



Politechnika
Wrocławska



Czyli jak technologia może ratować życie?

Prezentacja Zespołowego Przedsięwzięcia Inżynierijnego

Prezentują:

Mikołaj Kubś

Krzysztof Kulka

Martyna Łopianiak

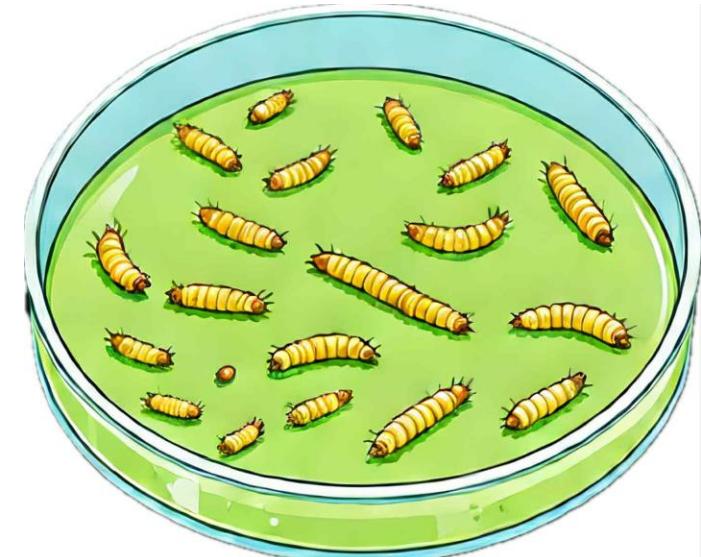
Patryk Łuszczek

Opiekun: dr inż. Natalia Piórkowska



Agenda

1. Dlaczego larwy?
2. Korzyści biznesowe
3. Rozwiązania konkurencji
4. Zarys architektury
 - 4.1 Symulacja
 - 4.2 Model ML
 - 4.3 Logika biznesowa
 - 4.4 Interfejs użytkownika
5. Harmonogram i dokonane postępy
6. Sesja Q&A
7. Zakończenie



Geneza pomysłu

- Sytuacja w szpitalu na ul. Borowskiej
- Problem:
 - brak wystarczająco szybkiej diagnozy przy niekomunikatywnym pacjencie
 - tabu społeczne związane z przedawkowaniem
 - toksyczność nieprawidłowych odtrutek
- Potrzeba:
 - szybki system diagnostyczny dla ofiar przedawkowania, dający odpowiedź w ciągu kilkunastu minut zamiast kilku godzin





Wizja

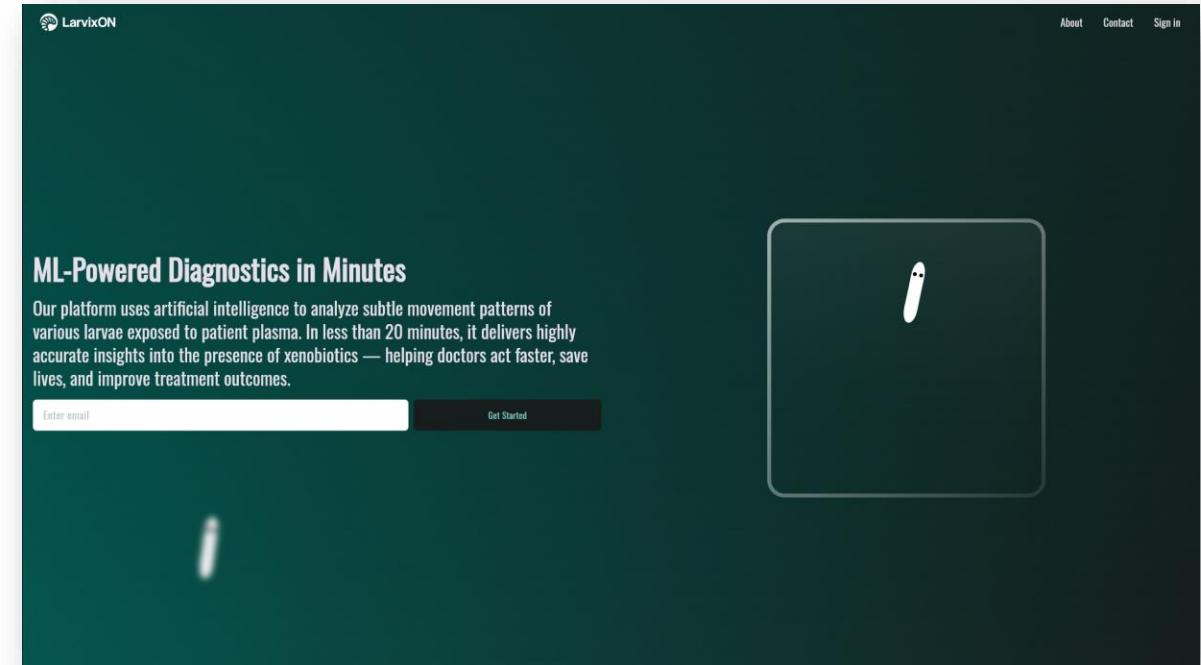
- Użycie larw aby szybko i skutecznie wspomóc działania diagnostyczne
- Model SI o wysokiej dokładności
- Prostota obsługi, nie wymagająca specjalistycznych zasobów
- Wspomaganie podejmowania decyzji krytycznych dla życia pacjentów





Wizja

- Nowoczesna i intuicyjna aplikacja
- Wynik w ciągu minut od wysłania do modelu
- Wykorzystanie najnowszych przełomów w dziedzinie SI i ML
- Wysoka jakość kodu i dokumentacji





Korzyści biznesowe / kliniczne

- Poprawa skuteczności leczenia
 - Szybsza i trafniejsza diagnoza pozwala na natychmiastowe wdrożenie odpowiedniej terapii
- Skrócenie czasu interwencji medycznej
 - Analiza w czasie <20 minut umożliwia reakcję w sytuacjach zagrożenia życia
- ***Ratowanie życia***
 - Wczesne wykrycie toksyn zwiększa szanse przeżycia pacjentów
- Wartość wdrożeniowa
 - System gotowy do integracji z infrastrukturą laboratoriów i szpitali
 - Potencjał komercjalizacji jako szybkie narzędzie diagnostyczne oparte na AI





Konkurencja

- Istniejące rozwiązania
 - Nie łączą analizy behawioralnej larw z uczeniem maszynowym
 - Koncentrują się głównie na ocenie przeżywalności lub toksyczności w badaniach eksperymentalnych
 - Brak systemów o charakterze diagnostycznym
- **LarvixON**
 - Pierwsze zastosowanie ML do analizy **wzorców ruchowych** larw po podaniu **osocza pacjentów**
 - Gotowość wdrożeniowa – projektowany z myślą o zastosowaniu w oddziałach medycznych i laboratoriach diagnostycznych
 - Innowacyjne podejście – analiza danych biologicznych w czasie rzeczywistym w celu wykrywania ksenobiotyków



Symulacja

*Pomysł na syntezę danych dla walidacji
architektury modelu*



Problem

- Niepewność, czy dane treningowe będą dostępne w odpowiednim czasie.
Ryzyko braku czasu na walidację architektury i działania modelu



Problem

- Niepewność, czy dane treningowe będą dostępne w odpowiednim czasie.
Ryzyko braku czasu na walidację architektury i działania modelu
- Rozwiązanie: generacja syntetycznych danych, możliwe zbliżonych do rzeczywistych



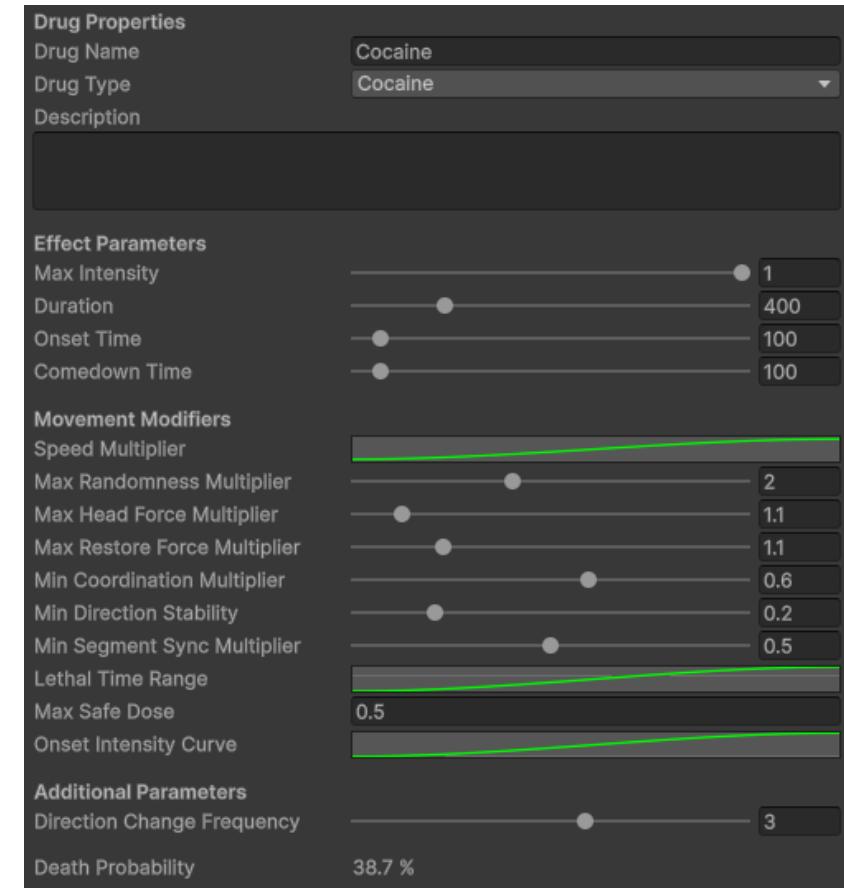
Założenia symulacji

- Na podstawie poprzednich badań i obserwacji ustalone główe parametry ruchu, na które wpływa ksenobiotyk
- Parametry:
 - Czas zgonu
 - Prędkość
 - Skrętność
 - Synchronizacja ruchu



Założenia symulacji narkotyków

- Z racji braku danych o realnych skutkach tych narkotyków, dobrano parametry zakładając podobny wpływ narkotyków na larwy jak na ludzi
- Symulowane ksenobiotyki:
 - Morfina: powolny, ale pewny ruch
 - Kokaina: bardzo szybki, ale chaotyczny ruch
 - Etanol: powolny i niepewny ruch
 - Ketamina: lekkie spowolnienie i bardzo chaotyczny ruch
 - Tetrodotoksyna: całkowity paraliż ruchu w krótkim czasie





Założenia techniczne symulacji

- Symulacja stworzona w silniku gier Unity, napisana w języku C#
- Model ruchu opierający się na maszynie stanów (leżenie, poruszanie się w kierunku celu, obserwowanie otoczenia)
- Technicznie larwa to 5 punktów w przestrzeni 2D, z preferowanymi odległościami między sobą
- Ruch to zwiększenie preferowanej odległości między głową larwy a jej poprzednim segmentem, następnie zmniejszenie odległości między tyłem larwy a jej przedostatnim segmentem, następnie wyluzowanie odległości
- Proste obliczenie prędkości wypadkowej każdego punktu na podstawie preferencji odległości z sąsiednimi punktami (plus dodatkowe siły jak prostowanie larwy przy mocnym wygięciu oraz kolizje fizyczne ze ścianą)



Generowanie danych

- Symulacja zachowania larwy na ograniczonym terenie (dość wierne odzwierciedlenie rzeczywistych warunków)
- Czas trwania symulacji to 30 symulowanych minut
- Realny czas wytworzenia filmiku treningowego: 7 minut
- Możliwość uruchomienia wielu kopii programu generującego dane naraz
- Wsparcie pliku konfiguracyjnego
- Członkowie projektu korzystają z różnych systemów operacyjnych – automatyczne budowanie symulacji na Windows, Linux i macOS zintegrowane z GitHub



Model

Czyli jak zamienić ruch larw w diagnozę?

Prezentacja tworzenia i działania



Przepływ szkolenia



Wyzwania - Model

- Jak podzielić wideo i wykryć prezencję larwy?
- Rozwiązanie: CSV2
- Skutek: pełne wykorzystanie informacji z danych





Wyzwania - Model

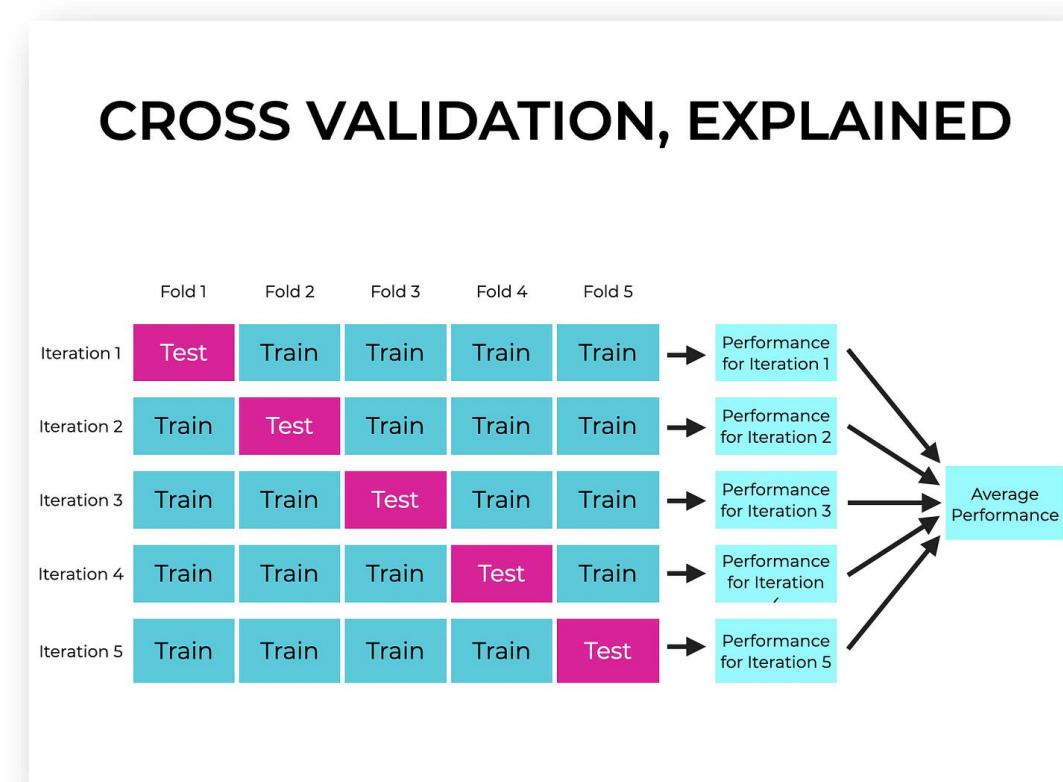
- Jak efektywnie szkolić model na ok. 300 Gigabajtach danych?
- Rozwiązanie: Wrocławskie Centrum Sieciowo-Superkomputerowe
- Skutek: czas szkolenia – 18 godzin -> ok. 1.5 godziny





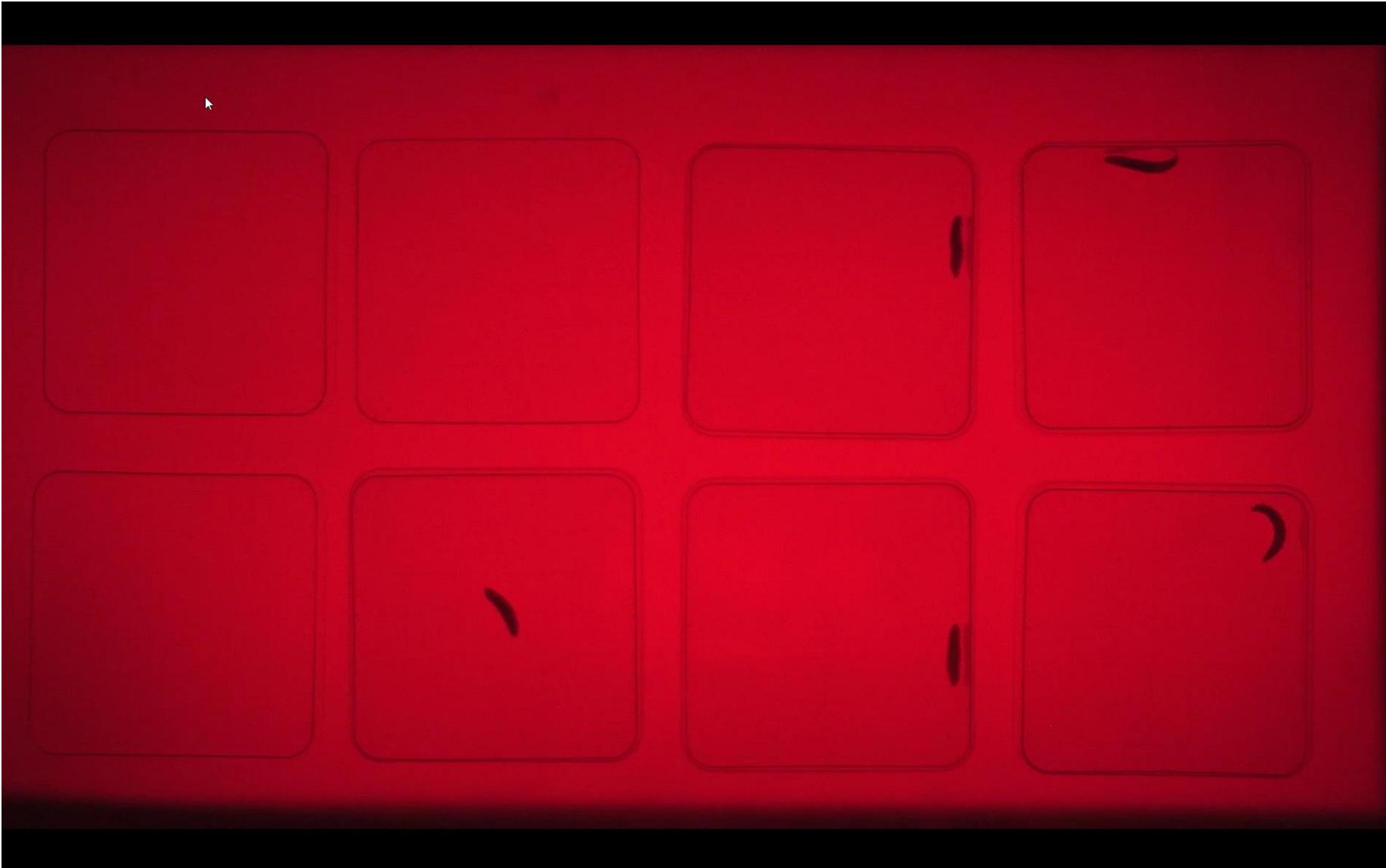
Wyzwania - Model

- Jak efektywnie sprawdzić skuteczność modelu?
- Rozwiązanie: Walidacja Krzyżowa
- Skutek: Brak możliwości weryfikacji skuteczności -> skuteczność na poziomie 85 – 95%





Potrzeba ML





Logika biznesowa

Połączenie między modelem a użytkownikiem

Struktura



Potrzeby

- Profil użytkownika (w tym uwierzytelnianie)
- Zarządzanie analizami (przesyłanie wideo do modelu i odbieranie wyników)
- Profil pacjenta (?)

accounts	
POST	/api/accounts/logout/
GET	/api/accounts/profile/
PUT	/api/accounts/profile/
PATCH	/api/accounts/profile/
GET	/api/accounts/profile/details/
PUT	/api/accounts/profile/details/
PATCH	/api/accounts/profile/details/
GET	/api/accounts/profile/stats/
analysis	
GET	/api/analysis/
POST	/api/analysis/
GET	/api/analysis/{id}/
PUT	/api/analysis/{id}/
PATCH	/api/analysis/{id}/
DELETE	/api/analysis/{id}/



Technologie

- Python
- Django (Django REST Framework)
- PostgreSQL





Wyzwania

- Model potrzebuje czasu na zwrócenie wyników - **konieczność komunikacji asynchronicznej**
- Przechowywanie wideo (**lokalnie -> w chmurze**)





Interfejs użytkownika

Interfejs diagnostyczny dla medycyny



Użytkownicy i ich potrzeby

- Lekarze i laboranci
 - Szybkie i intuicyjne rozpoczęcie procesu analizy
 - Przejrzysta prezentacja wyników
 - Minimalna krzywa uczenia się



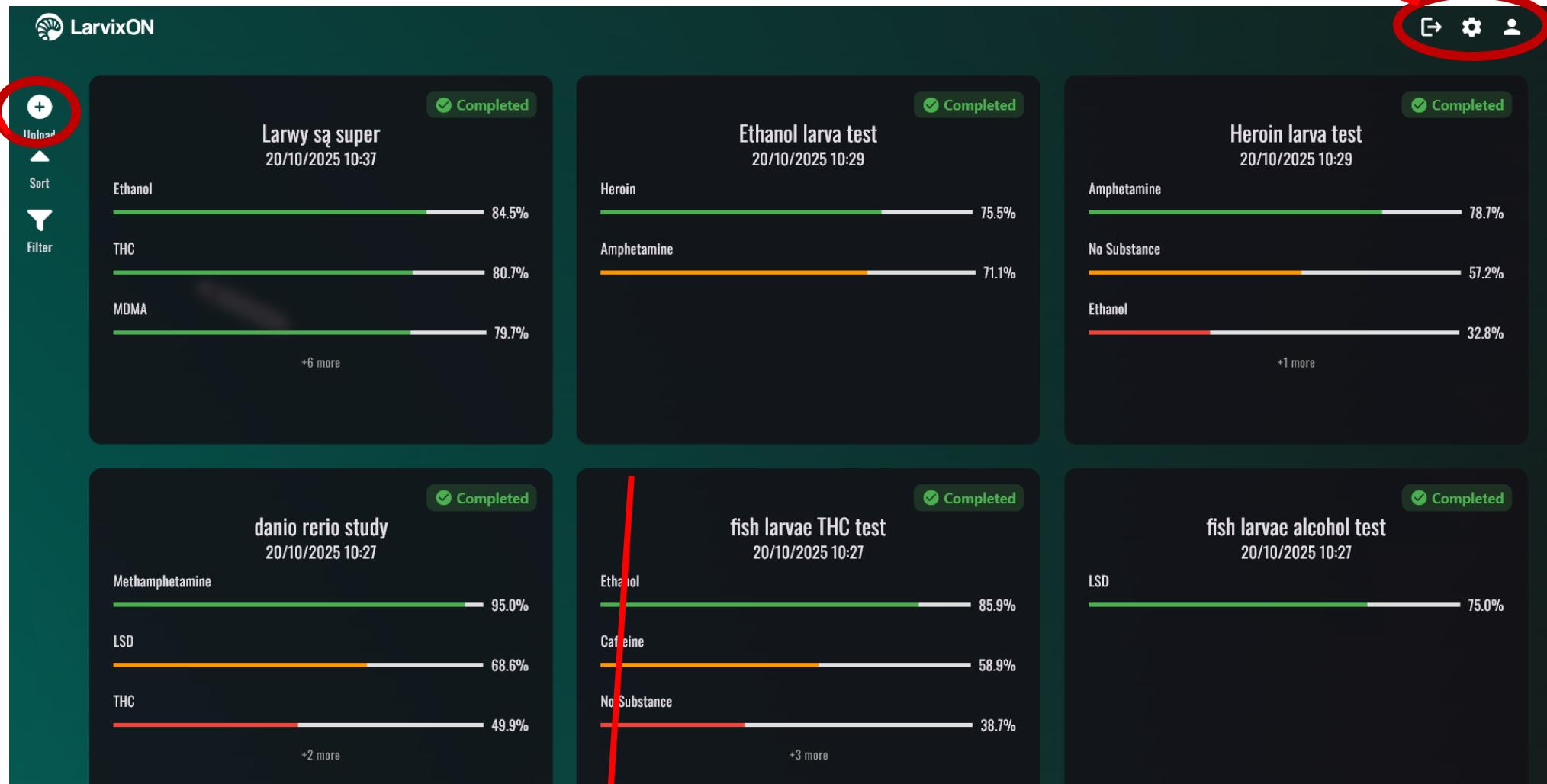
LarvixON – Interfejs Diagnostyczny

- Aplikacja skupiona na przeprowadzaniu analiz diagnostycznych
- Przejrzysty interfejs z intuicyjną nawigacją





Zlecenie nowej analizy



Nawigacja

Widok przeprowadzonych analiz



Kluczowe cechy

- Pipe-line analizy
 - Wczytanie nagrań larw
 - Uruchomienie analizy ML i monitorowanie postępu
 - Wizualizacja wyników predykcji
- Rekomendacje dotyczące neutralizowania wykrytych ksenobiotyków
- Przeglądanie wykonanych analiz
- Obsługa konta użytkownika – lekarza
- Możliwość wieloplatformowej integracji

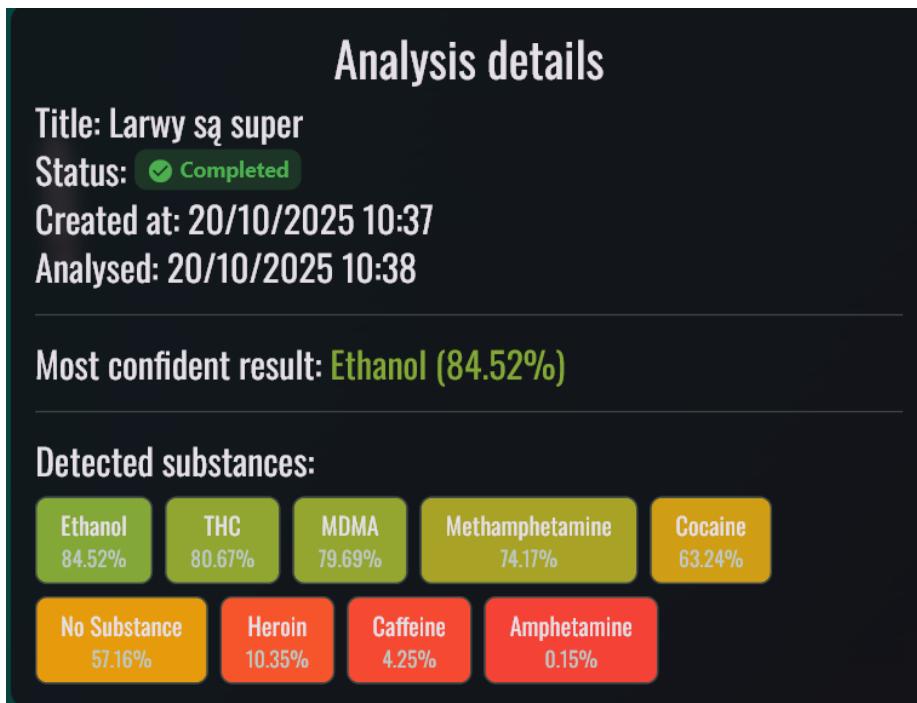
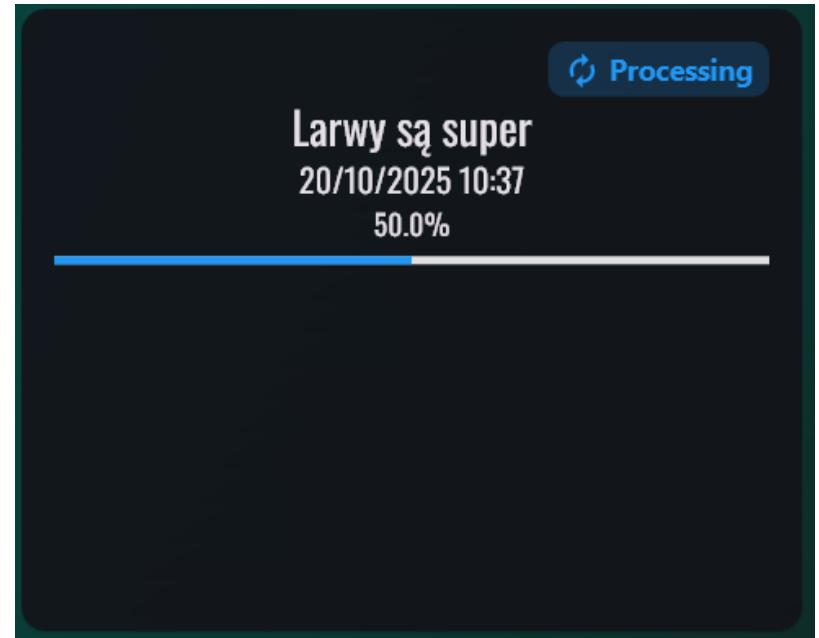


Upload new video
Upload a new video to the platform to analyze larval movements.

Larwy są super

Selected: ketamine[1].webm

Cancel Upload





Harmonogram





Podsumowanie

System **LarvixON** to nowatorskie narzędzie diagnostyczne, którego celem jest **wspomaganie lekarzy przy podejmowaniu decyzji klinicznych** poprzez analizę wzorców ruchowych larw po podaniu osocza pacjentów.





Q&A



Dziękujemy za uwagę

Bibliografia

- [A Deep Learning-Based Hybrid CNN-LSTM Model for Location-Aware Web Service Recommendation](#)
- [Dokumentacja – WCSS](#)
- [Practical Considerations and Applied Examples of Cross-Validation for Model Development and Evaluation in Health Care: Tutorial](#)
- [NVIDIA - AI Powered Medical Imaging](#)