	Architektura YOLO v4	47
2.3.3	Zestaw technik dla szkieletu	47
2.4	Podsumowanie	48
3	Przygotowanie danych i środowiska na potrzeby detekcji paneli i kolektorów słonecznych	49
3.1	Schemat przetwarzania danych	49
3.2	Dane źródłowe	50
3.2.1	Lokalizacja instalacji fotowoltaicznych	50
3.2.2	Zdjęcia lotnicze	53
3.3	Oznaczanie paneli i kolektorów na zdjęciach	57
3.4	Przygotowanie środowiska pracy	60
3.4.1	Infrastruktura	60
3.4.2	Konfiguracja Google Colab	62
3.4.3	Trenowanie modelu w Google Colab	63
3.5	Podsumowanie	68
4	Zastosowanie głębokich sieci neuronowych do detekcji paneli i kolektorów słonecznych	69
4.1	Procedura badawcza	69
4.1.1	Wykorzystana sieć neuronowa Yolo v4	69
4.1.2	Eksperymenty	69
4.2	Wyniki eksperymentów	71
4.2.1	Eksperiment 1	71
4.2.2	Eksperiment 2	73
4.2.3	Eksperiment 3	76
4.3	Wnioski płynące z eksperymentów	79
Zakończenie		83
Spis tabel		87
Spis rysunków		89
Spis programów		90

Wstęp

Odnawialne źródła energii towarzyszą ludziom od początku dziejów. Energię wiatru używano do napędzania łodzi czy spalanie biomasy jako źródło ciepła. Nasilające się zmiany klimatyczne wymusiły na rządach przyspieszenie dążenia do neutralności klimatycznej. Również Polska zdeklarowała się znacząco zwiększyć udział energii ze źródeł odnawialnych, dając do neutralności klimatycznej w 2050 roku.

Osiągnięcie powyższych celów będzie jedynie możliwe w momencie dokładnego monitorowania wyników oraz posiadania najaktualniejszych informacji dotyczących odnawialnych źródeł energii. Wykorzystać można dane pochodzące ze źródeł statystycznych (m.in. badań Głównego Urzędu Statystycznego) oraz niestatystycznych (m.in. rejesty administracyjne Urzędu Regulacji Energetyki czy big data). W niniejszej pracy skupiono się na ostatnim źródle w postaci zdjęć lotniczych (tzw. ortofotomapy) pozyskanych z Głównego Urzędu Geodezji i Kartografii.

Problem badawczy

Statystyka publiczna, w świetle rosnących problemów z realizacją badań częściowych oraz naciskiem na efektywność kosztową, coraz częściej sięga po nowoczesne, niestatystyczne źródła danych aby dostarczać rzetelnych informacji o procesach masowych. W kontekście statystyki energii odnawialnej naturalnym źródłem informacji mogą być zdjęcia lotnicze, które charakteryzują się bardzo dużą dokładnością.

Jednakże, ze względu na wolumen danych liczony w terabajtach, a także fakt, że poznanie dokładnej lokalizacji czy charakterystyk paneli i kolektorów słonecznych może być utrudnione, zachodzi potrzeba zastosowania nowoczesnych metod detekcji obiektów opartych na głębokich sieciach neuronowych.

W związku z tym **problemem badawczym** podnoszonym w niniejszej pracy magisterskiej jest **ocena możliwości wykorzystania algorytmów detekcji obiektów na zdjęciach lotniczych na potrzeby statystyki energii odnawialnej w Polsce**.

Cele i hipotezy badawcze

Głównym celem badania jest **opracowanie algorytmu detekcji paneli i kolektorów słonecznych na potrzeby statystyki energii odnawialnej w Polsce**. Algorytm ma na celu rozpoznanie ze zdjęć lotniczych w rozdzielcości 5 cm instalacji słonecznych na podstawie odpowiednio przygotowanego zbioru uczącego. Badaniu poddane zostaną lata 2018 i 2020 oraz miasto Poznań i wieś Skórzewo (powiat poznański).

Aby zrealizować ten cel zdefiniowano następujące cele szczegółowe:

- C1) identyfikacja paneli i kolektorów słonecznych na podstawie danych pozyskanych od operatora systemów dystrybucyjnych,
- C2) przygotowanie zbioru uczącego i testowego zawierającego fragmenty zdjęć lotniczych z oznaczonymi lokalizacjami,
- C3) nauczenie głębokiej sieci neuronowej opartej na algorytmie You Only Look Once w wersji 4 (YOLO v4) do detekcji instalacji,
- C4) weryfikacji skuteczności otrzymanej sieci neuronowej w czasie oraz przestrzeni w trzech eksperymentach.

W odniesieniu do tych celów postawiono trzy hipotezy badawcze:

1. algorytm YOLO v4 z wysoką skutecznością klasyfikuje i oznacza panele oraz kolektory słoneczne,
2. sieć zbudowana na danych z jednego okresu będzie się charakteryzowała słabszą skutecznością dla innego okresu,
3. sieć zbudowana na danych z jednej lokalizacji przestrzennej (np. miasta) będzie się charakteryzowała słabszą skutecznością dla innej lokalizacji (np. wsi).

Novum pracy

Novum niniejszej pracy magisterskiej polega na wykorzystaniu zdjęć lotniczych na potrzeby detekcji paneli i kolektorów słonecznych w Polsce oraz przeprowadzeniu eksperymentów celem

weryfikacji skuteczności głębokich sieci neuronowych w wymiarze przestrzennym oraz czasowym.

Praca magisterska jest rozszerzeniem wcześniejszego badania autora, które zostało opisane w pracy licencjackiej pt *Detekcja kolektorów i paneli słonecznych na podstawie zdjęć lotniczych miasta Poznania z wykorzystaniem głębokich sieci neuronowych*. Praca dyplomowa była przygotowane pod kierownictwem dr. Macieja Beręsewicza i jest dostępna na stronie <https://github.com/DepartmentOfStatisticsPUE/prace-diplomowe>.

Źródła danych

W pracy wykorzystano dwa źródła danych. Pierwszym były dane pochodzące od operatora systemów dystrybucyjnych zawierające przybliżoną lokalizację instalacji fotowoltaicznych w Wielkopolsce zgodnie ze stanem na koniec 2020 roku. Dane te zawierały przybliżone współrzędne geograficzne oraz unikalny identyfikator. W związku z ograniczeniami w dostępie do infrastruktury (w pracy korzystano wyłącznie z Google Colab) zdecydowano skupić się na dwóch lokalizacjach: mieście Poznań oraz wsi Skórzewo.

Drugim źródłem danych były zdjęcia lotnicze charakteryzujące się wysoką rozdzielcością (1 piksel = 5 cm) pozyskane z Głównego Urzędu Geodezji i Kartografii (GUGiK). W tym celu wykorzystano API przygotowane przez GUGiK, które umożliwiało automatyczne pobieranie ortofotomap dla wskazanych punktów w przestrzeni. Dostęp do API zapewnił pakiet `rgugik`, który zawiera szereg funkcji i usprawnia pracę z rejestrami udostępnionymi przez GUGiK.

Treść pracy

Praca składa się z czterech rozdziałów, wstępu i zakończenia. Praca zawiera również spis tabel, rysunków, programów oraz skrótów. W pierwszym rozdziale przedstawiono pojęcia związane z tematyką paneli fotowoltaicznych, a także informacje dotyczące produkcji energii odnawialnej w Polsce. Omówiono podstawowe akty prawne dotyczące tego rynku oraz źródła danych, które można wykorzystać do opisu energii odnawialnej w Polsce.

W drugim rozdziale opisano zagadnienia związane z głębokimi sieciami neuronowymi (ang. *deep learning*), w szczególności pojęcia sieci neuronowej, uczenia maszynowego, uczenia gę-

bokiego czy wstecznej propagacji błędu. Omówiono również wykorzystany w pracy algorytm YOLO w wersji 4 oraz miary jakości przydatne do oceny danej sieci.

Trzeci rozdział stanowi opis działań związanych z przygotowaniem danych oraz infrastruktury do uczenia głębokich sieci neuronowych. Na podstawie danych dystrybutora energii udało się pobrać lokalizacje potencjalnych instalacji fotowoltaicznych, a następnie dokonać przekształceń mających na celu zmniejszenie wolumenu zdjęć lotniczych. Tak przygotowane obrazy zostały następnie ręcznie oznaczone aby przygotować zbiór uczący oraz testowy. Rozdział kończy się technicznym opisem jak przygotować Google Colab do detekcji obiektów z wykorzystaniem algorytmu YOLO v4 i biblioteki darknet.

W rozdziale czwartym omówiono eksperymenty badawcze poświęcone weryfikacji skuteczności algorytmu YOLO v4 do detekcji obiektów w czasie i przestrzeni. Zaprojektowano trzy eksperymenty, w których sieć neuronowa była uczona tylko na danych z Poznania, wybranym roku (2018 lub 2020) lub obydwu latach (2018 oraz 2020), a weryfikowana na danych, których wcześniej nie widziała (wieś Skórzewo, Poznań ale inne lata). W ocenie skuteczności wykorzystano macierz błędów klasyfikacji oraz porównanie ze zdjęciami zawierającymi dane instalacje paneli i kolektorów słonecznych.

Rozdział 1

Odnawialne źródła energii w Polsce

1.1 Energetyka konwencjonalna i odnawialna

Do podstawowych zadań państwa należy zagwarantowanie obywatelom bezpieczeństwa we wszystkich dziedzinach życia. Z biegiem czasu zmieniły się obszary, w których państwo powinno zadbać o obywatele. W obecnym czasach zapewnienie bezpieczeństwa sektora elektroenergetycznego jest kluczowe nie tylko w działaniu przemysłu, ale również dla gospodarstw domowych. Sama zaś energia elektryczna stanowi kluczowy element w rozwoju gospodarczym państw.

Jednak wraz ze wzrostem podaży energii elektrycznej rośnie zanieczyszczenie środowiska będące wynikiem emisji gazów cieplarnianych, substancji zanieczyszczających środowisko czy wzrostu odpadów. Skutki nie ponosi tylko środowisko, ale pogorszenie jakości powietrza wpływa również na zdrowie ludzi jak podaje Krakowski Alarm Smogowy ([2020](#)). Z raportu wynika, że rocznie w Polsce z powodu zanieczyszczeń pyłem PM_{2,5} umiera ponad 44 tys. osób (dane dla 2012 roku).

Zwiększający zaś popyt na energię utrwała trend rosnącej emisji CO₂. Aktualnie większość państw uzyskuje energię elektryczną z paliw kopalnianych (węgiel, ropa naftowa czy gaz). W procesie spalania powyższych paliw, wydzielane jest ciepło, które następnie można przekształcić na energię. W przypadku spalania węgla, paliwo wchodzi w reakcję z tlenem emitując przy tym CO₂. Inne szkodliwe substancje emitowane w trakcie spalania węgla to tlenki azotu, dwutlenek siarki czy cząstki stałe jednak z biegiem lat dzięki udoskonalonym systemom oczyszczania powietrza udało się zmniejszyć emisję tych substancji.

Jak wynika z raportu Eurostat (2020), aktualnie około 34,2% produkcji energii pierwotnej w Europie pochodziło ze źródeł odnawialnych. Pozostałe 30,2% pochodziły z elektrowni jądrowej, stałe paliwa kopalne odpowiadały za 18,3%, gaz ziemny za 9,3%, ropa naftowa i produkty naftowe (3,9%) oraz odpady nieodnawialne wygenerowały 2,1% całej energii. Sektor energetyczny jest coraz częściej traktowany w polityce jako obszar priorytetowy. W ramach przyjętej strategii "Europa 2020" przez Unię Europejską w 2010 roku, jeden z wyznaczonych celów odnosił się do:

1. osiągnięcia 20% udziału energii pochodzącej ze źródeł odnawialnych w ogólnej ilości zużywanej energii w Europie,
2. zwiększenia efektywności wykorzystania energii o 20%.

Coraz większa troska o klimat, próba zahamowania globalnego ocieplenia oraz liczne strategie działania na przyszłość sprawiły, że w odnawialnych źródłach energii widzi się przyszłość energetyki.

W corocznym raporcie GUS (2020a) znajdziemy następującą definicję energii ze źródeł odnawialnych: *energię pochodzącą z naturalnych, powtarzających się procesów przyrodniczych, pozyskiwaną z odnawialnych niekopalnych źródeł energii*.

Odnawialne źródła energii (OZE) stanowią alternatywę do tradycyjnych metod pozyskiwania energii. Jak sama nazwa wskazuje, zasoby uzupełniają się ponownie w ramach naturalnych procesów co sprawia, że są niewyczerpalne.

Według corocznego raportu REN21 (2021) z 2021 roku, energia odnawialna odnotowała kolejny rekordowy wzrost w 2019 roku, ponieważ moc zainstalowana wzrosła o ponad 200 gigawatów (GW) – to największy wzrost w historii. W skali globalnej 32 kraje posiadały w 2019 r. co najmniej 10 GW mocy w energetyce odnawialnej, w porównaniu z zaledwie 19 krajami dekadę wcześniej. Łącznie zainstalowana moc energii odnawialnej wystarczyła do zapewnienia szacunkowo 27,3% (energia wodna 15,9%, wiatrowa 5,9%, słoneczna 2,8%, biomasa 2,2% oraz geotermalna 0,4%) globalnej produkcji energii elektrycznej do końca 2019 r.

Pomimo tych postępów, odnawialna energia elektryczna nadal boryka się z problemami związanymi z osiągnięciem większego udziału w globalnej produkcji energii elektrycznej, częściowo ze względu na utrzymujące się inwestycje w moce wytwórcze oparte na paliwach kopalnych (i jądrowych).

Według raportu International Energy Agency) (2020) ludzkość w 2018 roku zużyła 9938 MToe¹ (Megaton oleju ekwiwalentnego). W przeliczeniu na TWrok² wartość globalnej konsumpcji była równa 13,2. Według M. Perez i R. Perez (2015) zasób wszystkich odnawialnych źródeł energii wielokrotnie przekracza aktualne zużycie energii elektrycznej. Według autorów artykułu największy potencjał posiada energia ze słońca, która kilkuset krotnie przekracza aktualne globalne zużycie energii elektrycznej.

Tablica 1.1. Potencjał energetyczny z wybranych źródeł w skali roku w TWrok według M. Perez i R. 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: M. Perez i R. Perez (2015, s. 2). Objaśnienia: OTEC – Ocean Thermal Energy Conversion.

Efektywność energetyczna, jako idea oszczędzania energii i zwiększania efektywności energetycznej, zyskuje coraz większe znaczenie jako element koncepcji zrównoważonego rozwoju, bezpieczeństwa dostaw energii i zapobiegania globalnym zmianom klimatu. Według raportu Komisji Europejskiej z 2018 do roku 2050 zajdzie nie tylko zmiana w sposobach pozyskiwania energii, ale również dojdzie do zmniejszenia zużycia. Aktualnie państwa członkowskie Unii Europejskiej zmagają się z wyzwaniem sektora energetycznego. Zobowiązania wobec środowiska, dywersyfikacja źródeł pozyskiwania energii, uzależnienie od zewnętrznych dostaw ropy czy gazu ziemnego to kilka głównych problemów z jakimi muszą się zmierzyć się Europejczycy. W 2007 roku przywódcy krajów UE określili cele klimatyczne, które ostatecznie zostały przyjęte w 2009 roku. Przyjęty pakiet stanowił zbiór przepisów, które miały zapewnić osiągnięcie przez UE celów w zakresie energii i klimatu do 2020 roku. W ramach *Pakietu 2020* zdefiniowano następujące cele:

1. ograniczenie o 20% emisji gazów cieplarnianych (w stosunku do poziomu z 1990 r.),
2. 20% udział energii ze źródeł odnawialnych w całkowitym zużyciu energii w UE,
3. zwiększenie o 20% efektywności energetycznej.

W celu osiągnięcia powyższych założeń Unia Europejska działała w różnych obszarach m.in. wprowadzono system handlu uprawnieniami do emisji (ETS z ang. *Emissions Trading System*).

¹Tona oleju ekwiwalentnego (toe) – jest to energetyczny równoważnik jednej metrycznej tony ropy naftowej o wartości opałowej równej 10000 kcal/kg.

²TWrok to 8766 terawat godzin energii.

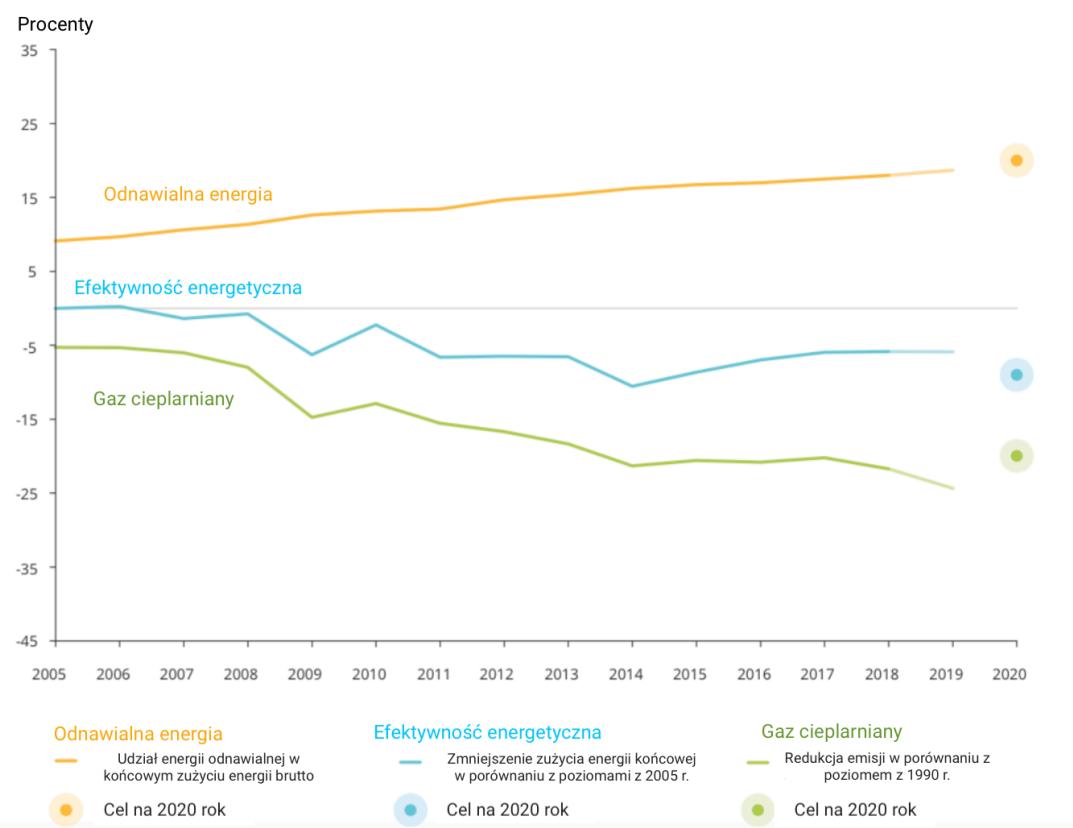
System ten został wprowadzany w 31 krajach (28 krajów UE oraz w Islandii, Liechtensteinie i Norwegii) i polegał na nałożeniu limitu łącznych emisji gazów na elektrownie, zakłady przemysłowe czy linie lotnicze. Narzucony limit z biegiem czasu maleje przez co łączne emisje w Unii Europejskiej spadają. W ramach działania ETS firmy kupują lub otrzymują uprawnienia do emisji, którymi mogą handlować. Rok do roku przedsiębiorstwo musi umorzyć liczbę uprawnień pokrywającą całkowitą emisję gazów. Dzięki wdrożeniu systemu ETS w branżach objętych programem tak jak elektrownie zanotowano spadek emisji o 9,1% w latach 2018-2019. Dla porównania, w sektorach nie objętych programem ETS nie odnotowano spadku emisji gazów między 2018, a 2019 rokiem.

W ramach drugiego punktu Pakietu 2020 państwa członkowie zdecydowały zwiększenie udziału energii ze źródeł odnawialnych. Cele zostały wyznaczone osobno dla państw w zależności od bogactwa i początkowego udziału czystej energii: od 10% na Malcie do 49% w Szwecji, zaś dla Polski wyznaczono cel na poziomie 15%.

W ramach raportu Europejskiej Agencji Środowiska z 2020 podsumowano rok 2019 w ramach wyznaczonych celów Pakietu 2020. Dzięki działaniom państw członkowskich oraz Unii Europejskiej udało się zmniejszyć emisję gazów cieplarnianych w stosunku do poziomu z 1990 roku o 26%. Przyjęty cel został osiągnięty już w 2014 roku kiedy po raz pierwszy emisje były mniejsze o ponad 20% względem roku bazowego. W obszarze energii odnawialnej wstępne dane EAŚ sugerują, że wszystkie państwa członkowskie UE osiągnęły całkowity udział energii zużywanej ze źródeł odnawialnych w wysokości 19,4% w 2019 r. Unia Europejska jest zatem na dobrej drodze do osiągnięcia celu na 2020 r. w postaci co najmniej 20% udziału energii z czystych źródeł energii. O ile udział energii elektrycznej, cieplnej i chłodniczej pochodzącej ze źródeł odnawialnych przyczynił się do osiągnięcia ogólnego celu UE, o tyle nadal odnotowywano gorsze wyniki w zakresie wzrostu udziału energii ze źródeł odnawialnych wykorzystywanej w sektorze transportu. Perspektywa osiągnięcia do 2020 r. celu 10% zapotrzebowania na energię w transporcie pochodzącej ze źródeł odnawialnych pozostaje niewielka. W 2019 roku trzynaście państw członkowskich przekroczyło krajowe cele ustalone na 2020 rok.

Dalej z raportu wynika, że jeśli chodzi o efektywność energetyczną, perspektywy dla celów na rok 2020 nie są pozytywne i często charakteryzują się tendencjami odbiegającymi od celu. W 2019 r. przybliżone wartości zużycia energii dla UE-28 wskazują na stabilizację końcowego zużycia energii i niewielki spadek zużycia energii pierwotnej o 1,4%. Ten ostatni jest następstwem marginalnego zmniejszenia w 2018 r. o 0,6% po kilku latach wzrostu zużycia energii

pierwotnej. Tendencje te są podobne do tych występujących w UE-27 i chociaż stabilizacja zużycia energii jest dobrym osiągnięciem, nie wykazano redukcji niezbędnych do osiągnięcia 20% spadku do 2020 r. Wstępne dane EAŚ dotyczące końcowego zużycia energii w 2019 r. pokazują, że tylko dziewięć państw członkowskich (Finlandia, Grecja, Włochy, Łotwa, Niderlandy, Portugalia, Rumunia, Słowenia i Hiszpania) było na dobrej drodze do osiągnięcia swoich odpowiednich celów w zakresie efektywności energetycznej na 2020 r. Oznacza to stały spadek w porównaniu z poprzednimi latami. Aktualne postępy w realizacji projektu Pakiet 2020 znajdują się na Rysunku 1.1.



Rysunek 1.1. Trendy historyczne i postępy w realizacji celów na lata 2020 w UE-28

Źródło: Opracowanie własne na podstawie Europejska Agencja Środowiska (2020).

Z końcem 2019 roku komisja europejska przyjęła koncept *Europejskiego Zielonego Ładu*, w ramach którego celem jest uczynienie wspólnoty Europejskiej pierwszym kontynentem neutralnym klimatycznie (Parlament Europejski 2019). Realizacja celu będzie tylko możliwa poprzez spełnienie kroków pośrednich, które zostały zdefiniowane jako plan klimatyczny na 2030 rok. W zawartym planie komisja zaproponowała redukcję do 2030 roku emisji gazów cieplarnianych

o co najmniej 55% względem 1990 roku. Innym celem będzie zwiększenie całkowitego nakładu energii pochodzącej z odnawialnych źródeł energii do poziomu co najmniej 32% całkowitego zużycia. Kluczowe będzie również zwiększenie efektywności energetycznej o co najmniej 32,5%. Realizacja wyżej wymienionych założeń umożliwi osiągnięcie neutralnej gospodarki europejskiej do roku 2050.

1.2 Ogniwa i kolektory słoneczne

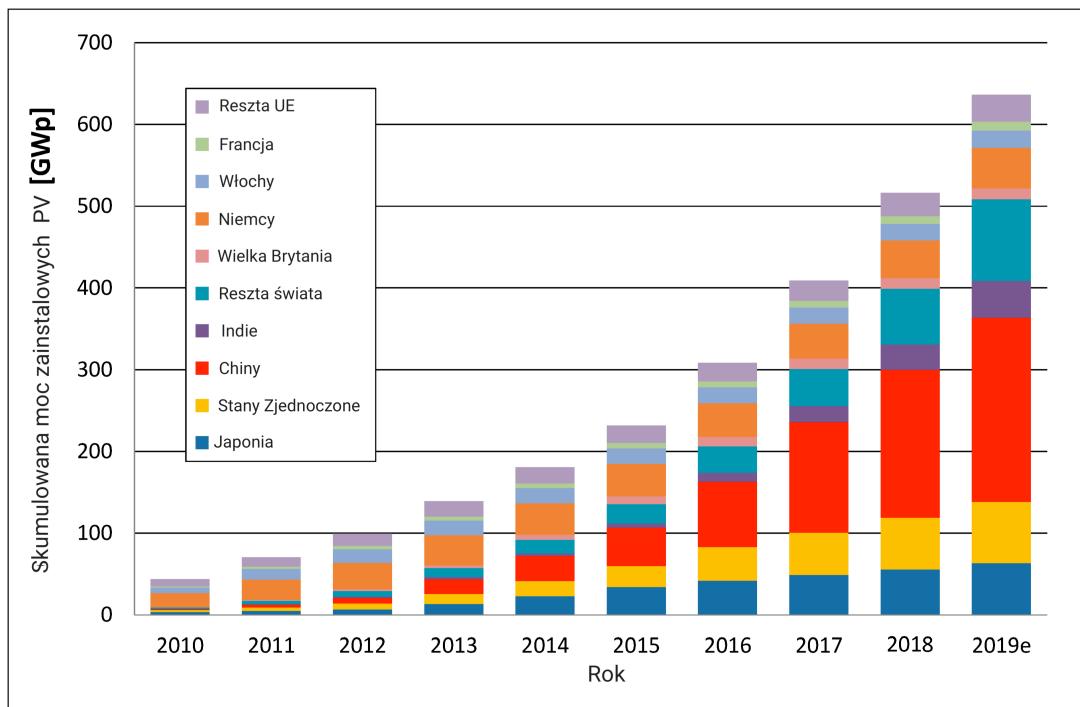
1.2.1 Ogniwa słoneczne

Słowo fotowoltaika pochodzi z połączenia dwóch słów pochodzących z fizyki: słowa "photo" oznaczające światło oraz "wolt" oznaczające jednostkę, którą wyrażamy napięcie prądu. Efekt fotowoltaiczny oznacza zaś zjawisko generujące prąd elektryczny pod wpływem padania promieni słonecznych. Ogniwa słoneczne bądź fotowoltaiczne stanowią podstawowy element budowy paneli fotowoltaicznych. Po dodaniu baterii całość można nazwać instalacją fotowoltaiczną.

Każde ogniwo słoneczne składa się z dwóch warstw: negatywnej i pozytywnej. W trakcie padania promieni słonecznych wytwarzane jest napięcie. Dzięki skumulowaniu wielu ogniw w panele możliwe jest generowanie większej mocy. Ostatecznie konieczne jest użycie falownika, który zamienia prąd stały na prąd zmienny, który używane jest w gospodarstwach domowych.

Początki fotowoltaiki sięgają roku 1839 kiedy Aleksander Edmund Becquerel, francuski fizyk odkrył efekt fotoelektryczny (Marszałek 2021). Kolejnym przełomowym wydarzeniem był eksperyment W.G Adams i R.E Day, którzy zaobserwowali zjawisko fotowoltaiczne w selenie. Pod wpływem światła wytworzono prąd, pozytywny wynik eksperymentu zaowocował powstaniem uproszczonego modelu ogniwa fotowoltaicznego. Pod koniec lat 50 XX wieku naukowcom udało się w znaczący sposób zwiększyć sprawność ogniw. Przykładowo, firma Hoffman Electronics w 1958 roku zbudowała ogniwo słoneczne o rekordowej sprawności 9%. W tym samym roku zastosowano panele słoneczne w satelicie Vanguard 1. Zastosowanie ogniw fotowoltaicznych stało się standardowym zastosowaniem w konstrukcjach kosmicznych. W latach 70 XX wieku dzięki licznym usprawnieniom udało się obniżyć koszty związane z produkcją paneli fotowoltaicznych o 80%. Przełom w dostępności przyniosły lata 90 XX wieku kiedy to w Japonii oraz w Niemczech wprowadzono pierwsze dotacje dla gospodarstw domowych, które zwiększyły

zainteresowanie mieszkańców, zaś w 2000 roku rozpoczęto masową produkcję ogniw fotowoltaicznych (Castellano 2012). Jak podaje Jäger-Waldau (2019), od 2010 roku na całym świecie następuje wykładniczy wzrost mocy pochodzącej z ogniw fotowoltaicznych.

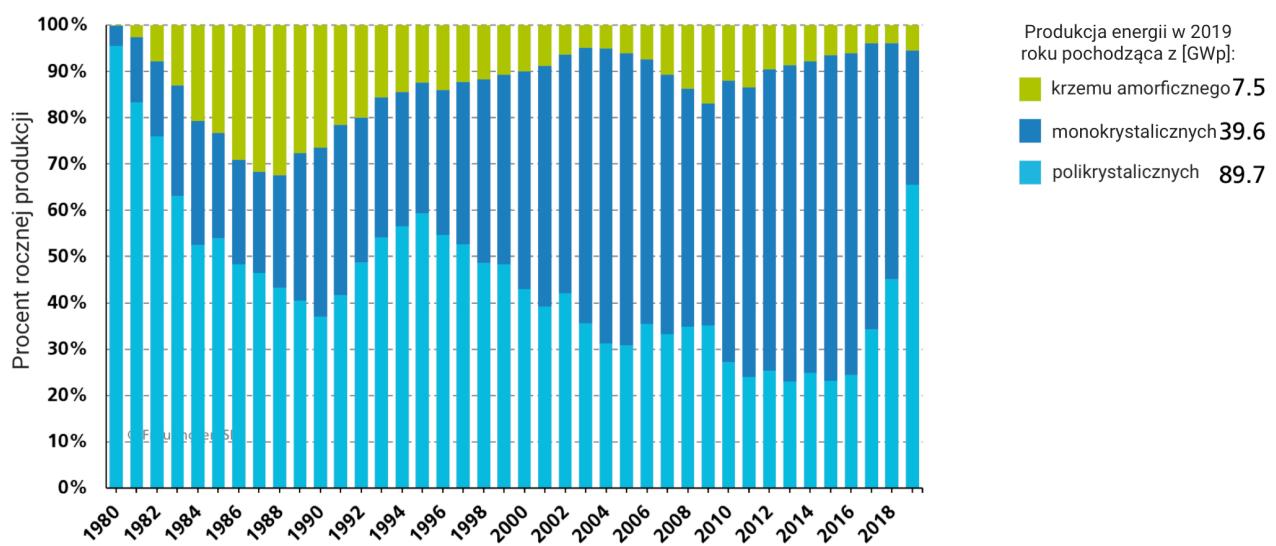


Rysunek 1.2. Moce PV zainstalowane w różnych krajach w latach 2010-2019

Źródło: Opracowanie własne na postawie Jäger-Waldau (2019).

Aktualnie można wyróżnić trzy najbardziej popularne rodzaje ogniw fotowoltaicznych: polikrystaliczne, monokrystaliczne oraz panele z krzemem amorficznego. Ogniwa różnią się między sobą właściwościami, ceną oraz wydajnością. Ogniwa polikrystaliczne są to moduły, które łączą w sobie wiele kryształów krzemu. Charakteryzują się niebieskim kolorem, a układ kwadratowych płytka wykorzystuje całą powierzchnię ogniska fotowoltaicznego. Sprawność takich ogniw wynosi od 14% do 16%. Ogniwa te są idealnym rozwiązaniem dla miejsc z mniejszym nasłonecznieniem, dlatego są najpopularniejszym typem ogniska w Polsce. Innymi zaletami są cena oraz długa żywotność, sięgająca nawet 25 lat. Według raportu Fraunhofer Institute (2020) w 2019 roku ogniska polikrystaliczne wyprodukowały około 39,6 GWp. Innym rodzajem ognisk są ogniska monokrystaliczne, które mają największą wydajność (od 17% do 22%). Jak nazwa wskazuje, produkowane są z pojedynczych kryształów krzemu, który sprawia że ognisko ma czarny kolor, a sam kształt płytka jest wielokątem. Rodzaj tego ogniska najlepiej sprawdza

się przy silnym nasłonecznieniu oraz ograniczonej powierzchni. Do głównych zalet należy również wysoka trwałość na warunki atmosferyczne, główną wadą jest zaś cena, która w przypadku tego typu ogniw jest najwyższa. Jak podaje Fraunhofer Institute (2020), w 2019 roku na świecie zostały przyłączone do sieci instalacje z ogniw monokrystalicznych o łącznej mocy 89,7 GWp. Ostatnim rodzajem ogniw są ogniska z krzemem amorficznego. Jest to najrzadziej spotykany rodzaj ogniska w Polsce, do głównych cech należy: niska wydajność, elastyczność, niska cena, niski stopień wrażliwości na wysokie temperatury. Ze względu na prosty montaż ogniska z krzemem amorficznego można montować na elewacji budynków. Z raportu tego wynika, że ze względu na niską wydajność, ogniska z krzemem amorficznego są najrzadziej stosowanym rodzajem ogniska, w 2019 na świecie ogniska te wygenerowały około 7,5 GWp.



Rysunek 1.3. Udział poszczególnych rodzajów ogniw w całkowitej produkcji energii ze słońca

Źródło: Opracowanie własne na podstawie Jäger-Walda (2019).

1.2.2 Kolektory słoneczne

Z pomocą kolektorów słonecznych można przekształcać energię słońca na ciepło. Docierająca energia słoneczna zamieniana jest na energię cieplną, którego nośnikiem może być powietrze lub ciecz. Jak podaje Tytko (2020) kolektory można podzielić ze względu na konstrukcję, dwie główne grupy kolektorów to: 1) skupiące; 2) płaskie;

W przypadku kolektorów słonecznych płaskich, następuje konwersja termiczna, przetwarzając energię promieniowania słonecznego na ciepło. Ciepło w przypadku tego rodzaju kolektorów transportowane jest poprzez płyn niezamarzający, wodę lub powietrze. Kolektor składa się z płyty połączonej rurkami miedzianymi poprzez które przepływa czynnik roboczy.

W kolektorach słonecznych skupiających stosowane są różne układy soczewek lub luster w celu zwiększenia promieniowania słonecznego. Rodzaj tych kolektorów musi być ustawiony prostopadle do kierunku padania promieni słonecznych, dlatego konieczne jest zastosowanie urządzeń obracających kolektor względem ruchu słońca.



Rysunek 1.4. Kolektor słoneczny skupiający

Źródło: Opracowanie na podstawie podstawie podstrony *Oferta* ze strony firmy Elfran
<http://www.elfran.com.pl>.

Według raportu EurObserv'ER (2020) rynek kolektorów słonecznych w Unii Europejskiej do produkcji ciepła, ogrzewania i ciepłej wody użytkowej korzystnie rozwijał się w 2019 roku. Wstępne szacunki określają całkowitą zainstalowaną powierzchnię kolektorów na nieco poniżej 2,3 mln m², co stanowi niewielki wzrost (1,5%) w stosunku do poziomu z 2018 roku.

Sytuacja w poszczególnych krajach jest jednak zróżnicowana, a sektor wciąż musi wymyślać nowe rozwiązania, aby sprostać ogromnemu wyzwaniu, jakim jest neutralność klimatyczna. Chociaż europejski rynek słonecznej energii cieplnej nie skurczył się od dwóch lat, główne trendy rynkowe przeważają z pozytywną dynamiką na rynkach greckim i cypryjskim (odpowiednio 10% i 23,7% wzrostu rok do roku). Dynamika rynku w Holandii również była optymistyczna (wzrost o 48% w 2019 r.), tj. 53 443 m² zainstalowanych, zgodnie z danymi Urzędu Statystycznego.

nego w Niderlandach. Według raportu rynek niemiecki ponownie odnotował spadek (o 10,9% między 2018 a 2019 rokiem) do 511 000 m². Rynek austriacki również mocno zmalał, schodząc poniżej progu 100 000 m² (90 810 m² w 2019 r. wg Statistics Austria).

Również w Polsce osiągnęła dynamika. Sektor kolektorów słonecznych skurczył się o 15% w stosunku do poziomu z 2018 roku (z 263 000 m² zainstalowanych w 2019 roku). Przyczyny spadku aktywności na tych rynkach są takie same jak w latach poprzednich. Indywidualne kolektory słoneczne lub małe systemy zbiorcze (hotele, budynek wielorodzinny itp.) często konkurują o dostępną powierzchnię dachową z bardzo prężnym rynkiem fotowoltaiki. Słoneczne systemy grzewcze wciąż borykają się ze słabą komunikacją z opinią publiczną i decydentami politycznymi, a także z brakiem rekomendacji ze strony instalatorów, którzy pod presją czasu skłaniają się ku rozwiązaniom szybszym w montażu, takim jak termodynamiczne podgrzewacze wody.

1.3 Prawne aspekty rozwoju energetyki odnawialnej w Unii Europejskiej i w Polsce

1.3.1 Regulacje w Unii Europejskiej

Zgodnie z *Traktat o Funkcjonowaniu Unii Europejskiej (2019)* z jednym z celów organizacji jest wspieranie odnawialnych źródeł energii. Do pierwszych dokumentów dotyczących energetyki należy Rezolucja Rady z 1986 roku dotycząca zbieżności państw członkowskich oraz nowych celów Wspólnoty w zakresie energetyki. W ramach Deklaracji Madryckiej z 1994 roku nałożono na państwa członkowskie warunek zastąpienia konwencjonalnych sposobów pozyskiwania energii na odnawialne źródła na poziomie 15%. Wcześniej założone cele zmusiły państwa członkowskie do opracowania, wdrożenia i ostatecznie wydania 11 listopada 1997 roku tzw. Białej Księgi Unii Europejskiej Energia dla przyszłości – odnawialne źródła energii. W ramach dokumentu omówiono korzyści płynące z wprowadzania energii odnawialnych oraz plan umożliwiający osiągnięcie udziału energii ze źródeł odnawialnych na poziomie minimum 12%. Rozwój technologii i wzrost świadomości społecznej spowodował dalsze zastrzeżenie wymogów UE w zakresie energii ze źródeł odnawialnych. Poprzez dyrektywę 2009/28/WE z dnia 23 kwietnia 2009 roku państwa członkowskie zobowiązali się do osiągnięcia udziału energii ze źródeł odnawialnych na poziomie 20% do 2020 roku oraz udziału biopaliw w odniesieniu

do paliw zużywanych w transporcie na poziomie 10%. Dyrektywa wskazywała, że weryfikacja postępów w realizacji celów krajowych odbywa się na podstawie dwuletnich sprawozdań z postępów w zakresie OZE. Kolejny dokument taki jak *Dyrektywa Parlamentu Europejskiego i Rady (UE) 2018/2001 (2018)* w sprawie promowania stosowania energii ze źródeł odnawialnych wprowadziły zaostrzone ramy dotyczące udziału energii ze źródeł odnawialnych. Zgodnie z dyrektywą, celem dla krajów członkowskich jest osiągnięcie w 2030 roku co najmniej 32% udziału energii odnawialnej w końcowym zużyciu energii brutto w UE. Ponadto ramy polityki w zakresie klimatu i energii obejmują ograniczenie emisji gazów cieplarnianych o co najmniej 40% w stosunku do poziomu z 1990 r. oraz zwiększenie efektywności energetycznej o co najmniej 32,5%. W kwietniu 2021 roku Rada i Parlament Europejski osiągnęły wstępne porozumienie co do prawa klimatycznego na kolejne lata. Celem będzie ograniczyć emisje netto o co najmniej 55% oraz osiągnięcie neutralności klimatycznej do roku 2050. Kolejnym krokiem będzie przedstawienie odpowiednich zmian w prawie w zakresie energii i klimatu, tak aby było ono spójne z uzgodnionym celem na rok 2030.

1.3.2 Regulacje w Polsce

Głównym aktem prawnym definiującym prawo energetyczne w Polsce jest Prawo Energetyczne (1997) (ostatnie zmiany z 07 lipca 2021 roku). W ustawie tej zdefiniowana odnawialne źródła energii jako: "źródła wykorzystujące w procesie przetwarzania energię wiatru, promieniowania słonecznego, geotermalną, fal, prądów i pływów morskich, spadku rzek oraz energię pozyskiwaną z biomasy, biogazu wysypiskowego, a także z biogazu powstałego w procesach odprowadzania lub oczyszczania ścieków albo rozkładu składowanych szczątków roślinnych i zwierzęcych". Zatem długotrwałe użycie nie wiąże się z zużyciem zasobów. Dokumentem konkretnie odnoszącym się do źródeł odnawialnych jest ustanowiona Ustawa o odnawialnych źródłach energii (2015) . Ustawa porusza tematy związane z: wytwarzaniem energii czy mechanizmami wspierającymi wytwarzanie energii. W ramach dyrektywy 2009/28/WE zostały przyjęte krajowe cele ogólne dla Polski, które wymagały osiągnięcia do 2020 roku poziomu:

1. 15% udziału energii ze źródeł odnawialnych w końcowym zużyciu energii brutto,
2. 10% udziału energii ze źródeł odnawialnych w transporcie.

Możliwości osiągnięcia powyższego planu zostały przedstawione w dokumencie (Minister Gospodarki 2010) opracowanego przez Ministra Gospodarki. Wedle założeń krajowe cele na

2020 rok zostaną osiągnięte, udział energii ze źródeł odnawialnych w końcowym zużyciu energii brutto ma wynieść 15,5% zaś udział energii ze źródeł odnawialnych w transporcie ma być na poziomie 10,14%. Z aktualnego raportu Głównego Urzędu Statystycznego (GUS 2020a) wynika, że wskaźnik udziału energii z odnawialnych źródeł energii w końcowym zużyciu energii brutto wyniósł w 2019 roku 12,16% (przy zakładanym poziomie 14,58% z Krajowego planu działania) czyli wzrost o 0,68 p.p. względem poprzedniego roku. Udział energii ze źródeł odnawialnych w końcowym zużyciu energii w transporcie na koniec 2019 roku wyniósł 6,12% (przy zakładanym poziomie 9,59%) czyli wzrost o 0,46 p. proc względem 2018 roku. Z powyższego raportu wynika, że w 2019 r., w porównaniu z rokiem 2012, moc zainstalowana w elektrowniach słonecznych [MW] oraz wyprodukowana przez nie energia elektryczna [GWh] wzrosła 1 184 razy, a energia elektryczna 623 razy. Powyższe tempo wzrostu zaczęło stabilizować się po 2015 r., i w 2019 r. (w stosunku do 2015 r.) odnotowano ponad czternastokrotny wzrost mocy zainstalowanej i ponad dwunastokrotny – pozyskanej energii elektrycznej. Wyhamowanie udziału energii odnawialnej w końcowym zużyciu energii brutto wynikało ze zwiększym zapotrzebowaniem na energię oraz wzrostem użycia oleju napędowego jako źródła pozyskiwania energii. Zmiany w prawie z 2016 roku (wprowadzenie ustawy o inwestycjach w zakresie elektrowni wiatrowych oraz z wprowadzeniem aukcji jako mechanizmu wsparcia) również miały wpływ na spowolnienie rozwoju alternatywnych źródeł pozyskiwania energii. W ramach Strategii na rzecz Odpowiedzialnego Rozwoju została określona polityka energetyczna Polski do roku 2020 z perspektywą do 2030 roku oraz polityka energetyczna Polski do 2040 r. Polska poprzez przyjęte ustawy zobowiązała się do uzyskania poziomu 23% udziału odnawialnych źródeł energii w końcowym zużyciu energii brutto do roku 2030 oraz produkcji połowy mocy ze źródeł odnawialnych do roku 2040.

1.3.3 Regulacje prawne dotyczące przyłączeń elektrowni

Z powodu różnorodności zagadnień dotyczących fotowoltaiki, akty prawne i rozporządzenia dotyczą prawa energetycznego, prawa o ochronie przeciwpożarowej, budowlanego, o odnawialnych źródłach energii czy prawie przeciwpożarowym. We wrześniu 2020 roku weszła w życie nowelizacja ustawy Ustawa o zmianie ustawy Prawo budowlane oraz niektórych innych ustaw (2020). Jak wynika z art. 29 ust. 4 pkt. 3, nie jest wymagana decyzja o pozwoleniu na budowę oraz zgłoszenia faktu zainstalowania "pomp ciepła, wolnostojących kolektorów słonecznych, urządzeń fotowoltaicznych o mocy zainstalowanej elektrycznej nie większej niż 50 kW

z zastrzeżeniem, że do urządzeń fotowoltaicznych o mocy zainstalowanej elektrycznej większej niż 6,5 kW stosuje się obowiązek uzgodnienia z rzecznikiem do spraw zabezpieczeń przeciwpożarowych pod względem zgodności z wymaganiami ochrony przeciwpożarowej". Jedynymi wyjątkami są instalacje: fotowoltaiczne na obiekcie wpisanym do rejestru zabytków, na terenie *Natura2000* oraz oddziałujące na środowisko.

Samo podłączenie do sieci jest regulowane poprzez dokument Rozporządzenie Komisji Europejskiej ([2016](#)) z 14.04.2016 roku. Zgłoszenie może zostać dokonane przez różne podmioty takie jak: osoby fizyczne, osoby fizyczne prowadzące działalność gospodarczą czy przedsiębiorstwa. Rozporządzenie wydziela różne typy jednostek wytwórczych bazując na poziomie napięcia. Sama procedura przyłączenia jest darmowa i różni się w zależności czy dany obiekt był już wcześniej przyłączony do sieci oraz czy moc instalacji nie przekracza mocy przyłączeniowej. Do najpopularniejszych przyłączeń należą mikroinstalacje czyli instalacje, w których moc szczytowa nie przekracza 50kWp oraz nie może przekroczyć mocy przyłączeniowej. W ustawie o Odnawialnych Źródłach Energii (OZE) zawiera liczne udogodnienia dla tzw. *prosumentów*. Wyrażenie prosument pochodzi z połączenia słów "producent" i "konsument". Prosumenci to klienci, którzy są również producentami energii.

1.4 Źródła danych o odnawialnych źródłach energii w Polsce

1.4.1 Statystyka publiczna

Jednym z głównych źródeł pozyskiwania wiedzy na temat stanu odnawialnych źródeł energii jest raport Głównego Urzędu Statystycznego (GUS) "Energia ze źródeł odnawialnych w 2019 roku" ([GUS 2020b](#)). Raport Głównego Urzędu Statystycznego określa energię ze źródeł odnawialnych (OZE) jako energię pochodząą z naturalnych, powtarzających się procesów przyrodniczych, pozyskiwaną z odnawialnych niekopalnych źródeł energii. Energia ze źródeł odnawialnych obejmuje energię z:

1. promieni słonecznych,
2. wody,
3. wiatru,
4. zasobów geotermalnych,
5. biopaliw stałych,

6. biogazu,
7. biopaliw ciekłych,
8. pomp ciepła.

Jak wynika z raportu GUS, energia promieniowania słonecznego jest przetwarzana na energię elektryczną lub ciepło poprzez wykorzystanie:

1. termicznych elektrowni słonecznych,
2. ogniw fotowoltaicznych do bezpośredniego wytwarzania energii elektrycznej,
3. płaskich, tubowo-próżniowych kolektorów słonecznych (cieczowych lub powietrznych) do podgrzewania ciepłej wody użytkowej, wody w basenach kąpielowych, ogrzewania pomieszczeń, w procesach suszarniczych lub chemicznych.

W ramach dokumentu (GUS 2016a) wydanego dnia 11.08.2016 określone zostały zasady zbierania danych o odnawialnych źródłach energii GUS (2016b). Wydany zestaw zasad stanowi uzupełnienie dokumentu wydanego w 2006 roku o nazwie (GUS 2006). Głównym źródłem danych statystycznych o OZE są sprawozdania i kwestionariusze w ramach badań statystycznych statystki publicznej. Jak wynika z (GUS 2006), informacje o OZE pozyskiwane są w badaniu "Rynek materiałowy i paliwowo-energetyczny", które jest prowadzone w ramach programu badań statystycznych statystyki publicznej, zaś dane opracowywane są na podstawie sprawozdawczości Prezesa Głównego Urzędu Statystycznego, Ministra do spraw energii oraz Prezesa Urzędu Regulacji Energetyki. Dane o odnawialnych źródłach energii pozyskiwane są z następujących kwestionariuszy sprawozdawczych oraz formularzy:

- Kwestionariusze i formularze GUSu:
 - Formularz G-03 – sprawozdanie o zużyciu paliw i energii,
 - Formularz G-02b – sprawozdanie bilansowe nośników energii i infrastruktury cieplowniczej,
 - Formularz G-02o – sprawozdanie o ciepłe ze źródeł odnawialnych,
 - Ankieta E-GD o zużyciu paliw i energii w gospodarstwach domowych.
- Formularze Ministerstwa Energii:
 - Formularz G-10.1(w)k – sprawozdanie o działalności elektrowni wiatrowej/wodnej,
 - Formularz G-10.2 – sprawozdanie o działalności elektrowni cieplnej zawodowej,

- Formularz G-10.3 – sprawozdanie o mocy i produkcji energii elektrycznej i ciepła elektrowni (elektrociepłowni) przemysłowej,
 - Formularz G-10.6 – sprawozdanie o mocy i produkcji elektrowni wodnych, wiatrowych i innych źródeł odnawialnych,
 - Formularz G-10.m – miesięczne dane o energii elektrycznej,
- Formularze Urzędu Regulacji Energetyki:
 - sprawozdanie producenta – stanowiące załącznik do rozporządzenia Ministra Gospodarki z dnia 9 grudnia 2014 r.,
 - URE-NCW – sprawozdanie podmiotu realizującego Narodowy Cel Wskaźnikowy.

Dane otrzymywane przez producentów i konsumentów energii elektrycznej i ciepła z odnawialnych źródeł to informacje głównie dotyczące:

- pozyskania,
- importu,
- eksportu,
- zmian zapasów,
- zużycia krajowego ogółem,
- zużycia na wsad przemian,
- zużycia własnego sektora energii,
- zużycia końcowego.

Pozyskiwane wyżej dane są następnie wykorzystywane do obliczenia udziału danego typu energii w całkowitym zużyciu energii ze wszystkich źródeł energii. Dokonywane obliczenia są zharmonizowane na poziomie międzynarodowym oraz stosowane jest specjalne narzędzie SHort Assessment of Renewable Energy Sources (SHARES) (Europejska 2018).

1.4.2 Rejestry administracyjne

W ramach źródeł niestatystycznych należy wymienić rejesty administracyjne, które są tworzone na potrzeby informacyjne Państwa. W kontekście niniejszej pracy należy wymienić Urząd Regulacji Energetyki (URE), Ministerstwo Klimatu i Środowiska (MEiŚ) oraz Główny Urząd Geodezji i Kartografii (GUGiK).

1.4.2.1 Urząd Regulacji Energetyki

URE jest urządkiem obsługującym Prezesa URE, który jest głównym organem administracji państowej, regulujący polski rynek energii. Urząd został powołany do życia 23 czerwca 1997 roku na mocy ustawy z dnia 10 kwietnia 1997 roku – Prawo energetyczne (Prawo Energetyczne 1997). URE zajmuje się:

- zapewnienie bezpieczeństwa energetycznego w Polsce,
- kontrolowanie podmiotów funkcjonujących na rynku energii (również nakładanie kar),
- zapewnienie usług związanych ze sprzedażą i dystrybucją energii oraz gwarantowanie jakości energii,
- ochrona odbiorców oraz dostawców energii.

URE monitoruje liczbę mikro i małych instalacji na terenie całego kraju. W przypadku mikroinstalacji mowa o instalacji o zainstalowanej mocy mniejszej niż 50kW, przyłączonej do sieci elektroenergetycznej o napięciu znamionowym niższym niż 110 kV albo o mocy osiągalnej cieplnej w skojarzeniu nie większej niż 150 kW, w której łączna moc zainstalowana elektryczna jest nie większa niż 50 kW. Według raportu URE z 16.03.2021 roku o nazwie "Raport dotyczący energii elektrycznej wytworzonej z OZE w mikroinstalacji i wprowadzonej do sieci dystrybucyjnej (art. 6a ustawy o odnawialnych źródłach energii)" (Urząd Regulacji Energetyki 2021a) na koniec 2020 roku energia elektryczna była wytwarzana w 459 168 mikroinstalacjach. Łączna zainstalowana moc wyniosła ponad 3GW. Najwięcej mikroinstalacji wykorzystywało promienie słoneczne (458 675) generując przy tym 3015,4 MW.

Tablica 1.2. Mikroinstalacje OZE w podziale na rodzaj odnawialnego źródła energii

Rodzaj mikroinstalacji	Moc	Liczba
wykorzystujące biogaz	0,9	50
wykorzystujące biomasę	0,3	18
wykorzystujące energię promieniowania słonecznego	3015,4	458 675
wykorzystujące energię wiatru	0,5	83
wykorzystujące hydroenergię	8,0	293
Łącznie	3025,1	459 119

Źródło: Opracowanie własne na podstawie: Urząd Regulacji Energetyki (2021a).

Jak wynika z Tabeli 1.2, prawie 100% wszystkich użytkowników mikroinstalacji stanowili prosumenci (452 307 prosumenckich mikroinstalacji ze 459 168 wszystkich). Dynamika przyrostu

w okresie 2018/2019 wyniosła 191% zaś w okresie 2019/2020 wyniosła około 202%. Z nowo przyłączonych mikroinstalacji, 2/3 zostały przyłączone do PGE Dystrybucja S.A oraz Tauron Dystrybucja S.A.

Tablica 1.3. Liczba prosumentów przyłączonych do sieci poszczególnych OSD w latach 2018-2020

Nazwa dostawcy	2018	2019	2020	Łączna moc
PGE Dystrybucja S.A.	18083	55140	154675	227 898
Tauron Dystrybucja S.A	15737	45186	142512	203 435
ENERGA-OPERATOR S.A.	9148	26696	86888	122732
ENEA Operator Sp. z o.o.	6285	18625	61675	86 585
innogy Stoen Operator Sp. z o.o.	1074	2560	5092	8726
Energoserwis Kleszczów Sp. z o.o.	806	1003	1132	2941
Pozostali	30	98	333	461

Źródło: Opracowanie własne na podstawie Urząd Regulacji Energetyki (2021a).

Jak wskazuje Tabela 1.3, najwięcej prosumentów w 2020 roku zostało przyłączonych do sieci firmy PGE Dystrybucja S.A. (154 675 prosumentów), następnie do Tauron Dystrybucja S.A. (142 512 prosumentów), a na trzecim miejscu znaleźli się prosumenci z przyłączeniami do ENERGA-OPERATOR S.A. (86 888 przyłączeń). W ramach art. 17 ustawy o odnawialnych źródłach energii powstał raport (Urząd Regulacji Energetyki 2021b). W przypadku małych instalacji, mowa o instalacji o łącznej mocy zainstalowanej elektrycznej większej niż 50 kW i mniejszej niż 500 kW.

Jak wynika z raportu, małe instalacje odpowiadają za około 2% całej wytwarzanej energii ze źródeł odnawialnych (183 MW z małych instalacji na ponad 10GW ze wszystkich rodzajów instalacji). Na posiadaczy małych instalacji czekają liczne ułatwienia związane z formalnościami (np. brak nakazu posiadania koncesji czy łatwiejsze przyłączenie do sieci). W przeciwieństwie do mikroinstalacji, właściciele małych instalacji, których działalność opiera się na wytwarzaniu energii są zobowiązani do przekazywania następujących informacji po zakończeniu każdego kwartału:

- łączna ilość wyprodukowanej energii,
- ilość sprzedanej energii pochodzącej z odnawialnych źródeł energii,
- całkowita ilość zużytych paliw dotworzenia energii elektrycznej,
- energii elektrycznej sprzedanej końcowym odbiorcom.

W 2020 roku energia była produkowana przez 898 małych instalacji, był to wzrost o 81 instalacji względem poprzedniego roku. Wzrost wytwarzanej energii wyniósł 20,3 MW rok do

roku, przez co całkowita zainstalowana moc z małych instalacji wyniosła 183 MW. Tabela 1.4 przedstawia sumaryczne ilości wytwarzanej energii oraz liczby małych instalacji z podziałem na rodzaj OZE.

Tablica 1.4. Małe instalacje OZE w podziale na rodzaj odnawialnego źródła energii

Rodzaj małej instalacji	Moc	Liczba
Instalacje wykorzystujące biogaz	32,10	117
Instalacje wykorzystujące biomasę	0,47	2
Instalacje wykorzystujące energię promieniowania słonecznego	66,86	328
Instalacje wykorzystujące energię wiatru	31,71	108
Instalacje wykorzystujące hydroenergię	51,96	343
Łącznie	183,1	898

Źródło: Opracowanie własne na podstawie: (Urząd Regulacji Energetyki 2021b).

1.4.2.2 Ministerstwo Klimatu i Środowiska

Ministerstwo Klimatu i Środowiska przygotowało program dofinansowań do instalacji fotowoltaicznych o nazwie "Mój Prąd". Początkowo program został zapowiedziany przez premiera Mateusza Morawieckiego dnia 23 lipca 2019 roku ³.

Głównym założeniem programu był wzrost produkcji energii z mikroinstalacji fotowoltaicznych, a początkowy budżet wyniósł 1 mld zł. Dofinansowanie obejmowało do 50% kosztów instalacji, maksymalnie 5000 zł. Wsparciem mogły zostać objęte jedynie gospodarstwa domowe dla instalacji o mocy między 2, a 10 kW. Termin nadsyłania wniosków dla pierwszej edycji programu został ustalony między 30 sierpnia, a 20 grudnia 2019 roku. Jak wynika z raportów Narodowy Fundusz Ochrony Środowiska i Gospodarki Wodnej (NFOŚiGW), organizatora programu "Mój prąd" pierwsza część programu zakończyła się dużym sukcesem. Wnioski składano od 30 sierpnia do 20 grudnia 2019 r. NFOŚiGW dofinansował ponad 27 tys. instalacji PV. Na ten cel wydatkowano prawie 132 mln zł, a zainstalowana moc PV to 152 MW. Drugi nabór do programu ruszył 13.01.2020 roku i miał potrwać do 18 grudnia 2020 roku. Jednak ze względu na wyczerpany budżet wynoszący 1,1 mld złotych, Ministerstwo Klimatu i Środowiska oraz NFOŚiGW postanowiły 7 grudnia ubiegłego roku zamknąć nabór do programu "Mój Prąd". Dzięki zwiększonemu budżetowi udało się ostatecznie dofinansować ponad 220 tys. mikroinstalacji. W końcowym etapie składania wniosków, liczba składanych wniosków na dzień wyniosła ponad 2 tysiące. Łączna moc dotowanych instalacji wyniosła ponad 1,2 GW. 1 lipca 2021 roku wystartowała trzecia wersja programu "Mój Prąd".

W najnowszej edycji programu można maksymalnie otrzymać zwrot wynoszący 50% kosztu instalacji, nie więcej niż 3000 zł. Sama dotacja jest zwolniona z podatku PIT zaś koszty nieopryte przez dotację można odliczyć od podatku (ulga termomodernizacyjna). Czas na składanie wniosków jest do 22.12.2021 lub do wyczerpania budżetu (na dzień 4 lipca 2021 roku wynoszący 51 789 998 zł). Według głównej strony programu (<https://mojprad.gov.pl>) wynika, że na dzień 4 lipca 2021 roku z programu "Mój prąd" dofinansowano 220 994 projektów fotowoltaicznych redukując przy tym CO₂ rocznie o 1 011 847 200 kg.

Rysunek 1.5 pokazuje liczbę dofinansowań oraz moc instalacji w MW z podziałem na województwa. Najwięcej dofinansowań z programu "Mój prąd" otrzymali prosumenti z województwa śląskiego (26712), następnie z województwa wielkopolskiego (26097), a na trzecim miejscu byli klienci indywidualni z województwa mazowieckiego (24652). Najmniej dofinanso-

³ Źródło: <https://www.gov.pl/web/klimat/moj-prad>



Rysunek 1.5. Dotacje z programu Mój Prąd

Objaśnienie: Kolor zielony (górsza wartość) to liczba przyznanych dofinansowań do projektów instalacji PV, kolor niebieski (dolna wartość) to łączna moc instalacji w kW Źródło:
<https://mojprad.gov.pl>.

wań otrzymali mieszkańcy województwa podlaskiego. W przypadku wzięcia pod uwagę liczby mieszkańców dla danego województwa wnioski się zmieniają, dla liczby przyznanych dofinansowań największy współczynnik jest dla województwa podkarpackiego (9,2 dofinansowania na 1000 mieszkańców), na kolejnym miejscu jest województwo wielkopolskie (7,5 dofinansowania na 1000 mieszkańców), zaś na trzecim miejscu znalazło się województwo małopolskie (7,2 dofinansowania na 1000 mieszkańców). Najniższy współczynnik (4 dofinansowania na 1000 mieszkańców) pochodził z województwa podlaskiego.

1.4.2.3 Główny Urząd Geodezji i Kartografii

Główny Urząd Geodezji i Kartografii (GUGIK) jest urzędem obsługującym i działającym pod kierownictwem Głównego Geodety Kraju od 1945 roku. Urząd kierowany jest przez Głównego Geodetę Kraju przy pomocy dyrektora generalnego, wiceprezesa i dyrektorów komórek organizacyjnych. GUGIK wykonuje zadania powierzone Głównemu Geodecie Kraju, a w szczególności odpowiedzialny jest za:

1. prowadzenie zasobu geodezyjnego i prowadzenie centralnego zasobu geodezyjnego i kartograficznego jak i dysponowanie środkami Funduszu Gospodarki Zasobem Geodezyjnym i Kartograficznym,
2. nadzorowanie realizacji polityki państwa w zakresie geodezji i kartografii,
3. nadawanie uprawnień zawodowych w dziedzinie geodezji i kartografii,
4. prowadzenie rejestru osób uprawnionych,
5. tworzenie i utrzymywanie geoportalu, jako punkt dostępu do usług związanych z działalnością GUGIK

1.4.3 Big data

1.4.3.1 Zdjęcia satelitarne

Zdjęcia satelitarne to obrazy pokazujące Ziemię zrobione przez satelity krążące wokół Ziemi, które są własnością rządów lub przedsiębiorstw takich jak np. Google. Historia zdjęć satelitarnych sięga 14 sierpnia 1959 roku kiedy to amerykańska satelita Explorer 6 uwieczniła pierwsze zdjęcie satelitarne. Aktualnie główne zastosowania zdjęć satelitarnych to meteorologia, oceanografia czy rolnictwo. Ze względu na rozdzielcość wynoszącą maksymalnie 31 cm na jeden piksel, zastosowanie zdjęć satelitarnych jest ograniczone. Przykładowy dach pochodzący ze zdjęcia satelitarnego został ukazany w Rysunek 1.6. Rozpoznawanie paneli na podstawie zdjęć satelitarnych, w niektórych sytuacjach byłoby możliwe jednak niska jakość zdjęć mogłaby negatywnie wpływać na skuteczność modelu.

1.4.3.2 Zdjęcia lotnicze

Ortofotomapy to mapa przekształcona metodą różniczkową oraz przedstawiona dla konkretnego układu współrzędnego. Zdjęcie lotnicze staje się ortofotomapą po procesie ortorektyfikacji. W procesie ortorektyfikacji zdjęcia są nakładane na model terenu, inaczej, dochodzi



Rysunek 1.6. Przykładowy wygląd dachu pochodzącego z Google Maps na bazie zdjęcia satelitarnego

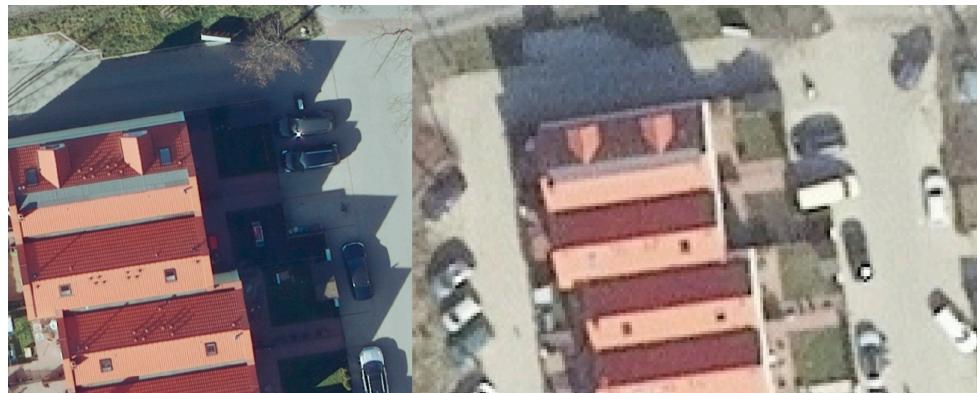
Źródło: Opracowanie własne na podstawie: maps.google.pl.

do zamiany rzutu śródkowego na rzut ortogonalny. W końcowym etapie należy połączyć liczne małe zdjęcia co nazywane jest mozaikowaniem.

Do głównych cech ortofotomapy należą:

1. rzut ortogonalny czyli prostopadły rzut względem terenu
2. równa skala terenu dla terenu
3. możliwie wysoka rozdzielczość, do 5cm terenu na każdy piksel

W zależności od pobieranej ortofotomapy, rozdzielczości mogą się różnić. W przypadku danych pozyskanych do pracy magisterskiej, wykorzystano jedynie ortofotomapy z najwyższą dostępną rozdzielczością, gdzie 5cm terenu odpowiada 1 pikselowi. W ramach danej lokalizacji dostępne były również ortofotomapy o mniejszej rozdzielczości takiej jak np. 25 cm na każdy piksel. Użycie mapy o takiej rozdzielczości znaczaco zmniejszyłaby ilość danych jednak ze względu na za niską rozdzielczość widoczną na Rysunku 1.7 zdecydowano wybrać ortofotomapy o największej rozdzielczości.



Rysunek 1.7. Porównanie jakości ortofotomap dla miasta Poznania

Z lewej strony fragment dachu z ortofotomapy o rozdzielczości 5 cm, a prawej strony fragment dachu z ortofotomapy o rozdzielczości 25 cm na jeden piksel
Źródło: Opracowanie własne.

1.5 Podsumowanie

Informacje dotyczące rozwoju energii ze źródeł odnawialnych pochodzą z badań częściowych oraz sprawozdań tworzonych przez Główny Urząd Statystyczny. Zmiany wynikające z regulacji prawnych oraz zobowiązania wobec Unii Europejskiej powinny wymusić zastosowanie dokładniejszych oraz szybszych metod estymacji progresu przechodzenia na zieloną energię. Dzięki wykorzystaniu nowoczesnych źródeł danych (big data) oraz algorytmów uczenia maszynowego możliwe będzie rozszerzenie informacji zawartej w rejestrach. W kolejnym rozdziale omówione zostaną głębokie sieci neuronowe, które można wykorzystać do monitorowania wykorzystania odnawialnych źródeł energii.

Rozdział 2

Detekcja obiektów z wykorzystaniem głębokich sieci neuronowych

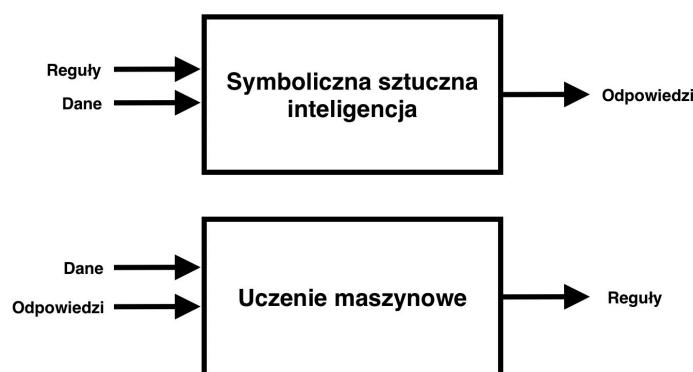
2.1 Sieci neuronowe – ujęcie historyczne

Badania nad mechanicznym lub "formalnym" rozumowaniem rozpoczęły filozofowie i matematycy w starożytności. Badania nad logiką matematyczną doprowadziły bezpośrednio do teorii obliczeń Alana Turinga, który zasugerował, że maszyna, poprzez tasowanie symboli tak prostych jak "0" i "1", może symulować każdy możliwy do pomyślenia akt matematycznej dedukcji. To spostrzeżenie, że komputery cyfrowe mogą symulować każdy proces formalnego rozumowania, znane jest jako teza Churcha-Turinga (Berlinski 2001). Wraz z równoległymi odkryciami w neurobiologii, teorii informacji i cybernetyce, doprowadziło to badaczy do rozważenia możliwości zbudowania elektronicznego mózgu. Turing zaproponował zmianę pytania z "czy maszyna jest inteligentna" na "czy jest możliwe, aby maszyna wykazywała inteligentne zachowanie". Pierwszą pracą, która jest obecnie powszechnie uznawana za początek sztucznej inteligencji, był formalny projekt McCulloucha i Pittsa z 1943 roku dla "sztucznych neuronów" o Turingowej kompletności (Russell i Norvig 2010).

Początkowa idea sztucznej inteligencji polegała na stworzeniu wystarczającej liczby reguł w celu opisywania rzeczywistości. Dopasowanie utworzonych reguł oraz wykorzystanie danych skutkuje utworzeniem odpowiedzi. Koncepcja ta nazywana jest symboliczną sztuczną inteligencją i dominowała od połowy lat 50 do końca lat 80 XX wieku. Jednym z ciekawszych przykładów implementacji symbolicznej sztucznej inteligencji jest czatbot Eliza, przedstawiony światu przez informatyka z MIT Josepha Weizenbauma w 1966 roku (Weizenbaum 1983). Eliza

to skrypt oparty na ograniczonym, z góry ustalonym przepływie, który mógł symulować rozmowę z psychoterapeutą. Program prowadził "rozmowy", wykorzystując metodykę dopasowywania wzorców i zastępowania, która dawała użytkownikom złudzenie zrozumienia ze strony programu, ale nie miała wbudowanych ram dla kontekstualizacji zdarzeń.

Weizenbaum stworzył Elizę, aby zademonstrować powierzchowność komunikacji między człowiekiem a maszyną, a przy okazji zbudował czatbota zdolnego do oszukiwania ludzi, że komunikują z psychoterapeutą. Osiągnięcie to zostało później ugruntowane przez zaliczenie algorytmu ograniczonego testu Turinga na inteligencję maszyny.



Rysunek 2.1. Uczenie maszynowe: nowy model programowania

Źródło: Opracowanie własne na podstawie: Chollet (2017, strona. 23).

Tak zdefiniowana sztuczna inteligencja jest ogólną dziedziną, na którą składa się również uczenie maszynowe jak i głębokie uczenie. Z biegiem czasu symboliczna sztuczna inteligencja została wyparta przez uczenie maszynowe.

W przypadku paradygmatu uczenia maszynowego, użytkownik na podstawie wprowadzanych danych oraz docelowych odpowiedzi wymaga od komputera stworzenia uniwersalnych reguł, możliwych do wykorzystania w przyszłości. Poprzez dostarczenie wystarczającej liczby przypadków, komputer jest w stanie stworzyć uniwersalne reguły.

Termin uczenia maszynowego został sformułowany w 1959 roku przez Arthura Samuela, amerykańskiego pracownika IBM i pioniera w dziedzinie gier komputerowych i sztucznej inteligencji (Samuel 1959). Kolejne użycia sformułowania miały miejsce w książce Nilssona o maszynach uczących się, zajmująca się głównie uczeniem maszynowym dla klasyfikacji wzorców. Następne wzmianki to praca Duda, Hart i in. (1973), gdzie autorzy opisywali zagadnienie związane z rozpoznawaniem wzorców. W 1981 roku Stevo Bozinovski przedstawił raport na temat

wykorzystania strategii nauczania w taki sposób, aby sieć neuronowa nauczyła się rozpoznawać 40 znaków (26 liter, 10 cyfr i 4 symbole specjalne) z terminala komputerowego. Mitchell i in. (1997) przedstawili szeroko cytowaną, bardziej formalną definicję algorytmów badanych w dziedzinie uczenia maszynowego: "O programie komputerowym mówi się, że uczy się na podstawie doświadczenia E w odniesieniu do pewnej klasy zadań T i miary wydajności P , jeśli jego wydajność w zadaniach w T , mierzona przez P , poprawia się wraz z doświadczeniem E ". Współczesne uczenie maszynowe ma dwa cele, jednym z nich jest klasyfikowanie danych na podstawie opracowanych modeli, drugim celem jest przewidywanie przyszłych wyników na podstawie tych modeli. Jako przedsięwzięcie naukowe, uczenie maszynowe powstało w wyniku prac nad sztuczną inteligencją, w których wykorzystywano samouczące się programy, zdolne do ekstrakcji wiedzy z informacji oraz później predykcji.



Rysunek 2.2. Sztuczna inteligencja, uczenie maszynowe i uczenie głębokie

Źródło: Opracowanie własne na podstawie: Chollet (2017, strona. 22).

W świecie otaczających nas pokładów danych dzięki wykorzystaniu uczenia maszynowego nie jest potrzebne zatrudnianie ludzi do ręcznego określania reguł. Sztuczna inteligencja już nie tylko zmienia branżę informatyczną, ale również ma coraz większy wpływ w życiu codziennym. Za pomocą uczenia maszynowego stworzono efektywne modele filtrowania spamu w pocztach elektronicznych, rozpoznawania twarzy, mowy czy samojezdzącymi pojazdami.

Głębokie uczenie (ang. *deep learning*) jest podgrupą uczenia maszynowego, które do tworzenia ogólnych reguł tworze sieci neuronowe. Głębokość odnosi się do liczby warstw modelu danych. Koncepcja uczenia modelu na podstawie głębokich sieci neuronowych pochodzi jako inspiracja z biologii i budowy ludzkiego mózgu.

Tabela 2.1 przedstawia główne różnice między uczeniem maszynowym, a głębokim uczeniem. Główną różnicą jest czas trenowania modelu, w przypadku uczenia głębokiego może mieć to miejsce przez kilka do nawet kilkudziesięciu godzin. W przypadku uczenia maszynowego, model najczęściej uczony jest w ciągu kilku godzin. Kluczową różnicą jest również infrastruktura, dla uczenia głębokiego wykorzystanie układów GPU (ang. *graphics processing unit*) jest wskazane zaś dla nauczenia modelu maszynowe układ CPU (ang. *central processing unit*) wystarcza.

Tablica 2.1. Różnice między uczeniem głębokim, a uczeniem maszynowym

Czynnik	Uczenie Głębokie	Uczenie maszynowe
Czas uczenia	Długi	Krótki
Dane	Wymaga dużych zbiorów danych	Nie wymaga dużych zbiorów danych; może uczyć się na mniejszych zbiorach
Infrastruktura	Może uczyć się na CPU, ale zalecane jest również GPU	Obliczenia wykonywane są na CPU

Źródło: Opracowanie własne na podstawie Chollet (2017).

2.2 Teoretyczne podstawy głębokich sieci neuronowych

Pierwsza koncepcja sztucznego neuronu powstała w 1943 roku przez badania Warrena McCullocha i Walter Pittsa. Idea ta zawierała uproszczony model komórki nerwowej, tzw. neuronu McCullocha-Pittsa (z ang. McCulloch-Pitts neuron – MCP; zob. McCulloch i Pitts (1943)). W przytoczonym źródle, McCulloch i Pitts opisali komórkę nerwową jako bramkę logiczną zwracającą 0 lub 1.

W 1957 roku Frank Rosenblatt bazując na modelu neuronu MCP opublikował pierwszą koncepcję reguły uczenia perceptronu (Rosenblatt 1957). Za pomocą reguły perceptronu Rosenblatt zaproponował algorytm, który automatycznie uczyłby się optymalnych wag, które następnie byłyby mnożone przez cechy wejściowymi w celu podjęcia decyzji, czy neuron uruchamia się (wysyła sygnał) czy nie.

2.2.1 Budowa sztucznego neuronu

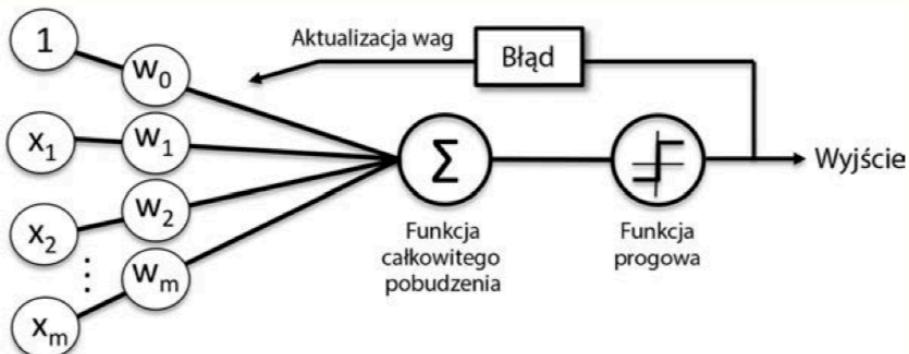
Formalnie rzecz biorąc, możemy umieścić ideę sztucznych neuronów w kontekście zadania klasyfikacji binarnej, gdzie dla uproszczenia odnosimy się do naszych dwóch klas jako 1 (klasa pozytywna) i -1 (klasa negatywna). Następnie możemy zdefiniować funkcję decyzyjną $\phi(z_i)$, która jako argument przyjmuje wektor z_i będący kombinacją liniową wektora danych wejściowych x_i oraz wag w_i , gdzie $i = 1, \dots, N$ to liczba obserwacji jak we wzorze (2.1). Wektor x_i dla każdej obserwacji ma wymiar m , gdzie $m = 1, \dots, M$ to liczba cech

$$z_i = w_{i1} \times x_{i1} + \dots + w_{iM} x_{iM} = \mathbf{w}_i^T \mathbf{x}_i. \quad (2.1)$$

Jeżeli dla danego zestawu danych całkowita suma jest wyższa od wartości progowej to θ to przedzielamy kombinację do klasy 1 w przeciwnym wypadku do klasy -1 . Funkcja decyzyjna ma postać funkcji skokowej dana jest wzorem (2.2)

$$\phi(z_i) = \begin{cases} 1 & \text{jeżeli } z_i \geq \theta \\ -1 & \text{jeżeli } z_i < \theta \end{cases} \quad (2.2)$$

Zwykle koncepcję działania perceptronu przedstawia się graficznie jak na Rysunku 2.3.



Rysunek 2.3. Ogólna budowa perceptronu

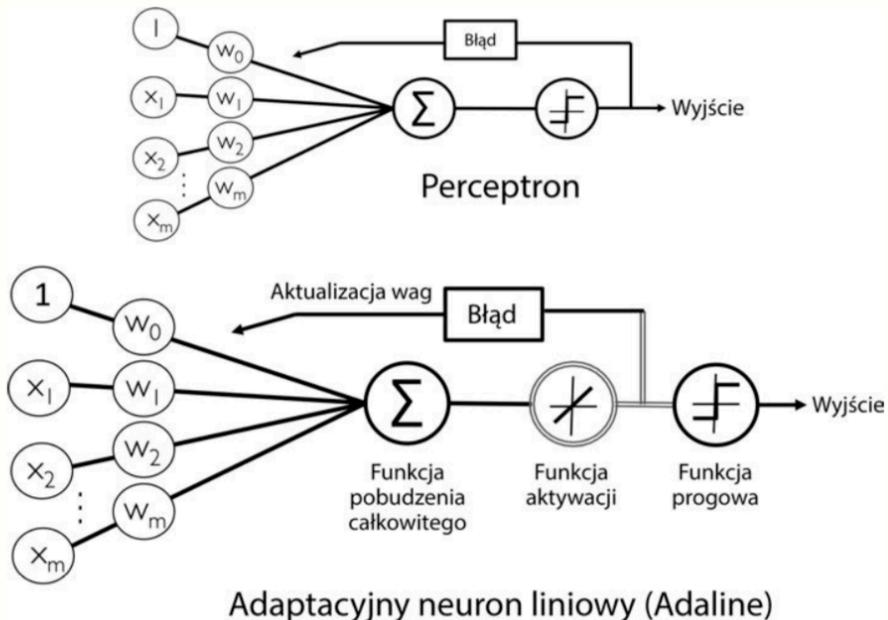
Źródło: Opracowanie własne na podstawie: Sebastian Raschka (2019, strona. 51).

Innym rodzajem jednowarstwowej sieci neuronowej jest ADALINE (ADApacyjny LINiowy NEuron). Koncepcja tej sieci powstała na początku lat 60-tych, po prezentacji perceptronu Franka Rosenblatta, autorem był Bernard Widrow oraz jego doktorant Tedd Hoff. Neuron liniowy Widrowa i Hoffa był rozwinięciem początkowego perceptronu, zaimplementowane zo-

stały kluczowe koncepcje definiowania i minimalizowania ciągłych funkcji kosztu. Tak zaimplementowany neuron tworzy podstawę bardziej zaawansowanych algorytmów klasyfikujących takich jak maszyny wektorów nośnych, modeli regresji czy regresji logistycznej. W przypadku nowej koncepcji wagi są aktualizowane na podstawie liniowej funkcji aktywacji, nie zaś funkcją skokową tak jak to było przy perceptronie Rosenblatta. Funkcja aktywacji w modelu Adaline przedstawia wzór (2.3)

$$\phi(z_i) = \mathbf{w}_i^T \mathbf{x}_i. \quad (2.3)$$

W przypadku Adaline, liniowa funkcja aktywacji zostanie wykorzystana do wyliczenia błędu modelu, a następnie aktualizacji wag. Porównanie obu modeli znajduje się na Rysunku 2.4.



Rysunek 2.4. Porównanie perceptronu oraz Adaline

Źródło: Opracowanie własne na podstawie Sebastian Raschka (2019, strona. 61).

Modelu neuronu Widrowa i Hoffa zapoczątkował erę wykorzystywania funkcji celu do optymalizacji w procesie nauki. W trakcie nauczania algorytmu najczęściej wykorzystywana jest funkcja kosztu, którą sieć próbuje minimalizować. Dla neuronu liniowego Adaline została wy-

znaczona funkcja kosztu $J(\mathbf{w})$, minimalizująca sumę kwadratów błędów pomiędzy predykcją, a rzeczywistą etykietą klas:

$$J(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N (y_i - \phi(\mathbf{z}_i))^2 = \frac{1}{2} \sum_{i=1}^N (y_i - \phi(\mathbf{w}_i^T \mathbf{x}_i))^2, \quad (2.4)$$

gdzie współczynnik $\frac{1}{2}$ został dodany w celu łatwiejszego wyprowadzenia pierwszych pochodnych (gradientu) funkcji kosztu.

Początkowa waga jest korygowana w przeciwnym kierunku do gradientu $\nabla J(\mathbf{w})$ do momentu jak osiągnie minimum lokalne lub globalne kosztu $J(\mathbf{w})$. Gradient funkcji kosztu zostanie otrzymany po obliczeniu pochodnej cząstkowej funkcji przy wykorzystaniu z każdej wagi w_m

$$\frac{\partial J}{\partial w_m} = - \sum_{i=1}^N (y_i - \phi(\mathbf{z}_i))^2 \mathbf{x}_{im}. \quad (2.5)$$

W przypadku algorytmu Adaline, wagi są aktualizowane na podstawie wszystkich obserwacji ze zbioru uczącego z wykorzystaniem czynnika skalującego η (współczynnika uczenia, zmieniający wagi przy każdym kroku uczenia) jak we wzorze (2.6)

$$\nabla \mathbf{w} = -\eta \nabla J(\mathbf{w}). \quad (2.6)$$

Aby uzyskać ostateczne wartości \mathbf{w} stosuje się algorytmy iteracyjne, które w każdej iteracji aktualizują wektor wag zgodnie ze wzorem (2.7)

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta \nabla J(\mathbf{w}^{(k)}), \quad (2.7)$$

gdzie k to dana iteracja algorytmu optymalizacyjnego, który kończy swoje działanie jak zostanie spełniony warunek stopu. Zwykle za warunek stopu przyjmuje się

$$\sum_{i=1}^N |\mathbf{w}_i^{(k+1)} - \mathbf{w}_i^{(k)}| < \epsilon, \quad (2.8)$$

lub

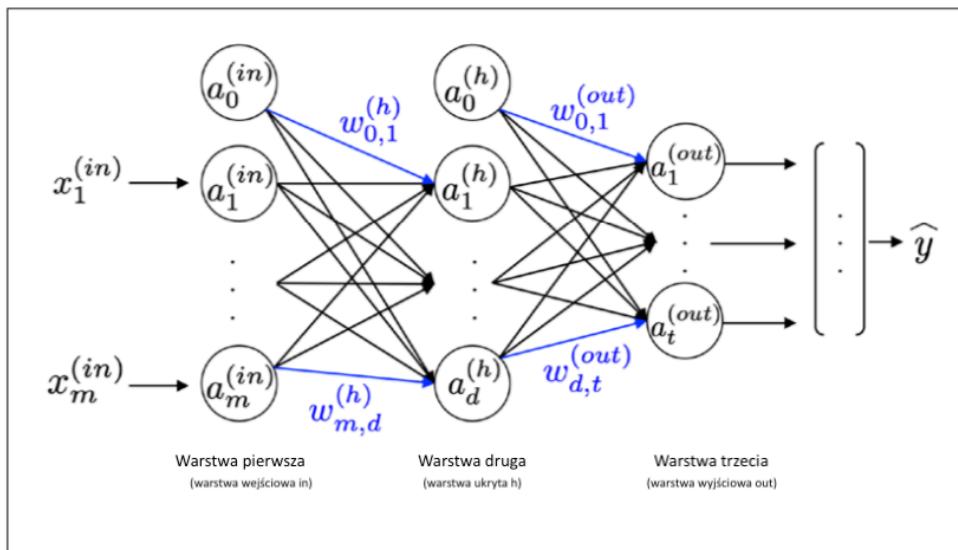
$$\nabla \mathbf{w} < \epsilon, \quad (2.9)$$

gdzie ϵ to odpowiednio mała wartość (np. 1×10^{-5}).

2.2.2 Wielowarstwowe sieci neuronowe

Sztuczny neuron omówiony w poprzedni podrozdziale jest podstawą wielowarstwowych sieci neuronowych. Pierwsze koncepcje sztucznego neuronu powstały w latach 50 na podstawie perceptronu Rosenblatta. Zainteresowanie wielowarstwowymi sieci zmalało ze względu na problemy uczenia wielowarstwowych sieci. Zagadnieniem ponownie zainteresowali się badacze David E. Rumelhart, Geoffrey E. Hinton i Ronald J. Williams w 1986 roku tworząc ideę wstecznej propagacji (z ang. *backpropagation algorithm*; Rumelhart i in. (1986)). Dzięki wykorzystaniu odkryć działaczy z lat 80-tych, zagadnienie sieci neuronowych nigdy nie było tak popularne jak w dzisiejszych czasach. Kluczowym aspektem jest zrozumienie nie tylko budulca (perceptronu), ale również całej architektury sieci, aby zrozumieć możliwy potencjał sieci neuronowych.

Wielowarstwowa sieć neuronowa nazywana również również wielowarstwowy perceptronem zawiera warstwę wejściową, ukrytą i warstwę wyjściową. Warstwa ukryta jest łącznikiem między warstwą wejściową, a wyjściową tak jak to przedstawia Rysunek 2.5. Sieć zawierająca co najmniej dwie warstwy ukryte nazywamy głęboką siecią neuronową.



Rysunek 2.5. Schemat wielowarstwowej sieci neuronowej

Źródło: Opracowanie własne na podstawie: Raschka i Mirjalili (2017, strona. 388).

Na Rysunku 2.5 rodzaj warstwy został zapisany w indeksie górnym, tak $a_m^{(in)}$ oznacza wartość dla m -tej cechy w warstwie wejściowej *in*¹, $a_m^{(h)}$ określa wartość dla m -tej cechy w tej

¹Symbol *in* od angielskiego *input*.

warstwie ukrytej h ², a $\mathbf{a}_m^{(out)}$ odnosi się do m -tej cechy w warstwie wyjściowej out ³. Każda wartość w warstwie l jest połączona ze wszystkimi wartościami z warstwy $l + 1$. W celu otrzymania wyniku, sieć wykorzystuje koncepcję propagacji w przód (z ang. *forward propagation*). W celu zrozumienia wpływu propagacji w przód, należy podzielić uczenie modelu na trzy podetapy:

1. od warstwy początkowej rozsyłane są wzorce danych uczących względem całej sieci, aby otrzymać odpowiedzi,
2. po otrzymaniu predykcji obliczany jest błąd, który jest minimalizowany
3. przesyłany jest wstecz błąd oraz liczona jest pochodna z błędu przy uwzględnieniu wag.

Powtórzenie powyższych punktów dla wielu epok oraz utworzenie wag modelu umożliwia wykorzystanie propagacji w przód, aby obliczyć predykcję. Wszystkie węzły ukryte są połączone z węzłami pochodząymi z warstwy wejściowej, zatem jednostkę aktywacji $\mathbf{a}_1^{(h)}$ można wyliczyć w następujący sposób:

$$\begin{aligned} \mathbf{z}_d^{(h)} &= a_0^{(in)} w_{0,1}^{(h)} + a_1^{(in)} w_{1,1}^{(h)} + \cdots + a_m^{(in)} w_{m,1}^{(h)}, \\ \mathbf{a}_d^{(h)} &= \phi(\mathbf{z}_d^{(h)}) , \end{aligned} \quad (2.10)$$

gdzie $\mathbf{z}_d^{(h)}$ jest pobudzeniem (z ang. *net input*), a ϕ funkcją aktywacji. Istotną cechą funkcji aktywacji jest różniczkowalność, aby móc zastosować procedurę uwzględniającą gradient. W zagadnieniach rozpoznawania obrazów często korzysta się z funkcji nieliniowych takich jak sigmoidalna. Przy wykorzystaniu takiej funkcji aktywacji każdy neuron zwraca wartość z przedziału od 0 do 1.

W celu zachowania zwięzości zapisu, funkcja aktywacji będzie wektoryzowana i będzie miała następującą strukturę:

$$\begin{aligned} \mathbf{z}^{(h)} &= \mathbf{W}^{(we)} \mathbf{a}^{(h)}, \\ \mathbf{a}^{(h)} &= \phi(\mathbf{z}^{(h)}) , \end{aligned} \quad (2.11)$$

gdzie $\mathbf{a}^{(h)}$ oznacza wektor cech o wymiarze $1 \times m$ dla próbki $\mathbf{x}^{(we)}$. $\mathbf{W}^{(h)}$ jest macierzą wag o wymiarze $m \times d$, gdzie d oznacza liczbę jednostek w ukrytej warstwie. Po działaniu mnożenia macierzowo-wektorowego, otrzymywany jest wektor o wymiarze $1 \times d$, który jest wektorem

²Symbol h od angielskiego słowa *hidden*.

³Symbol out od angielskiego słowa *output*.

pobudzenia $\mathbf{z}^{(h)}$. Otrzymany wektor można następnie wykorzystać w celu obliczenia aktywacji $\mathbf{a}^{(h)}$. Dla wszystkich próbek uczących wzór prezentowałby się następująco:

$$\mathbf{Z}^{(h)} = \mathbf{A}^{(we)} \mathbf{W}^{(h)}, \quad (2.12)$$

gdzie $\mathbf{A}^{(we)}$ oznacza $n \times m$ -wymiarową macierz. Wynikiem mnożenia dwóch macierzy będzie macierz całkowitego pobudzenia $\mathbf{Z}^{(h)}$ o wymiarach $n \times d$. W końcowym etapie, macierz całkowitego pobudzenia jest przemnażana przez funkcję aktywacji $\phi(\mathbf{z}_d^{(h)})$ otrzymując $n \times d$ -wymiarową macierz aktywacji $\mathbf{A}^{(h)}$. Analogicznie można zapisać zwektoryzowaną postać aktywacji warstwy wyjściowej dla danych uczących:

$$\mathbf{Z}^{(out)} = \mathbf{W}^{(h)} \mathbf{A}^{(out)}, \quad (2.13)$$

gdzie $\mathbf{W}^{(h)}$ ma postać $d \times t$ -wymiarowej macierzy, która następnie jest mnożona przez macierz $\mathbf{A}^{(h)}$ o wymiarach $n \times d$. W końcowym efekcie otrzymamy macierz $\mathbf{Z}^{(out)}$ o wymiarach $n \times t$. Wprowadzenie sigmoidalnej funkcji aktywacji sprawi, że otrzymamy wartości ciągłe:

$$\mathbf{A}^{(out)} = \phi(\mathbf{Z}^{(out)}), \quad (2.14)$$

2.2.3 Wsteczna propagacja

Pomimo zdefiniowania algorytmu wstecznej propagacji przez Rumelhart i in. (1986) w 1986 roku, algorytm nadal pozostaje jednym z najskuteczniejszych jeżeli chodzi o trenowanie sieci neuronowych. Metoda ta stanowi sposób obliczania pochodnych funkcji kosztu w przypadku sieci neuronowych. Następnie użyte są pochodne do aktualizowania wag, które stanowią kluczowy element przy wielowarstwowych sieciach neuronowych jak we wzorze (2.15) Problem wielowymiarowości funkcji kosztu w przypadku sieci neuronowych należy rozwiązać, aby spośród minimów lokalnych znaleźć minimum globalne

$$\frac{d}{dx}[f(g(x))] = \frac{df}{dg} \cdot \frac{dg}{dx}. \quad (2.15)$$

Podstawową operacją matematyczną propagacji wstecznej jest różniczkowanie funkcji złożonych. W zagadnieniu propagacji wstecznej zastosowano tzw. różniczkowanie automatyczne pozwalające na obliczanie długich funkcji złożonych. W ramach propagacji wstecznej przesy-

łany jest błąd od prawej do lewej strony (od wyjścia do wejścia). Obliczenia rozpoczynają się od obliczenia wektora błędu dla warstwy wyjściowej zgodnie ze wzorem (2.16)

$$\boldsymbol{\delta}^{(out)} = \mathbf{a}^{(out)} - \mathbf{y}. \quad (2.16)$$

W kolejnym etapie obliczany jest błąd dla warstwy ukrytej:

$$\boldsymbol{\delta}^{(h)} = \boldsymbol{\delta}^{(out)} \left(\mathbf{W}^{(out)} \right)^T \odot \frac{\partial \phi(\mathbf{z}^{(h)})}{\partial \mathbf{z}^{(h)}}, \quad (2.17)$$

gdzie

$$\frac{\partial \phi(\mathbf{z}^{(h)})}{\partial \mathbf{z}^{(h)}}$$

to pochodna sigmoidalnej funkcji aktywacji. Następnie obliczana jest macierz błędu warstwy ukrytej $\boldsymbol{\delta}^{(h)}$:

$$\boldsymbol{\delta}^{(h)} = \boldsymbol{\delta}^{(out)} \left(\mathbf{W}^{(out)} \right)^T \odot (\mathbf{a}^{(h)} \odot (1 - \mathbf{a}^{(h)})) \quad (2.18)$$

Po wyliczeniu wyrażeń $\boldsymbol{\delta}$ oblicza się pochodną oraz pochodne cząstkowe funkcji kosztu zapisaną jako:

$$\begin{aligned} \frac{\partial}{\partial w_{i,j}^{(out)}} J(\mathbf{W}) &= \mathbf{a}_j^{(h)} \boldsymbol{\delta}_i^{(out)} \\ \frac{\partial}{\partial w_{i,j}^{(h)}} J(\mathbf{W}) &= \mathbf{a}_j^{(in)} \boldsymbol{\delta}_i^{(h)} \end{aligned} \quad (2.19)$$

Po zliczeniu parametrów oblicza się gradient, dzięki któremu można zaktualizować wagi, przeciwnie do gradientu dla każdej z warstw.

$$\mathbf{W}^{(l)} := \mathbf{W}^{(l)} - \eta \Delta^{(l)} \quad (2.20)$$

2.3 Darknet i algorytm YOLO

W ramach pracy magisterskiej został wykorzystany YOLO v4 (YOLO z ang. *You only look once*) czyli systemem wykrywania obiektów. Pierwsza implementacja YOLO powstała w 2015 roku (Redmon, Divvala i in. 2015), a następnie rok później została zaprezentowana w trakcie konferencji Computer Vision and Pattern Recognition Conference (CVPR) przez Josepha Redmona, Santosha Divvala, Rossa Girshicka i Ali Farhandiego Redmon, Divvala i in. (2016a). Sama nazwa

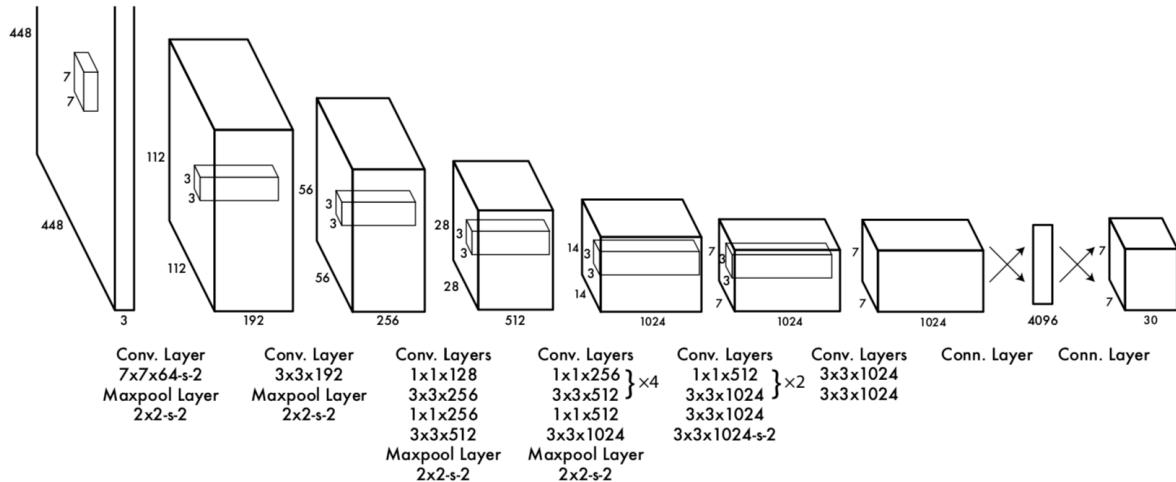
systemu wykrywania obiektów (YOLO) odnosi się do specyfiki modelu, jednorazowe przeprocesowanie obrazka poprzez sieć neuronową zwraca predykcję wszystkich obiektów znajdujących się na zdjęciu.

Pierwotna wersja YOLO bazowała na konwolucyjnej sieci neuronowej o nazwie Darknet Redmon (2013–2016). Autorem wykorzystanej sieci konwolucyjnej był również Joseph Redmon, który oparł rozwiązanie na języku C, aby optymalizować czas predykcji. Twórca Darknet i YOLO zaimplementował również sieć w CUDA (z ang. *Compute Unified Device Architecture*) czyli opracowanej przez Nvidię uniwersalnej architekturze procesorów wielordzeniowych, głównie przeznaczonych na karty graficzne. CUDA umożliwia przyspieszenie aplikacji wymagających dużej mocy obliczeniowej przy wykorzystaniu mocy procesorów graficznych (GPU) do równoległej części obliczeń.

Główną zaletą systemu YOLO nad innymi systemami wykrywania obiektów jest szybkość predykcji. W przypadku pierwszej wersji systemu, YOLO jest w stanie generować predykcje z predkością 45 FPSów (z ang. *frames per seconds*) czyli klatek na sekundę. Im większa wartość FPSów tym więcej zdjęć może dany model rozpoznać w czasie rzeczywistym. Częstotliwość klatek na sekundę w YOLO odpowiada czasowi detekcji wynoszącemu około 22 ms (milisekund). Porównaj to z czasem potrzebnym ludzkiemu mózgowi na wykrycie obrazów z około 13 ms – YOLO rozpoznaje obrazy w sposób nieznacznie wolniejszy, jak ludzie. Tak więc, daje to maszynom możliwość wręcz natychmiastowego wykrywania obiektów. Dodatkowo jedną z zalet YOLO jest utrzymanie wysokiego poziomu wskaźnika mAP (z ang. *mean Average Precision*) czyli średniej precyzji. mAP porównuje prawdziwe pole ograniczające do pola ograniczającego z predykcją i zwraca wynik. Im wyższy wynik, tym bardziej dokładny jest model w swojej detekcji.

Pierwsza generacja YOLO korzysta z konwolucyjnej sieci neuronowej o nazwie Darknet, składającej się z 24 warstw konwolucyjnych zakończonych 2 warstwami wierzchnimi. Zadaniem warstw konwolucyjnych jest ekstrakcja cech ze zdjęć zaś warstwy wierzchnie przewidują wyjściowe prawdopodobieństwa i współrzędne obiektów. Na wejściu sieć pobiera obrazek o rozmiarach 448px na 448px, przesyłając następnie dane poprzez 24 warstwy konwolucyjne. Końcowym etapem warstw konwolucyjnych jest tensor o rozmiarach (7, 7, 1024). Dzięki wykorzystaniu 2 warstw wierzchnich jako formę regresji liniowej, sieć zwraca tensor o wymiarach (7, 7, 30).

Poszczególne elementy systemu detekcji obiektów YOLO zostały wkomponowane w jedną sieć neuronową. Wszystkie pola ograniczające są jednocześnie przewidywane dla każdej z klas



Rysunek 2.6. Architektura Darknet

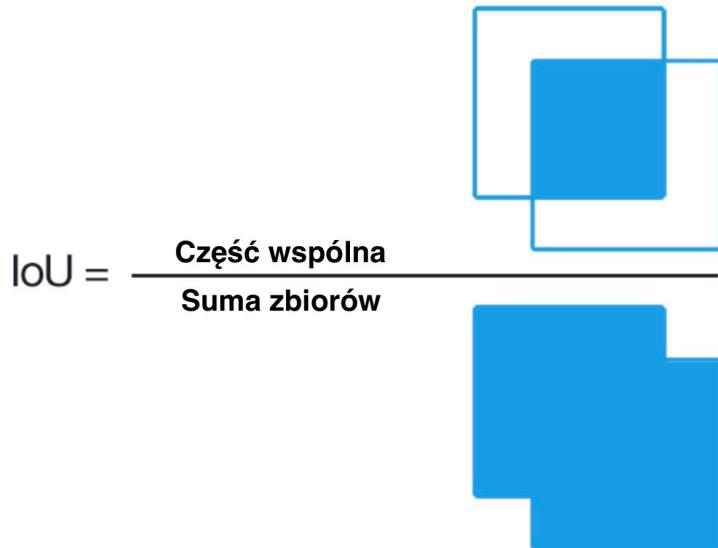
Źródło: Redmon, Divvala i in. (2016b).

obiektów. System YOLO dzieli obraz wejściowy na siatkę $S \times S$, gdzie S oznacza wielkość nakładanej siatki. Jeśli środek obiektu znajduje się w komórce siatki, to ta komórka siatki jest odpowiedzialna za wykrycie tego obiektu. Każda komórka siatki przewiduje B pól ograniczających (z ang. *bounding boxes*) i prawdopodobieństwo przynależności obiektu do określonej klasy. Prawdopodobieństwa te odzwierciedlają jak pewien jest model, że dane pole zawiera przedmiot i jak dokładne jest jego zdaniem pole ograniczające, które przewiduje. Prawdopodobieństwa te formalnie zapisujemy

$$\Pr(\text{Klasa}_i|\text{Obiekt}) \times \Pr(\text{Obiekt}) \times \text{IoU} = \Pr(\text{Klasa}_i) \times \text{IoU}, \quad (2.21)$$

gdzie $\Pr(\text{Klasa}_i|\text{Obiekt})$ stanowi prawdopodobieństwo, że dany obiekt należy do klasy_i, jeśli obiekt istnieje, $\Pr(\text{Obiekt}) \times \text{IoU}$ (z ang. *Intersection over Union*) jest prawdopodobieństwem oznaczenia obiektu, zaś $\Pr(\text{Klasa}_i) \times \text{IoU}$ jest prawdopodobieństwem znalezienia danej klasy i jest przedstawiony na Rysunku 2.7. Wyniki pokazują prawdopodobieństwo danej klasy w polu oraz jak do dokładnie pasuje przewidywane pole do obiektu.

Licznik powyższego wzoru stanowi obszar nakładania się prostokąta ręcznie nałożonego na obiekt i przewidywanego prostokąta ograniczającego. Mianownik zaś to obszar połączenia przewidywanego pola ograniczającego, jak i polem ograniczającym z ręcznie nałożonym polem ograniczającym.



Rysunek 2.7. Część wspólna przez sumę zbiorów – decyzja o wyborze określonego pola ograniczającego (bounding box)

Źródło: Opracowanie własne.

Podsumowując, wskaźnik IoU mierzy, jak blisko jest przewidywane pole ograniczające do prawdziwego pola ograniczającego. IoU waha się między 0, a 1, gdzie 1 jest wynikiem optymalnym. Gdy współczynnik IoU jest większy niż próg, wówczas ramka jest klasyfikowana jako pozytywna, ponieważ otacza dany obiekt. W przeciwnym razie ramka oznaczana jest jako negatywna.

Jeśli w danej komórce nie ma żadnego obiektu, wynik *confidence* powinien wynosić zero. W przeciwnym razie chcemy, aby wynik *confidence* był równy dla IoU między przewidywanym polem ograniczającym, a rzeczywistym polem ograniczającym.

W przypadku każdego pola ograniczającego przewidywane jest 5 elementów: x , y , w , h , i *confidence*. Współrzędne (x , y) reprezentują środek pola w stosunku do granic komórki siatki. Szerokość i wysokość są przewidywane w stosunku do całego obrazu. Ostatecznie predykcja *confidence* reprezentuje IoU pomiędzy przewidywanym polem ograniczającym, a polem wyznaczonym przez użytkownika.

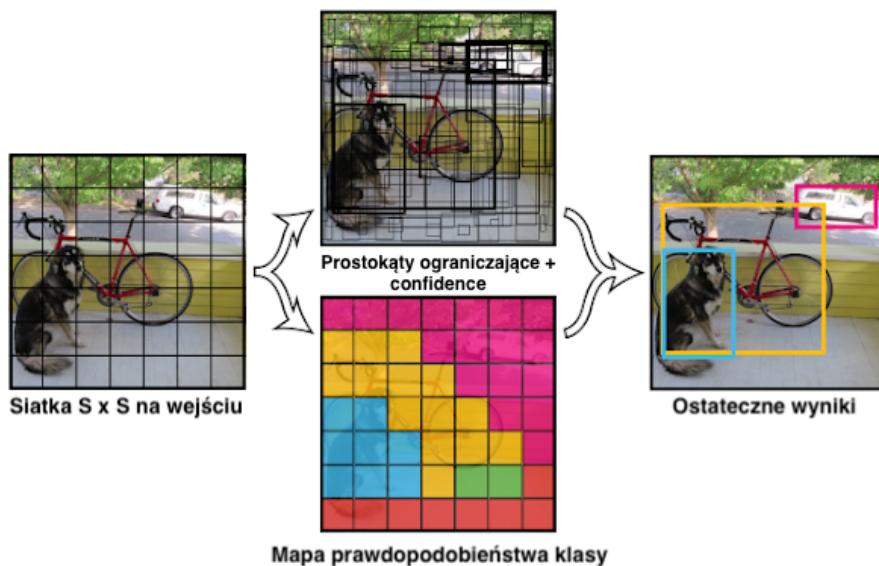
Każda komórka siatki przewiduje również klasę warunkową C , której formuła wygląda następująco:

$$\Pr(\text{Klasa}_i \mid \text{Obiekt}) \quad (2.22)$$

Prawdopodobieństwa te są uwarunkowane na komórce siatki zawierającej dany obiekt. Przewidywana jest tylko jedna klasa obiektu na komórkę siatki, niezależnie od liczby pól ograniczających B. Przemnożenie prawdopodobieństwa klas warunkowych i przewidywanych miar pewności (ang. *confidence*) poszczególnych pól dzięki czemu uzyskamy specyficzne dla danej klasy wyniki pewności dla każdego pola.

$$\Pr(\text{Klasa}_i \mid \text{Obiekt}) \times \Pr(\text{Obiekt}) \times \text{IoU}_{\text{predykcja}}^{\text{prawda}} = \Pr(\text{Klasa}_i) \times \text{IoU}_{\text{predykcja}}^{\text{prawda}} \quad (2.23)$$

Wyniki te kodują zarówno prawdopodobieństwo pojawienia się danej klasy w pudełku, jak i to, jak dobrze pudełko przewidywane pasuje do ręcznie nałożonego pola.



Rysunek 2.8. Przykładowe działanie YOLO

Źródło: <https://arxiv.org/pdf/1506.02640v5.pdf>.

W celu wyliczenia wcześniej wspomnianego wskaźnika mAP należy najpierw policzyć macierz błędu.

W celu przekształcenia wyników prawdopodobieństw w etykiety klas, używany jest tzw. próg (z ang. *threshold*). Gdy wynik jest równy lub wyższy od progu, próbka jest klasyfikowana jako dana klasa, w przeciwnym wypadku inna klasa jest przypisana obiekowi. Po przekształceniu prawdopodobieństw w konkretne klasy, można utworzyć tzw. macierz błędu.

		Klasa rzeczywista	
		pozytywna	negatywna
Klasa predykowana	pozytywna	prawdziwie pozytywna (TP)	fałszywie pozytywna (FP)
	negatywna	fałszywie negatywna (FN)	prawdziwie negatywna (TN)

Rysunek 2.9. Macierz błędu

Źródło: [https://pl.wikipedia.org/wiki/Tablica_pomyłek](https://pl.wikipedia.org/wiki/Tablica_pomy%C5%82ek).

Kolejnymi metryki koniecznymi do obliczenia jest precyza (ang. *precision*) i czułość (ang. *recall*). Precyza jest obliczana jako stosunek liczby prawidłowo zaklasyfikowanych próbek (prawdziwie pozytywnych) do całkowitej liczby próbek należących do danej klasy. Precyza mierzy dokładność modelu w klasyfikowaniu próbki jako pozytywnej i jest wyrażana następującym wzorem:

$$\text{Precyza} = \text{TP}/(\text{TP} + \text{FP}). \quad (2.24)$$

gdzie TP to prawdziwie pozytywne przypadki, a FP to fałszywie pozytywne przypadki z tablicy pomyłek

który kończy swoje działanie jak zostanie spełniony warunek stopu. Zwykle za warunek stopu przyjmuje się

Precyza jest wysoka jeżeli:

- Model dokonuje wielu poprawnych klasyfikacji Pozytywnych (maksymalizacja prawdziwie pozytywnych (TP)),
- Model sprawia, że mniej jest błędnych klasyfikacji pozytywnych (minimalizacja klasyfikacji fałszywie pozytywnych (FP)).

Podsumowując, precyza odzwierciedla, jak wiarygodny jest model w klasyfikowaniu próbek jako pozytywne. Wartości opisujące zdolność testu do wykrycia badanej cechy to czułość. Dokładniej jest to stosunek wyników prawdziwie pozytywnych do sumy prawdziwie pozytywnych (TP) i fałszywie negatywnych (FN). Stuprocentowa czułość w przypadku testu diagnostycznego

oznaczałaby, że wszystkie osoby chore zostałyby rozpoznane. Zatem czułość jest miernikiem ukazującym zdolność rozpoznawania przykładowo chorób tam, gdzie one występują. Ostatecznie, czułość wyrażana jest następującym wzorem:

$$\text{Czulosc} = \text{TP}/(\text{TP} + \text{FN}). \quad (2.25)$$

Decyzja o zastosowaniu precyzyji lub czułości zależy od rodzaju rozwiązywanego problemu. Jeżeli celem jest wykrycie wszystkich próbek pozytywnych (bez zwracania uwagi na to, czy próbki negatywne zostaną błędnie zaklasyfikowane jako pozytywne), wówczas należy zastosować czułość. Należy stosować precyzyję, jeżeli problem jest wrażliwy na zaklasyfikowanie próbki jako ogólnie pozytywnej, łącznie z próbками negatywnymi, które zostały fałszywie zaklasyfikowane jako pozytywne.

Ze względu na znaczenie zarówno precyzyji, jak i czułości, istnieje krzywa precyzyja-czułość, która pokazuje kompromis pomiędzy wartościami dla powyższych wskaźników z uwzględnieniem różnych progów. Krzywa ta pomaga wybrać najlepszy próg, aby zmaksymalizować obie metryki.

Potrzebne są następujące dane wejściowe do utworzenia krzywej precyzyji-czułości:

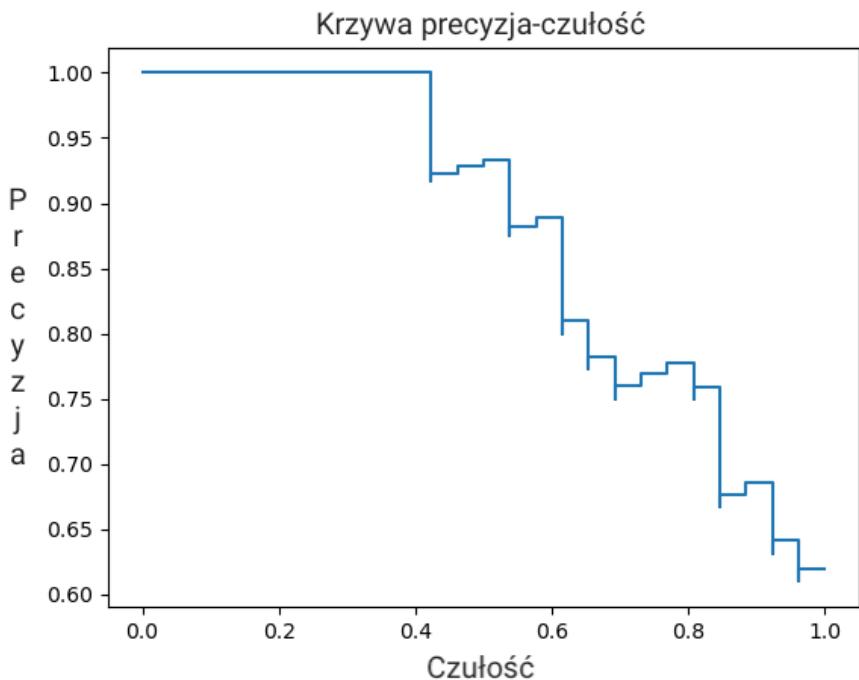
- prawdziwe pola ograniczające obiekty,
- predykcja prawdopodobieństw,
- progi do przeliczania wyników predykcji na etykiety klas.

Na podstawie prawdopodobieństw oraz konkretnych progów wyliczana jest precyzyja i czułość, na podstawie których można wyznaczyć krzywą. W celu wyznaczenia optymalnych wartości oblicza się wartość f_1 wyrażonej następującym wzorem:

$$f_1 = 2 \frac{\text{Precyza} \times \text{Czulosc}}{\text{Precyza} + \text{Czulosc}}. \quad (2.26)$$

Metryka f_1 mierzy równowagę pomiędzy precyzyją, a czułością. Gdy wartość f_1 jest wysoka, oznacza to, że zarówno precyzyja, jak i czułość są wysokie. Niższa wartość f_1 oznacza większą nierównowagę między precyzyją a czułością. Rysunek 2.10 przedstawia przykładową krzywą precyzyji i czułości wraz z wskaźnikiem f_1 .

Średnia precyza (AP, ang. *Average Precision*) jest sposobem na podsumowanie krzywej precyzyja-czułość w jedną wartość reprezentującą średnią wszystkich precyzyji. Za pomocą pętli,



Rysunek 2.10. Krzywa precyza-czułość

Źródło: Opracowanie własne.

która przechodzi przez wszystkie precyzje/czułości, obliczana jest różnica między bieżącą i następną czułością, a następnie mnożona przez bieżącą precyzję. Innymi słowy, AP jest ważoną sumą precyzji na każdym progu, gdzie wagą jest wzrost czułości.

$$\begin{aligned} AP &= \sum_{k=0}^{k=n-1} [Czulosc(k) - Czulosc(k+1)] \times \text{Precyza}(k), \\ Czulosc(n) &= 0, \quad \text{Precyza}(n) = 1, \\ n &= \text{Liczba progów}. \end{aligned} \tag{2.27}$$

W celu obliczenia mAP, należy obliczyć AP dla każdej klasy obiektów. Średnią wartością AP dla wszystkich klas jest mAP. Posiadając wartości AP dla każdej z klas można obliczyć mAP na podstawie poniższego wzoru:

$$\begin{aligned} mAP &= \frac{1}{n} \sum_{k=1}^{k=n} AP_k, \\ AP_k &= \text{AP danej klasy } k, \\ n &= \text{liczba klas obiektów}. \end{aligned} \tag{2.28}$$

Zagadnienie detekcji obiektów zostało uznane jako problem regresji względem pól ograniczających (ang. *bounding boxes*) i związanych z nimi prawdopodobieństw klasyfikacyjnych.

Detekcję obiektów kadrujemy jako pojedynczą regresję, począwszy od pikseli obrazu, a skończywszy na ograniczeniu pól koordynacji i prawdopodobieństw klasowych. Korzystając z naszego systemu, tylko raz (YOLO) patrzysz na obraz, aby przewidzieć, jakie obiekty są obecne i gdzie się znajdują.

Czwarta generacja YOLO korzysta z CSPDarknet-53 czyli ogólnodostępnej konwolucyjnej sieci neuronowej składającej się z 53 warstw.

Dzięki implementacji modelu Darknet w CUDA, czas predykcji modelu skraca się w sposób istotny. Yolo v4 to już czwarta generacja systemów wykrywania obiektów. W przypadku systemu YOLO, zagadnienie detekcji obiektów zostało uznane jako problem regresji względem pól ograniczających (ang. *bounding boxes*) i związanych z nimi prawdopodobieństw klasyfikacyjnych. Pojedyncza sieć neuronowa wstępnie określa pola ograniczające i prawdopodobieństwa klasowe całego obrazka na podstawie jednej iteracji sieci. Ponieważ cały system detekcji jest oparty na pojedynczej sieci neuronowej, można go zoptymalizować bezpośrednio pod kątem wydajności wykrywania.

2.3.1 YOLO v4

Główną zaletą systemu YOLO nad innymi systemami wykrywania obiektów jest szybkość predykcji. W przypadku czwartej wersji systemu, YOLO jak podaje Bochkovskiy i in. (2020) jest w stanie generować predykcje z prędkością 65 FPSów (ang. *frames per seconds*) czyli klatek na sekundę oraz średnią precyzją na poziomie 43,5% dla zbioru MS COCO. Im większa wartość FPSów tym więcej zdjęć może dany model rozpoznać. Częstotliwość klatek na sekundę w YOLO odpowiada czasowi detekcji wynoszącemu około od 6 do 22 ms (milisekund).

Porównując to z czasem potrzebnym ludzkiemu mózgowi na wykrycie obrazów z około 13 ms - YOLO rozpoznaje obrazy w podobny sposób, jak ludzie. Tak więc, daje to masynom możliwość natychmiastowego wykrywania obiektów.

2.3.2 Architektura YOLO v4

Rysunek 2.12 ukazuje dokładny opis architektury systemu rozpoznawania obiektów YOLO v4.

Obraz wejściowy jest przeprocesowany przez 4 bloki:

- Szkielet systemu (ang. *backbone*),
- Szyja (ang. *neck*),
- Gęsta predykcja (ang. *Dense Prediction*),
- Rzadka predykcja (ang. *Sparse Prediction*).

Szkielet systemu (ang. *backbone*) odnosi się tutaj do architektury ekstrakcji cech. W przypadku YOLO v4, szkieletem jest CSPDarknet53. Skrót CSP (z ang. *Cross-Stage-Partial*), który odnosi się do podzielenia warstwy na dwie części, jedna, która przejdzie przez blok konwolucyjny, a druga od razu do kolejnej warstwy. Celem bloku szyi (ang. *neck*) jest dodanie dodatkowych warstw pomiędzy szkieletem, a blokiem gęstej predykcji. Kolejny blok używany jest do lokalizacji pól ograniczających oraz klasyfikacji klasy obiektów. Autorzy pracowali nad technikami mającymi na celu poprawę dokładności modelu podczas uczenia i ewaluacji. Efektem są BoF (z ang. *Bag-Of-Freebies*) czyli zestaw technik, które pomagają podczas uczenia modelu, nie przedłużając czas nauczania. Niektóre popularne techniki obejmują zwiększenie próbki danych, losowe kadrowanie czy cieniowanie. Istnieją dwie grupy BoF:

- BoF dla szkieletu (CutMix, Mosaic data augmentation, DropBlock regularization czy Class label smoothing),
- BoF dla detекторa (CIoU-loss, CmBN, DropBlock regularization czy Mosaic data augmentation).

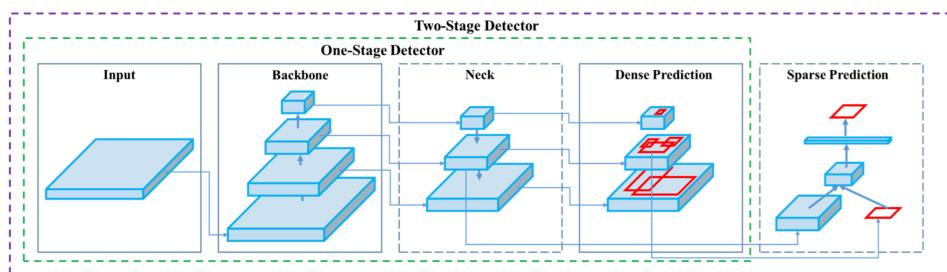
2.3.3 Zestaw technik dla szkieletu

Główną ideą CutMix jest łączenie obrazów poprzez wycinanie części z jednego obrazu i wklejanie ich w nowo utworzony obraz. Wycięcia w obrazie zmuszają model do nauki przewidywania w oparciu o konkretną ilość cech. W CutMix wycięcie jest zastępowane częścią innego obrazu wraz prawdziwym polem ograniczającym. Stosunek każdego obrazu jest ustalany w procesie generowania obrazu (np. 0,4/0,6). Na 3.10 można zobaczyć, jak autorzy CutMix demonstrują, że ta technika może działać.



Rysunek 2.11. Sposób działania CutMix

Źródło: Opracowanie własne na podstawie: <https://arxiv.org/pdf/1905.04899.pdf>.



Rysunek 2.12. Architektura YOLO v4

Źródło: <https://arxiv.org/pdf/2004.10934.pdf>.

2.4 Podsumowanie

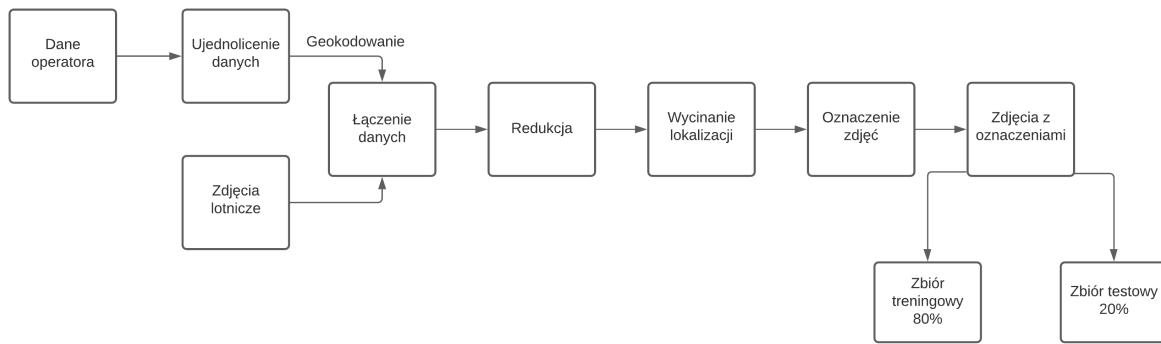
Przez ostatnie kilka lat dziedzina zajmująca się uczeniem głębokim przeżyła rewolucję. Ze względu na coraz to większe zasoby danych zagadnienie uczenia głębokiego powinno być coraz częściej brane pod uwagę jako możliwe narzędzie do rozwiązania problemu. Jednym z takich narzędzi może być algorytm Yolo w wersji czwartej służący do detekcji obiektów. Jednakże aby można było wykorzystać ten algorytm należy odpowiednio przygotować dane oraz środowisko. Szczegóły omówione zostaną w kolejnym rozdziale.

Rozdział 3

Przygotowanie danych i środowiska na potrzeby detekcji paneli i kolektorów słonecznych

3.1 Schemat przetwarzania danych

Proces przygotowania danych na potrzeby niniejszej pracy był wieloetapowy. W pierwszym etapie zostały otrzymane dane od operatora, które następnie zostały ujednolicone. Dalej, na podstawie współrzędnych geograficznych z pliku zostały pobrane ortofotomapy. Ze wszystkich możliwych ortofotomap zostały pobrane tylko te, które pochodzą z Poznania i ze Skórzewa dla 2018 i 2020 roku. W celu zmniejszenia ilości danych przekazywanych do trenowania modelu, wycięte zostały jedynie działki, na których znajdowały się panele. W kolejnym etapie ręcznie zaznaczono panele słoneczne, aby ukazać pozytywne przypadki. W ostatnim etapie przed trenowaniem modelu, wszystkie zdjęcia podzielono na dwie grupy, pierwszy zbiór składający się z 80% wszystkich zdjęć został użyty do trenowania modelu, zaś pozostałe 20% służyły do waliadacji modelu. W sekcji 3.2 opisano strukturę danych wejściowych, w sekcji 3.3 przedstawiono proces oznaczania zdjęć, a w sekcji 3.4 opisano konfigurację środowiska Colab na potrzeby detekcji obiektów z wykorzystaniem YOLO v4.



Rysunek 3.1. Schemat przygotowania danych

Źródło: Opracowanie własne.

3.2 Dane źródłowe

3.2.1 Lokalizacja instalacji fotowoltaicznych

W ramach pracy magisterskiej została podjęta współpraca z jednym z operatorów systemów dystrybucyjnych w ramach, której operator dostarczył przybliżoną lokalizację instalacji fotowoltaicznych swoich klientów na terenie województwa Wielkopolskiego. Otrzymany plik zawierał ponad 20 tys. instalacjach fotowoltaicznych zgodnie ze stanem na koniec 2020 roku. Przesłany plik w formacie `xlsx` zawierał następujące informacje:

- unikalny identyfikator instalacji,
- miasto,
- przybliżona szerokość geograficzna,
- przybliżona długość geograficzna.

Tablica 3.1. Liczba instalacji na koniec 2020 roku

Miasto	Liczba instalacji
Poznań	1 956
Kościan	448
Leszno	446
Piła	296
Luboń	285

Źródło: Opracowanie własne.

W Tabeli 3.1 zawarte są informacje o liczbie instalacji według miast. Na koniec 2020 roku najwięcej ich było w Poznaniu, następnie w Kościanie, Lesznie, Pile i Luboniu.



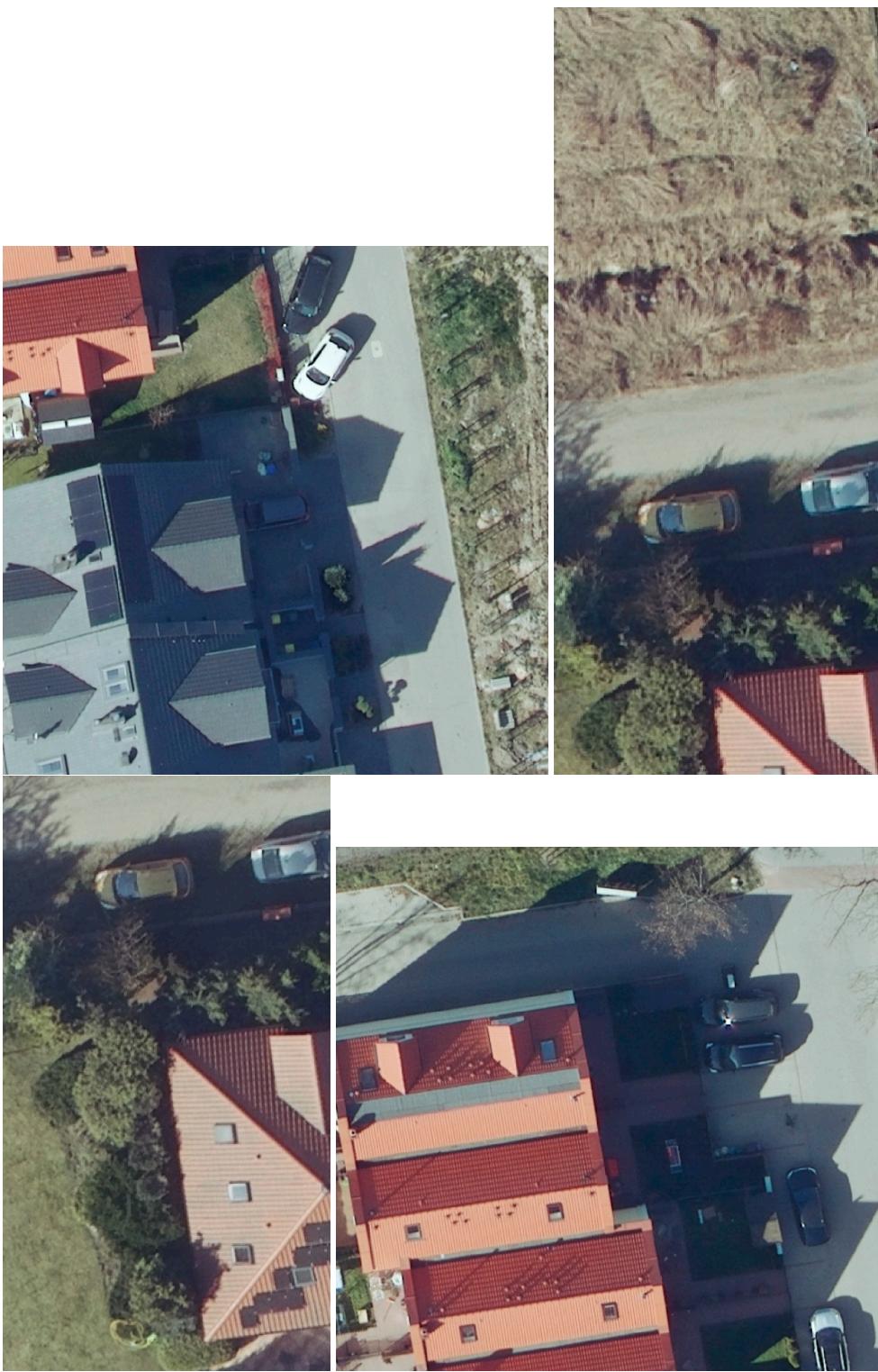
Rysunek 3.2. Rozkład instalacji fotowoltaicznej w Poznaniu (lewy) oraz Skórzewie (prawy)

Źródło: Opracowanie własne.

Rysunek 3.2 ukazuje rozkład instalacji względem miasta Poznania oraz Skórzewa. W przypadku miasta Poznania najwięcej paneli znajduje się w dzielnicach na obrzeżach Poznania takich jak Kiekrz, Strzeszyn, Krzyżowniki czy Szczepankowo, które charakteryzują się zabudową domów jednorodzinnych. Z drugiej strony dzielnice takie jak Wilda, Główna czy Rataje posiadają jedynie kilka instalacji ze względu na gęstą zabudowę wielorodzinną.

Rysunek 3.3 różne przykłady poprawnie i niepoprawnie dostarczonej lokalizacji. Zdjęcie w *lewym górnym rogu* ukazuje przykład poprawnie dostarczonej lokalizacji przez operatora co skutkowało dokładnym wycięciem dachu wraz z instalacją fotowoltaiczną.

W przypadku *prawego górnego* zdjęcia współrzędne nie tylko znalazły się na skraju ortofotomapy przez co samo wycięcie dachu nie jest kwadratowe, a prostokątne, ale również nie wskazały na istnienie instalacji fotowoltaicznej. Po dokładniejszej analizie pliku źródłowego udało się znaleźć instalację, którą chodziło operatorowi. Docelowa instalacja znalazła się kilkanaście metrów dalej, jednak ze względu na niedokładne współrzędne nie została wycięta co widać na *dolnym lewym zdjęciu*. W przypadku *prawego dolnego zdjęcia* nie ma instalacji fotowoltaicznej na wyciętym dachu oraz w pobliżu, może być to związane z faktem, że dana ortofotomapa mogła zostać utworzona we wcześniejszym terminie niż sama instalacja.



Rysunek 3.3. Wycięty dach wraz z instalacją fotowoltaiczną (lewy górny), wycięty dach bez instalacji fotowoltaicznej (prawy górny), faktyczne położenie instalacji fotowoltaicznej (lewy dolny) i przykład wyciętego dachu bez instalacji fotowoltaicznej w pobliżu (prawy dolny)

Źródło: Opracowanie własne.

3.2.2 Zdjęcia lotnicze

W otrzymanym pliku od znalazły się współrzędne geograficzne, które zostały następnie wykorzystane do pobrania ortofotomap. W tym celu zastosowano nowy pakiet w języku R o nazwie Dyba i Nowosad (2021)¹. Zawiera on wiele funkcji, które umożliwiają dostęp do zawartości udostępnionej przez GUGiK, w tym ortofotomapy. Do tego celu służy funkcja `ortho_request`, która ma jeden argument będący obiektem klasy `sf` (Pebesma 2018).

W pracy wykorzystano ramkę danych zawierającą współrzędne geograficzne poszczególnych lokalizacji. Następnie w pętli należało dla każdego punktu wywołać funkcję `ortho_request` oraz ograniczyć uzyskane wyniki do określonej jakości. Skrypt pobierający dane przedstawiony jest w Programie 3.1.

```
koscian_req <- data.frame()  
1  
2  
for (row in 1:nrow(data)) {  
3  
  data_ready <- data.frame(data[row, ] %>%  
4    st_buffer(., dist = 10) %>%  
5    bind_cols(ortho_request(.)) ) %>%  
6    filter(resolution=='0.05',  
7      composition=='RGB',  
8      CRS=='PL-2000:S6',  
9      year=='2020')  
10  koscian_req <- rbind(koscian_req, data_ready)  
11}  
12
```

Program 3.1. Przykład pobrania ortofotomapy z wykorzystaniem pakietu `rgugik` z rejestru GUGiK

Program 3.1 służył do pobierania ortofotomap na podstawie współrzędnych geograficznych. Do każdych współrzędnych geograficznych za pomocą funkcji `st_buffer` z pakietu `sf`, dodawane było 10 metrów jako bufor bezpieczeństwa. Za pomocą funkcji `ortho_request` wysyłane zostało zapytanie do GUGiK, aby otrzymać ortofotomapę.

Następnie zostały odfiltrowane wyniki tak, aby wyniki pochodziły z 2020 roku, wraz z rozdzielcością równą 5 cm na mapie = 1 piksel na zdjęciu. Po pobraniu informacji i zapisaniu do nowej zmiennej, otrzymana została ramka danych z 11 nowymi kolumnami. Do najważniejszych należą: URL do zdjęcia czy unikatowa nazwa pliku.

W kolejnym kroku został przygotowany kod, za pomocą którego zostały pobrane wszystkie pliki TIF (ang. *Tagged Image File Format*, do których prowadziły linki w odpowiednio przygotowanej ramce danych. Każdy został zapisany w następującym formacie: *identyfikator_miasto_rok*. Pobrane pliki były w formacie TIF, który to jest komputerowym formatem plików do przechowywania obrazów grafiki rastrowej, popularny wśród grafików, branży wydaw-

¹Szczegóły dotyczące pakietu można znaleźć na stronie <https://kadyb.github.io/rgugik/>

niczej i fotografów. Dla samego Poznania zostało pobranych 1 380 zdjęć dla danych z dwóch różnych lat (2018 i 2020), gdzie wielkość każdego pojedynczego pliku wyniosła około 30 MB (5000 na 8000 pikseli). Oznacza to, że wielkość wszystkich pobranych zdjęć wyniosła około 41 GB. Rozdzielcość ortofotomap była bardzo wysoka, 5 cm w terenie odpowiadało 1 pikselowi na zdjęciu. Każdy plik stanowił ortofotomapę terenu 250 metrów na 400 metrów. Przetwarzanie całych zdjęć byłoby bardzo czasochłonne dlatego zdecydowano się na wycięcie poszczególnych dachów względem podanych współrzędnych geograficznych przez operatorów. W teorii na każdym pobranym obrazku znajdowała się co najmniej jedna instalacja fotowoltaiczna.

Procedura wycinania budynków na podstawie współrzędnych wyglądała następująco:

- wgranie pojedynczego zdjęcia do RStudio za pomocą funkcji `rast` z pakietu `terra` (Hijmans 2021),
- znalezienie współrzędnych geograficznych na podstawie nazwy pliku zdjęcia, który zawierał unikatową nazwę instalacji fotowoltaicznej dostarczonej przez dostawcę,
- przeliczanie współrzędnych pomiędzy układami, tak aby nie było różnic między ortofotomapą, a współrzędnymi z pliku od operatora,
- poprzez funkcję `buffer` z pakietu `terra`, oblicz bufor wokół wszystkich komórek (w tym wypadku 15 metrów),
- wycięcie dachu na podstawie wcześniej obliczonego buforu.

Skrypt do wycinania fragmentu ortofotomapy zawierającej panel lub kolektor słoneczny zawiera Program 3.2.

```

1  for (i in orto_2020){
2      #wczytanie pliku
3      r <- rast(i)
4      #wczytanie tych wierszy od operatora, których nazwa pojawiła się w~plikach
5      to_check <- data %>%
6          select(-geometry) %>%
7          dplyr::filter(str_detect(URL, paste(path_file(i), collapse = "|")))) %>%
8          select(id)
9
10     #Pobranie długości i~szerokości geograficznej punktów
11     ppe <- instalacje %>%
12         filter(Numer.PPE %in% to_check$id) %>%
13         select(Numer.PPE, Długość.geograficzna,
14                Szerokość.geograficzna) %>%
15         rename(lat = Szerokość.geograficzna,
16                lon = Długość.geograficzna)
17
18     #przeliczanie współrzędnych pomiędzy układami
19     pj <- proj4::project(ppe[,c(2,3)], proj4string, inverse=FALSE)
20
21     #zapisanie do pliku przeliczonych współrzędnych
22     lonlat <- data.frame(ppe = ppe$Numer.PPE,lon=pj$x,lat=pj$y)

```

```

#stworzenie wektora ze współrzędnych
p <- vect(as.matrix(lonlat[,2:3]),crs=crs(r))
lonlat$vect<-list(p)
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

#dodanie bufora do każdej współrzędnej
p_buffer <- buffer(lonlat$vect[[1]], width=15, capstyle="round")

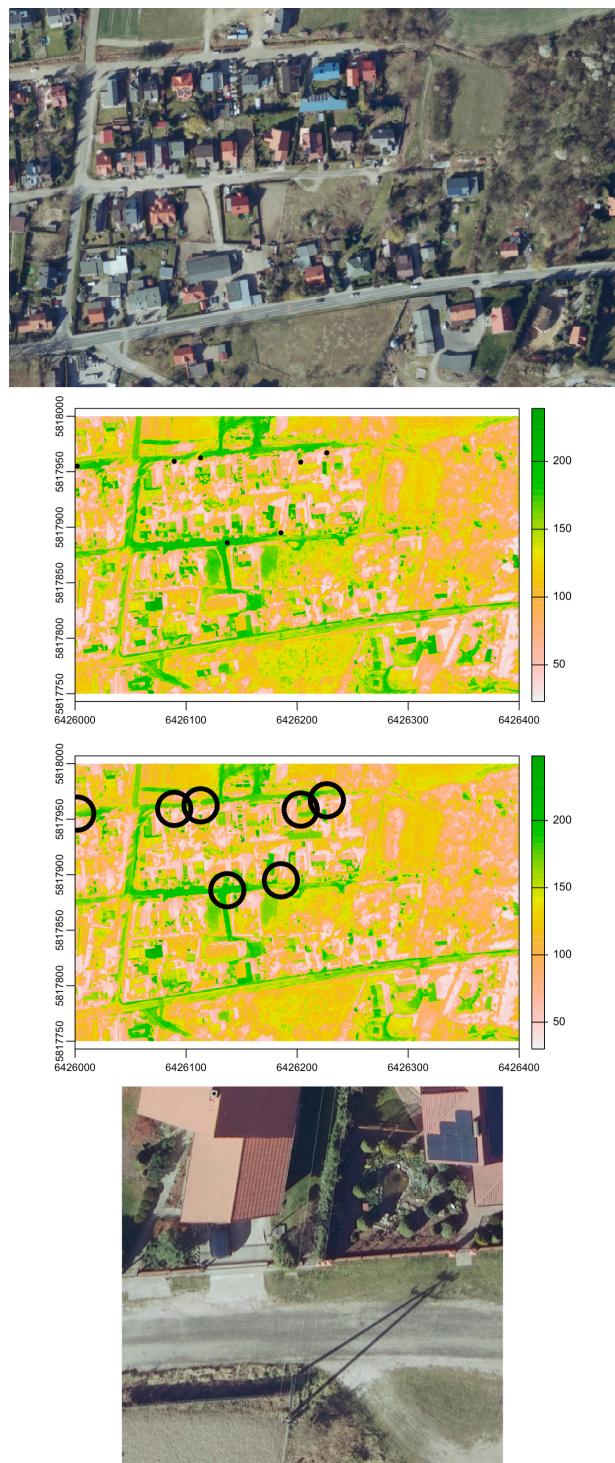
#wycięcie dachu dla każdej z instalacji i zapisaniu do osobnego pliku
for (i in 1:length(p_buffer)){
  r_c <- crop(r, p_buffer[i])
  writeRaster(r_c, paste(lonlat$ppe[i], '_poznan_2020.png', sep = ""),
              overwrite=TRUE)
  file.remove(paste(lonlat$ppe[i], '_poznan_2020.png.aux.xml', sep = ""))
}

```

Program 3.2. Wycinanie fragmentu ortofotomapy zawierającej panel lub kolektor

Wizualny skrót działania kodu został zaprezentowany na Rysunku 3.4. W pierwszym etapie do środowiska R wgrywana była ortofotomapa z danego regionu. Na podstawie nazwy wgrywanego pliku można było zidentyfikować jakie konkretnie instalacje fotowoltaiczne znajdowały się na wgrywanej ortofotomapie. Następnie przy wykorzystaniu pakietu proj4 oraz funkcji project dokonywano przekształcenia współrzędnych geograficznych do systemu PL-2000:S6, w jakich znajdująły się ortofotomapy. W przekształceniu pomogła strona <https://www.spatialreference.org/ref/epsg/2177/>, która zawierała wszystkie szczegółły do wykorzystania pakietu proj4. W ramach nowej ramki danych zapisano unikalne identyfikatory instalacji fotowoltaicznych wraz z przekształconymi współrzędnymi geograficznymi. Za pomocą funkcji vect z pakietu terra utworzony został wektor przestrzenny, który został dodany do unikalnego identyfikatora paneli. Ze względu na niedokładne dane otrzymane od operatora, do wszystkich współrzędnych dodano bufor o wielkości 15 metrów, aby zwiększyć prawdopodobieństwo uwzględnienia potencjalnej instalacji. W zmiennej p_buffer przechowywano wszystkie punkty z potencjalnymi instalacjami fotowoltaicznymi, dla których następnie iteracyjnie były wycinane kwadraty. Wycięte zdjęcia były następnie zapisywane względem unikalnego identyfikatora operatora, nazwy miasta oraz roku pochodzenia danej ortofotomapy. Wycięte zdjęcia były przechowywane w formacie PNG (ang. *Portable Network Graphics*).

Niestety dane dostarczone przez operatora, a w szczególności współrzędne geograficzne były niedokładne. Często wskazywały miejsca, w których nie było ani dachów, ani paneli słonecznych. W ramach Programu 3.2 udało się wyciąć 1958 lokalizacji, ostatecznie jedynie 408 instalacji fotowoltaicznych zostało znalezionych. W celu zbalansowania próbki i dostarczenia negatywnych przykładów do modelu głębokiego uczenia, liczba zdjęć, gdzie nie było paneli została zredukowana do 408 sztuk, tak aby stanowiła 50% całego zbioru.



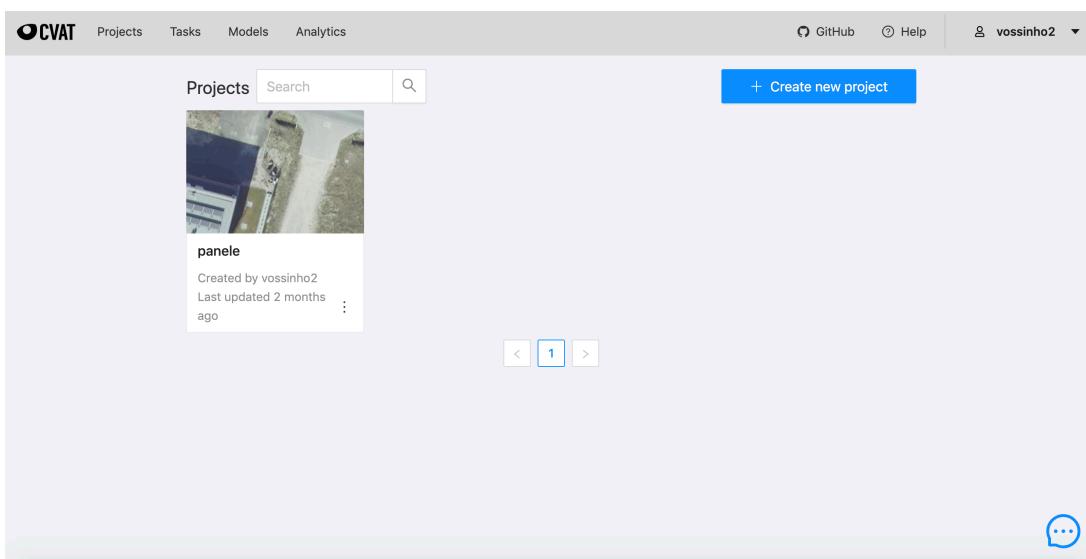
Rysunek 3.4. Wizualizacja procedury wycinania dachów z ortofotomap na podstawie danych dostarczonych przez operatora

Źródło: Opracowanie własne.

3.3 Oznaczanie paneli i kolektorów na zdjęciach

Wykorzystywany algorytm do uczenia (YOLO v4) wymaga podania ramek ograniczających. Ramki ograniczające powinny jak najdokładniej wskazywać umiejscowienie poszukiwanych obiektów, tak aby model mógł wzorować się na poprawnych danych. W celu oznaczenia zdjęć została wykorzystana strona cvat.org. CVAT został zaprojektowany, aby zapewnić użytkownikom zestaw wygodnych narzędzi do anotacji cyfrowych obrazów i filmów. Aktualnie jest używany przez takie firmy jak Intel do oznaczania swoich danych. Konieczne do oznaczania zdjęć jest utworzenie konta, a następnie projektu.

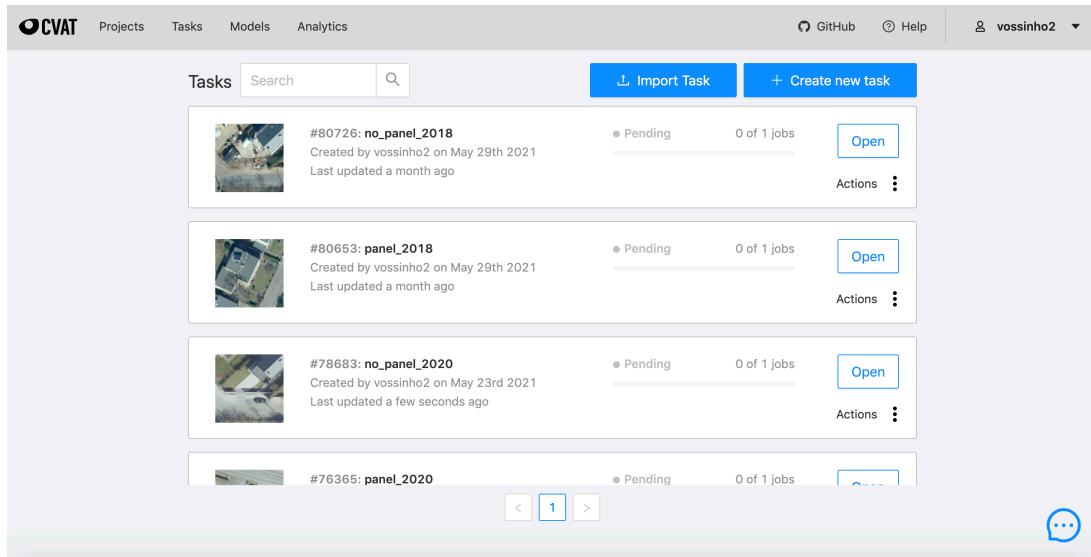
Użytkownik po wprowadzeniu swoich danych do logowania może utworzyć projekt. W ramach projektu ustalana jest nazwa całości oraz nazwa etykiet jakimi zdjęcia będą oznaczane tak jak jest to widoczne na Rysunku 3.5. W ramach konkretnego projektu dodawane są tzw. *tasks* czy zadania do wykonania. Zadanie w kontekście CVAT oznacza zbiór zdjęć, które muszą zostać oznaczone. W przypadku przygotowania danych do modelu uczenia maszynowego, utworzone zostały 4 zadania tak jak to widać na Rysunku 3.6 wraz z etykietą panel.



Rysunek 3.5. Wygląd CVAT po zalogowaniu

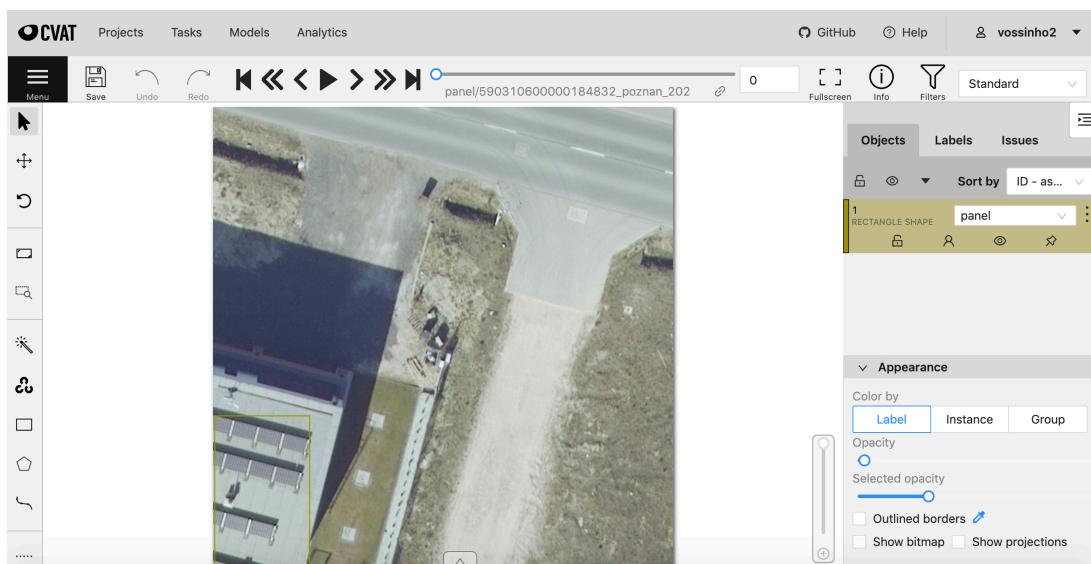
Źródło: Opracowanie własne.

Rysunek 3.7 prezentuje wygląd panelu, z którego oznaczane są zdjęcia. Założyciele narzędzia zadali o wygodę osoby oznaczającej i wprowadzili skróty klawiszowe, które znacznie przyspieszają oznaczanie zdjęć. Za pomocą klawisz *F* wybiera się kolejne zdjęcie, a przycisk *N* na klawiaturze uruchamia pole, którym użytkownik oznacza panele. Sam algorytm YOLO v4 wy-



Rysunek 3.6. Wygląd CVAT po utworzeniu zadań

Źródło: Opracowanie własne.

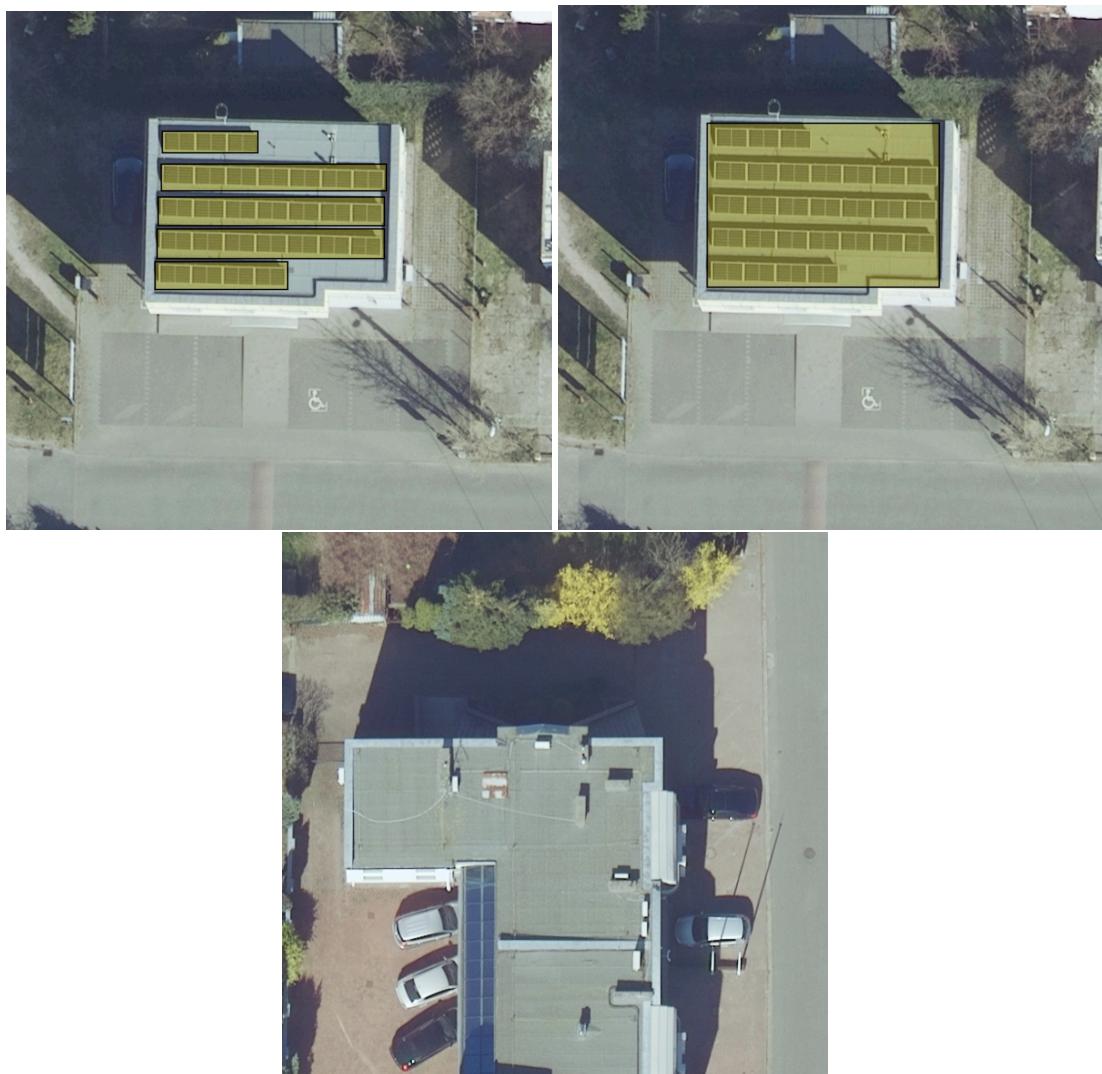


Rysunek 3.7. Wygląd CVAT w trakcie oznaczania

Źródło: Opracowanie własne.

maga przygotowanie pól ograniczających, które zostały zapisane przez nałożenie prostokąta na zdjęcie. Po oznaczeniu wszystkich pozytywnych przykładów należy pobrać adnotacje poprzez opcję "Dump annotations" i wybierając przy tym 1 z 19-stu możliwych formatów. Po wybraniu formatu zostaną pobrane skompresowane pliki, które posiadają te same nazwy co zdjęcia, dodatkowo w każdym pliku txt znajdują się wszystkie lokalizacje pozytywnych przykładów, które

znajdowały się na zdjęciach. Dla negatywnych przykładów, na których nie było instalacji fotowoltaicznych, należy wygenerować pusty plik.



Rysunek 3.8. Przykład zaznaczania instalacji fotowoltaicznej jako osobnych obiektów (lewy górny), przykład zaznaczenia instalacji fotowoltaicznej jako jednego obiektu (prawy górny), przykład dachu z cięko identyfikowaną instalacją (dolny)

Źródło: Opracowanie własne.

Sposób oznaczania paneli stanowi kluczowy element w przygotowaniu danych do modelu. Algorytm Yolo v4 wymaga od użytkownika stworzenia prawdziwych pól ograniczających za pomocą prostokątów. Prostokąty powinny jak najdokładniej oznaczać instalacje fotowoltaiczne, tak aby w danych było jak najmniej szumu. Rysunek 3.8 przedstawia technikę zaznaczania instalacji fotowoltaicznych. *Lewe górnne zdjęcie* przedstawia zaznaczenie całej instalacji fotowoltaicznej jako jednego obiektu. W przypadku tego podejścia problematyczne stają się duże instalacje,

których rozmiary mogą sięgać kilkudziesiąt metrów kwadratów. W przypadku wyboru takiego typu zaznaczania często dostarczane dane nie tylko zawierałyby same panele, ale również liczne okna, które wprowadzałyby szum. *Prawe górnne zdjęcie* pokazuje sposób zaznaczania paneli jaki został użyty w pracy magisterskiej. Obiekt identyfikowany jest jako nierożłaczne panele fotowoltaiczne. Takie podejście jest bardziej czasochłonne jednak znaczaco redukuje szum, który istniałby w przypadku pierwszego podejścia. W zbiorze przeznaczonym do oznaczenia znajdowały się również dachy, na których ciężko było rozpoznać panele. Przykładem takiej sytuacji jest *dolne zdjęcie* na Rysunku 3.8. Charakterystyczna budowa szklanego zadaszenia wyglądała łudząco podobnie do instalacji fotowoltaicznej. Ostatecznie przykład ten został wykluczony, aby nie wprowadzać szumu.

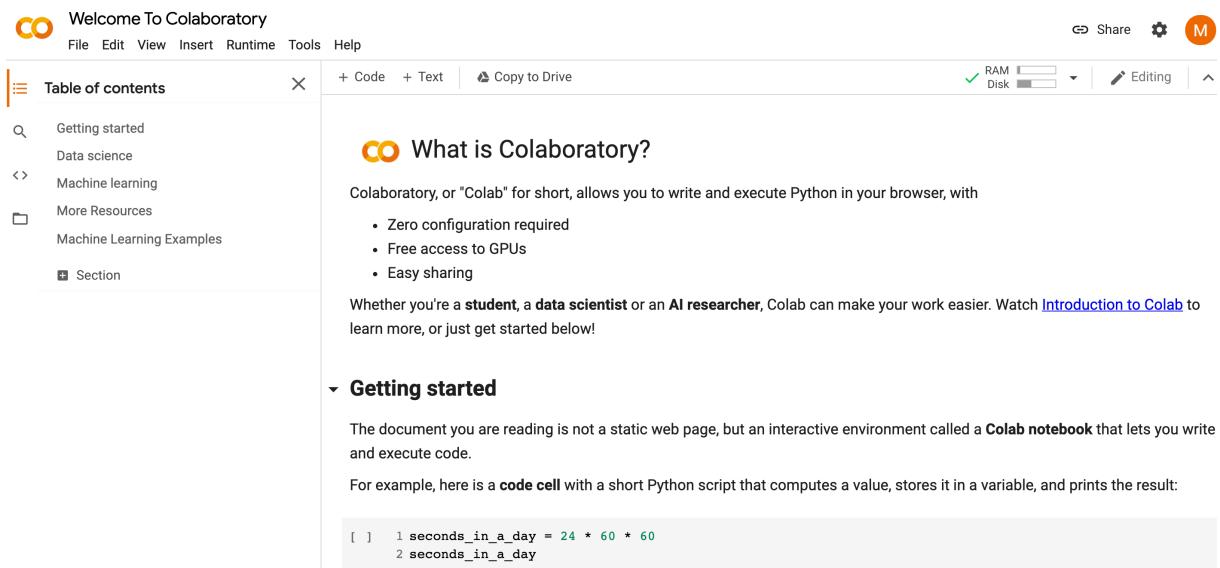
3.4 Przygotowanie środowiska pracy

3.4.1 Infrastruktura

W ramach pracy magisterskiej wykorzystano dwa środowiska do obliczeń – prywatny laptop oraz Google Colab. Pierwsze środowisko zostało wykorzystane w ramach przygotowania danych. Maszyna lokalna posiadająca 8GB RAM oraz procesor Intel i5 została użyta do ściągnięcia ortofotomap, a następnie wycięcia poszczególnych lokalizacji na bazie pliku dostarczonego przez OSD.

Na potrzeby stworzenia modelu wykorzystano infrastrukturę Google Colab². Google Colab zwany również Colaboratory, to środowisko umożliwiające uruchamianie kodu w języku Python czy R w przeglądarce. Do głównych zalet rozwiązania należy brak konieczności konfigurowania środowiska wirtualnego, dostęp do układów GPU (kart graficznych przyspieszających obliczenia) oraz łatwy dostęp (niezależność między systemami operacyjnymi). W ramach procesu trenowania modelu wykorzystano różne usługi Google Colab. Pierwsza z nich to darmowa usługa korzystania z układów GPU. W ramach darmowego konta użytkownicy mają dostęp do współdzielonych zasobów Google wraz z licznymi ograniczeniami takimi jak maksymalny czas działania kodu na poziomie 12 godzin, słabszy procesor (Intel(R) Xeon(R) CPU @ 2,20GHz) czy 12 GB RAM. Dla użytkowników wykupujących pakiet Colab Pro i płacących 10\$ miesięcznie czekają różne udoskonalenia takie jak: szybszy procesor (Intel(R) Xeon(R) CPU @ 2,30GHz), łatwiejszy

²Strona internetowa: <https://colab.research.google.com/>



Rysunek 3.9. Zrzut ekranu z serwisem Google Colab

Źródło: Opracowanie własne.

dostęp do mocniejszych kart graficznych czy 24GB RAM. Liczba kart graficznych jest ograniczona i składa się z następujących modeli:

Tablica 3.2. Dostępne układy graficzne w Google Colab [na dzień 6.07.2021]

Nazwa układu GPU	Liczba rdzeni CUDA	Pamięć w GB	Cena układu w zł
K80	4992	24	22 900
T4	2560	16	11 050
P4	2560	16	9 700
P100	3584	16	30 990

Źródło: Opracowanie własne.

Przedstawione w Tabeli 3.2 układy GPU są dostępne zarówno w darmowym koncie jak i płatnym planie, jednak w przypadku darmowego użycia dostępność układów GPU jest mocno ograniczona. Jak widać na podstawie Tabeli 3.2, koszt zakupu pojedynczej karty graficznej, której użycie może być nawet darmowe w Google Colab, jest kilkusetkrotnie wyższy niż płatna usługa do części infrastruktury posiadającej wyżej wymienione układ GPU. Posiadanie własnej infrastruktury niesie ze sobą liczne korzyści takie jak stały dostęp do mocy obliczeniowej, niezależność czy możliwość kontrolowania wykorzystania danego sprzętu. W przypadku jednak sporadycznego korzystania z infrastruktury w celu nauczenia kilku modeli, Google Colab spełni

swoją rolę lepiej niż własna infrastruktura. Zakup kart graficznych może zwrócić się w sytuacji ciągłego trenowania modeli i tworzenia setek predykcji.

3.4.2 Konfiguracja Google Colab

Chęć użycia układów GPU w trakcie uczenia modelu, wiąże się ze specjalnymi krokami potrzebnymi do podjęcia przed uruchomieniem fazy trenowania. Po utworzeniu nowego arkusza³ Google Colab należy wybrać typ środowiska wykonawczego. W tym celu należy wybrać z głównego menu "Środowisko wykonawcze", a następnie "Zmień typ środowiska wykonawczego". Po kliknięciu powinno wyskoczyć okno, w którym powinno wybrać się "GPU". Tak ustwiony notatnik będzie zoptymalizowany pod kątem uczenia głębokiego.

W ramach potwierdzenia, że układ GPU został przydzielony można uruchomić następujący kod: `!nvidia-smi`. Rezultat tej procedury został przedstawiony w Programie 3.3, który wskazuje m.in. jaka jest zainstalowana wersja CUDA (CUDA Version: 11,2) oraz jaką kartą graficzną zostanie wykorzystana na potrzeby obliczeń (Tesla T4).

+-----+ NVIDIA-SMI 470.42.01 Driver Version: 460.32.03 CUDA Version: 11.2 +-----+	1 2 3 4 5 6 7 8 9 10 11 12
GPU Name Persistence-M Bus-Id Disp.A Volatile Uncorr. ECC Fan Temp Perf Pwr:Usage/Cap Memory-Usage GPU-Util Compute M. MIG M. +-----+-----+-----+-----+-----+-----+-----+-----+ 0 Tesla T4 Off 00000000:00:04.0 Off 0% Default N/A 67C P8 12W / 70W 0MiB / 15109MiB +-----+-----+-----+-----+-----+-----+-----+-----+	13 14 15 16 17 18 19
+-----+ Processes: GPU GI CI PID Type Process name GPU Memory ID ID ID Usage +-----+-----+-----+-----+-----+-----+-----+ No running processes found +-----+-----+-----+-----+-----+-----+-----+	13 14 15 16 17 18 19

Program 3.3. Wydruk procedury `nvidia-smi` w serwisie Google Colab

Wykorzystanie poprawnie zasobów układów GPU będzie kluczowe dlatego należy sprawdzić wersję CUDA w notatniku poprzez komendę: `!/usr/local/cuda/bin/nvcc –version`. W wyniku otrzymujemy szczegółową informację dotyczącą wersji CUDA.

nvcc: NVIDIA (R) Cuda compiler driver Copyright (c) 2005-2020 NVIDIA Corporation Built on Wed_Jul_22_19:09:09_PDT_2020 Cuda compilation tools, release 11,0, V11.0.221 Build cuda_11.0_bu.TC445_37.28845127_0	1 2 3 4 5
---	-----------------------

Program 3.4. Informacje o wersji CUDA zainstalowanej w serwisie Google Colab

³De facto to Jupyter Notebook ponieważ ma rozszerzenie .ipynb

Dodatkowo, aktywowanie poprawnej infrastruktury CUDA następuje poprzez uruchomienie Programu 3.5, który automatycznie wykrywa jaki układ GPU został przydzielony użytkownikowi, a następnie łączy z CUDA. Przykładowe uruchomienie kodu z układem GPU może zwrócić następujący tekst: GPU Type: Tesla P100-PCIE-16GB. "GPU Type oznosi się karty graficznej, która została przydzielona w ramach sesji.

```

1 import os
2 os.environ['GPU_TYPE'] = str(os.popen('nvidia-smi --query-gpu=name --format
3 =csv,noheader').read())
4
5 def getGPUArch(argument):
6     try:
7         argument = argument.strip()
8         # All Colab GPUs
9         archTypes = {
10             "Tesla V100-SXM2-16GB": "-gencode arch=compute_70,code=[sm_70,
11                                         compute_70]",
12             "Tesla K80": "-gencode arch=compute_37,code=sm_37",
13             "Tesla T4": "-gencode arch=compute_75,code=[sm_75,compute_75]",
14             "Tesla P40": "-gencode arch=compute_61,code=sm_61",
15             "Tesla P4": "-gencode arch=compute_61,code=sm_61",
16             "Tesla P100-PCIE-16GB": "-gencode arch=compute_60,code=sm_60"
17         }
18         return archTypes[argument]
19     except KeyError:
20         return "GPU must be added to GPU Commands"
21     os.environ['ARCH_VALUE'] = getGPUArch(os.environ['GPU_TYPE'])
22
23 print("GPU Type: " + os.environ['GPU_TYPE'])
24 print("ARCH Value: " + os.environ['ARCH_VALUE'])

```

Program 3.5. Aktywowanie poprawnej wersji CUDA w języku Python

3.4.3 Trenowanie modelu w Google Colab

Po przygotowaniu notatnika do korzystania z kart graficznych, należy przejść do procedury trenowania konkretnego modelu. Konieczne będzie ściągnięcie Darknet czyli struktury sieci neuronowej, z której korzysta Yolo v4. W tym celu pobierane jest następujące repozytorium Github *!git clone https://github.com/roboflow-ai/darknet.git*. W ramach ściągniętego folderu należy ręcznie skonfigurować plik **Makefile**, który stanowi plik reguł dla programu make. Make jest programem systemowym, który automatyzuje komplikację programów. Na podstawie pliku **Makefile**, program make ustala, które pliki źródłowe wymagają komplikacji. Takie podejście znacznie przyspiesza tworzenie programów ponieważ nie jest potrzebne komplikowanie całego projektu. W przypadku pliku **Makefile** znajduje się kilka kluczowych informacji:

1. GPU=1 – oznacza wykorzystanie kart GPU,
2. CUDNN=1 – oznacza wykorzystanie CUDA do obliczeń,

3. CUDNN_HALF=0 - wyłączona opcja tzw. Mixed-precision czyli przyspieszenia obliczeń kosztem precyzji,
4. OPENCV=1 – oznacza wykorzystanie biblioteki OpenCV wspomagającej obróbkę wideo w czasie rzeczywistym,
5. AVX=0 i OPENMP=0 – AVX i OPENMP powinny posiadać wartość 1 w momencie wykorzystania CPU.

Następnie należy zainstalować środowisko na bazie pliku Makefile, przykładowa instalacji została ukazana w Programie 3.6.

```
!sed -i 's/OPENCV=0/OPENCV=1/g' Makefile           1
!sed -i 's/GPU=0/GPU=1/g' Makefile                 2
!sed -i 's/CUDNN=0/CUDNN=1/g' Makefile             3
!sed -i "s/ARCH= -gencode arch=compute_60,code=sm_60/ARCH= ${ARCH_VALUE}/g" Makefile
!make                                              5
```

Program 3.6. Skrypt zamieniający poszczególne wartości w pliku makefile

W kolejnym etapie konieczne będzie pobranie wag do sieci neuronowej dla Yolo v4, które mogą zostać pobrane z wykorzystaniem procedury wget: !wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.conv.137. Pobrane wagi zostały wyekstraktowane z modelu, który został nauczony na zbiorze danych COCO. Zbiór ten zawiera 330 tysięcy różnych zdjęć, dlatego model nauczony na takim zbiorze danych będzie dobrze rozpoznawał osobne instancje obiektów pomimo tego, że danych obiektów nigdy wcześniej nie widział. Poprzez wykorzystanie wcześniej utworzonych wag, model będzie w stanie szybciej dociągnąć się generalizowania paneli słonecznych.

W kolejnym kroku środowisko Google Colab zostanie połączone z usługą Google Dysk. Usługa Google Dysk służy do przechowywania i synchronizacji plików stworzona przez Google.

```
from google.colab import drive
drive.mount('/content/gdrive')
```

Program 3.7. Połączenie Google Colab i Google Drive

Wszystkie dane przeznaczone do trenowania bądź walidowania modelu zostały wcześniej do usługi Google, tak aby ułatwić i przyspieszyć przesył danych. Każderazowe uruchomienie notatnika Google Colab wymaga autoryzacji między notatnikiem, dyskiem. Czynność ta została udokumentowana w Programie 3.7.

Pobranie danych z dysku Google do notatnika Colab następuje po uruchomieniu kodu:

```
!unzip -q '/content/gdrive/MyDrive/train_test_poznan_rok_danych.zip'.
```

Komenda `unizp` oznacza, że archiwum zip zostanie rozpakowane i zostaną dodane dwa foldery: `train` i `test`. Algorytm Yolo v4 wymaga dostępu do danych z folderu `data` jak przedstawiono w Programie 3.8

```
#dodanie etykiet do folderu data 1
%cp /content/gdrive/MyDrive/obj.names data/obj.names 2

#utworzenie folderu obj 3
%mkdir data/obj 4
#dodanie wszystkich zdjęć oraz etykiet do folderu data/obj 5
%cp train/*.jpg data/obj/ 6
%cp test/*.jpg data/obj/ 7

%cp train/*.txt data/obj/ 8
%cp test/*.txt data/obj/ 9
```

Program 3.8. Kopiowanie danych z Google Drive na Google Colab

Następnie tworzone zostają pliki tekstowe zawierające ścieżki do zbioru testowego oraz treningowego na potrzeby uczenia głębokiej sieci neuronowej. Program 3.9 przedstawia skrypt w języku Python, który tworzy te pliki.

```
#utworzenie pliku pliku obj.data 1
with open('data/obj.data', 'w') as out: 2
    #dodanie liczby klas do predykcji jako 1 3
    out.write('classes = 1\n') 4
    #wskazanie, że w~lista zdjęć do trenowania modelu znajdzie się w~pliku 5
    #train.txt
    out.write('train = data/train.txt\n') 6
    #wskazanie, że w~lista zdjęć do testowania modelu znajdzie się w~pliku 7
    #test.txt
    out.write('valid = data/test.txt\n') 8
    #nazwy etykiet znajdują się pod data\obj.names 9
    out.write('names = data/obj.names\n') 10
    #wagi modelu zostaną zapisane do folderu backup 11
    out.write('backup = backup/')

import os 14
#stworzenie pliku train.txt, który będzie listą wszystkich zdjęć, które 15
#znajdują się w~folderze train
with open('data/train.txt', 'w') as out: 16
    for img in [f for f in os.listdir('train') if f.endswith('.jpg')]: 17
        out.write('data/obj/' + img + '\n') 18

#stworzenie pliku test.txt, który będzie listą wszystkich zdjęć, które 19
#znajdują się w~folderze test
with open('data/test.txt', 'w') as out: 20
    for img in [f for f in os.listdir('test') if f.endswith('.jpg')]: 21
        out.write('data/obj/' + img + '\n')
```

Program 3.9. Tworzenie zbioru dla próby uczącej i testowej

Kolejnym etapem przygotowania środowiska jest stworzenie pliku `config`. Plik `config` zawiera informacje dotyczące konfiguracji sieci neuronowej czy augmentacji danych. Pierwszym istotnym argumentem w pliku `config` jest `max_batches`, który powinien wynosić

`liczba_klas × 2000`. W przypadku `steps`, wartość dolna powinna wynosić 80%, a górna 90% `max_batches`.

1. `batch = 64` – określa liczbę zdjęć wgrywanych w ramach iteracji,
2. `subdivisions=24` – określa na ile zdjęć będzie dzielony dodatkowo batch,
3. `width = 416` – szerokość zdjęcia na wejściu
4. `height = 416` – wysokość zdjęcia na wejściu,
5. `channels = 3` – liczba kanałów zdjęcia, w przypadku 3 oznacza RGB, czyli użycie kolorowych zdjęć
6. `momentum = 0,949` – gradient dla kolejnej iteracji jest liczony jako `momentum × poprzedni_gradient + (1-momentum) × aktualny_gradient`,
7. `saturation = 1,5` – losowo zmienia nasycenie obrazów podczas treningu,
8. `exposure = 1,5` – losowo zmienia ekspozycję (jasność) obrazów podczas treningu,
9. `hue = 0,1` – losowo zmienia odcień (kolor) podczas treningu,
10. `blur = 1` – rozmycie będzie stosowane losowo w 50% przypadków,
11. `flip = 1` – stosowane będzie obracanie zdjęć dla zbioru treningowego,
12. `aspect = 0,75` – proporcja zdjęć będzie się zmieniała
13. `gaussian_noise = 1` – dodanie do zdjęć białego szumu

Rysunki 3.10 pokazują przykładowe zastosowania augmentacji danych w trakcie uczenia zbioru treningowego.



Rysunek 3.10. Przykład augmentacji próbki poprzez algorytm Cutmix (lewy górny), rozmycia zdjęcia (prawy górny) i zmianę odcienia koloru zdjęcia (dolny)

Źródło: Opracowanie własne.

3.5 Podsumowanie

Każde zagadnienie związane z uczeniem maszynowym wiąże się z przygotowaniem danych. W tym aspekcie kluczowe jest nie tylko zrozumienie z jakimi danymi ma się do czynienia, ale również sam fakt zastosowanych transformacji mających na celu minimalizację potencjalnych problemów, na które mógłby natrafić model uczenia maszynowego. W kolejnym rozdziale przedstawione zostaną wyniki eksperymentów związanych z rozpoznawaniem paneli i kolektorów słonecznych.

Rozdział 4

Zastosowanie głębokich sieci neuronowych do detekcji paneli i kolektorów słonecznych

4.1 Procedura badawcza

4.1.1 Wykorzystana sieć neuronowa Yolo v4

W pracy wykorzystano algorytm Yolo opisany w pliku custom-yolov4-detector.cfg, który znajduje się w repozytorium roboflow-ai/darknet w serwisie Github. Plik z rozszerzeniem cfg definiuje architekturę całej sieci neuronowej wykorzystywanej do trenowania modelu. Na wejściu sieć otrzymuje zdjęcie o rozmiarach 416x416. Zdjęcia przetwarzane są przez łącznie 110 warstw konwolucyjnych, 21 warstw łączących, 23 warstwy dodające, 3 warstwy maksymalizujące wartości cech oraz 3 warstwy yolo odpowiedzialne za przewidywanie lokalizacji pól ograniczających.

4.1.2 Eksperymenty

Aby odpowiedzieć na pytania badawcze stawiane w pracy zaprojektowano 3 eksperymenty, które zostały zdefiniowane poniżej

1. Eksperyment 1:

- Sieć uczona na danych dla Poznania na zdjęciach z 2020 roku (zbior uczący i testowy);

- Skuteczność sieci weryfikowana na nowych danych: Poznań 2018 oraz Skórzewo 2018 i 2020.

2. Eksperyment 2:

- Sieć uczona na danych dla Poznania na zdjęciach z 2018 roku (zbiór uczący i testowy);
- Skuteczność sieci weryfikowana na nowych danych: Poznań 2020 oraz Skórzewo 2018 i 2020.

3. Eksperyment 3:

- Sieć uczona na danych dla Poznania na zdjęciach z 2018 i 2020 roku (zbiór uczący i testowy);
- Skuteczność sieci weryfikowana na nowych danych: Skórzewo 2018 i 2020.

Liczebność poszczególnych zbiorów przedstawiono w tabeli 4.1. W przypadku 2018 i 2020 roku określono liczbę zdjęć w próbie uczącej i testowej na 326 i 82 aby uniknąć wpływu wielkości zbioru wejściowego. W przypadku modelu 3 wykorzystano wszystkie zdjęcia z 2018 i 2020 roku. Pomimo zastosowania tej samej liczby zdjęć liczba pojedynczych paneli różniła się między zbiorami. W przypadku 2018 roku, na wszystkich zdjęciach pochodzących z Poznania znajdowało się 416 paneli. Dla roku 2020, pomimo użycia tej samej liczby zdjęć, liczba paneli się różniła i wynosiła 586. Dla wsi Skórzewo, różnice w panelach były większe niż dla miasta Poznania. Pomimo użycia zbliżonej liczby zdjęć (w 2018 roku 207, a dla 2020 roku 222 zdjęcia), paneli w Skórzewie w 2020 roku było o 243 więcej niż dwa lata wcześniej. Taka różnica może wskazywać na dużą dynamikę instalacji paneli między 2018, a 2020 rokiem.

Tablica 4.1. Modele oraz liczebność zbiorów uczących i testowych

Numer modelu	Rok	Zbiór uczący	Zbiór testowy	Skórzewo
Liczba zdjęć				
1	2018	326	82	207
2	2020	326	82	222
3	2018 i 2020	652	164	429
Liczba instalacji				
1	2018	331	85	9
2	2020	455	131	243
3	2018 i 2020	786	216	252

Źródło: Opracowanie własne.

Biblioteka darknet zawiera algorytmy rozszerzające zbiór uczący poprzez modyfikację zdjęć wejściowych. Dlatego relatywnie niewielka liczba zdjęć wejściowych została w trakcie procedury uczenia zwiększona dziesięciokrotnie względem początkowego rozmiaru.

4.2 Wyniki eksperymentów

4.2.1 Eksperiment 1

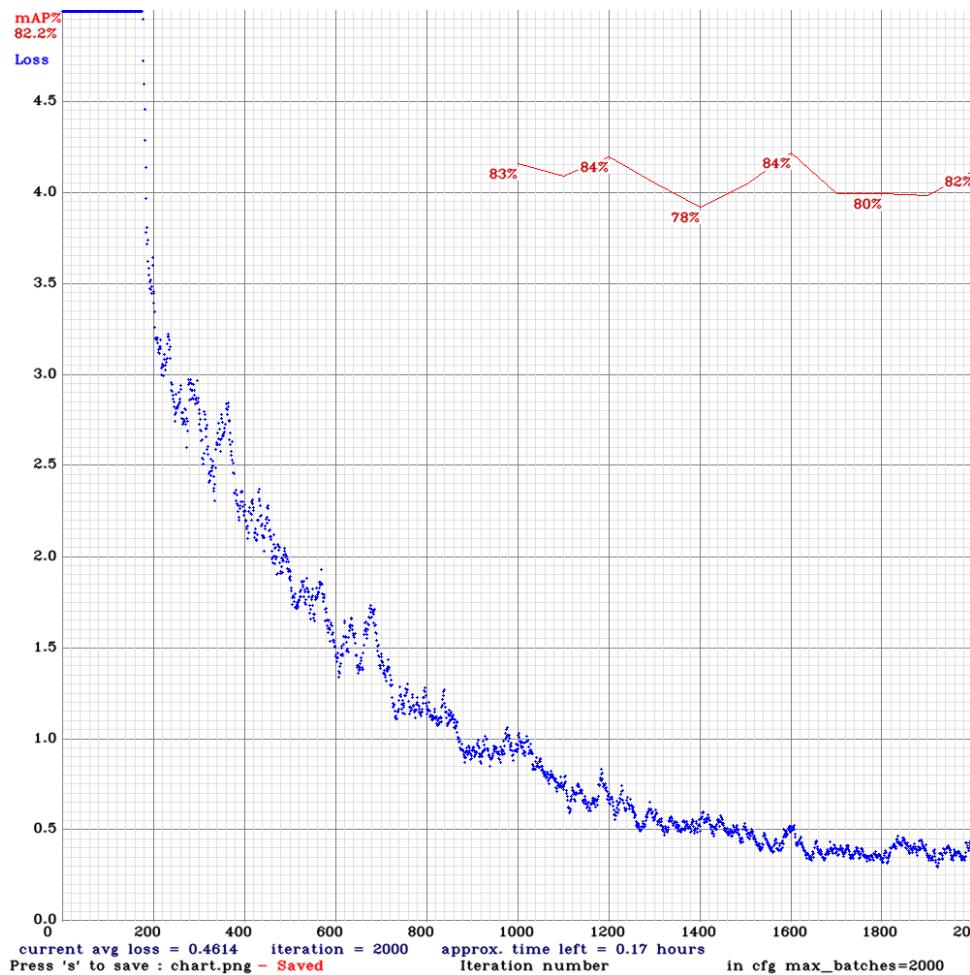
Tworzenie modeli rozpoczęto od miasta Poznania oraz roku 2020, gdzie się znajdowało najwięcej paneli. W procesie ekstrakcji dachów zostało znalezionych 864 zdjęć zawierających co najmniej jedną instalację fotowoltaiczną. Jednak ze względu na to, że dla roku 2018 z pobranych dachów jedynie 204 zdjęć posiadało co najmniej jeden panel, liczba zdjęć branych pod uwagę do stworzenia modelu dla danych pochodzących z 2020 roku została zrównana z próbką dla 2018 roku. Jako, że w przypadku modeli uczenia maszynowego im większa liczba zdjęć tym model może lepiej generalizować, późniejsze porównanie modeli nie miało sensu. Przy wykorzystaniu pakietu Google Colab PRO uczenie modelu trwało około 2 godziny 30 minut. W trakcie uczenia, Yolo v4 zapisuje średnią stratę i mAP (mean Average Precision) zdefiniowaną w rozdziale 2.

Rysunek 4.1 przedstawia wizualizację procedury treningu sieci z wykorzystaniem modułu darknet. Niebieskim kolorem przedstawiono funkcję straty, a kolorem czerwonym wartość mAP. Po 2000 iteracji wartość średniej straty spadła do poziomu 0,4614 z początkowego poziomu 5. W przypadku mAP pierwsze obliczenia miały miejsce dla 1000 iteracji, a następnie były powtarzane co setną iterację. Najwyższa wartość miary mAP została wyliczona dla 1600 iteracji i wynosiła 84%.

Tablica 4.2. Tabela klasyfikacji dla zbioru treningowego dla 2020 roku

Wartości przewidywane	Wartości prawdziwe	
	Instalacja	Instalacja
		Brak instalacji
Instalacja	417	129
Brak instalacji	38	0

Źródło: Opracowanie własne.



Rysunek 4.1. Krzywa średniej straty oraz mAP w trakcie trenowaniu modelu na danych z Poznania dla 2020 roku

Źródło: Opracowanie własne. Objaśnienie: niebieskim kolorem przedstawiono wartość funkcji straty, a czerwonym wartość mAP.

Tabela 4.2 pokazuje predykcję etykiet na zbiorze treningowym wraz z prawdziwymi etykietami. W przypadku zbioru treningowego, algorytm częściej widział panele w miejscach, gdzie ich nie było niż błędnie rozpoznawał istniejące panele.

Tablica 4.3. Tabela klasyfikacji dla zbioru testowego dla 2020 roku

Wartości przewidywane	Wartości prawdziwe	
	Instalacja	Instalacja
		Brak instalacji
Instalacja	107	31
Brak instalacji	24	0

Źródło: Opracowanie własne.

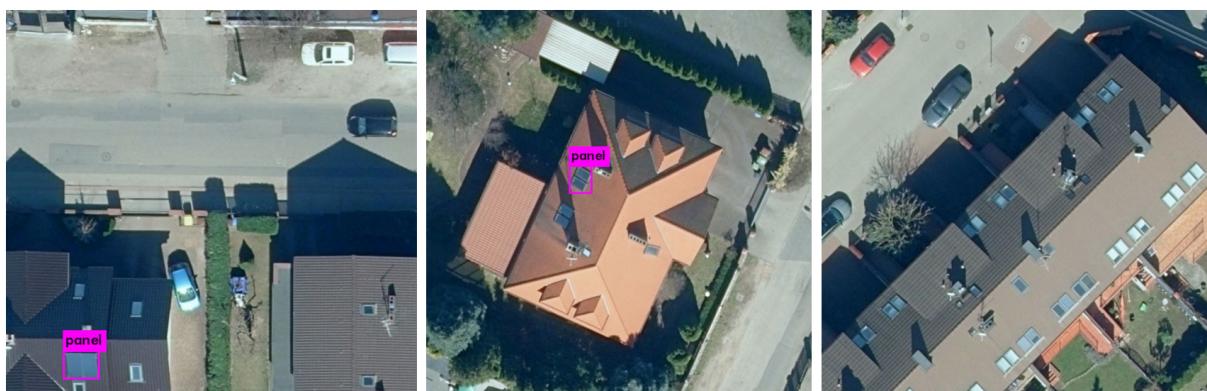
Tendencja utrzymuje się również dla zbioru testowego co widoczne jest w 4.3.

Tablica 4.4. Weryfikacja sieci z 2020 roku na danych z 2018 i Skórzewa

Model	mAP	TP	FP	FN	Precyza	Czułość	F1	IoU
Poznań 2018	79,86%	69	19	16	78%	81%	80%	59,74%
Skórzewo	82,52%	208	83	44	71%	83%	77%	55,11%

Źródło: Opracowanie własne.

W ramach stworzonego modelu na bazie zdjęć z 2020 roku, uruchomiono proces predykcji dla pozostałych zbiorów czyli danych z 2018 roku dla Poznania oraz zdjęć ze Skórzewa dla 2018 i 2020 roku. Wyniki wskazują, że model lepiej przewidaje lokalizacje paneli dla zbioru pochodzącego ze Skórzewa ze względu na większą skuteczność (82,52% do 79,86%). Predykcja na zbiorze z Poznania wykazała zaś wyższą precyzję niż dla zbioru ze Skórzewa.



Rysunek 4.2. Przykładowe predykcje modelu 2020 na zbiorze danych z 2018 roku

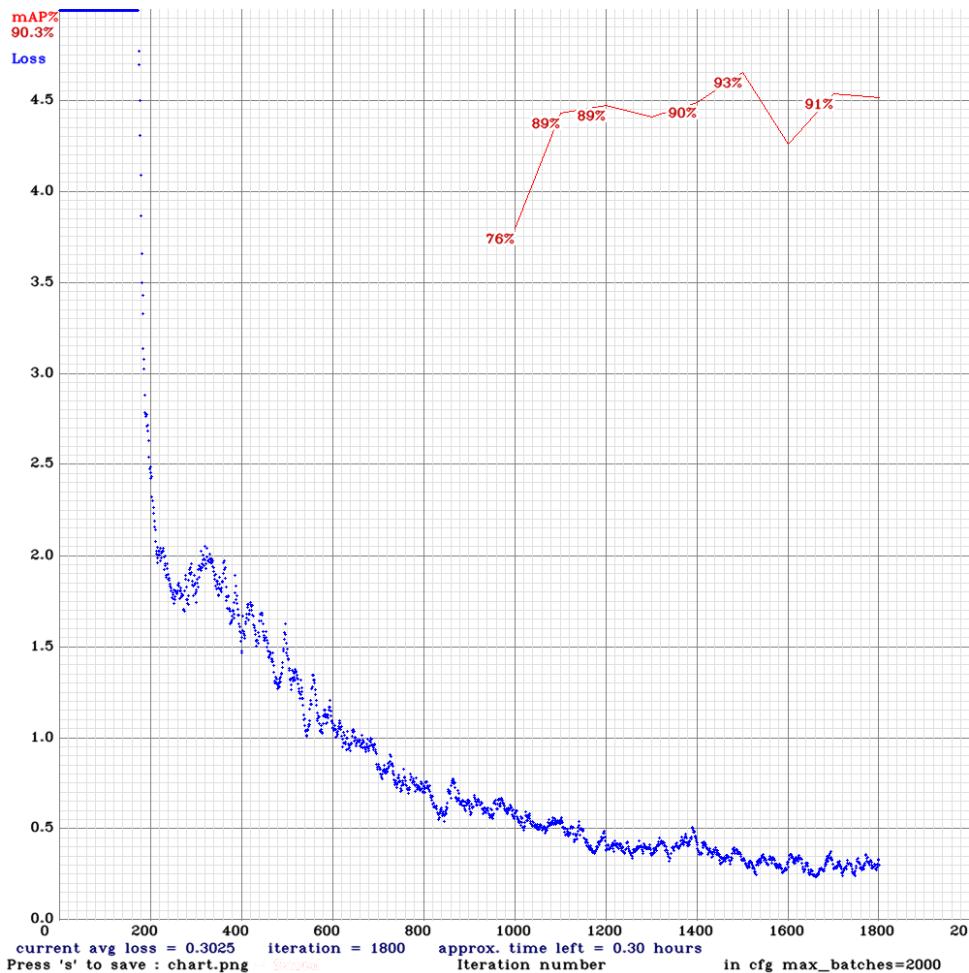
Źródło: Opracowanie własne.

Rysunek 4.2 ukazuje przykładowe predykcje modelu z 2020 roku na danych z 2018. Na lewym obrazku widać poprawnie zakwalifikowany panel, na środku jest błędna predykcja ponieważ model zaznaczył okno, a po prawej model nie uznał okien za panele co można uznać za sukces ze względu na wygląd okien.

4.2.2 Eksperyment 2

Kolejnym stworzonym modelem był model oparty na danych z 2018 roku. Ponownie, do zbioru treningowego zostało dodanych 204 zdjęcia z co najmniej jednym panelem oraz tyle

samo bez paneli. Model był uczyony również w notatniku Google Colab dla konta PRO. Czas uczenia wyniósł około 2 godziny.



Rysunek 4.3. Krzywa średniej straty oraz mAP w trakcie trenowania modelu na danych z Poznania dla 2018 roku

Źródło: Opracowanie własne.

W przeciwieństwie do pierwszego modelu, 4.3 pokazuje, że proces trenowania trwał 1800 iteracji ponieważ model uznał, że wartość straty nie spada wystarczająco zatem nie ma sensu kolejne 200 iteracji. Wartość średniej straty w przypadku świeżych danych była niższa niż w pierwszym modelu i wyniosła 0,3025, zaś maksymalna wartość mAP wyniosła 93,01% w momencie 1400 iteracji.

Następnie, przeprowadzono weryfikację modelu na danych z 2018 roku dla Poznania oraz Skórzewa za 2018 i 2020.

W przypadku zbioru treningowego dla danych z 2018 roku, uzyskano następujące wyniki dla klasyfikacji 4.5. Tak jak dla modelu z 2020 roku, również model dla danych z 2018 częściej prognozował istnienie panelu pomimo braku istnienia instalacji fotowoltaicznej.

Tablica 4.5. Tabela klasyfikacji dla zbioru treningowego dla 2018 roku

Wartości przewidywane	Wartości prawdziwe	
	Instalacja	Brak instalacji
Instalacja	322	26
Brak instalacji	9	0

Źródło: Opracowanie własne.

Wyniki przedstawione w 4.6 ukazują już dane klasyfikacyjne dla zbioru testowego. W przeciwieństwie do modelu z pierwszego eksperymentu, dla danych testowych występuje odwrotna sytuacja i model częściej nie rozpoznaje paneli tam, gdzie one istnieją.

Tablica 4.6. Tabela klasyfikacji dla zbioru testowego dla 2018 roku

Wartości przewidywane	Wartości prawdziwe	
	Instalacja	Brak instalacji
Instalacja	75	8
Brak instalacji	10	0

Źródło: Opracowanie własne.

Model z eksperymentu drugiego wykorzystano do stworzenia predykcji dla danych z Poznania pochodzących z 2020 roku oraz danych dla wsi Skórzewa dla lat 2018 i 2020. Podobnie jak w eksperymencie pierwszym, model uzyskał lepszą dokładność dla danych pochodzących ze Skórzewa (81,06% do 77,87%). Dodatkowo dla tego zbioru otrzymano również wyższą precyzję oraz taką samą czułość.

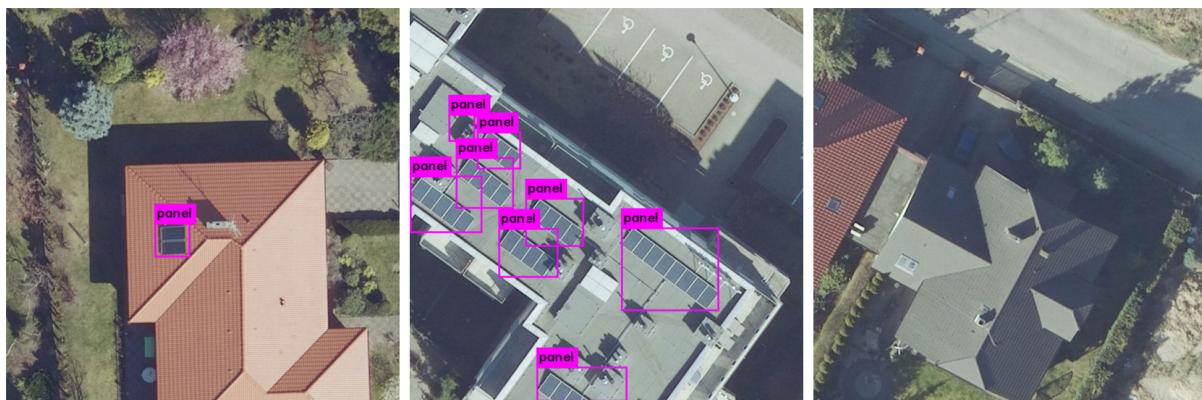
Tablica 4.7. Weryfikacja sieci z 2018 roku na danych z 2020 i Skórzewa

Model	mAP	TP	FP	FN	Precyza	Czułość	F1	IoU
Poznań 2020	77,87%	106	48	25	69%	81%	74%	53,86%
Skórzewo	81,06%	203	54	49	79%	81%	80%	59,62%

Źródło: Opracowanie własne.

Tabela 4.4 pokazuje przykładowe predykcje modelu z 2018 roku na danych z 2020. Po lewej stronie widać prawidłową predykcję panelu słonecznego. Dla środkowego zdjęcia jest jedna błędna predykcja, w lewym górnym rogu, która w rzeczywistości jest kominem, a nie panelem

słonecznym. Z prawej strony zdjęcia pokazano przykład dachu, na którym wygląd okna mocno przypomina panel słoneczny, jednak model nie rozpoznał na dachu instalacji fotowoltaicznej.



Rysunek 4.4. Przykładowe predykcje modelu 2018 na zbiorze danych z 2020 roku

Źródło: Opracowanie własne.

4.2.3 Eksperyment 3

W celu uzyskania możliwie najlepszego modelu, podjęta została decyzja o stworzeniu zbioru opartego na danych z 2018 i 2020 roku. Zwiększenie liczby zdjęć zarówno z panelami jak i bez paneli powinno spowodować, że model będzie w stanie lepiej generalizować rzeczywistość.

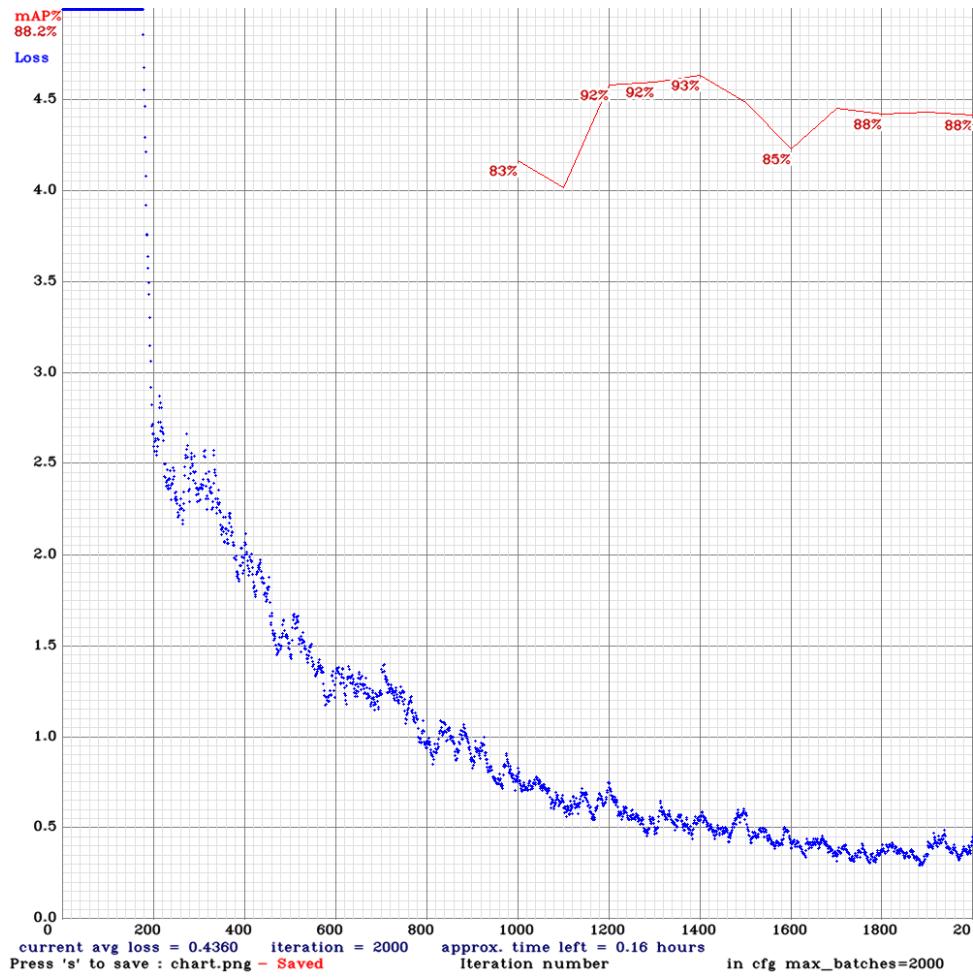
Jak ukazuje 4.5, średnia strata była wyższa niż dla drugiego modelu, ale porównywalna dla pierwszego i wyniosła 0,4360. W przypadku mAP, najwyższy współczynnik został uzyskany dla iteracji 1300 kiedy wyniósł 92,63%.

Tabela 4.8 przedstawia pomyłek dla zbioru testowego dla modelu uczącego się na danych zarówno z 2018 jak i 2020 roku. Jak w przypadku poprzednich modelu również i tu widać tendencję modelu do częstszej przewidywania paneli, tam gdzie ich nie ma niż nie wskazywania instalacji, tam gdzie rzeczywiście są.

Tablica 4.8. Tabela klasyfikacji dla zbioru treningowego dla 2018 i 2020 roku

Wartości przewidywane	Wartości prawdziwe	
	Instalacja	Brak instalacji
		745
Brak instalacji	41	0

Źródło: Opracowanie własne.



Rysunek 4.5. Krzywa średniej straty oraz mAP w trakcie trenowaniu modelu na danych z Poznania dla 2018 i 2020 roku

Źródło: Opracowanie własne.

Dla zbioru testowego wyniki przedstawiono w 4.9. Wnioski są podobne jak w przypadku pierwszego modelu, gdzie tendencja ze zbioru treningowego została zachowana. Model na zdjęciach, które wcześniej nie widział częściej, przewidywał istnienie panelu.

Tablica 4.9. Tabela klasyfikacji dla zbioru testowego dla 2018 i 2020 roku

Wartości przewidywane	Wartości prawdziwe	
	Instalacja	Instalacja
		Brak instalacji
Instalacja	187	52
Brak instalacji	29	0

Źródło: Opracowanie własne.

W ramach walidacji utworzonego modelu uruchomiona została predykcja na zbiorze ze Skórza dla lat 2018 i 2020. Model nauczony na danych z dwóch lat wykazał się najwyższą sku-

tecznością ze wszystkich modelu uzyskując mAP równe 86,11%. Precyzaja wyniosła 75%, zatem drugi model uzyskał lepszy wynik. W przypadku czułości, model z 3 eksperymentu uzyskał najwyższy wynik ze wszystkich modeli 86,11%.

Tablica 4.10. Weryfikacja sieci z 2018 i 2020 roku na danych z 2020 i Skórzewa

Model	mAP	TP	FP	FN	Precyzaja	Czułość	F1	IoU
Skórzewo	86,11%	216	71	36	75%	86%	80%	60,12%

Źródło: Opracowanie własne.

Dwie predykcje najlepszego modelu zostały ukazane w 4.6. Lewy obrazek pokazuje poprawne zidentyfikowane panele słoneczne, w przypadku prawego zdjęcia model ukazał o jeden panel za mało. Wytrenowany model powinien wskazać osobne instancje dla każdego z rzędu.



Rysunek 4.6. Przykładowe predykcje modelu 2018 i 2020 na zbiorze danych ze Skórzewa

Źródło: Opracowanie własne.

W przeciwieństwie do modeli z poprzednich przykładów, model zaagregowany dla danych z 2018 i 2020 roku częściej popełnia błąd pierwszego rodzaju zatem zaznacza panel tam, gdzie w rzeczywistości instalacji fotowoltaicznej nie ma.

4.3 Wnioski płynące z eksperymentów

W ramach pracy magisterskiej została dokonana obszerna analiza modeli uczenia głębo-kiego na podstawie różnych danych treningowych. Nie stanowi zaskoczenie fakt, że model oparty na największej liczbie zdjęć (dane dla 2018 i 2020 roku) potrafił najdokładniej rozpoznawać oraz zaznaczać panele uzyskując na zbiorze testowym dokładność na poziomie 92,63%. Dodatkowo w ramach walidacji modeli, uruchomiona została predykcja na niewidzianym wcześniejszym zbiorze zdjęć pochodzącym ze Skórzewa dla lat 2018 i 2020. Model oparty na największej liczbie danych uzyskał dokładność na poziomie 86,11%, gdzie dokładność pozostałych modeli wyniosła 82,52% oraz 81,06% odpowiednia dla modeli z lat 2020 i 2018. Stworzone modele dobrze rozpoznają instalacje słoneczne pochodzące z danych dla innych lat. Modele dla roku 2020 i 2018 uzyskały odpowiednio skuteczność na poziomie 79,86% i 77,87% dla danych z innych lat. Dokładność modeli dla danych z innych lat jest niższa jednak modele nadal potrafią poprawnie rozpoznać większość paneli. W przypadku walidacji modeli na danych z innej lokalizacji wszystkie nauczone modele uzyskały lepsze wyniki niż na danych z innych lat. Wnioski te sprawiają, że przy masowym użyciu modelu algorytmu Yolo v4 ważniejsze byłoby uzyskanie jak największej ilości zdjęć z różnych miast niż z różnych lat. Uzyskane wyniki stanowią potwierdzenie, że dla jak najlepszego odwzorowania rzeczywistości kluczowa jest nie tylko jakość danych, ale również dostarczana ilość zdjęć w ramach procesu uczenia.

Podsumowanie

Głównym problemem badawczym pracy była ocena możliwości wykorzystania algorytmów detekcji obiektów na zdjęciach lotniczych na potrzeby statystyki energii odnawialnej w Polsce. W tym celu określono cel główny, którym było opracowanie algorytmu detekcji paneli i kolektorów słonecznych na potrzeby statystyki energii odnawialnej w Polsce. Motywacją natomiast było uzyskanie bardziej szczegółowych informacji na temat wykorzystania energii słońca w Polsce na podstawie źródeł big data.

Dane wykorzystane do nauczenia i walidacji modelu pozyskano z Głównego Urzędu Geodezji i Kartografii w zakresie ortofotomapy dla lat 2018 i 2020 dla miasta Poznania i wsi Skórzewa. W ramach współpracy z jednym z operatorów sieci dystrybucyjnych otrzymano adresy klientów posiadających instalacje fotowoltaiczne. Na podstawie współrzędnych geograficznych od operatora, pobrano arkusze ortofotomapy, za pomocą pakietu rgugik w języku R, zawierające poszczególne lokalizacje instalacji. W celu ograniczenia ilości przetwarzanych danych, przy użyciu pakietu terra wycięto dokładne lokalizacje instalacji fotowoltaicznych względem współrzędnych od operatora. Konieczne było ręczne oznaczenie zdjęć ramkami ograniczającymi zawierających instalacje fotowoltaiczne czego dokonano na stronie cvat.org. Zdjęcia zawierające panele stanowiły 50% całej próby. Dodatkowo, zbiór został podzielony na treningowy (80%) oraz testowy (20%). Przygotowany zbiór do trenowania modelu można znaleźć na platformie Kaggle¹. Finalnym krokiem było zastosowanie głębokich sieci neuronowych, a dokładniej algorytmu Yolo v4 do detekcji paneli i kolektorów słonecznych.

W ramach pracy magisterskiej nauczono trzy modele algorytmu Yolo v4 na danych z Poznania dla wybranych lat (2018 lub 2020) lub dla obydwu latach (2018 oraz 2020). Zarówno walidując modele na zbiorze testowym na bazie danych z Poznania jak i danych ze wsi Skórzewo, najlepszy model oparty był na danych z lat 2018 i 2020. Na zbiorze testowym uzyskał on

¹https://www.kaggle.com/mivoss/photovoltaic-systems-on-aerial-images?select=train_test_poznan_2018_2020_second_model

dokładność równą 92,63%. Walidując model na niewidzianych wcześniej danych ze Skórzewa otrzymano dokładność wynoszącą 86,11%.

Na wstępie pracy zdefiniowane zostały cele szczegółowe, które miały pomóc w realizacji głównego celu. Pierwszy szczegółowy cel został zrealizowany poprzez wykorzystanie pakietu rgugik wraz z dostępem do API dostarczonego przez GUGiK wraz z licznymi transformacjami w języku R. W ramach drugiego szczegółowego celu, został przygotowany zbiór danych na podstawie wcześniej pobranych ortofotomap. Zdjęcia lotnicze zostały iteracyjnie przetwarzane w języku R dzięki czemu wycięto działki zawierające panele. Współrzędne prosumentów zostały dostarczone przez operatora w ramach współpracy. Niedokładność dużej części danych sprawiła, że cały zbiór musiał zostać ręcznie sprawdzony w celu wyeliminowania zdjęć niezawierających panele. Wytrenowanie trzech modeli na podstawie różnych danych spełniło cel szczegółowy C3. Późniejsza walidacja utworzonych modeli w czasie oraz przestrzeni pozwoliła odpowiedzieć na hipotezy zawarte we wstępie.

Model nauczony algorytmem Yolo v4 zbudowany na podstawie danych pochodzących z 2018 i 2020 roku z wysoką skutecznością przewidywał obiekty oraz dokładnie oznaczał pola ograniczające. Hipoteza oparta na niższej skuteczności dla innego okresu została po części odrejona. Zarówno dla modelu na bazie danych z 2018 i 2020 roku, skuteczność dla innego okresu była niższa niż dla zestawu testowego. Model oparty na danych z 2020 roku, uzyskał dokładność na zbiorze testowym na poziomie 92,63%, na zbiorze z 2018 roku zdobył dokładność równą 79,86%. W przypadku modelu na bazie danych z 2018 roku, który uzyskał dokładność równą 93,01% na zbiorze testowym, dane dla 2020 roku wykazały dokładność na poziomie 77,87%. W obu przypadkach zdolność do rozpoznawania paneli maleje wraz ze zmianą roku pochodzenia danych jednak nie w sposób drastyczny. Trzecia hipoteza została postawiona na bazie wniosków pochodzących z badania prowadzonego przez Statistics Netherlands o nazwie Deep Solaris (Curier i in. 2018). W ramach badania zostały stworzone liczne modele uczenia głębokiego na bazie danych z różnych miast. Jednym z wniosków płynących z pracy jest fakt, że w przypadku walidacji modelu na danych z innego miasta, zauważono znaczący spadek dla precyzyji oraz czułości. W celu weryfikacji tej hipotezy, wszystkie nauczone modele zostały uruchomione na danych ze wsi Skórzewo. W przypadku modeli opartych na danych z jednego roku, model poradził sobie lepiej z detekcją paneli słonecznych w Skórzewie niż dla innego roku, dla danych z Poznania. Dla modelu uczonego na zbiorze z lat 2018 i 2020, dokładność wyniosła 86,11% i była najwyższa ze wszystkich modeli.

Konkludując, modele oparte na algorytmie Yolo v4 z bardzo wysoką dokładnością rozpoznają panele i kolektory słoneczne. Dla poszczególnych modeli bardziej problematyczne stano-wiły procesy validacji na danych z różnych lat niż na danych z innej lokalizacji. Wykorzystanie modeli uczenia głębokiego w statystyce publicznej mogłoby znaczaco zmniejszyć koszty prowadzenia informacji publicznej dotyczącej zielonej energii.

Bibliografia

- Berlinski, D. (2001). *The Advent of the Algorithm: The 300-Year Journey from an Idea to the Computer*. Harcourt. <https://books.google.pl/books?id=gCswb92k74QC>
- Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection.
- Castellano, R. (2012). *Alternative Energy Technologies: Opportunities and Markets*. Éd. des archives contemporaines. <https://books.google.pl/books?id=RUqXHr3Wj0IC>
- Chollet, F. (2017). *Deep learning with Python*. Simon; Schuster.
- Curier, R., De Jong, T., Strauch, K., Cramer, K., Rosenski, N., Schartner, C., Debusschere, M., Ziemons, H., Iren, D. & Bromuri, S. (2018). Monitoring spatial sustainable development: Semi-automated analysis of satellite and aerial images for energy transition and sustainability indicators. *arXiv preprint arXiv:1810.04881*.
- Duda, R. O., Hart, P. E. i in. (1973). *Pattern classification and scene analysis* (Vol. 3). Wiley New York.
- Dyba, K. & Nowosad, J. (2021). rgugik: Search and Retrieve Spatial Data from the Polish Head Office of Geodesy and Cartography in R. *Journal of Open Source Software*, 6(59), 2948. <https://doi.org/10.21105/joss.02948>
- Dyrektyna Parlamentu Europejskiego i Rady (UE) 2018/2001. (2018).
- EurObserv'ER. (2020). Solar thermal and concentrated solar power barometer 2020.
- Europejska, K. (2018). SHARES Tool Manual.
- Europejska Agencja Środowiska. (2020). Trends and projections in Europe 2020.
- Eurostat. (2020). Energy, transport and environment statistics.
- Fraunhofer Institute. (2020). PHOTOVOLTAICS REPORT.
- GUS. (2006). Zasady metodyczne sprawozdawczości statystycznej z zakresu gospodarki paliwami i energią oraz definicje stosowanych pojęć.

GUS. (2016a). W Zasady metodyczne badań statystycznych z zakresu energii ze źródeł odnawialnych.

GUS. (2016b). ZASADY METODYCZNE BADAŃ STATYSTYCZNYCH Z ZAKRESU ENERGII ZE ŹRÓDEŁ ODNAWIALNYCH.

GUS. (2020a). Energia ze źródeł odnawialnych w 2019 r.

GUS. (2020b). Energia ze źródeł odnawialnych w 2019 roku.

Hijmans, R. J. (2021). *terra: Spatial Data Analysis* [R package version 1.3-4]. <https://CRAN.R-project.org/package=terra>

International Energy Agency). (2020). Key World Energy Statistics 2020.

Jäger-Waldau, A. (2019). Snapshot of Photovoltaics—February 2019. *Energies*, 12(5). <https://doi.org/10.3390/en12050769>

Komisja Europejska. (2018). A Clean Planet for all A European long-term strategic vision for a prosperous, modern, competitive and climate neutral economy.

Krakowski Alarm Smogowy. (2020). Wpływ zanieczyszczeń powietrza na zdrowie.

Marszałek, K. (2021). *Fotowoltaika*. Akademia Górnictwo-Hutnicza.

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.

Minister Gospodarki. (2010). Krajowy plan działań w zakresie energii ze źródeł odnawialnych.

Mitchell, T. M. i in. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37), 870–877.

Parlament Europejski. (2019). Europejski Zielony Ład.

Pebesma, E. (2018). Simple Features for R: Standardized Support for Spatial Vector Data. *The R Journal*, 10(1), 439–446. <https://doi.org/10.32614/RJ-2018-009>

Perez, M. & Perez, R. (2015). A Fundamental Look at Supply-Side Energy Reserves for the Planet (International Energy Agency, Solar Heating and Cooling Program Newsletter, vol 62).

Prawo Energetyczne. (1997). Ustawa z dnia 10 kwietnia 1997 r.

Raschka, S. & Mirjalili, V. (2017). Python Machine Learning: Machine Learning and Deep Learning with Python. *Scikit-Learn, and TensorFlow. Second edition ed.*

Redmon, J. (2013–2016). Darknet: Open Source Neural Networks in C.

Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection.

- Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016a). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016b). You Only Look Once: Unified, Real-Time Object Detection.
- REN21. (2021). RENEWABLES 2021 GLOBAL STATUS REPORT.
- Rosenblatt, F. (1957). *The Perceptron, a Perceiving and Recognizing Automaton Project Para*. Cornell Aeronautical Laboratory. https://books.google.pl/books?id=P%5C_XGPgAACAAJ
- Rozporządzenie Komisji Europejskiej. (2016). ROZPORZĄDZENIE KOMISJI (UE) 2016/631 z dnia 14 kwietnia 2016 r. ustanawiające kodeks sieci dotyczący wymogów w zakresie przyjęcia jednostek wytwarzających do sieci.
- Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088), 533–536.
- Russell, S. & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach* (3rd ed.). Prentice Hall.
- Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), 210–229. <https://doi.org/10.1147/rd.33.0210>
- Sebastian Raschka, V. M. (2019). *Python. Machine learning i deep learning. Biblioteki scikit-learn i TensorFlow 2*. Simon; Schuster.
- Traktat o Funkcjonowaniu Unii Europejskiej. (2019).
- Tytko, R. (2020). *Urządzenia i systemy energetyki odnawialnej*. Wydawnictwo i Drukarnia Towarzystwa Słowaków w Polsce.
- Urząd Regulacji Energetyki. (2021a). Raport dotyczący energii elektrycznej wytworzonej z OZE w mikroinstalacji i wprowadzonej do sieci dystrybucyjnej (art. 6a ustawy o odnawialnych źródłach energii).
- Urząd Regulacji Energetyki. (2021b). Wytwarzanie energii elektrycznej w Polsce w małych instalacjach OZE.
- Ustawa o odnawialnych źródłach energii. (2015). Ustawa z dnia 20 lutego 2015 r. o odnawialnych źródłach energii.
- Ustawa o zmianie ustawy Prawo budowlane oraz niektórych innych ustaw. (2020). Ustawa z dnia 13 lutego 2020 r. o zmianie ustawy - Prawo budowlane oraz niektórych innych ustaw.

Weizenbaum, J. (1983). ELIZA — a Computer Program for the Study of Natural Language Communication between Man and Machine. <https://doi.org/10.1145/357980.357991>

Spis tabelic

1.1	Potencjał energetyczny z wybranych źródeł w skali roku w TWrok według M. Perez i R. Perez (2015)	7
1.2	Mikroinstalacje OZE w podziale na rodzaj odnawialnego źródła energii	20
1.3	Liczba prosumentów przyłączonych do sieci poszczególnych OSD w latach 2018-2020	21
1.4	Małe instalacje OZE w podziale na rodzaj odnawialnego źródła energii	22
2.1	Różnice między uczeniem głębokim, a uczeniem maszynowym	31
3.1	Liczba instalacji na koniec 2020 roku	50
3.2	Dostępne układy graficzne w Google Colab [na dzień 6.07.2021]	61
4.1	Modele oraz liczliwość zbiorów uczących i testowych	70
4.2	Tabela klasyfikacji dla zbioru treningowego dla 2020 roku	71
4.3	Tabela klasyfikacji dla zbioru testowego dla 2020 roku	72
4.4	Weryfikacja sieci z 2020 roku na danych z 2018 i Skórzewa	73
4.5	Tabela klasyfikacji dla zbioru treningowego dla 2018 roku	75
4.6	Tabela klasyfikacji dla zbioru testowego dla 2018 roku	75
4.7	Weryfikacja sieci z 2018 roku na danych z 2020 i Skórzewa	75
4.8	Tabela klasyfikacji dla zbioru treningowego dla 2018 i 2020 roku	76
4.9	Tabela klasyfikacji dla zbioru testowego dla 2018 i 2020 roku	77
4.10	Weryfikacja sieci z 2018 i 2020 roku na danych z 2020 i Skórzewa	78
4.11	Spis skrótów używanych w pracy	91

Spis rysunków

1.1	Trendy historyczne i postępy w realizacji celów na lata 2020 w UE-28	9
1.2	Moce PV zainstalowane w różnych krajach w latach 2010-2019	11
1.3	Udział poszczególnych rodzajów ogniw w całkowitej produkcji energii ze słońca	12
1.4	Kolektor słoneczny skupiający	13
1.5	Dotacje z programu Mój Prąd	24
1.6	Przykładowy wygląd dachu pochodzącego z Google Maps na bazie zdjęcia satelitarnego	26
1.7	Porównanie jakości ortofotomap dla miasta Poznania	27
2.1	Uczenie maszynowe: nowy model programowania	29
2.2	Sztuczna inteligencja, uczenie maszynowe i uczenie głębokie	30
2.3	Ogólna budowa perceptronu	32
2.4	Porównanie perceptronu oraz Adaline	33
2.5	Schemat wielowarstwowej sieci neuronowej	35
2.6	Architektura Darknet	40
2.7	Część wspólna przez sumę zbiorów – decyzja o wyborze określonego pola ograniczającego (bounding box)	41
2.8	Przykładowe działanie YOLO	42
2.9	Macierz błędu	43
2.10	Krzywa precyza-czułość	45
2.11	Sposób działania CutMix	48
2.12	Architektura YOLO v4	48
3.1	Schemat przygotowania danych	50
3.2	Rozkład instalacji fotowoltaicznej w Poznaniu (lewy) oraz Skórzewie (prawy) . .	51

3.3	Wycięty dach wraz z instalacją fotowoltaiczną (lewy górny), wycięty dach bez instalacji fotowoltaicznej (prawy górny), faktyczne położenie instalacji fotowoltaicznej (lewy dolny) i przykład wyciętego dachu bez instalacji fotowoltaicznej w pobliżu (prawy dolny)	52
3.4	Wizualizacja procedury wycinania dachów z ortofotomap na podstawie danych dostarczonych przez operatora	56
3.5	Wygląd CVAT po zalogowaniu	57
3.6	Wygląd CVAT po utworzeniu zadań	58
3.7	Wygląd CVAT w trakcie oznaczania	58
3.8	Przykład zaznaczania instalacji fotowoltaicznej jako osobnych obiektów (lewy górny), przykład zaznaczenia instalacji fotowoltaicznej jako jednego obiektu (prawy górny), przykład dachu z ciężko identyfikowaną instalacją (dolny) . . .	59
3.9	Zrzut ekranu z serwisem Google Colab	61
3.10	Przykład augmentacji próbki poprzez algorytm Cutmix (lewy górny), rozmycia zdjęcia (prawy górny) i zmianę odcienia koloru zdjęcia (dolny)	67
4.1	Krzywa średniej straty oraz mAP w trakcie trenowania modelu na danych z Poznania dla 2020 roku	72
4.2	Przykładowe predykcje modelu 2020 na zbiorze danych z 2018 roku	73
4.3	Krzywa średniej straty oraz mAP w trakcie trenowania modelu na danych z Poznania dla 2018 roku	74
4.4	Przykładowe predykcje modelu 2018 na zbiorze danych z 2020 roku	76
4.5	Krzywa średniej straty oraz mAP w trakcie trenowania modelu na danych z Poznania dla 2018 i 2020 roku	77
4.6	Przykładowe predykcje modelu 2018 i 2020 na zbiorze danych ze Skórzewa . .	78

Spis programów

3.1 Przykład pobrania ortofotomapy z wykorzystaniem pakietu <code>rgugik</code> z rejestru GUGiK	53
3.2 Wycinanie fragmentu ortofotomapy zawierającej panel lub kolektor	54
3.3 Wydruk procedury <code>nvidia-smi</code> w serwisie Google Colab	62
3.4 Informacje o wersji CUDA zainstalowanej w serwisie Google Colab	62
3.5 Aktywowanie poprawnej wersji CUDA w języku Python	63
3.6 Skrypt zamieniający poszczególne wartości w pliku <code>makefile</code>	64
3.7 Połączenie Google Colab i Google Drive	64
3.8 Kopiowanie danych z Google Drive na Google Colab	65
3.9 Tworzenie zbioru dla próby uczącej i testowej	65

Dodatek: Skróty

Tablica 4.11. Spis skrótów używanych w pracy

Skrót	Znaczenie
GUS	Główny Urząd Statystyczny
GW	gigawat
YOLO	You Only Look Once
OZE	odnawialne źródła energii
MToe	Megaton oleju ekwiwalentnego
ETS	Emissions Trading System
SHARES	SHort Assessment of Renewable Energy Sources
URE	Urząd Regulacji Energetyki
IoU	Intersection over Union (pol. <i>Indeks Jaccarda</i>)
OSD	Operator Sieci Dystrybucyjnej