

Eryka PROBIERZ, Natalia BARTOSIAK, Martyna WOJNAR,
Kamil SKOWROŃSKI, Adam GAŁUSZKA, Tomasz GRZEJSZCZAK
Politechnika Śląska

ZASTOSOWANIE METOD TINY-ML DO ROZPOZNAWANIA TWARZY W ROBOTYCE SPOŁECZNEJ Z WYKORZYSTANIEM ROBOTÓW OHBOT

Streszczenie. Jednym z podstawowych problemów robotyki społecznej jest prawidłowe rozpoznawanie człowieka przez roboty. Do rozpoznawania twarzy wykorzystywane są rozwiązania z dziedziny Tiny Machine Learning (Tiny-ML). W artykule porównano sieć YOLOv4-tiny oraz YOLOv5s pod względem wydajności i czasu przetwarzania. Zaproponowane sieci zostały przetestowane na robotach społecznych typu OhBot rozszerzonych o moduły obliczeniowe Neural Sticks. Uzyskane wyniki wskazują na najwyższą wydajność sieci YOLOv5s przy użyciu Raspberry Pi wraz z akceleratorem.

APPLICATION OF TINY-ML METHODS FOR FACE RECOGNITION IN SOCIAL ROBOTICS USING OHBOT ROBOTS

Summary. One of the fundamental problems of social robotics is the proper recognition of humans by robots. For the purpose of face recognition, solutions from the Tiny Machine Learning (Tiny-ML) stream are used. The paper compares the YOLOv4-tiny and YOLOv5s networks in terms of efficiency and processing time. The proposed networks were tested on social robots of the OhBot type and with extended capabilities, by using Neural Sticks. The results obtained show the highest efficiency of YOLOv5s network using a Raspberry Pi along with an accelerator.

1. Wstęp

Robotyka społeczna to jedna z najszybciej rozwijających się gałęzi robotyki. Roboty towarzyszące, edukacyjne czy wspomagające terapię to roboty, które z hali przemysłowej wkraczają w codzienne życie zwykłych ludzi [9, 12]. Projektowanie, budowa i wdrażanie robotów społecznych to wyzwanie, które podejmują zarówno ośrodki badawcze, jak i wielkie korporacje technologiczne. Zmiany zachodzące w społeczeństwie przyczyniają się i wzmacniają potrzebę coraz częstszego wykorzystywania robotów w codziennych kontaktach z ludźmi [3]. Jednym z ważnych czynników jest starzenie się społeczeństwa, które prowadzi do zmiany odsetka osób starszych, których z roku na rok jest coraz więcej. Obecnie w krajach rozwijających się i wysoko rozwiniętych brakuje wyspecjalizowanego personelu, który mógłby zapewnić odpowiednią opiekę osobom starszym. Tutaj do gry wchodzi roboty rehabilitacyjne i towarzyszące [4]. Również globalne zmiany prowadzące do powszechnej migracji ludzi

przyczyniają się do zwiększonego wykorzystania robotów edukacyjnych [22]. Główną przyczyną problemów edukacyjnych dzieci jest brak adekwatnej edukacji w języku innym niż ojczysty, w którym edukacja jest obecnie prowadzona. Wykorzystanie osobistego robota edukacyjnego do przyspieszenia nauki języka prowadzi do równych szans i lepszej edukacji dzieci. Również dzieci z autyzmem i zespołem Aspergera mają szansę na realne korzyści. Według badań właściwe wykorzystanie robotów terapeutycznych przyczynia się do poprawy ich funkcjonowania w środowisku codziennym, w tym z rówieśnikami [8]. Jednak niezależnie od powodów, które powodują, że roboty społeczne są coraz szerzej stosowane, należy spełnić pewne podstawowe założenia, aby umożliwić komunikację człowiek-robot.

2. Powiązane prace

A. Roboty społeczne

Roboty społeczne to roboty, których zadaniem jest interakcja z ludźmi. Aby było to możliwe, roboty muszą postrzegać ludzi w czasie rzeczywistym, zarówno przed, w trakcie, jak i po wykonaniu zadania. Gdy robot prowadzi rozmowę z pojedynczym człowiekiem, powinien być w stanie rozpoznać twarz swojego rozmówcy i podążać za nią, nawet jeśli rozmówca zmienia lokalizację [8]. Wymagania te sprawiły, że rozpoznawanie ludzkiej twarzy, jako pośrednia cecha robotów społecznych, jest niezwykle ważne. Wśród stosowanych podejść ważne jest wyróżnienie tych, które opierają się na różnych rozwiązaniach technologicznych. Roboty społecznościowe rozpoznają osoby za pomocą czujników odległości, laserów, kamer czy mikrofonów. Różne podejścia technologiczne mają różny stopień skuteczności. W zakresie rozpoznawania obiektów wizualnych za pomocą nagrań wideo lub sekwencji zdjęć wykazuje się wysoką skutecznością. Jak wynika z badań prowadzonych przez innych badaczy, największym wyzwaniem staje się rozpoznawanie twarzy w różnych warunkach oraz ograniczenia sprzętowe i obliczeniowe robotów społecznych w stosunku do standardowych komputerów obliczeniowych [24]. Klasyczne rozwiązania o wysokiej wydajności mogą nie być możliwe do wdrożenia przy ograniczonej mocy robotów społecznościowych. Aby uniknąć tej sytuacji, rozwiązania wymagające szybkich czasów odpowiedzi i małej mocy obliczeniowej wykorzystują podejście Tiny Machine Learning, trend promujący budowanie szybkich, efektywnych, a przede wszystkim mało intensywnych obliczeniowo rozwiązań z zakresu uczenia maszynowego i sieci neuronowych. Podejście to reprezentuje nową ścieżkę rozwiązań, których połączenie obserwujemy od 2020 roku [30].

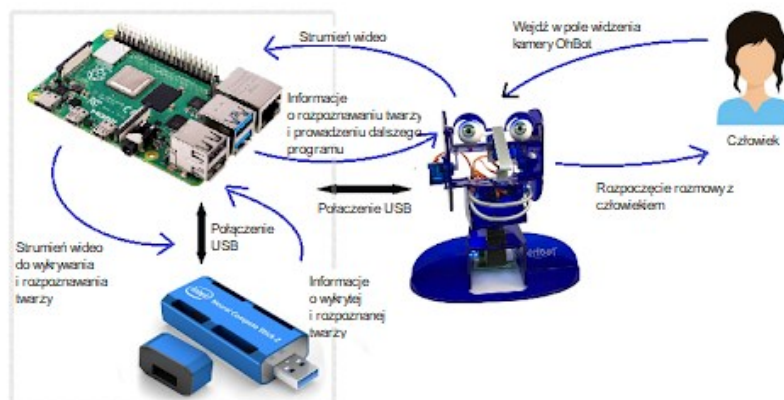
B. Tiny-ML - geneza i zastosowanie

Tiny-ML, czyli rozwiązania z zakresu uczenia maszynowego i sztucznej inteligencji to nowy trend obecny w technologii. Rozwiązania te są często określane jako Edge AI [26]. Ich głównym celem jest możliwość wykonywania obliczeń bezpośrednio na mikroprocesorach, nawet tych zasilanych bateriami. Oznacza to znaczącą zmianę w rozwoju technologii, która przechodzi od operacji przetwarzania w chmurze bezpośrednio do mikrotechnologii. Takie rozwiązanie ma kilka zalet. Jedną z najczęściej wymienianych jest poprawa bezpieczeństwa – dane nie są przesyłane, przetwarzane i przesyłane z powrotem. Daje to możliwość

konstruowania wielu rozwiązań z nowymi funkcjami, które wcześniej nie były dostępne. Udowodniono, że wykonywanie obliczeń bezpośrednio na mikroprocesorach pozwala zapewnić niskie opóźnienia, co w kontekście rozwoju robotów społecznych jest kwestią kluczową [27]. Rozwiązania Tiny-ML są już wykorzystywane w przemysłowej konserwacji predykcyjnej, umożliwiając ciągłe monitorowanie maszyn i przewidywanie usterek z wyprzedzeniem. Są również wykorzystywane w projektach opieki zdrowotnej, takich jak projekt Solar Scare Mosquito. Projekt ten wykorzystuje Tiny-ML do wykrywania warunków sprzyjających rozprzestrzenianiu się chorób przenoszonych przez komary, a dzięki stałej energii słonecznej może działać przez wiele lat. Podobne rozwiązania są również stosowane w rolnictwie, gdzie aplikacje pomagają wykrywać choroby roślin bez konieczności dostępu do internetu. Oznacza to, że modele uczenia maszynowego uruchamiane są bezpośrednio na urządzeniu, a wykrywanie chorób może odbywać się również w obszarach, w których nie ma zasięgu internetowego. Te i inne dotychczasowe aplikacje wskazują na szerokie możliwości wykorzystania Tiny-ML, w tym w robotyce społecznej [11,28]. Zastosowanie modeli uczenia maszynowego bezpośrednio do robotów to perspektywa, która oferuje nieocenione możliwości zarówno poprzez zwiększenie szybkości reakcji, jak i zmniejszenie konieczności utrzymywania ciągłego połączenia z internetem.

3. Architektura systemu

Architektura systemu składa się z robotycznej głowy typu OhBot, mikrokomputera typu Raspberry Pi i Neural Stick. Celem zaprojektowanego systemu jest wdrożenie rozpoznawania twarzy za pomocą rozwiązań Tiny-ML, co pozwoli na szybkie wykrycie osoby, z którą robot może nawiązać rozmowę lub akcję. To rozwiązanie pozwala robotowi niemal bez przerwy przeszukiwać otoczenie i uruchamiać kolejne programy aktywujące kontakt robota dopiero po wykryciu twarzy. Dzięki zastosowaniu Tiny-ML osoba, która pojawi się w zasięgu kamery umieszczonej w robocie społecznym, zostanie rozpoznana niemal natychmiast, robot ożyje i nastąpi kontakt człowiek-robot. Do projektu wykorzystano roboty społeczne OhBot, które podłączono do minikomputerów RaspberryPi 4, których moc obliczeniową wspierają Neural Sticks.



Rys. 1. Schemat architektury systemu na podstawie fotografii urządzeń producentów [16, 18, 19]

Robot przeprowadza ciągłą analizę pola kamery. W celu wykrycia osoby w polu widzenia kamery robota, dane wizualne przesyłane są do minikomputera, gdzie wstępnie dane są analizowane za pomocą sieci neuronowej (wspomaganej poprzez Neural Stick), której zadaniem jest wykrywanie i rozpoznawanie twarzy. Informacje zwrotne, które otrzymuje minikomputer, dotyczą wykrycia twarzy, w którym to przypadku minikomputer uruchamia kolejną sekwencję działań w celu komunikacji między robotem a człowiekiem i wysyła to polecenie do robota społecznego.

A. Roboty społeczne

Roboty społeczne OhBot to rozwiązanie technologiczne przypominające ludzką głowę. Składają się z serii serwomechanizmów, które pozwalają im poruszać oczami, powiekami, ustami, a nawet całą głową. Roboty są w pełni programowalne w Pythonie, co umożliwia implementację różnych rozwiązań Tiny-ML, również z wykorzystaniem biblioteki TensorFlow Lite. W pełni zintegrowane roboty z innymi dedykowanymi urządzeniami technologicznymi są w stanie odbierać bodźce wzrokowe, słuchowe i dotykowe, wykrywać przechylenie czy zmiany światła. Dodatkowo przy wykorzystaniu innych urządzeń możliwa jest komunikacja nie tylko przez internet, ale także przez sieć telefoniczną oraz analizowanie danych w oparciu o system LIDAR. Roboty są w pełni programowalne, co oznacza, że zarówno ich ruch, kolejność ich programów czy nakładanie się rozwiązań muszą być zaprogramowane i zaimplementowane w samym robocie. Umożliwia to ciągły rozwój robota społecznego i umożliwia analizę danych wizualnych, dźwiękowych i innych oraz stworzenie robota, który jest w stanie adekwatnie reagować na wyniki tych analiz [20].

B. Wykrywanie twarzy za pomocą sieci neuronowych

Wykrywanie twarzy należy do szerszej koncepcji wykrywania obiektów na obrazach. Najczęściej stosowanymi rozwiązaniami są te oparte na sieciach neuronowych. Dominującym wyborem są konwolucyjne sieci neuronowe (CNN), ze względu na ich predyspozycje do wyszukiwania wzorców z obrazów [14]. Struktura tych sieci pozwala na klasyfikację i etykietowanie wykrytych obiektów, co determinuje ich dalszy potencjał. Niektóre z najpopularniejszych sieci do wykrywania twarzy to R-CNN, Fast R-CNN, Faster R-CNN i YOLO [2]. Według przeglądu literatury sieci neuronowych YOLO są obecnie najpopularniejszym rozwiązaniem do wykrywania twarzy. Umożliwiają wykrywanie twarzy w czasie rzeczywistym i osiągają lepszy czas niż inne sieci oferujące tę funkcję [25]. Podstawowym rozwiązaniem istniejącym w YOLO jest generowanie regionów, na których tworzone są ramki ograniczające obraz. To właśnie wewnątrz tych ramek uruchamiany jest klasyfikator. Aby usprawnić wykrywanie ramek, w przetwarzaniu końcowym stosuje się metody ulepszające, dzięki czemu eliminowane są duplikaty [7]. Dana sieć analizuje wiele klatek i wiele podobieństw do wybranej klasy obiektów. Należy zauważyć, że istnieje wiele rozwiązań z rodziny YOLO, w tej pracy zastosowano rozwiązanie tinyYOLO, które umożliwia implementację w rozwiązaniach o ograniczonej mocy obliczeniowej. To rozwiązanie było wielokrotnie używane i testowane na różnych zbiorach danych i wykazuje zadowalające działanie w wykrywaniu twarzy [1]. Jako alternatywę dla tinyYOLO postanowiono zaproponować

własną sieć, utrzymywaną w dziedzinie Tiny-ML, która mogłaby łączyć wszystkie zalety sieci tinyYOLO zastosowanej w robotach społecznych.

C. Neural Stick

W celu optymalizacji i przyspieszenia robotów społecznych, rozwiązanie typu Intel Neural Compute Stick 2 (Intel NCS2) jest zintegrowane z minikomputerem w celu zwiększenia wydajności sieci neuronowej [10]. Rozwiązanie to umożliwia analizę i przetwarzanie w oparciu o rozwiązania uczenia maszynowego i sieci neuronowe bez konieczności przetwarzania w chmurze i jest kompatybilne z systemami operacyjnymi wymaganymi do programowania robotów społecznych. Ponadto rozwiązanie dystrybuuje zestaw narzędzi OpenVINO, które umożliwia implementację splotowych sieci neuronowych przy jednoczesnym zapewnieniu niskiego zużycia energii i pracy w czasie rzeczywistym. W rozwiązaniu dostępna jest obszerna biblioteka funkcji i wstępnie zoptymalizowanych jąder, a co najważniejsze, te zestawy narzędzi są w pełni kompatybilne z minikomputerami Raspberry Pi [6].

4. Projekt sieci neuronowej typu tiny

Do analiz zdecydowano się wykorzystać dwa rozwiązania oparte na sieciach neuronowych. Pierwsze rozwiązanie to proponowana sieć typu Tiny-ML, YOLOv4-tiny, druga sieć neuronowa to YOLOv5s. Obie sieci zostały przeszkolone na tych samych zestawach danych, a następnie przetestowane z wykorzystaniem rozwiązania składającego się z robota społecznego typu OhBot, minikomputera Raspberry Pi oraz akceleratora obliczeniowego Intel Neural Stick.

A. Architektura sieci neuronowych

Postanowiono wykorzystać i porównać ze sobą dwie architektury sieci YOLO. Pierwsza to YOLOv4 w wersji tiny, druga to YOLOv5 w małej wersji. Wersja YOLOv4 różni się od poprzednich drobnymi zmianami [13, 21]. Podstawowa architektura składa się z następujących elementów - Backbone typu CSPDarknet53, Neck opartego o Spatial pyramid pooling i Path Aggregation Network oraz Head opartej na podsieci Class i podsieci Box (prawie niezmienną od YOLOv3). Leaky RELs została użyta jako funkcja aktywacji dla tego modelu. YOLOv5 został zaproponowany przez innych autorów, w tym Glenna Jochera. YOLOv5 zawiera 5 modeli: s, m, l, x i n, które odpowiadają wielkości sieci i rosnącej dokładności. W tych badaniach wybrano wersję YOLOv5s, aby oferować najwyższą prędkość z proponowanych modeli [15]. Architektura YOLOv5 jest oparta na elementach Backbone typu CSPDarknet53, Neck na bazie Path Aggregation Network i Head na warstwie Yolo generującej trzy rozmiary map cech, aby osiągnąć wieloskalowość [29]. Detekcja wieloskalowa jest szczególnie ważna w kontekście wykrywania obiektów o zmieniającej się wielkości w wyniku zmiany skali. W przypadku robotów społecznych, zbliżający się człowiek zostanie najpierw wykryty w najmniejszym rozmiarze (18x18), a gdy człowiek zbliży się do robota, ten rozmiar wzrośnie do dużego rozmiaru (72x72).

B. Przygotowanie zbioru danych

Zbiory danych zostały przygotowane za pomocą narzędzia roboflow.

Przygotowane zestawy danych zostały następnie zaimportowane do sieci YOLOv4-tiny i YOLOv5s. Szkolenie sieci zostało zrealizowane z wykorzystaniem usług google colab. Szkolenie przeprowadzono na dwóch zbiorach danych.

Pierwszym zbiorem danych był MS COCO [23]. Jest to zbiór danych o dużej skali, który umożliwia wykrywanie i segmentację, a także oznaczanie i klasyfikowanie obiektów widocznych na obrazach. COCO to skrót od Common Objects in Context, a jego głównym celem jest poprawa rozpoznawania obrazów. Ze względu na szerokie zastosowanie tego zbioru danych dla wielu nowoczesnych sieci, zdecydowano się wykorzystać go również w niniejszym badaniu, tak aby wyniki uzyskane z treningu można było porównać z innymi wynikami obu tych samych trenowanych sieci przez innych badaczy i inne rozwiązania. Głównymi cechami kolekcji COCO jest ponad 200 000 zdjęć oznaczonych etykietami, przy czym łączna liczba wszystkich zdjęć to ponad 300 000. Kolekcja oferuje 80 kategorii obiektów, czyli instancje, które można łatwo oznaczyć (człowiek, samochód) oraz 91 kategorii rzeczy, które mogą nie mieć wyraźnych granic (np. niebo, trawa). W całej kolekcji znajduje się około 250 000 zdjęć osób z zaznaczonymi kluczowymi punktami niezbędnymi do wykrycia osoby.



Rys. 2. 17 kluczowych punktów do oznaczania osób w MS COCO [23]

Istotną z punktu widzenia proponowanego rozwiązania i zastosowania w robotyce społecznej jest obecność 17 punktów kluczowych pozwalających na oznaczenie punktów kluczowych wartościami (x,y,v) . Wartości x i y oznaczają współrzędne danego punktu, a v jego widoczność (widoczny, niewidoczny). 17 punktów obejmuje: nos, lewe oko, prawe oko, lewe ucho, prawe ucho, lewe ramię, prawe ramię, lewy łokieć, prawy łokieć, lewy nadgarstek, prawy nadgarstek, lewe biodro, prawe biodro, lewe kolano, prawe kolano, lewa kostka, prawa kostka. Pełny obraz pokazano na rysunku 2.

Drugim zbiorem danych był zestaw, który umożliwił wytrenowanie sieci w celu wykrycia, czy dana osoba nosiła maskę, czy nie nosiła maski, czy też nosiła ją niewłaściwie [17]. Kolekcja składała się z 853 obrazów. Zarówno jedna, jak i druga sieć zostały przeszkolone z podziałem danych wynoszącym 70:30, a łączne wyniki szkolenia podano w tabeli I. Aby przeanalizować wyniki, uwzględniono takie miary, jak precision,

recall and F1-measure (równanie 1).

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{tp}{tp + \frac{1}{2}(fp + fn)} \quad (1)$$

Tabela 1

Średnie i 95% przedziały ufności dla Precision, Recall i F-measure w 10 rundach szkolenia i oceny

Network + Dataset	Precision	Recall	F-measure
YOLOv4-tiny, MS COCO	0.44 (0.51-0.37)	0.37 (0.31-0.43)	0.42 (0.45-0.39)
YOLOv4-tiny. Face Mask	0.89 (0.81-0.96)	0.77 (0.74-0.80)	0.83 (0.77-0.89)
YOLOv5s, MS COCO	0.49 (0.54-0.44)	0.41 (0.44-0.37)	0.45 (0.49-0.41)
YOLOv5s, Face Mask	0.93^a (0.88-0.98)	0.81^a (0.76-0.86)	0.86^a (0.81-0.91)

^a. Best performing model and dataset for each metric.

Wyszkolone sieci zostały następnie opracowane jako pliki wykonywalne i przesłane do urządzenia docelowego, minikomputera Raspberry Pi. Wyniki uzyskane w tabeli 1 wskazują na wyższą wydajność analizowanych metryk dla sieci YOLOv5s. Ta sieć uzyskała lepszą wydajność zarówno w jednym, jak i drugim zestawie danych.

5. Wdrożenie w robotach OhBot

Wdrożenie zostało przeprowadzone z obydwoma sieciami z wykorzystaniem rozwiązania OpenVINO. Postanowiono wykorzystać dwa zaproponowane rozwiązania w celu przetestowania szybkości sieci na mikrokomputerze Raspberry Pi 4 model B WiFi DualBand Bluetooth 8GB RAM 1,5GHz. Testy przeprowadzono z akceleratorem Intel® Neural Compute Stick 2 (Intel® NCS2) i bez niego. Na podstawie wyników przeprowadzonych testów zdecydowano o ostatecznej implementacji przetestowanej eksperymentalnie na konkretnych scenariuszach dla głów robotycznych typu OhBot.

Tabela 2

Średnia liczba klatek na sekundę dla testowanych modeli

Network + Dataset	Raspberry Pi	Raspberry Pi + Intel Neural Stick 2
YOLOv4-tiny, MS COCO	4.3 FPS	13.1 FPS
YOLOv4-tiny. Face Mask	7.1 FPS	24 FPS
YOLOv5s, MS COCO	5.7 FPS	17.1 FPS
YOLOv5s, Face Mask	8.3 FPS	31.2 FPS

Najważniejszym parametrem dla zaimplementowanych modeli była wartość klatek na sekundę (FPS), której pełną listę można znaleźć w tabeli 2. Otrzymane wyniki wskazują na wyższość sieci YOLOv5s na obu zestawach danych. Najlepsze wyniki uzyskano z Intel Neural Stick 2, czyli przy użyciu akceleratora. Oznacza to, że do końcowych testów zostaną użyte YOLOv5s, a Intel Neural Stick 2 zostanie podłączony do minikomputera Raspberry Pi.

6. Wyniki eksperymentów

W celu wdrożenia prezentowanego rozwiązania postanowiono przeprowadzić szereg testów pozwalających na wykorzystanie sieci YOLOv5s pozwalającej na detekcję określonych kategorii zawartych w MS COCO oraz w zestawie pozwalającym na detekcję masek. Ze względu na priorytetowe traktowanie wykrywania osób przez roboty społeczne, w oparciu o inne możliwości sieci zaprojektowano i rozszerzono serię testów skupionych wokół tego tematu.

A. Wykrywanie osoby na obrazie w celu rozpoczęcia rozmowy

Celem tego zadania było zaimplementowanie funkcji automatycznego rozpoczynania rozmowy po wykryciu osoby na poziomie 0,7. Należy zauważyć, że osoby są wykrywane przez sieć od poziomu 0,5 wzwyż, ale wartości mniejsze niż 0,7 wskazują na większą odległość między osobą a robotem. Po wykryciu osoby uruchamiany jest skrypt, który składa się z następujących kroków:

1. Wykrycie osoby na poziomie 0,7.
2. Odpowiedź głowy robota: Dzień dobry, chciałbyś porozmawiać z OhBotem?
3. Oczekiwanie na odpowiedź.
4. Analiza poziomu wykrywalności osoby, zwiększanie wartości oznacza zbliżanie się osoby, zmniejszanie oddalania się.
5. W przypadku pozytywnej odpowiedzi przełączenie na automatyczny skrypt czatu.
6. Gdy nie ma odpowiedzi lub jest niejasna, robotyczna głowa odpowiada: Podejdź bliżej, abym mógł Cię lepiej zrozumieć.
7. W momencie, gdy robot przestanie wykrywać osobę, skrypt jest automatycznie anulowany.

B. Wykrywanie maski na twarzy

Celem tego zadania było zaimplementowanie funkcji automatycznego wykrywania, czy ludzie mają założoną maskę, czy nie, lub czy mają ją założoną nieprawidłowo. Tylko wartości powyżej 0,7 są analizowane z tych samych powodów, co przedstawiono w scenariuszu 1. Skrypt składa się z następujących kroków:

I schemat:

1. Wykrycie osoby noszącej maskę na poziomie 0,7.
2. Reakcja głowy robota: Dzień dobry, dziękuję za noszenie maski!
3. Odpowiedź głowy robota: Czy chciałbyś porozmawiać z OhBotem?
4. Jeśli otrzymasz pozytywną odpowiedź, przełącza się na automatyczny skrypt czatu.
5. W przypadku braku odpowiedzi lub odpowiedzi negatywnej, skrypt zostaje zakończony.

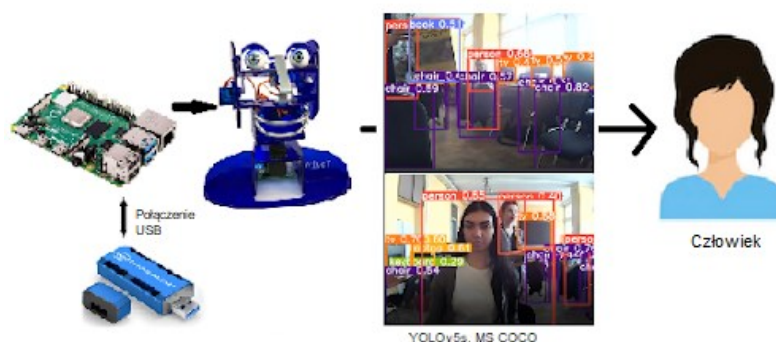
II schemat:

- 1) Wykrycie osoby noszącej nieprawidłowo założoną maskę na poziomie 0,7.
- 2) Odpowiedź głowy robota: Witam, mam wrażenie, że twoja maska nie jest założona prawidłowo. Czy mógłbyś to poprawić?
- 3) Analiza obrazu. Jeśli osoba nosząca maskę zostanie wykryta na poziomie 0,7, uruchom pierwszy schemat.
- 4) Jeśli OhBot ponownie wykryje nieprawidłowo założoną maskę na poziomie 0,7, powtórz żądanie (punkt 2).
- 5) W przypadku wykrycia braku maski na poziomie 0,7 przejdź do schematu trzeciego.

III schemat:

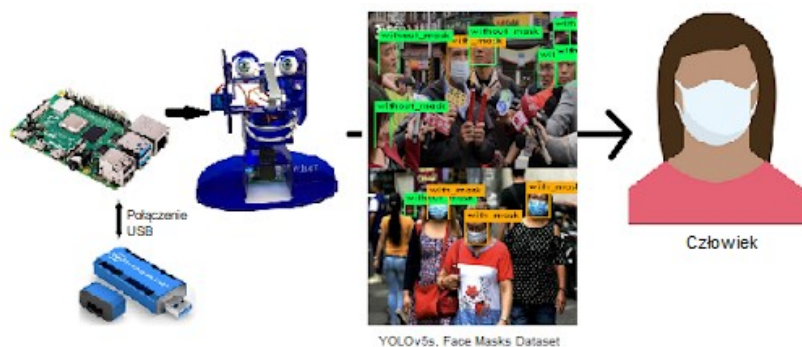
- 1) Wykrycie osoby bez maski na poziomie 0,7.
- 2) Reakcja głowy robota: Dzień dobry, myślę, że nie masz założonej maski, czy mógłbyś ją założyć?
- 3) Analiza obrazu. Jeśli osoba nosząca maskę zostanie wykryta na poziomie 0,7, uruchom pierwszy schemat.
- 4) Jeśli osoba zostanie ponownie wykryta bez maski na poziomie 0,7, powtórz prośbę (pkt 2).
- 5) W przypadku wykrycia niepoprawnie założonej maski na poziomie 0,7 uruchamiający jest drugi schemat.

W przypadku pierwszego scenariusza eksperymentu przeprowadzono 40 prób. 36 razy robot poprawnie wykrył osobę i nawiązał z nią rozmowę. 11 osób nie podjęło dalszej rozmowy, natomiast pozostałe 25 osób prowadziło dalszą rozmowę z czatbotem.



Rys. 3. Wykrywanie osoby na obrazie w celu rozpoczęcia rozmowy – schemat architektury na podstawie fotografii urządzeń producentów [16, 18, 19]

W przypadku drugiego scenariusza eksperymentu przeprowadzono 120 prób. 58 razy robot poprawnie wykrył maskę, z czego 33 osoby (57%) zdecydowały się kontynuować rozmowę z OhBotem. 7 razy robot nie wykrył prawidłowo noszonej maski. 27 razy robot wykrył nieprawidłowo założoną maskę, z czego 17 osób (63%) poprawiło maskę. 10 osób nie podjęło rozmowy z robotem ani nie poprawiło maski. 6 razy robot nie wykrył nieprawidłowo założonej maski. 13 razy robot poprawnie wykrył osoby bez masek, z czego 4 osoby (31%) założyły maskę i kontynuowały rozmowę z robotem, a pozostałe 9 odeszło. 6 razy robot błędnie wykrył brak maski.



Rys. 4. Wykrywanie maski na twarz - schemat architektury na podstawie fotografii urzędzeń producentów [16, 18, 19]

7. Wnioski

Celem przeprowadzonych badań była analiza i wdrożenie dostępnych sieci do wykrywania osób i twarzy w robotach społecznych OhBot. Przetestowano dwie sieci i dwie architektury rozwiązań. Siecią wykazującą najwyższą wydajność i jakość działania okazały się sieci YOLOv5s, które najlepsze wyniki uzyskano na architekturze łączącej minikomputer i akcelerator obliczeniowy. Takie rozwiązanie zostało zintegrowane z robotem społecznym, a następnie przeprowadzono szereg testów sprawdzających działanie danego rozwiązania w praktyce. Uzyskane wyniki wskazują, że w większości przypadków zaimplementowane schematy sprawdziły się i mogły być wykorzystane w czynnościach związanych z komunikatywnością i interaktywnością robotów społecznych. Proponowane rozwiązania to tylko jedna z wielu funkcjonalności zaimplementowanych w robotach społecznościowych, a szczególnie istotne ze względu na możliwość nawiązania i otwarcia rozmowy z człowiekiem. W dalszych pracach planowane jest wdrożenie schematów robotów społecznościowych dla innych klas i rozwiązań pozwalających na rozmowę z więcej niż jedną osobą.

PODZIĘKOWANIA

Praca PE była wspierana częściowo przez Unię Europejską poprzez Europejski Fundusz Społeczny jako stypendium w ramach Grantu POWR.03.02.00-00-I029, a częściowo przez Politechnikę Śląską poprzez dotację na utrzymanie i rozwój potencjału badawczego w 2022 roku dla młodych naukowców numer 02/060/BKM22/0036. Praca NB i MW była wspierana przez Program Mentorski realizowany przez Politechnikę Śląską (Program Mentorski – „Rozwiń Skrzydła”) i opłacany z rezerwy Prorektora ds. Studenckich i Kształcenia: MPK: 60/ 001 GŻF: SUBD. Praca KS została wsparta przez Politechnikę Śląską poprzez dotację na utrzymanie i rozwój potencjału badawczego w 2022 roku dla młodych naukowców numer 02/060/BKM22/0040. Praca AG była wspierana przez Politechnikę Śląską poprzez dotację na utrzymanie i rozwój potencjału badawczego w 2022 roku. Praca TG została częściowo wsparta przez tą samą dotację oraz poprzez dotację na utrzymanie i rozwój potencjału badawczego w 2022 roku dla młodych naukowców numer 02/060/BKM22/0041.

LITERATURA

1. Abbasi S, Abdi H, Ahmadi A, "A Face-Mask Detection Approach based on YOLO Applied for a New Collected Dataset," In 2021 26th International Computer Conference, Computer Society of Iran (CSICC) IEEE, 2021 Mar 3, pp. 1-6.
2. Abdullah SM, Abdulazeez AM, "Facial expression recognition based on deep learning convolution neural network: A review," Journal of Soft Computing and Data Mining, 2021 Apr 15, 2(1), pp. 53-65.
3. Belpaeme T, Kennedy J, Ramachandran A, Scassellati B, Tanaka F, "Social robots for education: A review," Science robotics, 2018 Aug 15; 3(21):eaat5954.
4. Broekens J, Heerink M, Rosendal H, "Assistive social robots in elderly care: a review," Gerontechnology, 2009 Apr; 8(2), pp.94-103.
5. Cabibihan JJ, Javed H, Ang M, Aljunied SM, "Why robots? A survey on the roles and benefits of social robots in the therapy of children with autism," International journal of social robotics, 2013 Nov, 5(4), pp. 593-618.
6. Cao Q, Irimiea AE, Abdelfattah M, Balasubramanian A, Lane ND, "Are Mobile DNN Accelerators Accelerating DNNs?" In Proceedings of the 5th International Workshop on Embedded and Mobile Deep Learning, 2021 Jun 25, pp. 7-12.
7. Chen W, Huang H, Peng S, Zhou C, Zhang C, "YOLO-face: a real-time face detector," The Visual Computer, 2021 Apr, 37(4), pp. 805-13.
8. Cifuentes CA, Pinto MJ, Céspedes N, Múnera M, "Social robots in therapy and care," Current Robotics Reports, 2020 Sep, 1(3), pp. 59-74.
9. Dautenhahn K, Werry I, Rae J, Dickerson P, Stribling P, Ogden B, "Robotic playmates," In Socially intelligent agents, Springer, Boston, MA, 2002, pp.117-124.
10. Demidovskij A, Tugaryov A, Fatekhov M, Aidova E, Stepyreva E, Shevtsov M, Gorbachev Y, "Accelerating Object Detection Models Inference within Deep Learning Workbench," In 2021 International Conference on Engineering and Emerging Technologies (ICEET) IEEE, 2021 Oct 27, pp. 1-6.
11. Dutta L, Bharali S, "TinyML Meets IoT: A Comprehensive Survey," Internet of Things, 2021 Dec 1, 16, pp.100461.
12. Erel H, Shem Tov T, Kessler Y, Zuckerman O, "Robots are always social: robotic movements are automatically interpreted as social cues," In Extended abstracts of the 2019 CHI conference on human factors in computing systems 2019 May 2, pp. 1-6.
13. Guo C, Lv XL, Zhang Y, Zhang ML, "Improved YOLOv4-tiny network for real-time electronic component detection," Scientific reports, 2021 Nov 23, 11(1), pp.1-3.
14. Hasan MK, Ahsan M, Newaz SH, Lee GM, "Human face detection techniques: A comprehensive review and future research directions," Electronics, 2021 Jan, 10(19), pp. 2354.
15. https://pytorch.org/hub/ultralytics_yolov5/ (access: 14.05.2022)
16. <https://www.intel.com/content/www/us/en/developer/tools/neural-compute-stick/overview.html> (access: 14.05.2022)

17. <https://www.kaggle.com/andrewmvd/face-mask-detection> (access: 14.05.2022)
18. <https://www.ohbot.co.uk/> (access: 14.05.2022)
19. <https://www.raspberrypi.com/> (access: 14.05.2022)
20. Janiaczyk, W. A., Probierz, E., & Gałuszka, A, "On the recognition and analysis of selected emotional states in the artificial intelligence of social robots," In A. Nketsai, C. Baron, & C. Foucher, A. Nketsai, C. Baron, & C. Foucher (Red.), *Modelling and simulation 2020: The European Simulation and Modelling Conference 2020*. ESM '2020, October 21-23, 2020, Toulouse, France, pp. 223–228.
21. Jiang Z, Zhao L, Li S, Jia Y, "Real-time object detection method based on improved YOLOv4-tiny," arXiv preprint arXiv:2011.04244. 2020 Nov 9.
22. Kirstein F, Risager RV, "Social robots in educational institutions they came to stay: Introducing, evaluating, and securing social robots in daily education," In 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), IEEE, 2016 Mar 7, pp. 453-454.
23. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL, "Microsoft coco: Common objects in context," In *European conference on computer vision 2014* Sep 6, pp. 740-755.
24. Martin-Rico F, Gomez-Donoso F, Escalona F, Garcia-Rodriguez J, Cazorla M., "Semantic visual recognition in a cognitive architecture for social robots," *Integrated Computer-Aided Engineering*, 2020 Jan 1, 27(3), pp. 301-16.
25. Mbunge E, Simelane S, Fashoto SG, Akinnuwesi B, Metfula AS, "Application of deep learning and machine learning models to detect COVID-19 face masks-A review," *Sustainable Operations and Computers*, 2021 Jan 1, 2, pp. 235-45.
26. Ray PP, "A review on TinyML: State-of-the-art and prospects," *Journal of King Saud University-Computer and Information Sciences*, 2021 Nov 30.
27. Reddi VJ, Plancher B, Kennedy S, Moroney L, Warden P, Agarwal A, Banbury C, Banzi M, Bennett M, Brown B, Chitlangia S, "Widening Access to Applied Machine Learning with TinyM" arXiv preprint arXiv:2106.04008, 2021 Jun 7.
28. Shafique M, Theocharides T, Reddy VJ, Murmann B, "TinyML: Current Progress, Research Challenges, and Future Roadmap," In 2021 58th ACM/IEEE Design Automation Conference (DAC) IEEE, 2021 Dec 5, pp. 1303-1306.
29. Tan S, Lu G, Jiang Z, Huang L, "Improved YOLOv5 network model and application in safety helmet detection" In 2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR) IEEE, 2021 Mar 4, pp. 330-333.
30. Warden P, Situnayake D, "TinyM," O'Reilly Media, Incorporated; 2019.