

Animal Emotion Recognition

Kacper Rabczewski, Patryk Przybysz, Aleksander Trzcíński

9 czerwca 2025

Cel Projektu

Celem projektu jest opracowanie systemu umożliwiającego automatyczną klasyfikację emocji u zwierząt domowych na podstawie analizy obrazu rejestrowanego przez kamerę wideo wbudowaną w laptop. System ten wykorzystuje metody z zakresu widzenia komputerowego oraz uczenia maszynowego do identyfikacji sygnałów wizualnych związanych z emocjonalnym stanem zwierzęcia.

Cele szczegółowe projektu

- Wykrywanie obecności zwierząt na obrazie wideo z kamery laptopa przy użyciu metod analizy obrazu.
- Klasyfikacja emocji zwierząt na podstawie wizualnych cech zarejestrowanych w czasie rzeczywistym.
- Wykorzystanie techniki transfer learningu w celu zwiększenia skuteczności i skrócenia czasu trenowania modeli.
- Integracja systemu z kamerą wbudowaną w laptop, umożliwiającą łatwe i bezpośrednie korzystanie bez dodatkowego sprzętu.
- Zapewnienie działania aplikacji w czasie rzeczywistym, umożliwiające natychmiastowe wykrywanie i klasyfikację emocji zwierzęcia.

Wymagania funkcjonalne

Musi być

- System musi wykrywać obecność zwierzęcia w obrazie z kamery laptopa.

- System musi klasyfikować emocje zwierzęcia w czasie rzeczywistym.
- System musi działać lokalnie, bez konieczności połączenia z Internetem.
- System musi rozróżniać co najmniej dwie emocje (np. radość i smutek).

Powinno być

- System powinien klasyfikować emocje z dokładnością co najmniej 85%.
- Interfejs użytkownika powinien być prosty i intuicyjny w obsłudze.
- Aplikacja powinna umożliwiać ręczne zatrzymywanie i wznowianie analizy.

Może być

- System może zapisywać historię wykrytych emocji w lokalnej bazie danych.
- Aplikacja może umożliwiać eksport danych do pliku CSV.
- Aplikacja może zawierać tryb edukacyjny opisujący interpretację emocji.

Nie będzie

- System nie będzie integrowany z usługami chmurowymi w obecnej wersji.
- Aplikacja nie będzie obsługiwać wielu kamer jednocześnie.
- System nie będzie wspierać analizy dźwięku ani rozpoznawania szczekania/miauczenia.
- Aplikacja nie będzie dostępna na urządzeniach mobilnych w tej wersji.

Technologie

Projekt wykorzystuje szereg nowoczesnych technologii z zakresu sztucznej inteligencji, przetwarzania obrazu i architektury systemów.

Sztuczna inteligencja

- **YOLOv10 Nano** – detekcja obiektów w czasie rzeczywistym; implementacja oparta na bibliotece Ultralytics.
- **MobileNetV3 Large** – lekka sieć neuronowa do klasyfikacji emocji zwierząt; wykorzystana z frameworkiem PyTorch i biblioteką TIMM.
- **PyTorch** – główny framework do uczenia głębokiego i inferencji modeli.
- **TIMM (PyTorch Image Models)** – kolekcja nowoczesnych architektur sieci do zadań vision.

Przetwarzanie obrazu

- **OpenCV** – akwizycja obrazu z kamery, wstępne przetwarzanie oraz wizualizacja wyników detekcji i klasyfikacji.
- **PIL / Pillow** – biblioteka do manipulacji i konwersji obrazów.
- **NumPy** – obsługa tablic numerycznych i operacji na danych obrazu.

Architektura systemu

- **Python Multiprocessing** – umożliwia podział systemu na niezależne procesy (np. akwizycja, przetwarzanie, wizualizacja).
- **Multiprocessing.Queue** – kolejki do komunikacji między procesami.
- **Event-based synchronization** – synchronizacja i kontrola działania współbieżnych komponentów systemu.

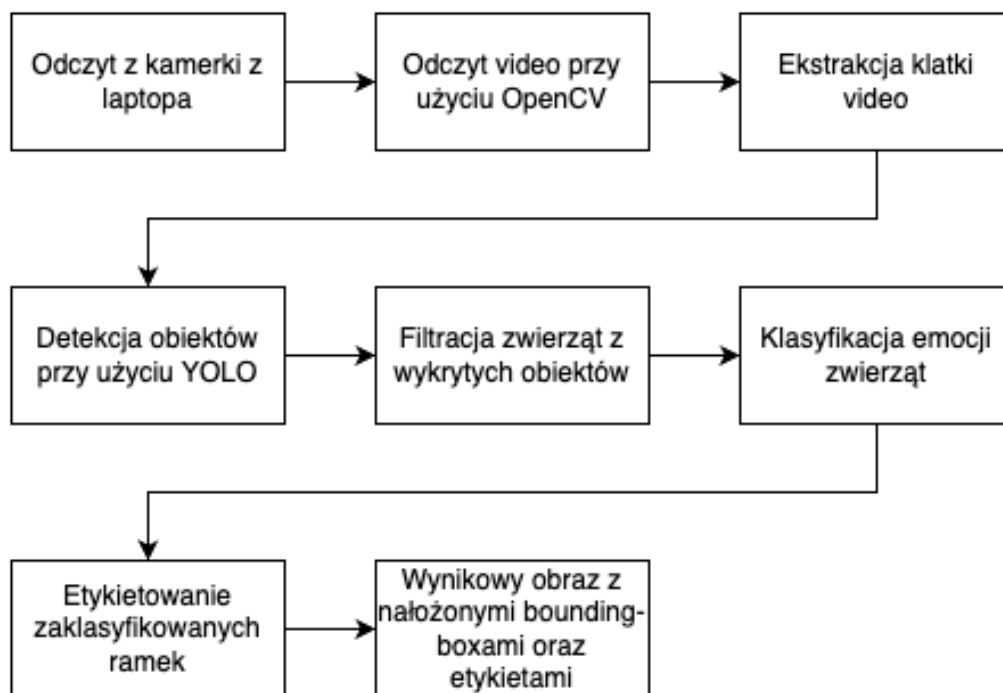
Zbiór danych

W projekcie wykorzystaliśmy poniższe zbiory danych:

- [1] Pet's Facial Expression Image Dataset - zawiera 1000 zdjęć twarzy różnych zwierząt (psy, koty, króliki, chomiki, owce, konie, ptaki) oraz różne emocje (radość, złość, smutek, itp.)
- [2] Dog Emotion - zawiera 4000 zdjęć psów, po 1000 zdjęć dla każdej emocji (złość, szczęście, spokój, smutek)

Architektura pipeline

Rysunek 1 przedstawia architekturę pipeline odpowiedzialnego za klasyfikację i rozpoznanie emocji zwierząt. Uwzględnia on zarówno etap ekstrakcji obrazu, przetwarzania oraz samej klasyfikacji.



Rysunek 1: Architektura pipeline

Przykład użycia

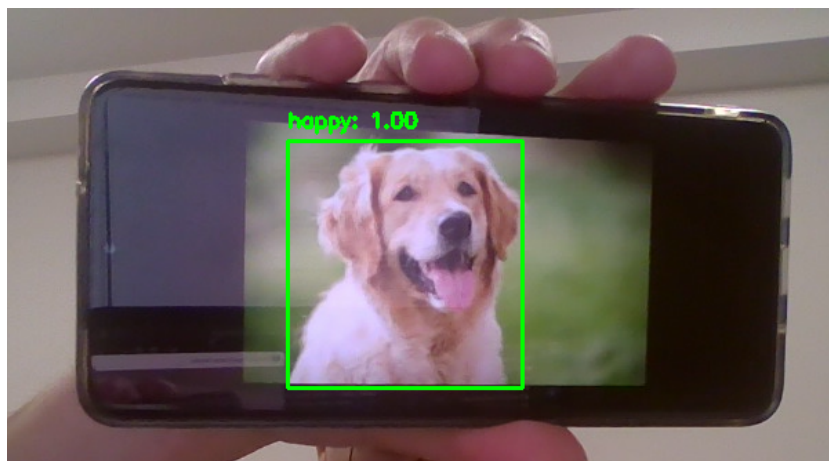
Poniżej przedstawiono przykłady działania systemu w praktyce. Na zrzutach ekranu widać, jak system identyfikuje zwierzęta na obrazie z kamery i klasyfikuje ich emocje w czasie rzeczywistym.

Rysunek 2 ilustruje detekcję emocji "szczęśliwy" u psa rasy golden retriever. System poprawnie zidentyfikował zwierzę oraz przypisał mu etykietę emocji wraz z pewnością predykcji.

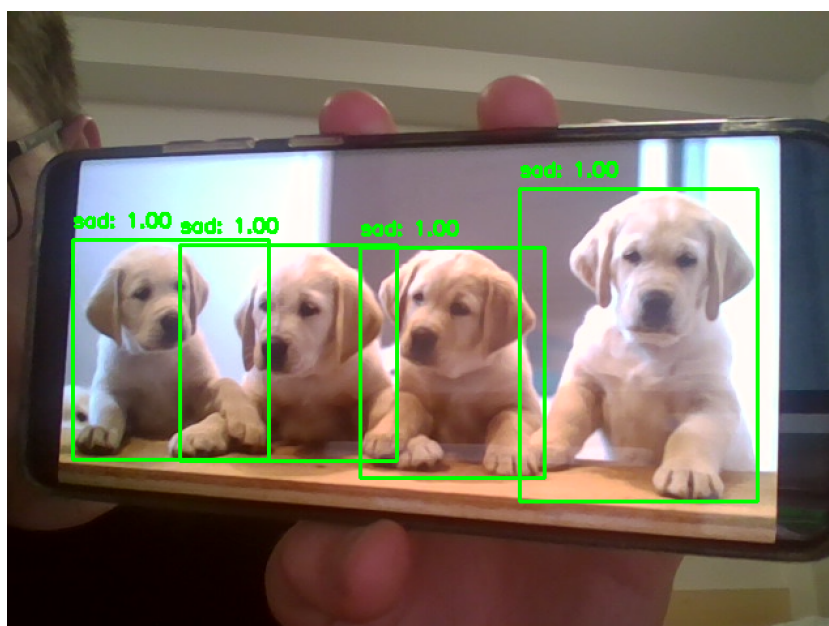
Na rysunku 3 widać przykład działania systemu na obrazie zawierającym wiele zwierząt. System poprawnie wykrył cztery szczenięta i dla każdego z nich określił emocję jako "smutny".

Na rysunku 4 przedstawiono przykład detekcji i klasyfikacji emocji "zły" u psa rasy chihuahua. Podobnie jak w poprzednich przypadkach, zwierzę zo-

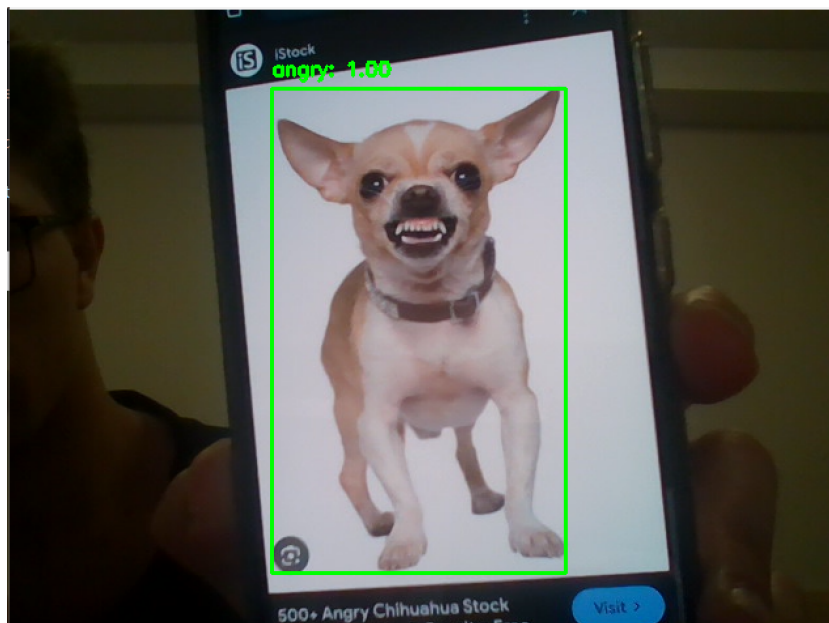
stało poprawnie zlokalizowane, a jego emocja sklasyfikowana.



Rysunek 2: Przykład detekcji emocji “szczęśliwy“



Rysunek 3: Przykład detekcji emocji “smutny“ u wielu zwierząt



Rysunek 4: Przykład detekcji emocji “zły”

1 Instalacja

1.1 System operacyjny

Projekt został przetestowany na systemach:

- Windows 11
- Ubuntu 24.04.1 LTS

1.2 Wymagania

Przed rozpoczęciem instalacji upewnij się, że masz zainstalowane następujące oprogramowanie:

- Menedżer pakietów ‘pipx’

2 Uruchamianie projektu

Poniżej przedstawiono kroki niezbędne do pobrania i uruchomienia projektu.

2.1 Pobieranie projektu

Najpierw należy sklonować repozytorium projektu z GitHub oraz przejść do utworzonego katalogu:

1. Sklonuj repozytorium:

```
git clone https://github.com/RabaDaba1/animal-emotion-classification.git
```

2. Przejdź do katalogu projektu:

```
cd animal-emotion-classification
```

2.2 Instalacja narzędzi i konfiguracja środowiska

Projekt wykorzystuje ‘uv’ do zarządzania środowiskiem Python i zależnościami. Zalecany sposób instalacji ‘uv’ jest użycie ‘pipx’, co zapewnia izolowaną instalację narzędzi CLI.

2.2.1 Instalacja pipx (jeśli nie jest zainstalowany)

‘pipx’ pozwala na instalację i uruchamianie aplikacji Python w izolowanych środowiskach. Jeśli nie posiadasz ‘pipx’, możesz go zainstalować za pomocą ‘pip’. Upewnij się, że Python oraz ‘pip’ są zainstalowane i dostępne w Twojej ścieżce systemowej.

```
python -m pip install --user pipx
```

Następnie, dodaj katalog binariów ‘pipx’ do ścieżki systemowej PATH:

```
python -m pipx ensurepath
```

Po wykonaniu tych komend może być konieczne ponowne uruchomienie terminala lub otwarcie nowej sesji, aby zmiany w PATH zostały uwzględnione i komenda ‘pipx’ była rozpoznawana.

2.2.2 Instalacja uv za pomocą pipx

Gdy ‘pipx’ jest już zainstalowany i dostępny w PATH, zainstaluj ‘uv’ za pomocą następującego polecenia:

```
pipx install uv
```

Po tej operacji również zaleca się ponowne uruchomienie terminala lub środowiska IDE, aby upewnić się, że komenda ‘uv’ jest poprawnie rozpoznawana.

2.2.3 Konfiguracja środowiska Python i instalacja zależności projektu (za pomocą uv)

Po pomyślnym sklonowaniu repozytorium i zainstalowaniu ‘uv’, wykonaj poniższe kroki w głównym katalogu projektu (‘animal-emotion-classification’):

1. **Instalacja interpretera Python w wersji 3.12:** ‘uv’ może zarządzać wersjami Pythona dla projektu. Jeśli nie masz Pythona 3.12, ‘uv’ może go pobrać i skonfigurować dla tego projektu.

```
uv python install 3.12
```

2. **Utworzenie wirtualnego środowiska i instalacja zależności projektu:** To polecenie utworzy wirtualne środowisko (jeśli jeszcze nie istnieje) i zainstaluje wszystkie zależności zdefiniowane w pliku ‘pyproject.toml’ (korzystając z ‘uv.lock’, jeśli istnieje, do zapewnienia powtarzalności).

```
uv sync
```

2.3 Uruchamianie aplikacji

Aby uruchomić główną aplikację, należy wykonać polecenie:

```
uv run python -m src.main
```

2.4 Uruchamianie notatników Jupyter

Upewnij się, że środowisko Jupyter (np. VS Code, Jupyter Lab) jest skonfigurowane do używania kernela ‘.venv’ stworzonego przez ‘uv’.

1. Otwórz notatnik ‘notebooks/fine_tuning.ipynb’. Wybierz kernel z ‘.venv’.
2. Uruchom komórki notatnika.

2.5 Uruchamianie testów

Aby uruchomić testy automatyczne, użyj następującego polecenia z głównego katalogu projektu:

```
uv run pytest
```


Literatura

- [1] <https://www.kaggle.com/datasets/anshtanwar/pets-facial-expression-dataset>
- [2] <https://www.kaggle.com/datasets/danielshanbalico/dog-emotion>