**Politechnika Wrocławska**

**Wydział Informatyki i Telekomunikacji**

Kierunek: IST

**ZESPOŁOWE PRZEDSIĘWZIĘCIE INFORMATYCZNE**

**System Larvix-On AI – szybka diagnostyka toksyczności osocza z wykorzystaniem analizy behawioralnej larw i algorytmów uczenia maszynowego**

Kubś Mikołaj

Kulka Krzysztof

Łopianiak Martyna

Łuszczek Patryk

Opiekun pracy

**dr inż. Piórkowska Natalia**

Słowa kluczowe: medycyna, AI, diagnostyka

WROCŁAW 2025

Spis treści

[1. Syntetyczny opis projektu 3](#_Toc199233427)

[2. Technologia 3](#_Toc199233428)

[3. Roadmapa projektu 4](#_Toc199233429)

[4. Kluczowe ryzyka: 5](#_Toc199233430)

[5. Założenia i ograniczenia: 5](#_Toc199233431)

# Syntetyczny opis projektu

Celem projektu jest opracowanie szybkiego systemu diagnostycznego opartego na sztucznej inteligencji, umożliwiającego wykrywanie ksenobiotyków w osoczu pacjentów. System ten ma analizować wzorce ruchowe larw Galleria mellonella, którym podano próbki osocza, w celu identyfikacji charakterystycznych reakcji organizmu na toksyczne substancje. Kluczowym założeniem jest osiągnięcie wysokiej czułości i specyficzności klasyfikacji przy czasie analizy nieprzekraczającym 20 minut. Projekt dąży do stworzenia algorytmów uczenia maszynowego zdolnych do rozpoznawania subtelnych zmian w motoryce larw wywołanych obecnością leków, narkotyków, toksyn lub patogenów. Ostatecznym celem jest wdrożenie nowatorskiej metody diagnostycznej, która przyczyni się do poprawy skuteczności leczenia i skrócenia czasu interwencji medycznej w stanach zagrożenia życia.

Zakres projektu obejmuje zarówno badania nad algorytmami uczenia maszynowego, jak i opracowanie aplikacji diagnostycznej wspierającej kliniczne wykorzystanie systemu. W części badawczej projekt koncentruje się na analizie danych ruchowych larw Galleria mellonella, ekstrakcji cech behawioralnych oraz trenowaniu modeli klasyfikacyjnych. Równolegle realizowana jest część inżynierska, w której powstaje aplikacja integrująca cały pipeline: od wczytania danych wideo, przez analizę i predykcję, aż po prezentację wyniku użytkownikowi. Aplikacja będzie wyposażona w prosty interfejs graficzny, umożliwiający lekarzowi lub laborantowi szybkie przeprowadzenie analizy i uzyskanie wyniku w czasie rzeczywistym. Całość systemu zostanie przetestowana pod kątem wydajności, dokładności predykcji oraz użyteczności klinicznej.

# 2. Technologia

**Technologie użyte w projekcie:**

* **Wrocławskie Centrum Superkomputerowe (WCSS):** zapewnia moc obliczeniową do intensywnych zadań obliczeniowych, takich jak analiza danych wideo i trenowanie modeli uczenia maszynowego. Umożliwia przeprowadzanie eksperymentów na dużą skalę oraz testowanie wydajności systemu.
* **Python:** główny język używany w części badawczej. Służy do przetwarzania i analizy danych ruchowych larw Galleria mellonella, ekstrakcji cech behawioralnych, implementacji i trenowania modeli klasyfikacyjnych. Wykorzystywane biblioteki to m.in. NumPy, pandas, scikit-learn, PyTorch.
* **React:** użyty do budowy frontendowej części aplikacji diagnostycznej. Umożliwia stworzenie lekkiego i intuicyjnego interfejsu użytkownika, który pozwala lekarzowi/laborantowi na wczytanie danych wideo, uruchomienie analizy oraz szybkie uzyskanie wyniku predykcji. Komunikuje się z backendem napisanym w Pythonie przez API.
* **Roboflow:** narzędzie wykorzystywane do anotacji danych wideo – służyło do oznaczania obiektów (np. larw) na klatkach filmu, co umożliwiło trenowanie modeli detekcji i analizy ruchu.

# 3. Roadmapa projektu

 **Do 15.06** – Finalizacja zbioru danych

* Zebranie i wstępne oczyszczenie nagrań wideo
* Anotacja danych w Roboflow

 **Do 30.06** – Ekstrakcja cech i przygotowanie danych

* Analiza ruchu larw
* Ekstrakcja cech behawioralnych
* Przygotowanie zestawów treningowych/testowych

 **Do 31.07** – Trenowanie i walidacja modeli ML

* Testowanie różnych podejść klasyfikacyjnych
* Wybór najlepszego modelu
* Wstępna ocena wydajności na WCSS

 **Do 31.08** – Rozwój backendu aplikacji

* Implementacja pipeline: wczytanie wideo → analiza → predykcja
* API do komunikacji z frontendem

 **Do 30.09** – Rozwój frontendowej aplikacji w React

* Interfejs do obsługi analizy i prezentacji wyników
* Integracja z backendem
* Testy UX z udziałem użytkowników

 **Do 15.10** – Testy końcowe i optymalizacja

* Testy wydajności na WCSS
* Poprawa dokładności i czasu działania
* Usprawnienia interfejsu

 **Do 31.10** – Walidacja kliniczna

* Testy systemu z udziałem ekspertów
* Zbieranie opinii i analiza wyników

 **Do 30.11** – Finalizacja projektu

* Dokumentacja techniczna i użytkowa
* Ostateczne poprawki
* Przygotowanie wersji końcowej do wdrożenia/prezentacji

# 4. Kluczowe ryzyka:

* **Niska jakość danych wideo** – możliwe trudności w ekstrakcji cech; rozwiązanie: standaryzacja nagrań, filtrowanie danych.
* **Brak wyraźnych wzorców ruchowych** – trudność w klasyfikacji; rozwiązanie: wzbogacenie zbioru danych, zaawansowane metody ekstrakcji cech (np. deep learning).
* **Błędy przy adnotacji** – nieprawidłowa identyfikacja przodu larwy
* **Przekroczenie limitu czasu analizy (20 min)** – ryzyko nieoptymalnej implementacji; rozwiązanie: profilowanie kodu, optymalizacja algorytmów i równoległość obliczeń.
* **Niska generalizacja modelu** – model może nie działać dobrze na nowych próbkach; rozwiązanie: walidacja krzyżowa, regularizacja, augmentacja danych.
* **Problemy z integracją systemu** – błędy na etapie łączenia modułów

# 5. Założenia i ograniczenia:

* Analiza musi zakończyć się w 20 minut.
* Dane wejściowe to nagrania wideo larw Galleria mellonella.
* System oparty na ML, głównie klasyfikacji ruchu.
* Aplikacja z prostym GUI, możliwa do obsługi przez osoby nietechniczne.
* Środowisko: Python + biblioteki ML (np. scikit-learn, PyTorch), dostępne przez cały czas trawania projektu