Doggy monitor

Raport z postępów w pracy w tygodniach 1 i 2

<u>Zadanie 1:</u> Rozwój modeli rozpoznawania emocji

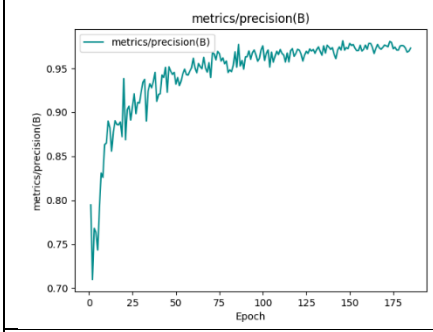
- Trenowanie modeli odczytujących punkty charakterystyczne oraz opracowanie wstępnego algorytmu odczytującego emocje.
- Sprawdzenie możliwości konsultacji z behawiorystą.
- Dokończenie przygotowania datasetu z poprzedniego etapu (tygodnie 1 i 2).
- **Kryterium akceptacji:** 75% skuteczności każdego z modeli.

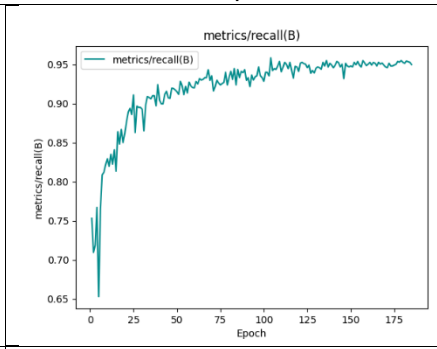
<u>Postępy w pracy:</u>

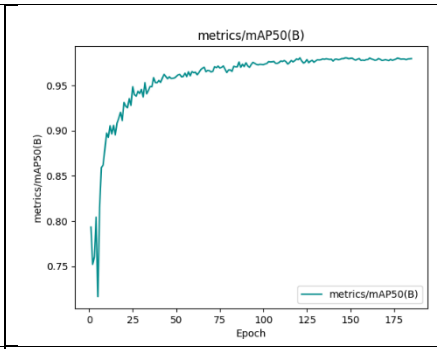
- Początkowo planowano trenować modele ViT, jednak na komputerach z systemem Windows oraz stronie Kaggle napotkano wiele problemów z kompatybilnością bibliotek. Aby zaoszczędzić czas, jedna osoba próbowała rozpocząć trenowanie na systemie Linux, a reszta przeszła na model YOLO11n-pose. Modele YOLO już przy pierwszych treningach dawały obiecujące rezultaty, dlatego zdecydowano o kontynuacji pracy z nimi. Ostatecznie udało się rozpocząć trening ViT, więc jeśli zdążymy, będziemy mogli porównać jego działanie z YOLO.
- W trakcie trenowania YOLO11 zauważono, że model dla grupy drugiej nie uzyskuje tak dobrych rezultatów jak pozostałe. Powodem okazała się niedokładna anotacja części grupy 2A. Obecnie jest ona w trakcie poprawek, a rezultaty zapisane w tym raporcie dotyczą treningu modelu dla grupy 2B.
- Dataset każdej z grup podzielono na części treningową (70%), walidacyjną (20%) i testową (10%).
- **Rezultaty YOLO11n-pose:**
    - <u>Pojęcia:</u>
    - Precision – precyzja; określa, ile z wykrytych obiektów należy do pozytywnych przykładów
    - Recall – czułość; określa, jaki odsetek rzeczywistych obiektów został poprawnie wykryty
    - mAP50 – Mean Average Precision; średnia precyzja przy progu IoU (Intersection over Union) 0.5 dla wykrycia obiektów
    - box loss – strata dotycząca dopasowania granic b-boxa
    - pose loss – strata dotycząca dopasowania parametrów pozycji
    - kobj loss – strata dotycząca wykrywania kluczowych obiektów
    - cls loss – strata dotycząca przypisania wykrytego obiektu do właściwej klasy
    - dfl loss – distance focal loss; strata dotycząca precyzji dopasowania współrzędnych obiektów
    - P lub B przy wykresach precyzji, czułości i mAP50 oznacza, że wykres dotyczy odpowiednio pozycji lub b-boxa.
    - Accuracy – skuteczność – nie jest ona bezpośrednio obliczana w trakcie treningu i walidacji w przypadku modelu YOLO, dlatego napisano skrypt testujący, który porównuje obiekty wykryte przez model z anotacjami. Wykorzystywany do tego jest próg określany jako 7% długości psa w pikselach. Na tej podstawie zliczane są przypadki prawdziwie pozytywne (TP) i negatywne (TN) oraz fałszywie pozytywne (FP) i negatywne (FN), a skuteczność obliczana jest ze wzoru:

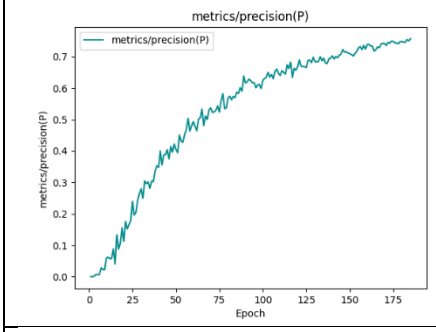
$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

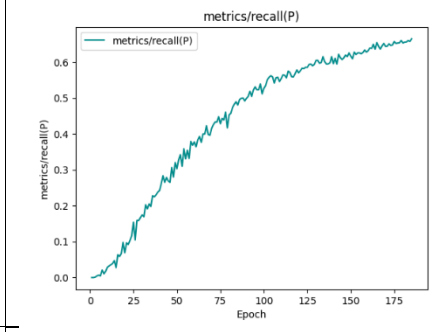
Grupa 1

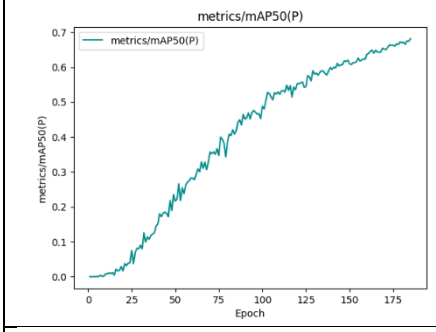


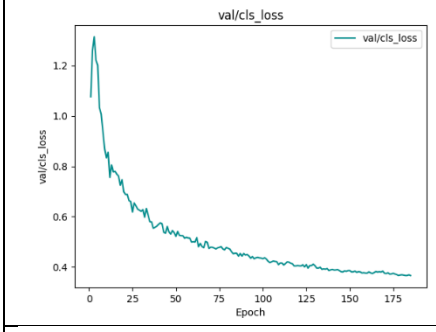


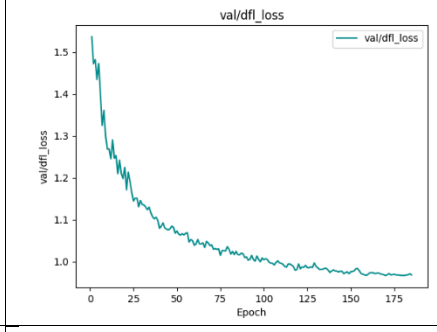


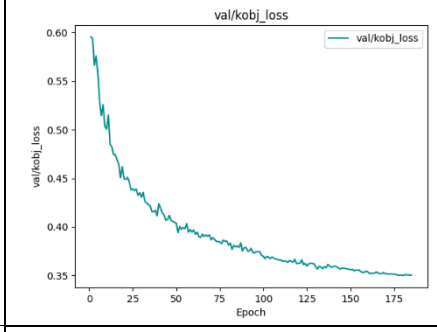


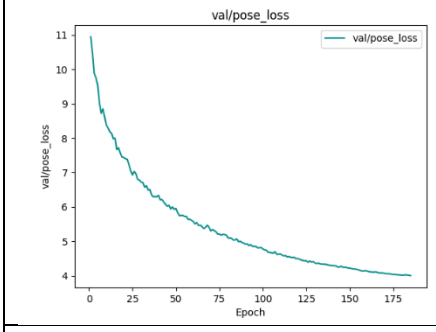


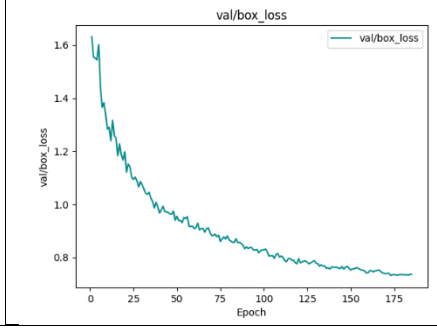










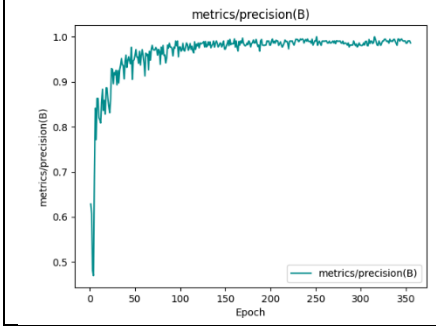


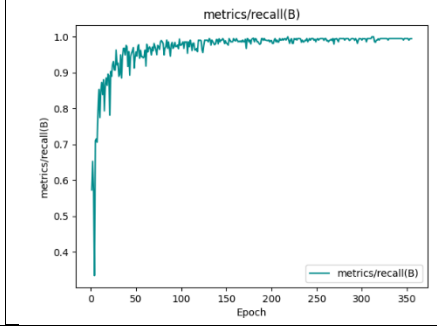
Wyniki testowania modelu:

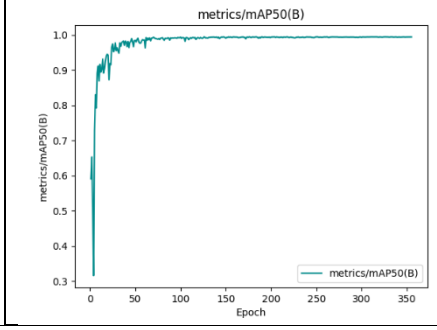
TP = 4381
TN = 4380
FP = 1755
FN = 986

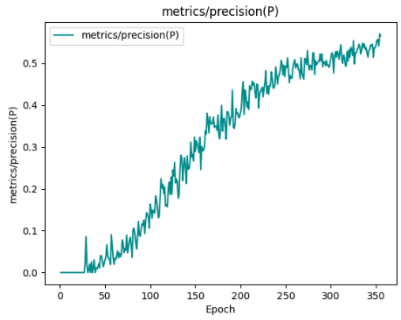
Accuracy = 76,17%

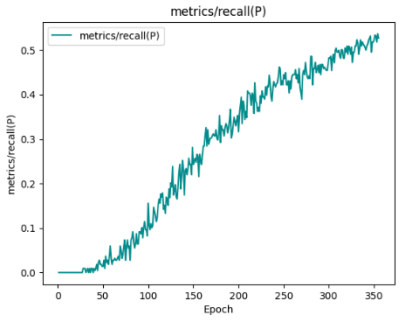
Grupa 2

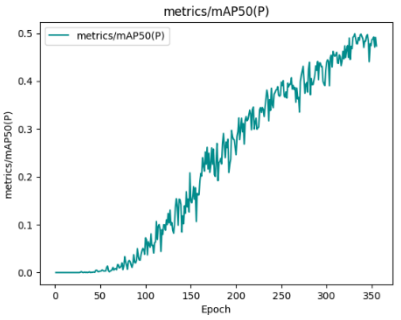


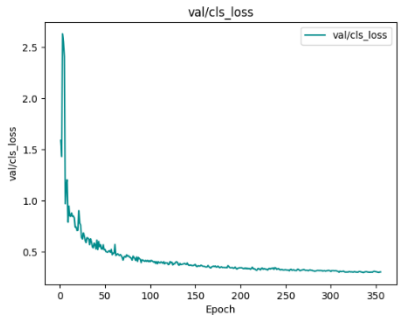


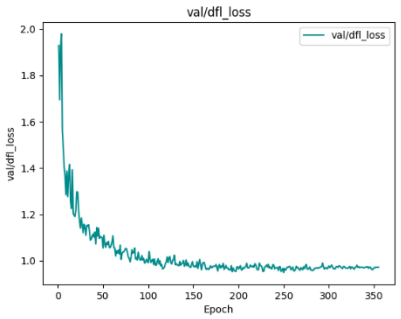


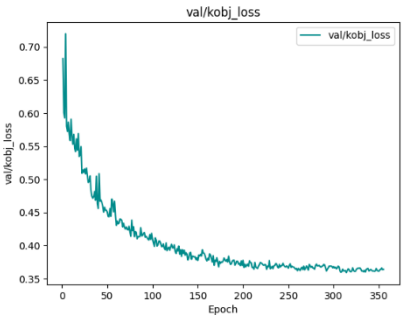


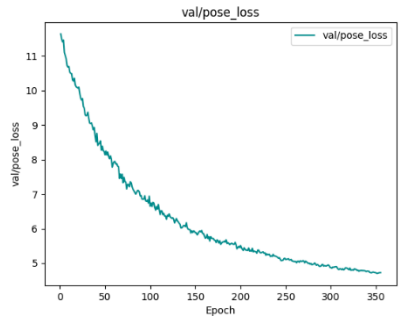


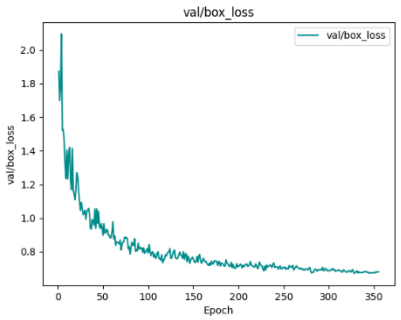










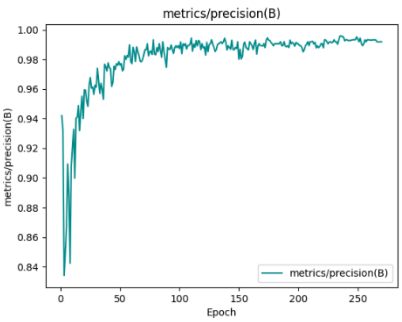


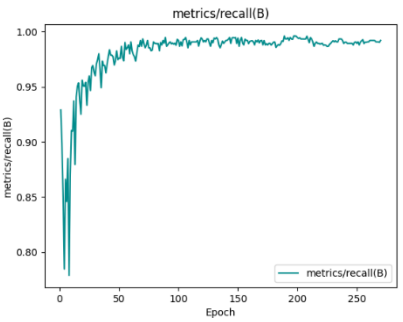
Wyniki testowania modelu:

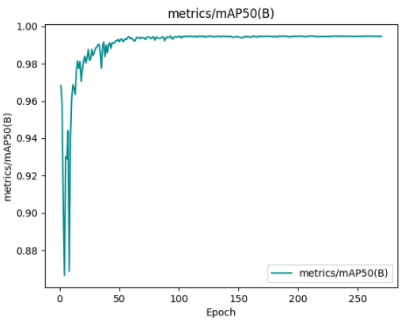
TP = 727
TN = 749
FP = 745
FN = 398

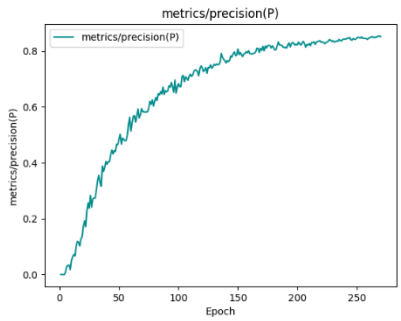
Accuracy = 56,36%

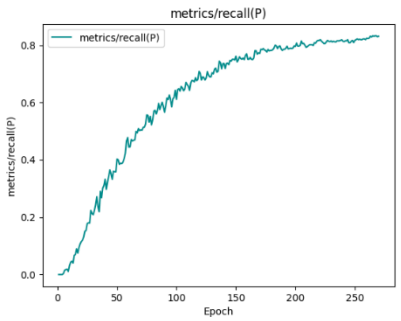
Grupa 3

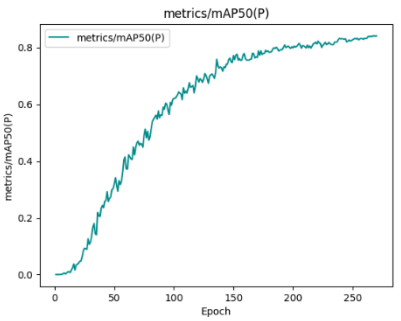


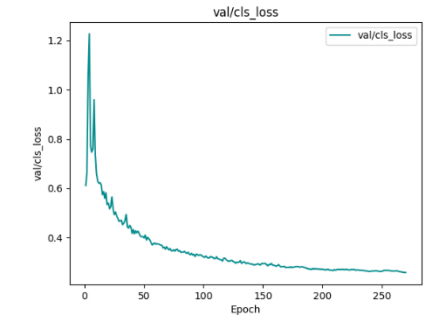
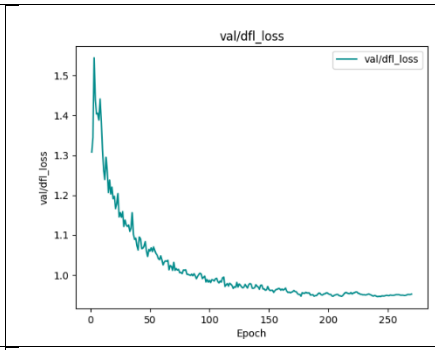
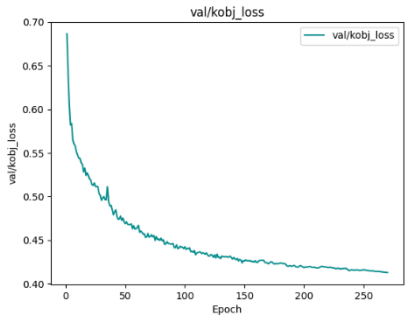
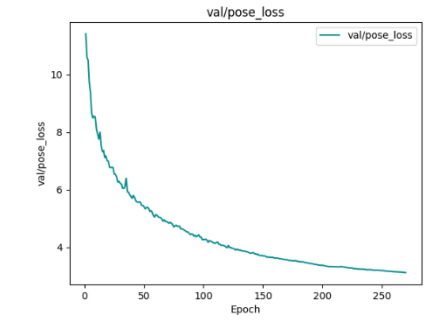
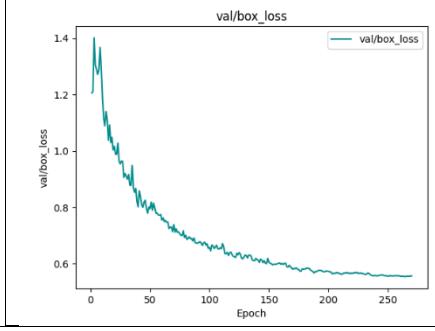










<table>
<tr>
<td> 
</td>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td> 
</td>
<td> 
</td>
<td>Wyniki testowania modelu:

TP = 4912
TN = 2883
FP = 1641
FN = 716

Accuracy = 76,78%</td>
</tr>
</table>

- Wstępny algorytm odczytujący emocje – wykorzystano algorytm napisany w poprzednim semestrze na potrzeby przedmiotu Projekt grupowy i dostosowano go do obecnie wykorzystywanych modeli.
- Sprawdzenie możliwości konsultacji z behawiorystą – otrzymamy analizę poprawności odczytywania emocji od studentów kierunku Zwierzęta w rekreacji, edukacji i terapii Uniwersytetu Warmińsko-Mazurskiego. W razie potrzeby będziemy się również kontaktować ze studentką psychologii będącą w trakcie kursu psiego behawiorysty.
- **Podział zadań:**
    - Zofia Drozdowska – anotacja 2B i 3E, skrypty przygotowujące dane do treningu, trening YOLO dla grupy 2 oraz dotrenowanie tego dla grupy 3.
    - Joanna Ryś – anotacja 3B-E, skrypty przygotowujące dane do treningu, skrypty do testowania modeli, trening YOLO dla grupy 3.
    - Tomasz Sekrecki – anotacja 1A-D, trening YOLO dla grupy 1 oraz sprawdzenie w jaki sposób można trenować model ViT.
    - Marcel Czerwiński – anotacja 1E i 2A oraz dostosowanie algorytmu odczytującego emocje do nowych modeli.
    - Łukasz Marcinkowski i Michał Kruszewski – pomoc w anotacji (okazała się na tyle czasochłonna, że wymagała zaangażowania dwóch kolejnych osób).
- **Spełnienie kryterium akceptacji:** uzyskano wymagane 75% dla modeli grup 1 i 3. Ze względu na błędy w datasecie model grupy 2 wymaga poprawek.

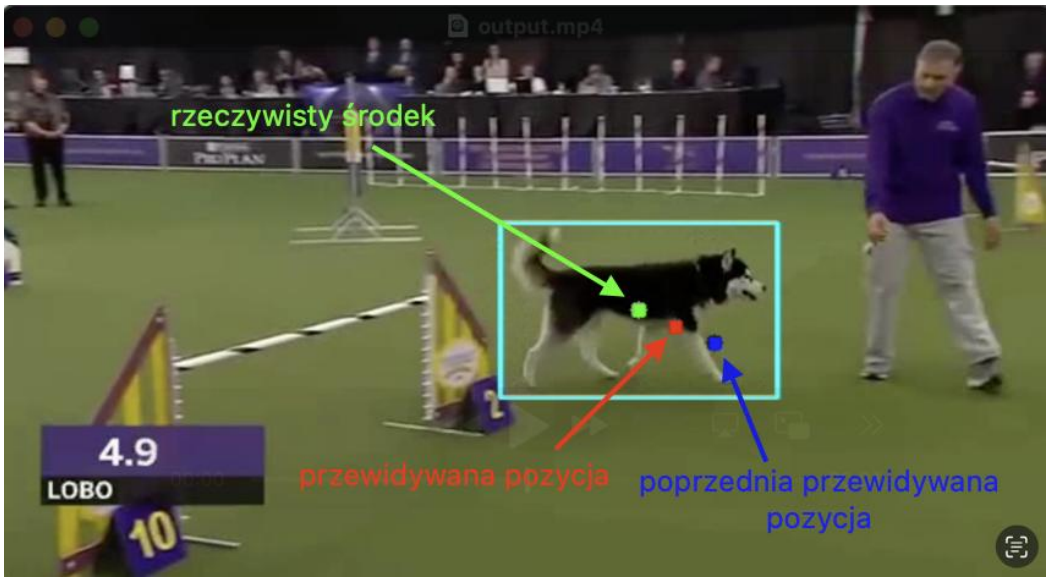
<u>Zadanie 2:</u> Implementacja filtru Kalmana

- Wdrożenie filtru Kalmana w celu szacowania przyszłych ruchów psa, aby zapewnić nadążanie kamery za zwierzęciem.
- **Kryterium akceptacji:** 70% skuteczności w przewidywaniu przyszłych pozycji.

<u>Postępy w pracy:</u>

- **Przewidywanie pozycji**
    - Sterowanie realizowane jest na podstawie współrzędnych obiektu na obrazie
    - Obliczana jest różnica między środkiem kadru a współrzędnymi obiektu, która reprezentuje błąd pozycji.
    - Regulacja pozycji serwomechanizmów odbywa się za pomocą regulatora PID

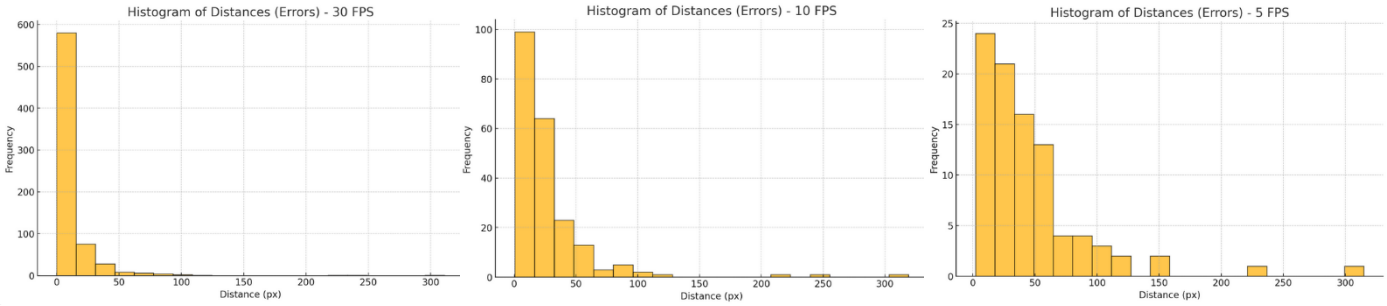
- Wartości wewnętrzne PID (pochodna i całka) są ograniczone górną i dolną wartością tak aby zapobiec występowaniu wind-up
- Nowe kąty serwomechanizmów są obliczane i ograniczane do bezpiecznych zakresów (czyli takich które nie zniszczą układu)
- Zastosowanie filtru Kalmana w celu przewidzenia pozycji, kiedy pozycja obiektu nie została znaleziona
- Wprowadzenie obszaru “deadzone”, aby kamera nie wykonywała małych ruchów w momencie, kiedy obiekt znajduje się blisko środka obrazu



- **Testowanie -** w kontekście przewidywania pozycji można zauważyć, że przewidywana pozycja jest uzależniona od ilości klatek na sekundę. Nasza kamera ma w zależności od prędkości łącza ma 5-10 FPS. Poniżej przedstawiono wyniki testów filtru Kalmana, badających procent pozycji przewidywanych, które znalazły się w b-boxie oraz odległość przewidywanej pozycji od właściwej wyrażoną w pikselach.

Ilość klatek na sekundę	% przewidzianych pozycji w b-boxie	Dolna granica dla górnych 10% błędów
30FPS	98.16%	26px
10FPS	92.49%	53px
5FPS	86.81%	93px

  - W teście przewidywania pozycji obiektu (psa) w nagraniach wideo stosowane są metody detekcji i śledzenia, a wyniki analizowane są pod kątem precyzji przewidywania pozycji.
  - **Obliczanie różnic** - w każdej klatce wideo obliczana jest różnica między przewidywaną pozycją środka obiektu a jego rzeczywistą pozycją. Ta różnica, zwana błędem, wyraża odległość euklidesową między punktami, a jej wartość wskazuje na skuteczność predykcji. Różnica jest wyrażona wzorem $\delta = \sqrt{(x_p - x_r)^2 + (y_p - y_r)^2}$, gdzie (x_p, y_p) to przewidziana pozycja, a (x_r, y_r) to rzeczywista pozycja.
  - **Analiza rozkładu błędów** - zebrane błędy tworzą zbiór danych, którego rozkład można analizować, by określić, jak często i jak znaczne są błędy w przewidywaniach. Histogram rozkładu błędów pozwala na szybki wgląd w dokładność przewidywania, ukazując, czy model jest systematycznie dokładny czy pojawiają się duże odchylenia.
  - **Redukcja i normalizacja danych** - błędy mogą być agregowane w grupy (uśredniane w blokach klatek), by ułatwić analizę trendów i uniknąć szumu w danych. Następnie dane są skalowane, aby lepiej pasowały do porównań, co pozwala łatwo zobaczyć zmiany precyzji przewidywania w czasie.
  - **Wizualizacja błędów** - wykres błędów pokazuje, jak zmienia się odległość przewidywanej pozycji od rzeczywistej w kolejnych klatkach. Taki wykres może wskazać, czy przewidywanie ma tendencję do dryfu (narastania błędu) lub czy systematycznie wraca do rzeczywistej pozycji, co świadczy o jego stabilności.



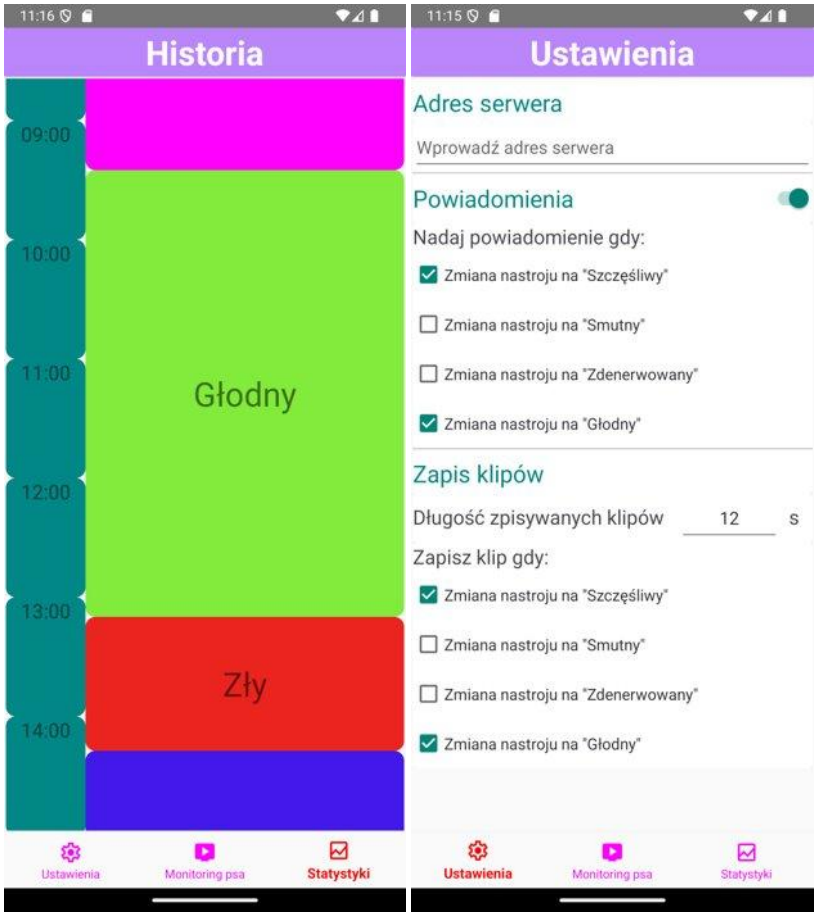
- **Podział zadań:**
    - Olaf Łogin - planowanie zadań (razem z milestonem), setup projektu, poszukiwanie i kupienie odpowiednich elementów do projektu (gimball z serwami, kamera i odpowiedni konwerter), przygotowanie narzędzi do programowania raspberry pi razem z poradnikiem, opracowanie schematów UML dla projektu, obsługa kamery w python, code review
    - Aleksander Fuks – Testowanie protokołów streamujących wideo. Ostateczne skupienie się na RTMP i napisanie skryptu uruchamiającego stream na serwer globalny. Opóźnienie wynosiło około 20 sekund, co prawdopodobnie było spowodowane korzystaniem z darmowej wersji konta w usłudze serwerowej. Przy streamingu na localhost opóźnienie nie przekraczało 3 sekund. Zainstalowanie i konfiguracja systemu linux na rpi.
    - Maksymilian Burdziej – sterowanie serwami za pomocą RPI. Napisanie programu umożliwiającego poruszanie serwami (na zasadach PID) w sposób, który umożliwia śledzenie obiektu w czasie rzeczywistym. Wykorzystanie filtru Kalmana do przewidywania następnej pozycji obiektu, w przypadku, kiedy nie została znaleziona rzeczywista pozycja. Napisanie programu oceniającego dokładność filtru Kalmana na podstawie filmów z psami.

<u>Zadanie 3:</u> Prototyp front-endu dla aplikacji właściciela

- Zaprojektowanie i stworzenie podstawowej aplikacji dla właściciela psa.
- **Kryterium akceptacji:** nie dotyczy.

<u>Postępy w pracy:</u>

- Przygotowano wstępny projekt aplikacji w języku kotlin. Na chwilę obecną są zapewnione:
    - Responsywne menu i layout
    - Wyświetlanie pliku video z serwera w głównym ekranie
    - Zapis ustawień i preferencji użytkownika
    - Layout historii



- Całe zadanie zrealizował Maciej Rogowicz.

Podsumowanie

Każdy z członków zespołu wykonał przydzielone mu zadania i nie występowały problemy z komunikacją. Pojawiły się błędy w anotacji filmów związane z gorszą precyzją, jednak większa część projektu jest już gotowa i mamy czas na poprawki i inne udoskonalenia.

Oceny		
Członkowie zespołu	**Tygodnie 1-2**	**Tygodnie 3-4 (milestone 1)**
Zofia Drozdowska	5	5
Joanna Ryś	5	5
Marcel Czerwiński	5	4.5 (-0.5 oceny za będy w datasecie)
Tomasz Sekrecki	5	5
Olaf Łogin	5	5
Aleksander Fuks	5	5
Łukasz Marcinkowski	5	5
Maksymilian Burdziej	5	5
Micha Kruszewski	5	5
Maciej Rogowicz	5	5