



Politechnika Łódzka

**Wydział Fizyki Technicznej, Informatyki
i Matematyki Stosowanej**

Marcin Mazur
242467

PRACA DYPLOMOWA
inżynierska
na kierunku Informatyka Stosowana

**Wykorzystanie oprogramowania Open-Source do
współpracy z kamerami TP-Link TAPO**

Instytut Informatyki I72
Promotor: dr inż. Krzysztof Lichy

ŁÓDŹ 2026

Spis treści

Streszczenie	3
Słowa kluczowe	3
Wstęp	4
Cel i zakres pracy	5
1 Wprowadzenie technologiczne Kamer IP	7
1.1 Zastosowanie Kamer IP	7
1.1.1 Monitoring	9
1.1.2 Kontrola Dostępu	9
1.1.3 Zarządzanie Procesami Biznesowymi	10
1.1.4 Technologie Smart	10
1.1.5 Analiza Danych	10
1.2 Budowa	11
1.2.1 Budowa Fizyczna - Hardware	11
1.2.2 Oprogramowanie - Software	11
1.3 Zasada działania	11
1.3.1 Zarządzanie Zasilaniem	11
1.3.2 Komunikacja	12
1.3.3 Provisioning	12
1.3.4 Proces przetwarzania obrazu	12
1.3.5 Proces przetwarzania dźwięku	12
1.3.6 Streamowanie	12
1.4 Funkcje	12
1.4.1 Obrót PTZ	12
1.4.2 Wykrywanie obiektów i zdarzeń - AI	12
1.4.3 Wykrywanie ruchu	12
1.4.4 Noktowizja i termowizja	12
1.4.5 Dwukierunkowe audio	12
1.4.6 Zapis danych	12
1.4.7 Integracja z Inteligentnymi Systemami	12
1.4.8 Powiadomienia push	12
1.5 Ograniczenia	12
1.5.1 Przepustowość i zużycie danych	12
1.5.2 Zależność od sieci	12
1.5.3 Bezpieczeństwo	12
1.5.4 Zależność od producenta / Chmura	12

1.6	Wnioski - Analiza	12
2	Analiza Kamery TP-Link TAPO C200	13
2.1	Specyfikacja techniczna i architektura funkcjonalna	13
2.2	Analiza Protokołów Komunikacyjnych i Potencjału Integracji Open Source	13
2.3	Wbudowane funkcje AI a Wyzwanie Open Source (Vendor Lock-in) . . .	14
3	Metodologia i implementacja rozwiązania	15
3.1	Metoda Badawcza	15
3.1.1	Double Diamond	15
3.1.2	Projektowanie i implementacja	15
3.2	Założenia	15
3.3	Zastosowane narzędzia i technologie	15
3.3.1	Język Programowania: Python 3.13	15
3.3.2	Konteneryzacja: Docker	15
3.3.3	Package Manager: uv	15
3.3.4	Audio, Video i Streaming	15
3.3.5	Web Server	15
3.3.6	Kontrola Kamery: PyTapo	15
3.4	Architektura rozwiązania	15
3.5	Proces implementacji	16
3.5.1	Serwer http	16
3.5.2	Implementacja połączenia z kamerą	16
3.5.3	Client	16
3.5.4	API	16
3.5.5	Przechwytywanie audio	16
3.5.6	Przechwytywanie wideo	16
3.5.7	Sterowanie kamerą - PTZ	16
3.5.8	Algorytm wykrywania ruchu	16
3.5.9	Nagrywanie	16
3.5.10	Zapis	16
3.6	Podsumowanie	16
4	Testowanie i Analiza wyników	17
4.1	Zakres Testów	17
4.2	Środowisko Testowe	17
4.3	Wyniki testów i Analiza	17
4.4	Podsumowanie	17
	Wnioski Końcowe	18

Kierunki dalszego rozwoju	18
Podsumowanie pracy	18
Spis rysunków	20
Spis tabel	21

Streszczenie

Słowa kluczowe

IoT, Kamera IP, TP-Link Tapo, Open Source, PyTapo, Docker, RTSP, Detekcja Ruchu, Flask.

Wstęp

Globalny rynek systemów monitoringu przechodzi dynamiczną transformację, będącą efektem rozwoju **Internetu Rzeczy (IoT)**. Kamery IP stały się wszechobecnym elementem infrastruktury cyfrowej, pełniąc funkcje od podstawowego dozoru, aż po zaawansowaną analizę danych. Równolegle z postępem technologicznym, pojawia się wyzwanie o charakterze inżynierskim, jakim jest dominacja systemów opartych na **zamkniętym oprogramowaniu (proprietary software)**.

Wybór tematu pracy wynika z konieczności zaadresowania problemu **vendor lock-in** w kontekście popularnych kamer konsumenckich, na przykładzie urządzeń TP-Link Tapo. Zjawisko to, polegające na uzależnieniu pełnej funkcjonalności sprzętu od infrastruktury chmurowej i aplikacji mobilnej producenta, ogranicza **dostępność danych** oraz **możliwości integracji** z otwartymi systemami automatyki i bezpieczeństwa. Problem ten jest szczególnie istotny w kontekście **cyber bezpieczeństwa**, gdzie zamknięte i często nieaudytowane firmware może stanowić potencjalny wektor ataku.

W pracy zastosowano **metodykę Double Diamond**, dzieląc proces projektowy na fazy eksploracji i definiowania problemu (analiza protokołów kamery) oraz fazy rozwoju i dostarczania rozwiązania. Warstwa aplikacyjna została zaimplementowana w języku **Python 3.13** z wykorzystaniem **konteneryzacji Docker** dla zapewnienia izolacji i wysokiej **reprodukowalności środowiska**. Komunikacja z kamerą odbywa się poprzez bibliotekę **PyTapo**, natomiast przetwarzanie strumienia wideo RTSP realizują narzędzia **FFmpeg** i **OpenCV**. Taki zestaw narzędzi, osadzony w architekturze serwera **Flask** z protokołem **WebSocket's**, pozwolił na stworzenie systemu o niskim opóźnieniu (*low latency*).

Niniejsza praca ma za zadanie stanowić nie tylko dowód kompetencji inżynierskich, ale także praktyczny wkład w rozwój otwartych technologii w dziedzinie monitoringu IoT.

Cel i zakres pracy

Cel

Celem głównym niniejszej pracy inżynierskiej jest opracowanie oraz implementacja kompletnego, modułowego rozwiązania programistycznego opartego wyłącznie na **otwartym oprogramowaniu (Open Source)**, które umożliwi pełne wykorzystanie funkcjonalności kamery IP TP-Link Tapo C200 w środowisku lokalnym i niezależni użytkownika od zamkniętej infrastruktury producenta (problem *vendor lock-in*).

Osiągnięcie celu głównego jest weryfikowane poprzez realizację następujących, **konkretnych i mierzalnych** celów szczegółowych:

- Umożliwienie stabilnego **wyświetlania obrazu w czasie rzeczywistym** w przeglądarce internetowej. Weryfikacja nastąpi poprzez pomiar **opóźnienia strumienia wideo (*Latency*)** oraz wskaźnika **klatek na sekundę (FPS)**, celem osiągnięcia płynności monitoringu porównywalnej z dedykowanymi systemami VMS.
- Zdalne sterowanie kluczowymi funkcjami kamery, w tym **ruchem PTZ** (Pan/Tilt/Zoom), za pośrednictwem własnego API.
- Implementacja **algorytmu wykrywania ruchu**, z poziomu serwera hostującego. Weryfikacja nastąpi poprzez analizę **efektywności algorytmów** mierzoną w kategoriach czasu przetwarzania klatki oraz minimalizacji błędów detekcji.
- Zbudowanie rozwiązania w oparciu o technologię **Docker** w celu zapewnienia **skalowalności systemu** oraz **reprodukowalności środowiska** na platformach mikserwerowych IoT (np. Raspberry Pi).
- Implementacja funkcjonalności **zapisu nagrań wideo** na serwerze hostującym z możliwością ich późniejszego **odtwarzania** poprzez interfejs webowy.
- Osiągnięcie architektury gotowej do **integracji** z otwartymi platformami automatyki domowej (np. Home Assistant).

Zakres Pracy

Zakres pracy inżynierskiej obejmuje projektowanie, implementację oraz testowanie modułowego systemu klient-serwer. Praca stanowi odpowiedź na problem *vendor lock-in* w segmencie kamer IoT, uzasadniając wybór tematu rosnącą potrzebą na otwarte i bezpieczne systemy zarządzania danymi.

Aspekty objęte zakresem pracy

- Projekt trójwarstwowej architektury kontenerowej (Docker) dla warstwy dostępu do sprzętu, logiki biznesowej (Flask/WebSocket's) oraz warstwy prezentacji (Web Client).
- Wykorzystanie biblioteki PyTape do obsługi nieudokumentowanego API komend sterujących kamery (PTZ, tryb nocny, aktywacja usług).
- Praca skupia się na przechwytywaniu jednokierunkowego strumienia wideo i audio. Implementacja pełnej komunikacji zwrotnej w czasie rzeczywistym jest **poza zakresem** projektu.
- Przeprowadzenie **testów wydajnościowych** skupiających się na **zużyciu zasobów (CPU/RAM)** hosta podczas ciągłej analizy strumienia wideo.

Wyłączenia z zakresu pracy

W celu zachowania osiągalności i weryfikowalności celów w ramach pracy inżynierskiej, poniższe aspekty zostały wykluczone, ze względu na ich złożoność badawczą lub techniczną:

- Protokół inicjalizacji - **Provisioning** - kamery Tapo w trybie Access Point (AP) jest nieudokumentowany, szyfrowany i opiera się na wymianie kluczy sesyjnych, co wymaga weryfikacji po stronie chmury TP-Link. Odtworzenie tego protokołu jest zadaniem na poziomie projektu badawczego i **wykracza poza ramy niniejszej pracy**. W konsekwencji, praca zakłada, że **kamera została jednorazowo skonfigurowana w sieci Wi-Fi** przy użyciu oficjalnej aplikacji mobilnej.
- Implementacja modeli **uczenia maszynowego** (np. rozpoznawanie twarzy, klasyfikacja obiektów - YOLO), ze względu na wysokie wymagania obliczeniowe i złożoność czasową, **została wykluczona**. Praca koncentruje się na detekcji ruchu opartej na różnicy klatek (OpenCV).

1 Wprowadzenie technologiczne Kamer IP

Rozdział ten ma za zadanie ugruntować zrozumienie złożoności systemów kamer IP i precyzyjnie wskazać na luki w otwartych standardach, które musi wypełnić zaprojektowane rozwiązanie.

Współczesne systemy monitoringu wizyjnego oparte na kamerach IP stanowią kluczowy element infrastruktury bezpieczeństwa, wykraczając funkcjonalnością poza tradycyjne, analogowe systemy CCTV. Ewolucja ta jest ściśle związana z rozwojem sieci komputerowych i koncepcji IoT, gdzie urządzenia peryferyjne uzyskują zdolność do przetwarzania i autonomicznej komunikacji w ramach sieci. Z inżynierskiego punktu widzenia, kamera IP jest zaawansowanym systemem wbudowanym, łączącym optykę, cyfrowe przetwarzanie sygnału, kompresję danych oraz kompleksowy stos protokołów sieciowych.

1.1 Zastosowanie Kamer IP

Na podstawie raportu Hanwha Vision z 2025 roku, można wyróżnić następujące, główne obszary zastosowań kamer IP **Hanwha:2025**:

Tabela 1.1: Główne obszary zastosowań kamer IP (na podstawie raportu Hanwha Vision, 2025)

Obszar zastosowania	Przykłady wykorzystania kamer IP
Bezpieczeństwo publiczne	Monitorowanie ulic, placów, obiektów strategicznych; automatyczne wykrywanie zagrożeń i incydentów.
Transport i logistyka	Monitoring lotnisk, dworców, portów; analiza przepływu pasażerów; automatyczne rozpoznawanie tablic rejestracyjnych.
Przemysł	Kontrola procesów produkcyjnych, wykrywanie awarii maszyn, nadzór nad pracownikami i bezpieczeństwem pracy.
Handel detaliczny	Zapobieganie kradzieżom, analiza zachowań klientów, optymalizacja układu sklepu.
Edukacja	Zwiększanie bezpieczeństwa uczniów i nauczycieli, kontrola dostępu do budynków szkolnych.
Ochrona zdrowia	Nadzór nad pacjentami i personelem, zabezpieczenie pomieszczeń szpitalnych, kontrola dostępu do stref wrażliwych.
Smart City	Analiza ruchu drogowego, inteligentne sterowanie sygnalizacją świetlną, planowanie urbanistyczne na podstawie danych z kamer.

Zastosowanie monitoringu wizyjnego opartego na kamerach IP jest obecnie wielosektorowe i dynamiczne. Urządzenia te, integrujące funkcje sensora i procesora danych, stały się podstawą **systemów analitycznych** w kluczowych obszarach gospodarki i bezpieczeństwa. W kontekście dalszego rozwoju monitoringu wizyjnego, szczególne znaczenie zyskuje **sztuczna inteligencja (AI)** i **uczenie maszynowe (ML)**. Nowoczesne algorytmy pozwalają na automatyczną detekcję zagrożeń, eliminację fałszywych alarmów oraz identyfikację i śledzenie obiektów w czasie rzeczywistym. Integracja tych zaawansowanych technik z **otwartym oprogramowaniem** — co jest celem niniejszej pracy — otwiera drogę do stworzenia bardziej zaawansowanych, konfigurowalnych i niezależnych narzędzi wspierających bezpieczeństwo oraz analitykę zdarzeń. Rozwój kamer IP, szczególnie w kontekście inteligentnego monitoringu, jest ściśle powiązany z ewolucją **Narzędzi Kognitywnych (Cognitive Tools)**.

Narzędzia kognitywne w monitoringu wizyjnym działają na zasadzie mechanizmów inferencji, które imitują procesy decyzyjne i percepcyjne ludzkiego mózgu. Umożliwiają one systemom na przechodzenie od prostej detekcji ruchu do **zrozumienia kontekstu i intencji** obserwowanych zdarzeń **Fan:Stanford:2020**. Dzięki temu, system monitorujący może automatycznie filtrować szum wizualny i koncentrować uwagę na zdarzeniach o wysokim prawdopodobieństwie zagrożenia lub anomalii. Technologie te transformują surowe dane wideo w zorganizowane i użyteczne metadane, co jest fundamentalne dla automatyki i bezpieczeństwa. W przeciwieństwie do tradycyjnej detekcji ruchu opartej na różnicy pikseli (algorytm *frame differencing*), nowoczesne systemy wizyjne wykorzystują głębokie sieci neuronowe (DNN) do zaawansowanej analizy obrazu. Pozwala to na realizację funkcji inżynierskich o wysokiej wartości:

Efektywna Analiza Danych wymaga interoperacyjności. Jest to główny powód, dla którego w niniejszej pracy inżynierskiej dąży się do uwolnienia strumienia danych z kamery Tapo C200. Tylko otwarty dostęp do strumienia wideo i metadanych umożliwia ich integrację z zaawansowanymi platformami analitycznymi (np. platformy IoT, systemy Business Intelligence), co jest niemożliwe w zamkniętych ekosystemach producentów. Kamera IP, połączona z narzędziami kognitywnymi (AI), przestaje być pasywnym urządzeniem rejestrującym, a staje się aktywnym sensorem generującym **metadane strukturalne**. W kontekście systemów Big Data, strumień wideo jest przetworzony w chmurze lub na urządzeniu brzegowym (*edge computing*) na użyteczne informacje, takie jak:

1. Liczba wykrytych obiektów (ludzie, pojazdy), ich zagęszczenie (tzw. *heatmaps*) oraz czas przebywania w określonej strefie (np. w handlu detalicznym) znane jako **dane statystyczne**.
2. Analiza ścieżek ruchu, wykrywanie nietypowych wzorców zachowania (np. bieganie w strefie zakazu, pozostawienie bagażu) oraz trendów sezonowych w na-

tężeniu ruchu znane jako **dane behawioralne**.

3. Na podstawie historycznych i bieżących danych, systemy AI mogą przewidywać prawdopodobne incydenty. Przykładowo, zwiększone zagęszczenie osób w metrze w połączeniu z nietypowymi wzorcami ruchu może wygenerować alarm o potencjalnym zatorze lub wypadku, zanim ten nastąpi, nazywane **analitiką predykcyjną**.

1.1.1 Monitoring

Podstawowym i historycznym zastosowaniem kamery IP jest **nadzór wizyjny (monitoring)**. W odróżnieniu od analogowego CCTV, monitoring oparty na protokole internetowym umożliwia przesyłanie strumienia wideo wysokiej rozdzielczości (np. 1080p w Tapo C200) oraz metadanych poprzez standardowe sieci LAN/WLAN. Z technicznego punktu widzenia, monitoring realizowany jest poprzez ciągłe kodowanie wideo (standardy H.264/H.265), strumieniowanie za pomocą protokołów czasu rzeczywistego (**RTSP**) oraz zapis cyfrowy na nośnikach lokalnych (microSD, serwer NVR) lub w chmurze. Zaawansowane funkcje, takie jak **PTZ (Pan-Tilt-Zoom)**, dają inżynierom możliwość dynamicznego dostosowania pola widzenia i śledzenia obiektów bez ingerencji fizycznej, co jest kluczowe w monitorowaniu dużych obszarów (np. hal magazynowych) **Al-Fuqaha:2015**.

1.1.2 Kontrola Dostępu

Kamery IP są coraz częściej integrowane z systemami **Kontroli Dostępu (Access Control Systems - ACS)**. Ich rola wykracza poza zwykłe weryfikowanie tożsamości. Dzięki wykorzystaniu AI, kamery stają się kluczowym sensorem w bezdotykowej autoryzacji. Przykłady zastosowań inżynierskich obejmują:

1. **Rozpoznawanie Twarzy (Facial Recognition)**: Zastosowanie algorytmów głębokiego uczenia do identyfikacji i weryfikacji osób uprawnionych, automatycznie odblokowując wejścia.
2. **Rozpoznawanie Tablic Rejestracyjnych (ANPR)**: Automatyczne zezwalanie na wjazd pojazdów do strzeżonych stref (np. parkingów pracowniczych) na podstawie analizy obrazu z kamery.

Takie rozwiązania minimalizują ryzyko błędów ludzkich i zwiększają bezpieczeństwo poprzez ciągłe logowanie zdarzeń wejścia i wyjścia, stanowiąc integralną część zabezpieczeń fizycznych i sieciowych **Bou-Harb:2024**.

1.1.3 Zarządzanie Procesami Biznesowymi

Wykorzystanie kamer IP w zarządzaniu procesami (**Business Process Management - BPM**) koncentruje się na optymalizacji operacyjnej poprzez zbieranie danych o efektywności i bezpieczeństwie pracy. W sektorach takich jak produkcja i logistyka, kamery są używane do:

1. **Kontroli Jakości (Quality Assurance - QA):** Monitorowanie linii produkcyjnych w celu automatycznego wykrywania defektów, niezgodności montażu lub nieprawidłowej sekwencji działań.
2. **Optymalizacji Przepływu Pracy (Workflow Optimization):** Analiza ścieżek ruchu pracowników i pojazdów w celu identyfikacji wąskich gardeł w magazynach i centrach dystrybucyjnych.

Te zastosowania wymagają wysokiej precyzji metadanych i niskiego opóźnienia, co stawia wysokie wymagania przed **algorytmami analizy brzegowej (Edge Analytics)**, które muszą działać na poziomie procesora kamery lub serwera lokalnego Abdalla, 2020.

1.1.4 Technologie Smart

Kamery IP są fundamentalnym elementem **ekosystemów Smart Home i Smart City**. W tych kontekstach, kamera pełni rolę czujnika behawioralnego, dostarczając danych do zautomatyzowanych systemów decyzyjnych. W budownictwie inteligentnym, Tapo C200, podobnie jak inne urządzenia IoT, jest zintegrowana za pomocą protokołów API z platformami takimi jak **Google Assistant i Amazon Alexa** (jak wskazano w dokumentacji Tapo). Przykłady zastosowań to:

1. **Automatyzacja Zdarzeniowa:** Detekcja ruchu lub dźwięku (np. wykrywanie płaczu dziecka w Tapo C200) uruchamia inne urządzenia (np. włącza światło, wysyła alert do systemu zarządzania domem).
2. **Zarządzanie Energią:** Wykrycie braku obecności osób w pomieszczeniu może prowadzić do automatycznego obniżenia temperatury lub wyłączenia niepotrzebnych urządzeń, przyczyniając się do zwiększenia efektywności energetycznej.

Ten obszar ilustruje potrzebę **interoperacyjności**, która jest blokowana przez zamknięte protokoły chmurowe, co stanowi główną motywację dla niniejszej pracy inżynierskiej.

1.1.5 Analiza Danych

Kamera IP, połączona z narzędziami kognitywnymi (AI), przestaje być pasywnym urządzeniem rejestrującym, a staje się aktywnym sensorem generującym **metadane struk-**

turalne. W kontekście systemów Big Data, strumień wideo jest intensywnie przetwarzany, stanowiąc bazę dla analityki w czasie rzeczywistym i prognozowania zdarzeń.

Efektywne wykorzystanie danych wizyjnych do celów analitycznych obejmuje trzy główne poziomy inżynierskie:

1. **Ekstrakcja Danych Statystycznych:** Dotyczy pomiarów ilościowych, takich jak gęstość obiektów, liczenie przepływu (*flow counting*) oraz generowanie map ciepła (*heatmaps*) Minerva, Biru i Rotondi, 2021.
2. **Analiza Behawioralna i Wzorce Trendów:** Identyfikacja nietypowych sekwencji zdarzeń, które mogą sugerować incydent bezpieczeństwa (np. pozostawiony pakunek) **Al-Fuqaha:2015.**
3. **Analityka Predykcyjna (Predictive Analytics):** Przewidywanie potencjalnych przyszłych zdarzeń na podstawie historycznych i bieżących metadanych. Wymaga to integracji i walidacji danych z wielu źródeł IoT Alaba i in., 2017.

Możliwość pełnej i niezależnej **Analizy Danych (Data Analytics)** jest ściśle powiązana z problemem *vendor lock-in*. Uwolnienie strumienia z kamery Tapo C200 jest podstawowym warunkiem inżynierskim dla realizacji zaawansowanej analityki danych.

1.2 Budowa

1.2.1 Budowa Fizyczna - Hardware

- Matryca
- Mikrofon
- Zasilanie
- Układ scalony (SoC)
 - CPU
 - Pamięć
 - Network

1.2.2 Oprogramowanie - Software

1.3 Zasada działania

1.3.1 Zarządzanie Zasilaniem

- POE
- Zasilanie

1.3.2 Komunikacja

- Wifi
- HTTP

1.3.3 Provisioning

1.3.4 Proces przetwarzania obrazu

1.3.5 Proces przetwarzania dźwięku

1.3.6 Streamowanie

1.4 Funkcje

1.4.1 Obrót PTZ

1.4.2 Wykrywanie obiektów i zdarzeń - AI

1.4.3 Wykrywanie ruchu

1.4.4 Noktowizja i termowizja

1.4.5 Dwukierunkowe audio

1.4.6 Zapis danych

1.4.7 Integracja z Inteligentnymi Systemami

1.4.8 Powiadomienia push

1.5 Ograniczenia

Ograniczenia wynikające z technologii i modelu biznesowego producentów.

1.5.1 Przepustowość i zużycie danych

1.5.2 Zależność od sieci

1.5.3 Bezpieczeństwo

1.5.4 Zależność od producenta / Chmura

1.6 Wnioski - Analiza

2 Analiza Kamery TP-Link TAPO C200

Kamera TP-Link Tapo C200 stanowi reprezentatywny przykład współczesnego, konsumenckiego urządzenia klasy **IoT (Internet of Things)** przeznaczonego do monitoringu wewnętrznego. Ze względu na swoją popularność i uwarunkowania techniczne, jest idealnym studium przypadku dla oceny możliwości integracji z **otwartymi ekosystemami (Open Source)**, co stanowi główny cel niniejszej pracy inżynierskiej.

2.1 Specyfikacja techniczna i architektura funkcjonalna

Kamera Tapo C200 to urządzenie typu **Pan/Tilt (PTZ)**, oferujące zdalny obrót w zakresie horyzontalnym (360°) oraz wertykalnym. Kluczowe parametry w kontekście przetwarzania obrazu i wydajności sieciowej bazują na oficjalnej specyfikacji technicznej:

- **Maksymalna Rozdzielczość:** 1080P HD (1920x1080 px).
- **Klatkaż:** 30 fps, co zapewnia płynny obraz.
- **Kompresja Video:** H.264, standard przemysłowy dla efektywnego przesyłania wideo.
- **Łączność Bezprzewodowa:** 2.4 GHz Wi-Fi (IEEE 802.11b/g/n).
- **Pamięć Lokalna:** Slot microSD (do 512 GB), umożliwiający zapis nagrań w trybie offline.

Architektura urządzenia jest oparta na wbudowanym układzie **SoC (System-on-a-Chip)**, który odpowiada za przetwarzanie obrazu, sprzętową kompresję H.264 oraz obsługę protokołów sieciowych.

2.2 Analiza Protokołów Komunikacyjnych i Potencjału Integracji Open Source

Mimo przynależności do zamkniętego ekosystemu, kamera udostępnia kluczowe protokoły umożliwiające **lokalną, niezależną integrację**, co jest warunkiem koniecznym dla realizacji niniejszego projektu:

1. **RTSP (Real-Time Streaming Protocol):** Protokół ten jest obsługiwany przez kamerę i umożliwia pobieranie surowego strumienia wideo i audio w czasie rzeczywistym. Jest to fundament, który pozwala na wykorzystanie otwartych narzędzi, takich jak FFmpeg i OpenCV, do analizy i przetwarzania obrazu poza zamkniętą aplikacją.
2. **ONVIF (Open Network Video Interface Forum):** Choć standard jest obsługiwany, pełna funkcjonalność i integracja często wymaga dodatkowych działań inżynierskich, zwłaszcza w zakresie sterowania urządzeniem.

3. **Protokół Komend Własnościowych:** Sterowanie zaawansowanymi funkcjami, w tym **PTZ (Pan/Tilt/Zoom)**, odbywa się za pomocą zamkniętego API komunikującego się w sieci lokalnej. Osiągnięcie **pełnej kontroli** w środowisku Open Source wymagało analizy i inżynierii wstecznej tego protokołu (co zrealizowano w bibliotece PyTapo).

2.3 Wbudowane funkcje AI a Wyzwanie Open Source (Vendor Lock-in)

Kamera Tapo C200 posiada wbudowane funkcje **AI Detection**, takie jak wykrywanie osób i płaczu dziecka, realizowane za pomocą uczenia maszynowego wewnątrz urządzenia. Ograniczeniem jest tu zjawisko **Vendor Lock-in**, które blokuje dostęp użytkownika do parametrów konfiguracyjnych tych funkcji oraz do surowych danych o zdarzeniach, uniemożliwiając personalizację i rozszerzenie funkcjonalności. W konsekwencji, niezbędne staje się opracowanie własnego, otwartego i konfigurowalnego algorytmu detekcji ruchu na serwerze hostującym, co jest kluczowym elementem pracy.

3 Metodologia i implementacja rozwiązania

3.1 Metoda Badawcza

3.1.1 Double Diamond

Model ten umożliwia skuteczną transformację problemu (zamknięty ekosystem Tapo) w konkretne rozwiązanie inżynierskie.

3.1.2 Projektowanie i implementacja

3.2 Założenia

Główne założenia dotyczą pracy w sieci lokalnej (LAN), minimalizacji opóźnień oraz 100% wykorzystania narzędzi Open Source.

3.3 Zastosowane narzędzia i technologie

3.3.1 Język Programowania: Python 3.13

3.3.2 Konteneryzacja: Docker

Zapewnia hermetyzację środowiska i reprodukowalność wdrożenia na różnych platformach sprzętowych.

3.3.3 Package Manager: uv

3.3.4 Audio, Video i Streaming

- ffmpeg
- OpenCV
- MoviePy

3.3.5 Web Server

- Flask
- WebSocket's

3.3.6 Kontrola Kamery: PyTapo

3.4 Architektura rozwiązania

Przedstawienie architektury opartej na trzech warstwach: dostępu do kamery, logiki biznesowej (Flask/SocketIO) i warstwie prezentacji.

3.5 Proces implementacji

3.5.1 Serwer http

3.5.2 Implementacja połączenia z kamerą

3.5.3 Client

3.5.4 API

3.5.5 Przechwytywanie audio

3.5.6 Przechwytywanie wideo

3.5.7 Sterowanie kamerą - PTZ

3.5.8 Algorytm wykrywania ruchu

3.5.9 Nagrywanie

3.5.10 Zapis

3.6 Podsumowanie

4 Testowanie i Analiza wyników

4.1 Zakres Testów

Opis metodologii testowania funkcjonalności PTZ, streamingu, detekcji ruchu i nagrywania.

4.2 Środowisko Testowe

Specyfikacja sprzętu (np. Raspberry Pi lub PC hostujący Docker), wersji oprogramowania i konfiguracji sieciowej.

4.3 Wyniki testów i Analiza

Prezentacja kluczowych metryk, w tym:

- **Opóźnienie (Latency):** Porównanie opóźnień strumienia RTSP przetworzonego przez Flask/WebSockets z oficjalną aplikacją.
- **Skuteczność Detekcji Ruchu:** Wyniki testów algorytmu OpenCV (np. metryki True Positive Rate, False Positive Rate).
- **Wydajność Systemu:** Obciążenie procesora hosta w warunkach ciągłego streamingu i detekcji.

4.4 Podsumowanie

Wnioski Końcowe

Niniejsza praca inżynierska zrealizowała postawiony cel główny, tworząc kompletne i modułowe rozwiązanie Open Source dla kamer TP-Link Tapo C200. Projekt udowodnił, że bariera **vendor lock-in** może być skutecznie przełamana za pomocą inżynierii wstecznej protokołów własnościowych i integracji sprawdzonych narzędzi (Docker, FFmpeg, OpenCV).

Osiągnięte wyniki potwierdzają:

1. Pełną i stabilną kontrolę PTZ z poziomu autorskiego interfejsu.
2. Możliwość lokalnego streamingu wideo z minimalnymi opóźnieniami.
3. Skuteczną implementację autorskiej detekcji ruchu, oferującej większą konfigurowalność niż wbudowane funkcje kamery.

Kierunki dalszego rozwoju

Dalsze prace mogą koncentrować się na:

- Integracji z protokołami Smart Home (np. MQTT) w celu komunikacji z systemami takimi jak Home Assistant.
- Zastosowaniu WebRTC (Web Real-Time Communication) w celu dalszej redukcji opóźnień streamingu.
- Wdrożeniu lekkich modeli uczenia maszynowego (np. YOLOv5 Nano) dla zaawansowanej detekcji obiektów, wykorzystujących akcelerację sprzętową GPU hosta.

Podsumowanie pracy

Opracowany prototyp stanowi w pełni funkcjonalną i otwartą alternatywę dla zamkniętego ekosystemu producenta, oferując użytkownikowi pełną suwerenność nad gromadzonymi danymi i możliwością rozbudowy systemu.

Bibliografia

- Abdalla, P. (2020). *Testing IoT Security: The Case Study of an IP Camera*. ResearchGate PDF. URL: https://www.researchgate.net/publication/342184780_Testing_IoT_Security_The_Case_Study_of_an_IP_Camera.
- Alaba, F.A. i in. (2017). „Internet of Things security: A survey”. W: *Future Generation Computer Systems* 79, s. 273–283. DOI: 10.1016/j.future.2017.03.005. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC7506579/>.
- Minerva, Roberto, Abyi Biru i Domenico Rotondi (2021). „A survey on IoT platforms: Communication, security, and more”. W: *Computer Networks* 193, s. 108040. DOI: 10.1016/j.comnet.2021.108040. URL: <https://www.sciencedirect.com/science/article/pii/S1389128621001444>.

;

Spis rysunków

Spis tabel

1.1 Główne obszary zastosowań kamer IP (na podstawie raportu Hanwha Vision, 2025)	7
---	---