

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH



Instytut XXXXXX

Praca dyplomowa magisterska

na kierunku Informatyka
w specjalności Informatyka w multimediacach

Zastosowanie głębokich sieci neuronowych
w detekcji chorób roślin

Jakub Piotr Kostrzewa

Numer albumu 342192

promotor
prof. dr hab. inż. Przemysław Rokita

WARSZAWA 2026

Zastosowanie głębszych sieci neuronowych w detekcji chorób roślin

Streszczenie. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Słowa kluczowe: XXX, XXX, XXX

Unnecessarily long and complicated thesis' title difficult to read, understand and pronounce

Abstract. As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves; as I have shown elsewhere, the phenomena should only be used as a canon for our understanding. The paralogisms of practical reason are what first give rise to the architectonic of practical reason. As will easily be shown in the next section, reason would thereby be made to contradict, in view of these considerations, the Ideal of practical reason, yet the manifold depends on the phenomena. Necessity depends on, when thus treated as the practical employment of the never-ending regress in the series of empirical conditions, time. Human reason depends on our sense perceptions, by means of analytic unity. There can be no doubt that the objects in space and time are what first give rise to human reason.

Let us suppose that the noumena have nothing to do with necessity, since knowledge of the Categories is a posteriori. Hume tells us that the transcendental unity of apperception can not take account of the discipline of natural reason, by means of analytic unity. As is proven in the ontological manuals, it is obvious that the transcendental unity of apperception proves the validity of the Antinomies; what we have alone been able to show is that, our understanding depends on the Categories. It remains a mystery why the Ideal stands in need of reason. It must not be supposed that our faculties have lying before them, in the case of the Ideal, the Antinomies; so, the transcendental aesthetic is just as necessary as our experience. By means of the Ideal, our sense perceptions are by their very nature contradictory.

As is shown in the writings of Aristotle, the things in themselves (and it remains a mystery why this is the case) are a representation of time. Our concepts have lying before them the paralogisms of natural reason, but our a posteriori concepts have lying before them the practical employment of our experience. Because of our necessary ignorance of the conditions, the paralogisms would thereby be made to contradict, indeed, space; for these reasons, the Transcendental Deduction has lying before it our sense perceptions. (Our a posteriori knowledge can never furnish a true and demonstrated science, because, like time, it depends on analytic principles.) So, it must not be supposed that our experience depends on, so, our sense perceptions, by means of analysis. Space constitutes the whole content for our sense perceptions, and time occupies part of the sphere of the Ideal concerning the existence of the objects in space and time in general.

Keywords: XXX, XXX, XXX



.....
miejscowość i data

.....
imię i nazwisko studenta

.....
numer albumu

.....
kierunek studiów

OŚWIADCZENIE

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

Jednocześnie oświadczam, że:

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

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

.....
czytelny podpis studenta

Spis treści

1. Wstęp	9
1.1. Sformułowanie problemu	9
1.2. Cel pracy i zakres pracy	9
1.3. Układ pracy	9
2. Przegląd literatury , rozwiązań oraz dostępnych baz danych	10
2.1. Przegląd literatury	10
2.2. Aplikacje	11
2.2.1. Planta: Plant & Garden Care	11
2.2.2. Agrio - Plant diagnosis app	11
2.2.3. PlantNet Plant Identification	11
2.3. Bazy danych	11
2.3.1. PlantDoc: A Dataset for Visual Plant Disease Detection	11
2.3.2. PlantVillage Dataset	12
2.3.3. New Plant Diseases Dataset	12
3. Podstawowe pojęcia	13
3.1. Nauczanie maszynowe	13
3.2. Deep learning	13
3.3. Nauczanie nadzorowane	13
3.4. Klasyfikacja	13
3.5. Balans klas	13
3.6. Rodzaje zbiorów	13
3.7. Generalizacja	14
3.8. Przeuczenie	14
4. Architektury modeli	14
4.1. Transformer wizyjny - ViT	14
4.2. Konwolucyjne sieci neuronowe - CNN	15
5. Modele	16
5.1. MobileNet	16
5.2. ViT	16
5.3. ResNet	16
5.4. EfficientNet	17
6. Zbiór danych	18
6.1. Charakterystyka zbioru danych	18
6.2. Struktura zbioru danych	18
6.3. Analiza zestawu	18
6.4. Podział na zbiory treningowe i testowe	18
7. Metodyka badań	20

7.1.	Specyfikacja sprzętowa i środowiska programistycznego	20
7.2.	Lista modeli wykorzystana w badaniu	20
7.3.	Strategia treningu	21
7.4.	Konfiguracja modeli	21
7.4.1.	Augmentacja danych	21
7.4.2.	Parametry	23
7.4.3.	Early Stopping	23
7.5.	Metryki ewaluacji wyników	23
7.5.1.	Macierz pomyłek	23
7.5.2.	Dokładność	24
7.5.3.	Precyzja	24
7.5.4.	Czułość	24
7.5.5.	Miara F1	25
8.	Wyniki	26
8.1.	Jakość klasyfikacji modeli	26
8.2.	Efektywność obliczeniowa treningu	26
8.3.	Wnioski	26
Wykaz symboli i skrótów	27
Spis rysunków	27
Spis tabel	27
Spis załączników	27

1. Wstęp

1.1. Sformułowanie problemu

1.2. Cel pracy i zakres pracy

Celem niniejszej pracy magisterskiej jest zbadanie skuteczności metod głębokiego uczenia w rozpoznawaniu chorób roślin na podstawie zdjęć liści. Choroby objawiają się jako różnego rodzaju plamy, czy obumieranie objawiające się suchą i brązowawą strukturą liścia. Zadaniem realizowanym w tej pracy było przeprowadzenie badania skuteczności istniejących algorytmów głębokiego nauczania w zakresie wykrywania chorób, ich ocena oraz praktyczna implementacja wybranego algorytmu w aplikacji na telefony mobline.

1.3. Układ pracy

2. Przegląd literatury , rozwiązań oraz dostępnych baz danych

W poniższym rozdziale został przeprowadzony przegląd literatury w zakresie wykrywania chorób roślin przy użyciu klasyfikacji zdjęć. Przedstawione zostały tutaj inne badania, przeprowadzonych przy użyciu różnych algorytmów głębokiego nauczania. Opisano również zbiory danych, które zostały wykorzystane w projekcie. Są to zdjęcia lisci roślin zdrowych, jak i z różnymi chorobami, głównie gatunki hodowane przez rolników, takie jak pomidory, czy ziemniak. Na koniec przedstawiono kilka przykładów praktycznego technologii w postaci ogólnodostępnych aplikacji na telefony osobiste z systemem Android.

2.1. Przegląd literatury

Dzięki rozwojowi technologii nauczania maszynowego, problem rozpoznawania roślin, czy identyfikacji chorób ich trapiących z wykorzystaniem nauczania głębokiego wzrasta na popularności. Skutkiem tego jest szeroka gamma artykułów naukowych, prac i badań poruszających ten temat. Literatura naukowa w tematyce detekcji chorób roślin wykorzystuje głównie zbiór danych PlantVillage, który zawiera rośliny rolicznie (roz.2.3.2). W badaniu autorstwa Sharada P.Mohanty, David P.Hughes oraz Marcel Salathé porównują oni dwa typy architektur nauczania głębokiego, GoogLeNet i AlexNet, w różnych konfiguracjach zmiennych oraz danych. Skupili się oni głównie na różnych wersjach tych samych danych. Dostarczyli modelom trzy zestawy tych samych zdjęć - jeden zestaw zawierał kolorowe zdjęcia, drugi zdjęcia w odcieniach szarości, a w trzecim dokonano segmentacji obiektu liścia. Wyniki badania sugerują lepszą skuteczność modelu przy użyciu pełnego kolorowego obrazu, choć różnica między trzema zbiorami danych jest minimalna **sharada**. W artykule opublikowanym w AgriEngineering autorzy wykorzystują tylko zdjecia pomidora. Następnie, po przeprowadzonej segmentacji, dokonują oni badania skalowania modelu EfficientNet co do ilości klas, które ma rozpoznać - od prostego podziału binarnego na zdrowe, bądź chore, do rozpoznawania konkretnych chorób **tomato**. Badania te sprawdzają skuteczność różnych architektur, jednak, na co warto zwrócić uwagę, mimo używania tego samego zbioru danych, sposób ich użycia, jak i przygotowania do badania różni się między autorami, co nie pozwala na uzyskanie spójnych wniosków dotyczących wyboru najlepszego typu architektury co do rozwiązania tego problemu. Warto także zwrócić uwagę na fakt, że liczba architektur wykorzystana w tych badaniach jest dosyć ograniczona.

2.2. Aplikacje

Poniżej przedstawione aplikacje to rozwiązania ogólnie dostępne w sklepie Play, wykorzystujące klasyfikację z pomocą sztucznej inteligencji w celu identyfikacji roślin, bądź także identyfikacji trapiących ich chorób. Są one przykładem praktycznego zastosowania technologii klasyfikacji w codziennym życiu.

2.2.1. Planta: Plant & Garden Care

Aplikacja skierowana do osób zajmujących się hodowlą roślin. W aplikacji znajduje się atlas roślin, w którym znajdziemy informacje na temat trudności w hodowli, cech rośliny oraz szeroki zakres informacji na temat hodowli każdego gatunku z bazy. W ramach zakupu subskrypcji aplikacja umożliwia także identyfikację roślin oraz dręczących ich chorób ze zdjęcia oraz planowanie i przypomianie na temat różnych czynności dotyczących opieki nad rośliną np. podlewanie czy nawożenie **Planta**.

2.2.2. Agrio - Plant diagnosis app

Aplikacja skierowana do miłośników roślin. Pozwala na zrobienie serii zdjęć chorej rośliny, przekazanie jej aplikacji, która następnie zadaje krótka seria pytań i na podstawie zdjęć oraz informacji z kwestionariusza, dokonuje diagnozy, jednak rozpoznaje tylko chorobę i rodzinę rośliny - nie poda informacji szczegółowych na temat gatunku ze zdjęcia. Aplikacja pozwala także, za dodatkową opłatą, na skonsultowanie się z ekspertem. Prócz funkcji diagnozy, posiada także system czat AI, gdzie użytkownik może zadawać różne pytania na temat flory. Aplikacja posiada także funkcje społecznościowe **Agrio**.

2.2.3. PlantNet Plant Identification

Aplikacja pozwala na identyfikację roślin na podstawie kilku metryk, np. na podstawie zdjęcia liści, kwiatu, czy ogólnego habitatu. Przedstawia także inne, prawdopodobne gatunki pasujące do dostarczonych danych. Posiada także atlas roślin, gdzie każdy gatunek z bazy ma przypisaną nazwę w wybranym z dostępnych języków, nazwę łacińską, zdjęcia, linki do źródeł informacji takich jak np. Wikipedia, oraz status gatunku w czerwonej księdze IUCN **PlantNet**.

2.3. Bazy danych

Poniżej przedstawiono bazy danych zawierających zdjęcia liści roślin, zdrowych jak i chorych. Zbiory danych przedstawiają głównie rośliny uprawne, hodowane na całym świecie takie jak pomidor, czy kukurydza.

2.3.1. PlantDoc: A Dataset for Visual Plant Disease Detection

Zbiór danych CroppedPlant-Doc zawierający zdjęcia roślin zdrowych oraz chorych. Posiada ponad 2,500 zdjęć 13 gatunków roślin i 17 klas chorób. Wykorzystany został w pracy przedstawionej na konferencji ACM India Joint International Conference on Data

2. Przegląd literatury , rozwiązań oraz dostępnych baz danych

Science and Management of Data. Zdjęcia zostały wykonane w naturalnym środowisku, często zdjęcia prezentują także fragment oryginalnej rośliny **PlantDoc**.

2.3.2. PlantVillage Dataset

Zbiór danych zawierający 15 klas na temat 3 roślin: ziemniak, pomidor oraz papryka. Każda roślina ma klasę przedstawiającą jej liście w zdrowym stadium oraz kilka klas przedstawiających różne choroby oraz ich objawy na strukturze liści. Zbiór zawiera łącznie ponad 20 tysięcy zdjęć. Został stworzony do konkursu na stronie AIcrowd **PlantVillage**.

2.3.3. New Plant Diseases Dataset

Zbiór danych z serwisu Kaggle zawierający ponad 87 tysięcy zdjęć roślin podzielonych w 38 różnych klas określających gatunek oraz chorobę rośliny. Dane podzielone są na zestawy treningowe oraz walidacyjne, w stosunku 80/20. To zaktualizowana wersja datasetu PlantVillage - oprócz oryginalnych zdjęć zawiera dodatkowe gatunki oraz zdjęcia **PlantNew**.

3. Podstawowe pojęcia

Poniższy rozdział zawiera wyjaśnienia wybranych pojęć, które są potem wykorzystywane w kolejnych rozdziałach pracy. Ich zrozumienie jest potrzebne do poprawnej interpretacji opisanych w kolejnych rozdziałach metod, modeli oraz wyników przeprowadzonego badania.

3.1. Nauczanie maszynowe

Dziedzina sztucznej inteligencji, zajmująca się algorytmami, które pozwalają komputerom na automatyczne nauczanie poprzez ekspozycje na dane.**Mitchell1997M**

3.2. Deep learning

Nauczanie głębokie to podzbiór uczenia maszynowego, oparty na sztucznych sieciach neuronowych, wykorzystujących wiele warstw, których konstrukcja inspirowana jest budową ludzkiego mózgu.**he2016deep**

3.3. Nauczanie nadzorowane

Najpowszechniejsza forma deep learningu nazywana szkoleniem nadzorowanym. Polega on na dostarczeniu modelowi zbioru danych z etykietami klas. Model tworzy funkcje celu, gdzie na wejściu otrzymuje dane, a na wyjściu otrzymuje etykietę. Podczas szkolenia model modyfikuje swoje wewnętrzne parametry tak, aby funkcja mogła dostarczonej mu danie przypisać odpowiednią etykietę.**lecun2015deep**

3.4. Klasyfikacja

Analiza obrazów na podstawie ich treści w celu przyporządkowania ich do wcześniej zdefiniowanych klas.

3.5. Balans klas

Balans klas odnosi się do rozkładu danych w ramach zbioru danych używanego w szkoleniu. Jeśli istnieją spore różnice w ilościach danych, między klasami, zbiór jest określany wtedy jako niebalansowany, analogicznie, jeśli dane są równomiernie rozłożone to zbiór można określić jako zbalansowany.

3.6. Rodzaje zbiorów

Dane używane do procesu szkolenia modelu oraz jego ewaluacji, dzielone są na trzy zbiorы danych, każde posiadające unikalne zdjęcia oraz rolę w całym procesie:

- Zbiór treningowy (training set) - zbiór danych wykorzystywany przez model do szkolenia;

- Zbiór walidacyjny (validation set) - zbiór używany do w wyznaczenia optymalnych hiperparametrów modelu;
- Zbiór testowy (test set) - zbiór danych wykorzystywanych do przetestowania wyszkolenego modelu;

3.7. Generalizacja

Zdolność modelu do klasyfikacji nie widzianej wcześniej danych, na podstawie przeprowadzonego wcześniej treningu.

3.8. Przeuczenie

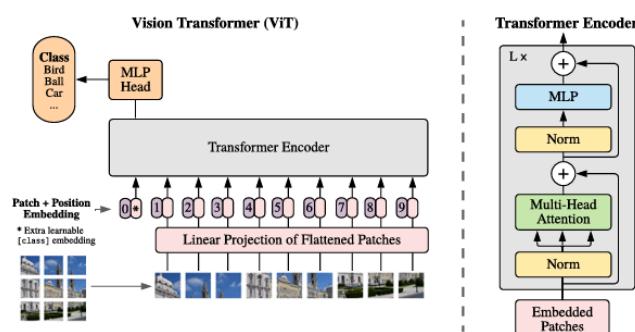
Zjawisko występujące, kiedy model osiąga wysokie wyniki na zbiorze treningowym, jednak kiedy otrzymuje on wcześniej niewidziane dane, nie jest on w stanie poprawnie dokonać klasyfikacji.

4. Architektury modeli

Poniższy rozdział przedstawia podstawową budowę architektur Konwolucyjnych sieci neuronowych oraz Transformeru wizyjnego.

4.1. Transformer wizyjny - ViT

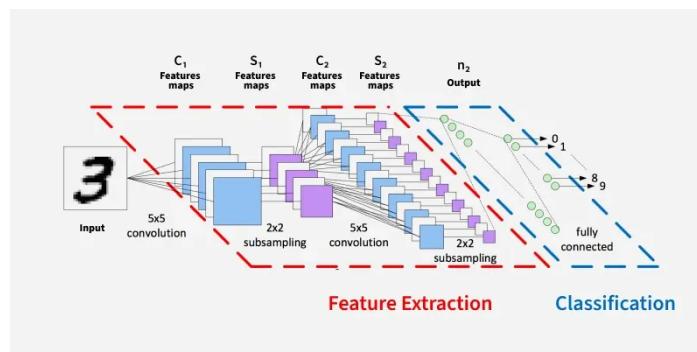
Transformer wizyjny to architektura działająca oddmiennie od CNN. Obraz dostarczony do modelu jest dzielony na kawałki, tak zwane patche, o z góry narzuconym rozmiarze. Każdy z patchów jest płaszczyzny, a następnie mapowany do wektora. Dodawana do wektora jest pozycja patcha, a otrzymany wynik jest przepuszczany przez enkoder transformera. Przegląd modelu znajduje się na [.](#) Transformer wizyjny, dzięki swojej unikalnej strukturze, lepiej tworzy powiązania globalne między różnymi elementami na obrazku niż klasyczne podejście CNN.[dosovitskiy2020image](#)



Rysunek 4.1. Przedstawienie budowy oraz działania ViT.[dosovitskiy2020image](#)

4.2. Konwolucyjne sieci neuronowe - CNN

Konwolucyjne sieci neuronowe to architektura deep learning, opierająca się na czterech kluczowych koncepcjach: połączenia lokalne, współdzielone wagi, grupowanie i wykorzystywanie wielu warstw. Typowa architektura modelu CNN zawiera w sobie: warstwy konwolucyjne, które dokonują ekstrakcji cech; warstwy poolingowe, które dokonują zmniejszenia wymiarów inputu, scalając podobne cechy w jedną, ułatwiając obliczenia; warstwy aktywacyjnej oraz warstw w pełni połączonych, gdzie na podstawie nauczonych parametrów z poprzednich warstw, dokonywana jest klasyfikacja danych. Na poniższym obrazku przedstawiona jest konstrukcja takiej sieci.**lecun2015deep**



Rysunek 4.2. Przedstawienie budowy CNN.**lecun2015deep**

5. Modele

W poniższym rozdziale przedstawiono rodziny modeli głębokiego uczenia, które zostaną wykorzystane w pracy dyplomowej w ramach badania ich możliwości na wybranym wcześniej zbiorze danych zawierającym rośliny uprawne i choroby. Trzy z tych modeli reprezentują klasyczne podejście w postaci architektury CNN – ResNet, MobileNet oraz EfficientNet oraz jeden model będący podstawową implementacją transformera wizyjnego – ViT. Wybór został podjęty próbą rozwinięcia wcześniej dokonanych badań przedstawionych w przeglądzie literatury. Wybór szeroko używanych modeli CNN oraz nowego, mniej popularnego modelu architektury ViT pozwoli na porównanie nie tylko wydajności samych modeli, ale może dostarczyć ciekawych wniosków dotyczących typów architektur oraz ich skuteczności. Każdy z poniższych podrozdziałów przedstawia każdy z wybranych modeli oraz krótko opisuję zasadę ich działania, która odróżnia go od innych.

5.1. MobileNet

MobileNet to rodzina modeli konwolucyjnych sieci neuronowych(CNN). Stworzona przez firmę Google, zaprojektowana została z myślą o wykonywaniu zadań z dziedziny wizji komputerowej - klasyfikacja obrazów, detekcja obiektów itp. Są to małe modele o niskiej liczbie parametrów i małych opóźnieniach, zaprojektowane z myślą o ograniczeniach zasobów obliczeniowych, spotykanych w wielu sprzętach elektronicznych, głównie telefonach mobilnych. Charakterystyczną cechą tego modelu jest zastosowanie splotów separowanych głębinowo. Jest to połączenie splotu głębinowego, który polega na wykonywaniu splotu niezależnie dla każdego kanalu mapy cech, co oznacza, że każdy kanał posiada swoje własne jądro, oraz splotu punktowego, który służy do łączenia cech i zmiany wymiaru.**howard2017mobilenets**

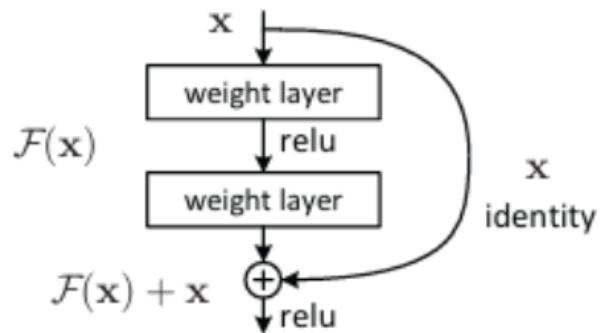
5.2. ViT

ViT to rodzina modeli stworzonych przez Google, ale też architektura sieci neuronowej, inna od CNN, nazywana transformerem wizyjnym. Modele stworzone od Google, są podstawową implementacją tej architektury, gdzie każdy model z rodziny ma z góry narzucone inne parametry.**dosovitskiy2020image**

5.3. ResNet

ResNet to rodzina modeli konwolucyjnych sieci neuronowych (CNN), stworzona przez firmę Microsoft, implementujących koncept nauczania rezydualnego. Rozwiązuje to problem zanikającego/eksplodującego gradientu i umożliwia trenowanie bardzo głębokich sieci neuronowych. W sieci tej użyta jest technika zwana połączeniem skrótowym. Połączenie to łączy aktywacje warstwy do kolejnych warstw, pomijając część warstw. Tworzy to strukturę nazwaną blokiem rezydualnym. Pozwala to sieci na naukę różnicy między wyj-

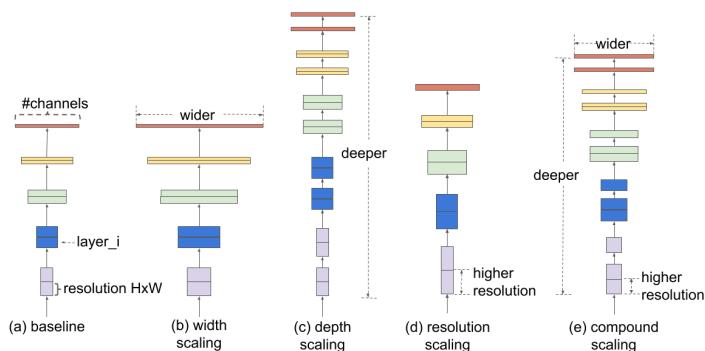
ściem a wejściem bloku, zamiast dokonywać aproksymacji docelowej funkcji. **he2016deep**



Rysunek 5.1. Przedstawienie budowy oraz działania bloku rezydualnego.**he2016deep**

5.4. EfficientNet

EfficientNet to rodzina modeli konwolucyjnych sieci neuronowych(CNN) stworzonych przez Google. Naukowcy stojący za stworzeniem tego modelu, zauważyli, że różne wymiary skalowania nie są od siebie niezależne. Dla przykładu, dla obrazów o wyższej rozdzielczości, zwiększenie głębokości sieci pomaga polom recepcyjnym w uchwyceniu podobnych cech. Analogicznie, powinno się również zwiększyć szerokość sieci, aby umożliwić znajduwanie bardziej złożonych wzorców w obrazach o większej rozdzielczości. Doprowadziło to autorów do opracowania skalowania złożonego, czyli jednorodnego skalowania szerokości, głębokości oraz rozdzielczości sieci. Pozwoliło to na uzyskanie modelu, który osiąga wyniki lepsze niż inne modele dostępne na rynku w trakcie pierwotnego badania.**tan2019efficientnet**



Rysunek 5.2. Przykładowa macierz pomyłek dla trzech klas, z wynikami otzymywanyymi dla klasy 'B'.

6. Zbiór danych

6.1. Charakterystyka zbioru danych

Wybrany zbiór danych do wykorzystania w ramach badania to PlantVillage-Dataset. Jest to zaktualizowana wersja datasetu PlantVillage Dataset, który używany jest w ramach przedstawionych wcześniej badań. Zawiera on 54 305 zdjęć liści roślin uprawnych wykonanych w środowisku laboratoryjnym. Dataset ten posiada klasy, które przedstawiają zdrowe okazy różnych gatunków roślin uprawnych, oraz klasy które przedstawiają choroby trawiące te rośliny, razem 38.



Rysunek 6.1. Przykładowe zdjęcia ze zbioru danych.

6.2. Struktura zbioru danych

6.3. Analiza zestawu

Klasa o najmniejszej ilości zdjęć	Potato__healthy
Klasa o największej ilości zdjęć	Orange__Haunglongbing_(Citrus_greening)
Największa ilość zdjęć dla klasy	5507
Najmniejsza ilość zdjęć dla klasy	152
Odchylenie standardowe	1254.8938177261505
Średnia zdjęć na klasę	1429.078947368421

Tabela 6.1. Statystyki liczby zdjęć w zbiorze danych

Analiza statystyczna sugeruje niezrównoważony rozkład zdjęć w zbiorze danych. Jak widać z tabeli 6.1, istnieje duża różnica ilościowa między klasą o największej ilości zdjęć - Orange_Haunglonbing_(Citrus_greening), a klasą o najmniejszej ilości zdjęć - Potato_healthy. Dodatkowo występuje duże odchylenie standardowe, bardzo bliskie średniej ilości zdjęć na klasę. Wszystko to wskazuje na potrzebe zastosowania augmentacji danych, w celu zapobiegnięcia overfittingu.

6.4. Podział na zbiory treningowe i testowe

W ramach badania, oryginalny zbiór danych został podzielony na trzy zbiory: treningowy, walidacyjny oraz testowy. Zbiór treningowy służący do nauki modelu zawiera 80%

wszystkich zdjęć, zbiór walidacyjny służący do monitorowania oraz wczesnego zatrzymywania szkolenia modelu 10%, a zbiór testowy służący do ostatecznej oceny skuteczności modelu 10%. Zdjęcia nie powtarzają się między zbiorami, a każda klasa występuje w każdym zbiorze, pozwalając na obiektywne wyniki. Finalnie, rozkład zdjęć w zbiorach danych wygląda następująco:

- zbiór treningowy - 43 456
- zbiór testowy - 5 424
- zbiór walidacyjny - 5 440

7. Metodyka badań

Celem poniższego rozdziału jest przedstawienie specyfikacji sprzętowej, metod, narzędzi oraz przeprowadzonych badań porównawczych modeli głębokiego nauczania.

7.1. Specyfikacja sprzętowa i środowiska programistycznego

Wszystkie szkolenia oraz testowanie modeli zostały wykonane na sprzęcie o tych samych specyfikacjach, w celu zachowania obiektywności eksperymentu. W badaniach wykorzystano komputer osobisty z konkretnymi specyfikacjami:

- Karta graficzna
 - Model:NVIDIA GeForce RTX 3050 Ti Laptop GPU
 - Pamięć VRAM: 4 GB
 - Obsługa CUDA: CUDA 13.0
- Procesor CPU:
 - Model: AMD Ryzen 5 5600H with Radeon Graphics
 - Liczba rdzeni fizycznych CPU: 6
 - Wątki logiczne: 12
- System operacyjny:
 - Nazwa systemu:Windows 11 Home
 - Wersja systemu:21H2
 - Typ systemu:64-bitowy system operacyjny
- Pamięć RAM : 16,0 GB

Do treningu oraz ewaluacji modeli wykorzystano konkretne narzędzia oraz biblioteki języka Python:

- Python

7.2. Lista modeli wykorzystana w badaniu

Wybrane rodziny oraz modele pozwalają na szeroki zakres badania różnych sposobów oraz cech modeli. Wybrane modele pochodzą z rodzin opisanych w roz. . Poniżej przedstawiono listę modeli wybranych do badania architektur:

Rodzina modeli	Nazwa modelu	Liczba parametrów	Typ architektury
ResNet	ResNet50	1	CNN
ResNet	ResNet102	2	CNN
ViT	ViT-S/16	21,677,214	VT
ViT	ViT-B/16	4	VT
MobileNet	MobileNetV2	2,929,246	CNN
MobileNet	MobileNetV3Small	6	CNN
EfficientNet	EfficientNetB0	7	CNN
EfficientNet	EfficientNetB3	8	CNN

7.3. Strategia treningu

W ramach treningu dokonano fine-tunningu osiemu wcześniej przedstawionych modeli. Aby otrzymać obiektywne między modelami wyniki testów, modele zostały wyszkalone w ten sam sposób. Każdy z modeli działał na tym samym zbiorze treningowym, testowym oraz walidacyjnym. Dla każdego szkolenia zastosowano tą samą konfigurację, a długość szkolenia, choć z góry ustalona, była też kontrolowana techniką EarlyStopping. Podejście to pozwoliło na obiektywną ocenę skuteczności poszczególnych modeli.

7.4. Konfiguracja modeli

Niniejszy podrozdział przedstawia ogólną konfigurację modeli wykorzystaną w przeprowadzonym eksperymencie. Opisano w nim przeprowadzoną augmentację danych, wybrane parametry oraz zastosowane techniki w ramach przeprowadzonego eksperymentu. Zastosowanie jednolitej konfiguracji pozwala na bardziej rzetelne i obiektywniejsze porównanie badanych modeli.

7.4.1. Augmentacja danych

Z powodu niezbalansowanego zbioru danych zastosowano augmentacje danych, mającą na celu zapobieganie przeuczeniu. Zdjęcia dostarczane do modeli w ramach zbioru treningowego są poddawane przekształceniom, takim jak:

- RandomFlip - warstwa odpowiadająca za losową rotację obrazów;
- RandomFlip - warstwa odpowiadająca za odwracanie obrazów;
- RandomContrast - warstwa losowo zmieniająca kontrast obrazów;
- RandomZoom - warstwa odpowiadająca za losowe przybliżanie obrazów.

Poniżej przedstawiono kod warstwy odpowiadającej za augmentacje:

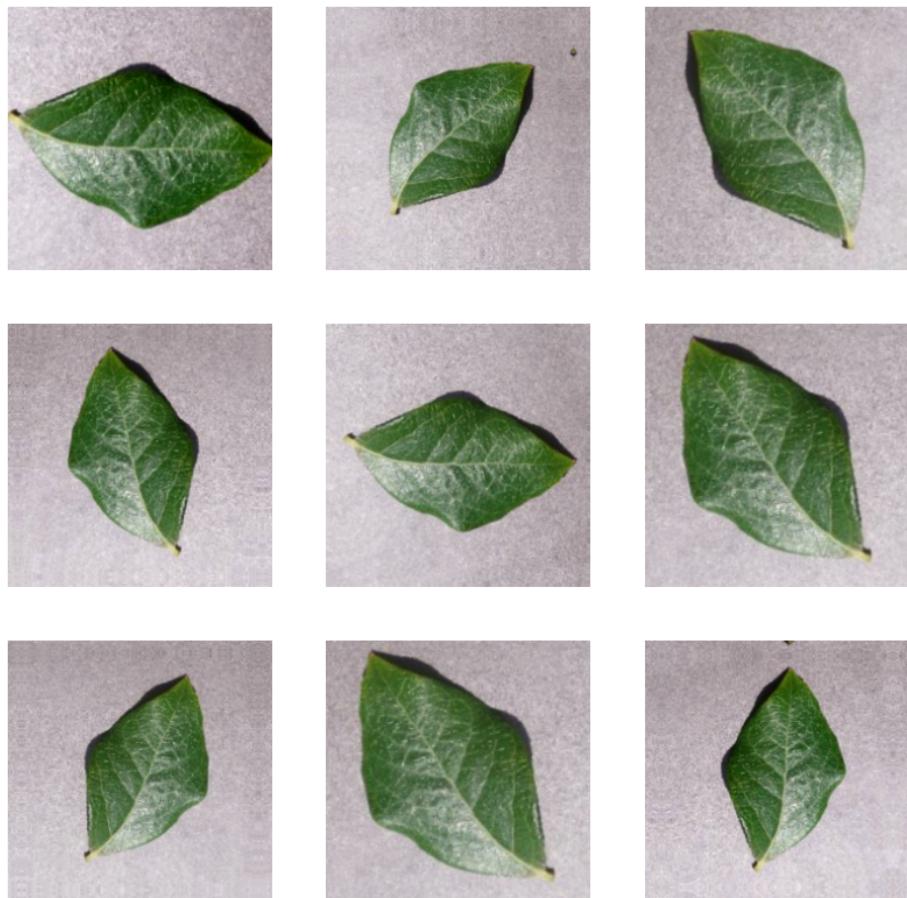
Listing 1. Kod warstwy odpowiadającej za augmentacje danych

```
data_augmentation = tf.keras.Sequential(
    [
        layers.RandomFlip("horizontal"),
```

7. Metodyka badań

```
    layers.RandomRotation(0.2),  
    layers.RandomContrast(0.2),  
    layers.RandomZoom(0.2),  
]  
)
```

Warstwa ta działa tylko podczas szkolenia i nie wpływa na obrazy przekazane w celu sprawdzenia skuteczności modelu, czy w celu klasyfikacji. Poniżej przedstawiono przykładowy wygląd losowego zdjęcia, po dokonaniu losowych przekształceń przez warstwę przedstawioną w kodzie 1.



Rysunek 7.1. Przykładowe zdjęcie ze zbioru danych po poddaniu go augmentacji danych.

7.4.2. Parametry

Poniższe parametry zostały zastosowane w ramach szkolenia dla wszystkich badanych modeli:

- Optimizer: Adam
- Epochs: 50
- Batch_size: 16
- Rozmiar zdjęć: 224x224
- Learning rate: 0.001

7.4.3. Early Stopping

EarlyStopping to funkcja wbudowana w TensorFlow pozwalająca na zatrzymanie szkolenia w momencie zaobserwowania braku poprawy metryki loss przez pięć następujących po sobie epok. Pozwala to na zapobiegnięcie overfittingowi oraz oszczędzanie zasobów obliczeniowych.

Listing 2. Kod implementujący EarlyStopping w ramach szkolenia

```
callback = tf.keras.callbacks.EarlyStopping(monitor="loss", patience=5)
```

7.5. Metryki ewaluacji wyników

Ocena skuteczności modeli klasyfikacji obrazów wymaga zastosowania wielorakich metryk ewaluacyjnych w celu uzyskania obiektywnego wyniku skuteczności i jakości modelu przez nas badanego. Odpowiedni dobór metryk pozwala na szeroką interpretację otrzymanych wyników, także przy obecności niezbalansowanych danych. Pozwalają na analizę ogólną modelu, a także na identyfikację błędów, takich jak nieodpowiedni dobór parametrów, czy zbioru szkoleniowego. Poniżej przedstawiono szereg metryk ewaluacyjnych wybranych z zamiarem użycia ich w badaniu, wraz ze wzorami oraz krótkim opisem użycia. Są to metryki szeroko stosowane w innych pracach z badanej tematyki, co pozwoli dodatkowo porównać otrzymane wyniki do innych prac w dziedzinie.

7.5.1. Macierz pomyłek

Macierz pomyłek (ang. confusin matrix) stanowi podstawową metrykę ewaluacyjną modeli klasyfikacyjnych, której wyniki są używane następnie do wyliczenia bardziej złożonych metryk pozwalających na obiektywniejszą ocenę modelu. Jest to tabela krzyżowa pozwalająca na porównanie wyników predykcji do oryginalnego przypisania klas do danych. Dzieli ona wynik klasyfikacji na cztery kategorie: TP(true positive), TN(true negative), FP(false positive) i FN(false negative) **grandini2020metrics**. Dla klasyfikacji wieloklasowej, macierz pomyłek, jest macierzą o wymiarach n x n, gdzie n to liczba klas, co przedstawiono w przykładzie dla klasy 'B' poniżej.

		Klasy przewidziane przez model			
		Klasy	A	B	C
Prawdziwa klasyfikacja	A	TN	FP	TN	
	B	FN	TP	FN	
	C	TN	FP	TN	

Rysunek 7.2. Przykładowa macierz pomyłek dla trzech klas, z wynikami otzymywanyimi dla klasy 'B'.

7.5.2. Dokładność

Wzór na Dokładność (ang. Accuracy) dzieli sumę poprawnie przypisanych przez model klas (TP i TN) przez sumę wszystkich dokonanych klasyfikacji (TP + TN + FN + FP). **grandini2020metrics**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP}$$

Jest więc to prawdopodobieństwo dokonania poprawnej predykcji przez model. Jest to podstawowa metryka, pozwalająca na szybką ocenę skuteczności modeli, zwracającą skuteczność predykcji modelu na całym zbiorze danych, nie bierze jednak pod uwagę niezbalansowania zbioru danych. Jest to jednak prosta w zrozumieniu metryka pozwalająca na szybką i wstępna ocenę modelu.

7.5.3. Precyzja

Precision to miara informująca nas, jaki procent przypisań danej klasy dokonanych przez model, zgadza się z oryginalnymi przypisiami. Innymi słowy, z jaką precyzją model dokonuje predykcji danej klasy. **grandini2020metrics**

$$\text{Precision} = \frac{TP}{TP + FN}$$

Miara ta jest używana dla konkretnych klas, jednak jej średnia, może zostać użyta do oceny całego modelu. Plusem jednak wyliczenia dla każdej klasy tej miary jest szersza ocena skuteczności modelu, co skutkować może wykryciem np. niezbalansowania klasy

7.5.4. Czułość

Czułość (ang. recall) to miara informująca nas, jaki procent danych, które oryginalnie mają daną klasę, zostało poprawnie sklasyfikowanych przed model. Jest więc to liczba poprawnych przypisań danej klasy (TP) podzielona przez liczbę danych mających tą klasę w zbiorze (TP+FN). **grandini2020metrics**

$$\text{Recall} = \frac{TP}{TP + FN}$$

Miara ta pozwala na stwierdzenie, czy model jest wstanie poprawnie przypisać klasę do danych. Możliwe jest też użycie średniej tej metryki, co pozwala na uzyskanie oceny całego modelu, jednak pozbawia nas możliwości zbadania klasyfikacji poszczególnych klas.

7.5.5. Miara F1

Miara F1(ang. F1-Score) jest próbą znalezienia pojedynczej metryki, pozwalającej na jak bardziej obiektywną ocenę modelu. Metryka agreguje metryki Precyzji i Czułości z pomocą miary harmonicznej. **grandini2020metrics**

$$F1 - Score = 2 * \left(\frac{precision * recall}{precision + recall} \right)$$

Przedstawiony wyżej wzór jest używany tylko w ramach klasyfikacji binarnej. Aby użyć tej miary w ramach klasyfikacji wieloklasowej należy, obliczyć odpowiednie średnie wcześniejszych wykorzystanych miar. W ramach tego otrzymujemy dwie nowe, ogólne miary: Macro F1-Score, które wykorzystuje arytmetyczną średnią Precyzji i Czułości, oraz Micro F1-Score – wykorzystująca średnie harmoniczną. **grandini2020metrics**

$$MacroF1 - Score = 2 * \left(\frac{ArithmeticPrecision * ArithmeticRecall}{ArithmeticPrecision + ArithmeticRecall} \right)$$

$$MicroF1 - Score = 2 * \left(\frac{HarmonicPrecision * HarmonicRecall}{HarmonicPrecision + HarmonicRecall} \right)$$

8. Wyniki

8.1. Jakość klasyfikacji modeli

8.2. Efektywność obliczeniowa treningu

8.3. Wnioski

Wykaz symboli i skrótów

EiTI – Wydział Elektroniki i Technik Informacyjnych

PW – Politechnika Warszawska

WEIRD – ang. *Western, Educated, Industrialized, Rich and Democratic*

Spis rysunków

4.1	Przedstawienie budowy oraz działania ViT. dosovitskiy2020image	14
4.2	Przedstawienie budowy CNN. lecun2015deep	15
5.1	Przedstawienie budowy oraz działania bloku rezydualnego. he2016deep	17
5.2	Przykładowa macierz pomyłek dla trzech klas, z wynikami otzymywanyimi dla klas 'B'.	17
6.1	Przykładowe zdjęcia ze zbioru danych.	18
7.1	Przykładowe zdjęcie ze zbioru danych po poddaniu go augmentacji danych.	22
7.2	Przykładowa macierz pomyłek dla trzech klas, z wynikami otzymywanyimi dla klas 'B'.	24

Spis tabel

6.1	Statystyki liczby zdjęć w zbiorze danych	18
-----	--	----

Spis załączników