

Aplikacja mobilna do rozpoznawania ras kur

Projekt naukowo-wdrożeniowy

Politechnika Wrocławska
Wydział Informatyki i Telekomunikacji

Semestr letni 2023/2024

Weronika Belniak (249048)

Spis treści

1. Cel projektu	2
1.1. Cele szczegółowe:	2
2. Plan działania	2
3. Przegląd literatury	3
4. Wybór technologii	4
4.1. Python.....	4
4.2. Pandas.....	4
4.3. NumPy.....	4
4.4. OpenCV.....	4
4.5. scikit-learn	4
4.6. TensorFlow i Keras	4
4.7. ResNet50V2	5
5. Implementacja	5
5.1. Przygotowanie Danych	5
5.1.1. Zebranie Danych	6
5.1.2. Anotacja Danych.....	6
5.1.3. Przekształcenie i Augmentacja Danych	6
5.2. Budowa Modelu	7
5.3. Kompilacja i trenowanie modelu	7
5.4. Wizualizacja wyników	7
5.5. Uzyskane wyniki	7
5.5.1. Testy na Zbiorze z 6 Rasami i 40 Zdjęciami dla Każdej Rasy.....	7
5.5.2. Testy na Zbiorze z 4 Rasami i 100 Zdjęciami dla Każdej Rasy.....	7
5.5.3. Testy na Zbiorze z 6 Rasami i 100 Zdjęciami dla Każdej Rasy.....	8
5.5.4. Podsumowanie wyników	8
6. Podsumowanie	9
Bibliografia.....	10
Spis rysunków	11
Spis tabel	11

1. Cel projektu

Celem projektu jest stworzenie aplikacji mobilnej wykorzystującej sztuczną inteligencję do automatycznego rozpoznawania ras kur na podstawie zdjęć. Aplikacja ma być narzędziem wspomagającym hodowców, naukowców oraz entuzjastów drobiu ozdobnego w identyfikacji i klasyfikacji różnych ras kur. Automatyzacja tego zadania za pomocą technologii rozpoznawania obrazów usprawni i ułatwi proces identyfikacji konkretnych ras dla osób bez wiedzy eksperckiej.

1.1. Cele szczegółowe:

- Zebranie obszernej bazy danych zdjęć kur różnych ras.
- Opracowanie algorytmów rozpoznawania obrazów za pomocą sztucznej inteligencji.
- Implementacja i testowanie aplikacji mobilnej.

2. Plan działania

Semestr 1:

- Zgromadzenie zdjęć 16 ras (średnio 100 zdjęć na rasę)
- Wybór i edycja zdjęć
- Rozeznanie w modelach
- Wstępne próby stworzenia modelu z części zdjęć

Semestr 2:

- Zakończenie procesu przygotowywania zdjęć (gotowa cała baza danych 1600 zdjęć)
- Dopracowanie modelu w oparciu o wszystkie zdjęcia
- Rozeznanie w tworzeniu aplikacji mobilnych

Semestr 3:

- Projekt i implementacja aplikacji na telefon z systemem Android

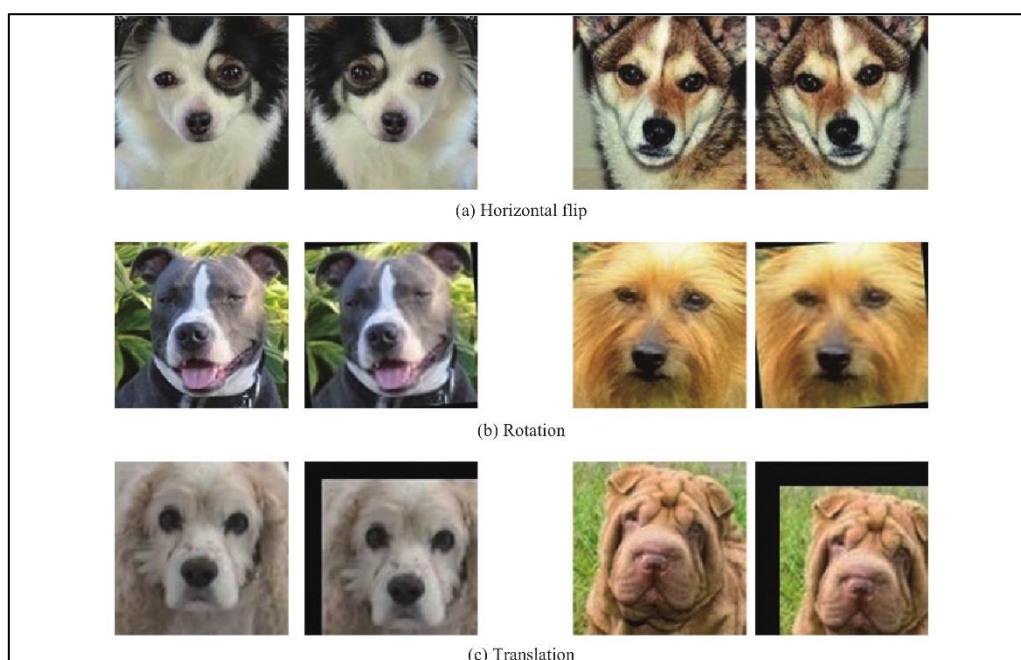
3. Przegląd literatury

Rozpoznawanie obrazów przy użyciu technologii uczenia maszynowego, a zwłaszcza konwolucyjnych sieci neuronowych (CNN), stało się jedną z najważniejszych dziedzin badań w sztucznej inteligencji. Systemy te znajdują zastosowanie w różnych obszarach, od medycyny po rolnictwo. W kontekście rozpoznawania ras kur, literatura naukowa natomiast nie dostarcza licznych przykładów i badań, które pokazują, jak technologia ta może być wykorzystana do automatyzacji i usprawnienia procesów klasyfikacji ras drobiu. W związku z tym wiedzę zgromadzono w oparciu o popularniejszy problem, jakim jest klasyfikacja ras psów.

Przykładem odniesienia do tej tematyki w literaturze naukowej jest publikacja *Dog Breed Identification using ResNet Model* [1], gdzie za pomocą Keras i TensorFlow tworzony jest, testowany i szkolony zbiór danych pod kątem wykrywania ras psów za pomocą RESNET. Zbiór danych obejmujący 20600 zdjęć psów obejmuje około 120 różnych ras psów.

W artykule *Dog Breed Identification Using Deep Learning* [2] przedstawiono problem rozpoznawania obrazu, mający charakter wieloklasowy, polegający na określeniu rasy psa na danym obrazie. W prezentowanym systemie zastosowano innowacyjne metody głębokiego uczenia, w tym splotowe sieci neuronowe. Dwie różne sieci są szkolone i oceniane na podstawie zbioru danych Stanford Dogs.

W artykule *Knowing Your Dog Breed: Identifying a Dog Breed with Deep Learning* [3] przedstawiono rozwiązanie umożliwiające identyfikację ras psów na podstawie wizerunków ich pysków. W proponowanej metodzie zastosowano podejście oparte na głębokim uczeniu się w celu rozpoznania ich ras. Ważnym tematem poruszonym jest augmentacja danych, przedstawiona na Rysunek 1, która zostanie wykorzystana w projekcie.



Rysunek 1. Przykłady rozszerzenia zbioru danych: oryginalne obrazy (po lewej) i obrazy rozszerzone (po prawej).

Proponowany przez autorów model osiąga obiecującą dokładność na poziomie 89,92% w opublikowanym zbiorze danych obejmującym 133 rasy psów.

Przegląd literatury pokazuje, że konwolucyjne sieci neuronowe są potężnym narzędziem do rozpoznawania obrazów, w tym do klasyfikacji ras kur. Zastosowanie tych technik w rolnictwie i hodowli zwierząt jest obiecujące i może prowadzić do znacznych usprawnień w procesach identyfikacji i zarządzania. Pomimo wyzwań, takich jak potrzeba dużych zbiorów danych, istnieje wiele możliwości dalszego rozwoju i optymalizacji tych technologii.

4. Wybór technologii

4.1. Python

Python jest jednym z najpopularniejszych języków programowania na świecie, co oznacza, że posiada ogromne wsparcie społeczności oraz bogaty ekosystem bibliotek i narzędzi. To sprawia, że jest to język pierwszego wyboru dla wielu zadań, w tym uczenia maszynowego i analizy danych.

4.2. Pandas

Pandas jest biblioteką, która umożliwia łatwe i efektywne przetwarzanie, manipulowanie i analizowanie danych tabelarycznych. Umożliwia ona szybkie wczytywanie, filtrowanie, grupowanie i agregowanie danych.

4.3. NumPy

NumPy dostarcza wszechstronne narzędzia do pracy z wielowymiarowymi tablicami oraz do wykonywania złożonych operacji matematycznych i statystycznych. Jest podstawą dla wielu innych bibliotek, takich jak Pandas i TensorFlow.

4.4. OpenCV

OpenCV to wszechstronna biblioteka do przetwarzania obrazów i analizy wideo. Umożliwia operacje takie jak filtrowanie, transformacje, detekcja krawędzi i wiele innych, co jest szczególnie przydatne w zadaniach związanych z widzeniem komputerowym.

4.5. scikit-learn

Scikit-learn to biblioteka, która dostarcza szeroki zakres algorytmów uczenia maszynowego do klasyfikacji, regresji, klasteryzacji oraz narzędzia do przetwarzania wstępnego danych, selekcji cech i oceny modeli. Jest często wykorzystywana w procesie przygotowywania danych i oceny modeli.

4.6. TensorFlow i Keras

TensorFlow to platforma do uczenia maszynowego opracowana przez Google. Keras jest wysokopoziomowym API do TensorFlow, które upraszcza tworzenie i trenowanie modeli głębokiego uczenia. Razem umożliwiają budowanie skomplikowanych sieci neuronowych i pracę z dużymi zestawami danych.

4.7. ResNet50V2

ResNet50V2 to pretrenowany model konwolucyjny, który jest bardzo skuteczny w ekstrakcji cech z obrazów. Użycie pretrenowanych modeli pozwala na wykorzystanie wcześniejszej wiedzy zgromadzonej podczas trenowania na dużych zestawach danych, co znacząco poprawia wyniki i skraca czas treningu.

5. Implementacja

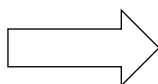
Implementacja opisuje wszystkie kroki niezbędne do przetwarzania danych, budowania modelu, trenowania go oraz wizualizacji wyników. Każdy krok jest starannie zaplanowany i realizowany przy użyciu odpowiednich bibliotek Pythona, co pozwala na stworzenie efektywnego i dokładnego modelu klasyfikacji obrazów.

5.1. Przygotowanie Danych

Początkowo udało się znaleźć 2 bazy danych powiązane z tematyką projektu. Pierwsza baza zdjęć posiadała zdjęcia sklasyfikowane według rasy, ale po dogłębnym przejrzeniu bazy uznano, że posiada on zbyt dużo błędów i niepoprawnie przyporządkowanych etykiet. W związku z licznymi błędami porzucono próby jej wykorzystania do realizacji projektu. Druga baza natomiast składała się z samych zdjęć bez etykiet, do tego nie było nigdzie informacji na temat wykorzystanych w niej rasach, co całkowicie uniemożliwiło jej wykorzystanie do uczenia modelu. W związku z napotkanymi problemami postawiono na własnoręczne skompletowanie zbioru danych z wykorzystaniem własnej oraz zdobytej z forów tematycznych [4] wiedzy. Poniżej przedstawiono proces przygotowania zbioru danych.



Rysunek 2. Oryginalne zdjęcie w formacie JPG.



Rysunek 3. Zdjęcie odpowiednio przystosowane do wprowadzenia do modelu.

5.1.1. Zebranie Danych

Pierwszym krokiem było zebranie zdjęć przedstawiających różne rasy kur. Kilka przykładowych ras zostało przedstawionych w tabeli poniżej.



Rysunek 4. Przykładowe zdjęcia ze zbioru danych na podstawie 6 wybranych ras.

Dane te zostały zgromadzone z różnych źródeł:

- Zdjęcia zebrane własnoręcznie
- Zdjęcia uzyskane z zamkniętych grup tematycznych na portalach społecznościowych za zgodą ich twórców.

5.1.2. Anotacja Danych

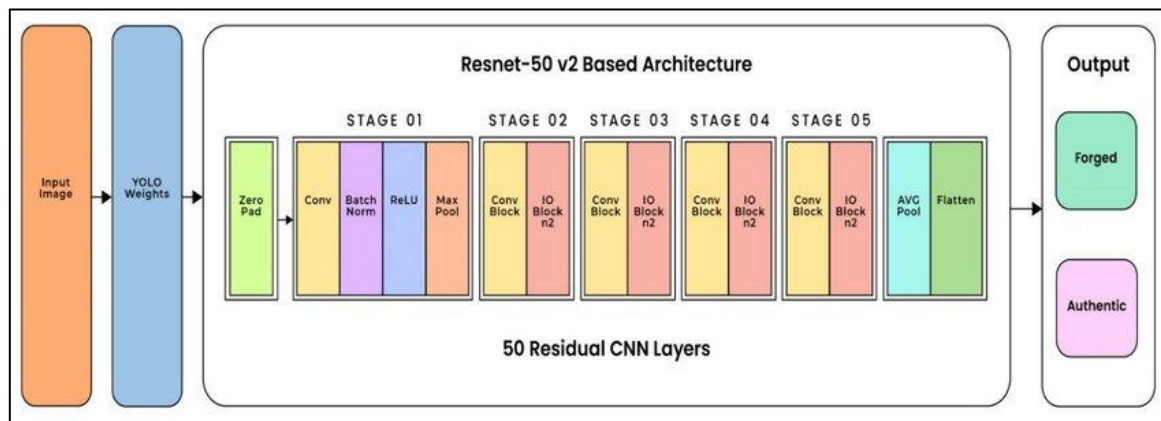
Zebrane zdjęcia zostały ręcznie oznakowane, tj. przypisano im odpowiednie etykiety (nazwy ras). Proces ten był kluczowy, aby dane były poprawnie przygotowane do trenowania modelu.

5.1.3. Przekształcenie i Augmentacja Danych

Aby zwiększyć różnorodność danych treningowych, zastosowano techniki augmentacji obrazów, takie jak: obrót i przycięcie.

5.2. Budowa Modelu

Model jest budowany przy użyciu pretrenowanej sieci ResNet50V2, na którą nakładane są dodatkowe warstwy specyficzne dla danego zadania. Warstwy te obejmują globalne średnie łączenie, warstwy gęste, dropout oraz normalizację.



Rysunek 5. Architektura ResNet50v2 [5].

Model został zaimplementowany w Pythonie z wykorzystaniem bibliotek TensorFlow i Keras.

5.3. Kompilacja i trenowanie modelu

Model jest kompilowany przy użyciu optymalizatora RMSprop, a następnie trenowany na danych treningowych. Ustalane są odpowiednie kroki na epokę i kroki walidacyjne w zależności od liczby próbek i rozmiaru partii.

5.4. Wizualizacja wyników

Na końcu, historia treningu jest wizualizowana za pomocą Matplotlib, co umożliwia analizę postępów modelu w trakcie trenowania.

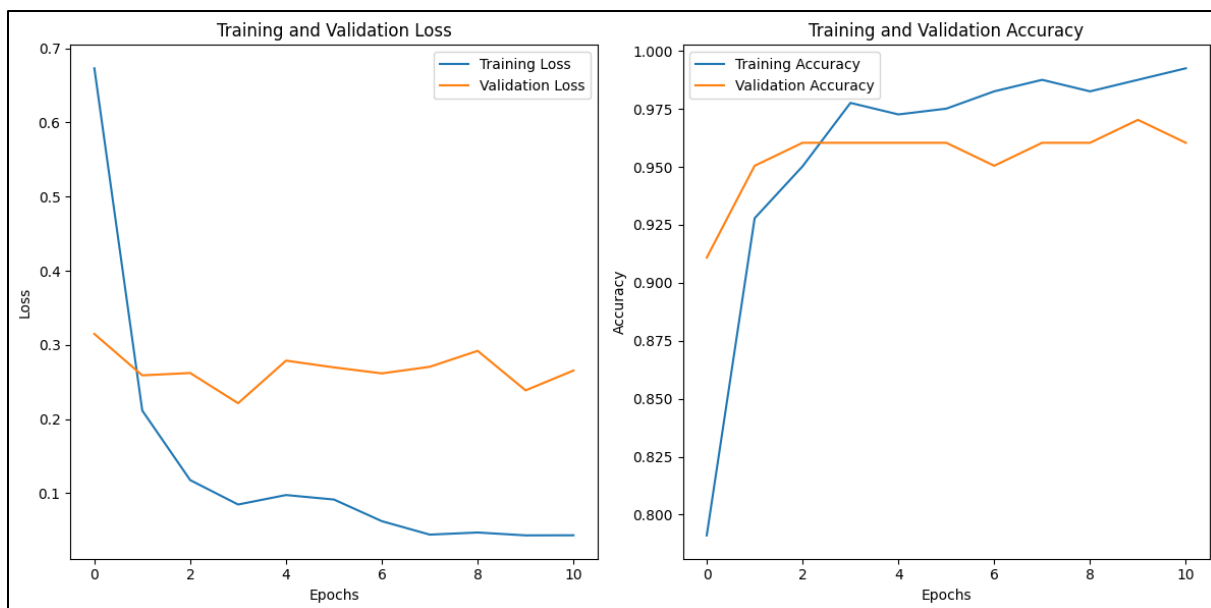
5.5. Uzyskane wyniki

5.5.1. Testy na Zbiorze z 6 Rasami i 40 Zdjęciami dla Każdej Rasy

W pierwszym etapie przeprowadzono testy modelu na zbiorze danych zawierającym 6 ras kur, z których każda była reprezentowana przez 40 zdjęć. Wyniki testów wykazały, że model osiągnął dokładność rozpoznawania na poziomie 78,75%. Mniejsza liczba zdjęć na rasę mogła ograniczyć zdolność modelu do efektywnego rozróżniania poszczególnych ras.

5.5.2. Testy na Zbiorze z 4 Rasami i 100 Zdjęciami dla Każdej Rasy

Następnie przeprowadzono testy na mniejszym zbiorze danych zawierającym 4 rasy, jednak z większą liczbą zdjęć dla każdej rasy (100 zdjęć na rasę). W tym przypadku model osiągnął bardzo wysoką dokładność rozpoznawania, wynoszącą 99,07%. Zwiększona liczba zdjęć dla każdej rasy znacznie poprawiła zdolność modelu do precyzyjnego rozpoznawania.



Rysunek 6. Wykres przedstawiający wyniki Loss i Accuracy dla 4 ras.

5.5.3. Testy na Zbiorze z 6 Rasami i 100 Zdjęciami dla Każdej Rasy

Ostateczne testy przeprowadzono na większym zbiorze danych zawierającym 6 ras, z których każda była reprezentowana przez 100 zdjęć. Wyniki testów wykazały, że model osiągnął dokładność rozpoznawania na poziomie 97,52%. Pomimo większej liczby ras, duża liczba zdjęć dla każdej rasy pozwoliła modelowi na efektywne rozróżnianie poszczególnych ras z wysoką dokładnością.



Rysunek 7. Wykres przedstawiający wyniki Loss i Accuracy dla 6 ras.

5.5.4. Podsumowanie wyników

Liczba zgromadzonych zdjęć umożliwiła przeprowadzenie skutecznych testów modelu, osiągając zadowalającą dokładność rozpoznawania ras kur powyżej 90%.

Tabela 1. Zestawienie uzyskanych wyników

Wstępne wyniki testów dla ResNet50		
Liczba ras	Liczba zdjęć dla 1 rasy	Dokładność rozpoznawania ras
6	40	78,75%
4	100	99,07%
6	100	97,52%

Obecnie posiadana baza danych jest gotowa do dalszych testów, a dalsze prace będą koncentrować się na optymalizacji modelu i zwiększeniu liczby zdjęć dla każdej rasy w celu jeszcze większego poprawienia dokładności.

6. Podsumowanie

W pierwszym semestrze udało się zrealizować wszystkie ustalone zadania. Zgromadzono ponad 2000 zdjęć 16 wybranych ras kur, po 100 zdjęć dla każdej pojedynczej rasy. z czego 600 zdjęć poddane zostało już odpowiedniej obróbce i zostały połączone w jedną bazę danych. Następnie wybrano i edytowano zdjęcia dla 6 ras, co pozwoliło na zgromadzenie bazy danych zawierającej 600 zdjęć oraz etykiety, wykorzystane później w procesie uczenia maszynowego. Podjęto także rozeznanie w różnych modelach uczenia maszynowego oraz przeprowadzono wstępne próby stworzenia modelu z wykorzystaniem tej części zgromadzonych zdjęć. Przy obecnym zbiorze zdjęć model osiąga zadowalające wyniki, wykazując skuteczność powyżej 90% w rozpoznawaniu ras kur. Na najbliższy czas, czyli okres między-semestralny oraz semestr drugi planowane jest ukończenie prac nad kompletowaniem zbioru danych (planowo ~1600 zdjęć) oraz udoskonalanie modelu, działającego już na pełnym zbiorze zdjęć.

Bibliografia

- [1] Y. S. K. S. M. S. C. M. Yarram Abhilash Reddy, „Dog Breed Identification using ResNet Model,” w *International Journal on Recent and Innovation Trends in Computing and Communication*, 2023.
- [2] C. S. Z. V. A. Z. Zalan Raduly, „Dog Breed Identification Using Deep Learning,” w *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*, 2018.
- [3] W. K. S. K. K. T. Punyanuch Borwarnginn, „Knowing Your Dog Breed: Identifying a Dog Breed with Deep Learning,” *Machine Intelligence Research*, 2021.
- [4] STAR MILLING, „Breeds of Chickens,” [Online]. Available: <https://starmilling.com/poultry-chicken-breeds/>. [Data uzyskania dostępu: Czerwiec 2024].
- [5] T. Z. A. A. Emad Ul Haq Qazi, „Deep Learning-Based Digital Image Forgery Detection System,” *Applied Sciences*, Marzec 2022.
- [6] J. Le, „How to easily build a Dog breed Image classification model,” 15 Marzec 2019. [Online]. Available: <https://medium.com/nanonets/how-to-easily-build-a-dog-breed-image-classification-model-2fd214419cde>. [Data uzyskania dostępu: Czerwiec 2024].
- [7] D. Wood, „Using CNNs to Classify Dog Breeds (Python, PyTorch),” 30 Marzec 2019. [Online]. Available: <https://dwood3.github.io/machinedeeplearning/2019/03/30/dog-breed-classifier.html>. [Data uzyskania dostępu: Czerwiec 2024].

Spis rysunków

Rysunek 1. Przykłady rozszerzenia zbioru danych: oryginalne obrazy (po lewej) i obrazy rozszerzone (po prawej).	3
Rysunek 2. Oryginalne zdjęcie w formacie JPG.	5
Rysunek 3. Zdjęcie odpowiednio przystosowane do wprowadzenia do modelu.	5
Rysunek 4. Przykładowe zdjęcia ze zbioru danych na podstawie 6 wybranych ras.	6
Rysunek 5. Architektura ResNet50v2 [5].	7
Rysunek 6. Wykres przedstawiający wyniki Loss i Accuracy dla 4 ras.	8
Rysunek 7. Wykres przedstawiający wyniki Loss i Accuracy dla 6 ras.	8

Spis tabel

Tabela 1. Zestawienie uzyskanych wyników	9
--	---