Informatyka techniczna i sztuczna inteligencja
Technical informatics and Artificial Intelligence
Engineer of XXI Century'2024

Face authenticity detection system

Rafał Klinowski[1*], Mirosław Kordos [2]

[1] *University of Bielsko-Biala, Poland, d28@student.ubb.edu.pl*
[2] *University of Bielsko-Biala, Poland, mkordos@ubb.edu.pl*
**corresponding author*

Abstract: The paper presents a passive face authenticity detection system implemented as a stacking committee of models. Each of the four models included in the committee recognizes whether the camera view shows the real face of a live person or whether the face is shown in a photo or video. These models include: a convolutional network and Bezel algorithm for detecting portable devices on which the face can be shown, face context analysis, and a convolutional network for image analysis. The outputs from these models are fed to the inputs of a neural network that makes the final decision.

Keywords: face spoofing detection; convolutional neural network; stacking ensemble

System detekcji autentyczności twarzy

Rafał Klinowski[1*], Mirosław Kordos [2]

[1] *Uniwersytet Bielsko-Bialski, Polska, d28@student.ubb.edu.pl*
[2] *Uniwersytet Bielsko-Bialski, Polska, mkordos@ubb.edu.pl*
**corresponding author*

Streszczenie: Praca przedstawia pasywny system detekcji autentyczności twarzy zaimplementowany w formie komitetu modeli typu stacking. Każdy z czterech modeli wchodzących w skład komitetu rozpoznaje, czy w widoku kamery znajduję się autentyczna twarz żywej osoby, czy też twarz ta została podstawiona do kamery na zdjęciu lub filmie. Te modele to: sieć konwolucyjna oraz algorytm Bezela dokonujące detekcji urządzeń, na których mogła być pokazana twarz, analiza kontekstu twarzy oraz sieć konwolucyjna dokonująca analizy obrazu. Wyjścia z tych modeli podawane są na wejścia sieci neuronowej, która dokonuje końcowej decyzji.

Słowa kluczowe: rozpoznawanie autentyczności twarzy; konwolucyjna sieć neuronowa; komitet modeli

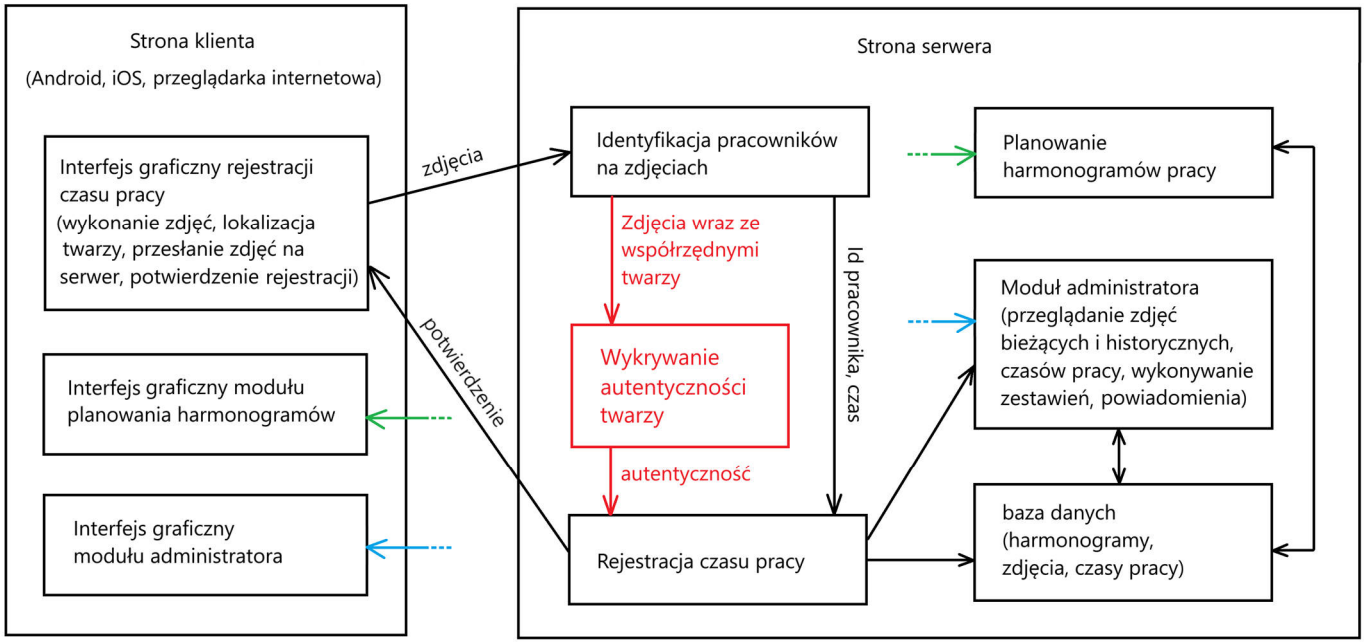
1. Wstęp

Autentyczna twarz w kontekście niniejszego artykułu rozumiana jest jako twarz żywej osoby znajdującej się przed kamerą. Natomiast twarz nieautentyczna to twarz wyświetlona na ekranie urządzenia elektronicznego, znajdująca się na materiale wideo lub wydrukowana na kartce papieru.

Niniejszy artykuł przedstawia proponowaną metodę rozpoznawania autentyczności twarzy, przeznaczaną do zastosowania w systemie rejestracji i zarządzania czasem pracy. Celem tej metody jest wykrywanie sytuacji, w których jeden pracownik podstawia zdjęcie lub wideo innego pracownika do kamery, celem zarejestrowania czasu pracy osoby, która w rzeczywistości do pracy nie przyszła. Zanim zostanie omówione działanie proponowanej metody, przedstawiony zostanie po krótce system zarządzania czasem pracy, narzucone z góry autorom prezentowanej metody umiejscowienie metody w całym systemie oraz otrzymane założenia, które przedstawiona metoda musi spełniać.

DOI: https://doi.org/10.53052/9788367652292.23

Omawiany system zarządzania czasem pracy działa aktualnie w wielu firmach i zgodnie z otrzymanymi założeniami należy dodać nowy moduł realizujący rozpoznawanie autentyczności twarzy i połączyć go z całością systemu w sposób wskazany na Rys. 1. Moduł rozpoznawania autentyczności twarzy ma otrzymywać z modułu identyfikacji pracowników zdjęcia rozpoznanych pracowników wraz ze współrzędnymi twarzy na tych zdjęciach. Następnie powinien wyznaczać prawdopodobieństwo autentyczności twarzy i przekazywać wynik do modułu rejestracji czasu pracy. Prawdopodobieństwo to zostanie zapisane w bazie danych wraz ze zdjęciem, dla którego zostało wyznaczone i następnie informacja ta zostanie wykorzystana do wyróżnienia bardziej podejrzanych zdjęć w programie administratora, aby mógł je przeglądnąć w pierwszej kolejności. Nie dla każdego zdjęcia człowiek jest w stanie stwierdzić, czy przedstawiona na nim twarz jest autentyczna. Jednakże w takich sytuacjach administrator może sprawdzić, czy dana osoba rzeczywiście znajduje się w pracy.



Rysunek 1. Schemat systemu zarządzania czasem pracy. Na czarno zaznaczono bloki istniejącego systemu, na czerwono miejsce działania proponowanej metody wykrywania autentyczności twarzy

Cały system składa się z części realizowanej po stronie klienta i po stronie serwera. Aplikacja po stronie klienta jest zaimplementowana w systemach Android, iOS oraz w przeglądarce internetowej i wykorzystuje najczęściej proste przednie kamery smartfonów, rzadziej kamery USB, prawie w całości bez podczerwieni, do wykonania zdjęć osób rejestrujących czas pracy. Zgodnie z otrzymanymi założeniami autorzy artykułu nie mają możliwości ingerowania w inne moduły systemu, niż przez nich rozwijany, w tym nie mają możliwości zainstalowania kamer wyposażonych w podczerwień, a więc i możliwości zastosowania technologii TrueDepth Camera firmy Apple [1], do rozpoznawania twarzy. Założenia te wynikają z braku zgody zarówno klientów używających sytemu jak i jego twórców na zmiany w dobrze działającym systemie i na ponoszenie związanych z tym kosztów.

Rozwiązanie TrueDepth Camera [1] rejestruje dane twarzy, wyświetlając i analizując tysiące niewidocznych punktów, aby utworzyć mapę głębi twarzy, a także rejestruje obraz twarzy w podczerwieni, dzięki temu działa nawet w ciemności. Następnie tak uzyskana mapa głębi twarzy jest porównywana z mapą uzyskaną podczas pierwszej rejestracji twarzy w systemie. Jednakże autorzy artykułu nie mają wpływu na ewentualne zastosowanie tej metody i aktualnie muszą się dostosować do istniejącego sposobu wykonywania zdjęć.

Aplikacja po stronie klienta ta jest każdorazowo aktywowana kodem QR pracownika zaczynającego, lub kończącego pracę, aby nie wykonywała niepotrzebnych zdjęć osobom, które znajdują się w widoku kamery, ale nie zaczynają, ani nie kończą pracy, lub pustemu pomieszczeniu. Następnie kamera wykonuje zdjęcia osób rejestrujących czas pracy i przy pomocy algorytmu BlazeFace[2] lokalizuje na nich twarz. Tylko zdjęcia ze zlokalizowaną twarzą są dalej

kompresowane do formatu webp do rozmiaru 5-6 kB celem zapewnienia szybkiej transmisji i przesyłane na serwer, gdzie są dalej przetwarzane i muszą być gromadzone w bazie danych na potrzeby ewentualnej późniejszej analizy.

Na serwerze działa moduł identyfikacji pracowników, który otrzymuje zdjęcia z aplikacji klienta i identyfikuje osoby na nich przedstawione, sprawdzając zgodność osoby z kodem QR. Moduł ten bazuje na algorytmie ArcFace[3]. Działa bardzo dobrze z niemal 100% dokładnością.

Zgodnie z otrzymanymi założeniami, autorzy niniejszego artykułu nie mogą modyfikować istniejących modułów, dlatego m. in. nie jest możliwe zastosowanie algorytmu RetinaFace[4] do wykrywania lokalizacji twarzy, ponadto wobec bardzo wysokiej dokładności istniejącego rozwiązania nie wydaje się to konieczne.

Zgodnie z otrzymaną specyfikacją metoda proponowana w tym artykule ma otrzymywać zdjęcia wraz ze współrzędnymi twarzy z działającego już po stronie serwera bloku identyfikacji pracowników na podstawie twarzy. Jakość otrzymywanych zdjęć nie jest najlepsza (webp w rozmiarze 5-6kB, często nieoptymalne warunki oświetleniowe, osoby rejestrujące czas pracy w pośpiechu, nie ustawione dokładne na wprost kamery), ale z taką jakością musi radzić sobie przedstawiona tu metoda.

Typowo problem rozpoznawania autentyczności twarzy jest w literaturze analizowany w kontekście logowania się użytkownika do systemu operacyjnego lub jakiejś aplikacji. Metody rozpoznawania autentyczności twarzy w takim kontekście były już przedmiotem wielu badań naukowych i zostały opisane w licznych artykułach. W niniejszym artykule prezentowany jest proponowany sposób rozpoznawania autentyczności twarzy w systemie rejestracji czasu pracy, gdzie są całkowicie odmienne warunki i wymagania (które już zostały opisane powyżej), a przetestowane metody rozpoznawania autentyczności w systemach logowania się zupełnie nie spawdziły, dając dokładności rzędu 50%. Większość autentycznych twarzy była rozpoznawana jako nieautentyczne. Dotyczyło to także pewnego znanego rozwiązania komercyjnego w cenie ok. 31.000 zł. Możliwe było w nim przetestowanie tylko kilkunastu zdjęć na koncie testowym bez zakupu oprogramowania. Wykorzystanie dwóch kont testowych wystarczyło w pełni by przeanalizować działanie tego rozwiązania na zdjęciach z systemu rejestracji czasu pracy. Rozwiązanie ta działało niemal idealnie w idealnych warunkach, gdy w dobrym oświetleniu starannie została podstawiona twarz lub jej zdjęcie pod kamerę USB na komputerze. Jednakże fakt, że większość twarzy z systemu rejestracji czasu pracy była błędnie kwalifikowana jako nieautentyczne naprowadził nas na trop, że ważnym czynnikiem, które taki system bierze pod uwagę jest jakość zdjęcia. Wykonano więc starannie kilka zdjęć twarzy dobrym pełnoklatkowym aparatem fotograficznym i podstawiono je pod kamerę na ekranie notebooka z rozdzielczością 4K. Zdjęcia te zostały w większości rozpoznane jako autentyczne. Wyniki analizy tego systemu dla zdjęć z rejestracji czasu pracy przedstawiono w Tabeli 4. Ponieważ nie udało się znaleźć metody dobrze działającej w opisanych warunkach, dlatego koniecznością było opracowanie własnych metod, z których jedną przedstawiono w niniejszym artykule.

2. Przegląd literaturowy metod rozpoznawania autentyczności twarzy.

Jak już wspomniano we wstępie, autorom nie udało się znaleźć żadnego rozwiązania sprawdzającego się w warunkach analizowanego systemu rejestracji czasu pracy. Zatem w tej sekcji opisane zostaną metody rozpoznawania autentyczności twarzy przeznaczone do logowania się do systemów, gdzie użytkownik ma czas i chęć współpracować z takim systemem, lub opracowane celem dawania dobrych wyników na testowych bazach danych.

Metody rozpoznawania autentyczności twarzy można podzielić pasywne, które nie wymagają interakcji użytkownika i aktywne, które takiej interakcji wymagają. W metodach aktywnych użytkownik jest proszony, aby przykładowo zamrugał oczami lub się uśmiechnął, lub pokręcił głową. Jednakże w niektórych zastosowaniach metody aktywne są nieakceptowalne. Przykładem tego jest system rejestracji czasu pracy, gdzie sprawdzana jest autentyczność pracowników, ponieważ nie ma tam czasu na tego typu interakcje, gdy dużo pracowników przychodząc do pracy rejestruje się w jednym punkcie i czas na rejestrację jest ograniczony do najwyżej 1 sekundy, żeby nie powstawały kolejki. Dodatkowo też sami pracownicy nie wyrażają często zgody na współpracę z aktywnym systemem. Dlatego w niniejszej pracy skupiono się na rozwiązanych pasywnych.

W początkowym okresie badań (do ok. 2016 roku) nad detekcją autentyczności twarzy zaproponowano wiele rozwiązań, w których ręcznie zostały określone cechy, które są badane, aby stwierdzić, czy twarz zaprezentowana do kamery jest autentyczna [5,6]. Używano w tym celu m.in. deskryptorów cech jak SIFT [7], LBP [8] czy SURF [9].

Jednakże dalej w rzeczywistych warunkach niektóre z tych metod mogą się sprawdzać lepiej od metod opartych na nowszych rozwiązaniach głębokiego uczenia i dalej są rozwijane. W pracy [10] z tego roku przedstawiono wykorzystanie heterogenicznych metod podobieństwa w różnych przestrzeniach barw w połączeniu z modele SVM, które daje bardzo dobre wyniki na testowych bazach danych. W przyszłych pracach tego typu rozwiązanie będzie prawdopodobnie warte wprowadzenia jako dodatkowy algorytm w opracowanej metodzie, ze względu na powtarzalność położenia kamery.

W ostatnim okresie dominującym kierunkiem w badaniach nad wykrywaniem autentyczności twarzy stały się metody hybrydowe [11-14], gdzie cechy, które należy wykryć są danej projektowane ręcznie, natomiast na tych cechach są uczone modele głębokiego uczenia, oraz metody w których modele głębokiego uczenia są wykorzystywane jako jedyny mechanizm w całym procesie wykrywania autentyczności twarzy [15-20]. Ręczne projektowanie cech służy dwóm celom; po pierwsze umożliwia ono zmniejszenie wymiarowości problemu, po drugie zapobiega wykrywaniu przez sieć konwolucyjną nieistotnych właściwości na skutek specyficznych cech zbioru uczącego.

Oprócz stosowanych szeroko sieci konwolucyjnych, w ostatnich latach popularność w różnych zagadnieniach rozpoznawania obrazów zyskują modele Vision Transformers (VITs). W pracach [21] i [22] przeanalizowano możliwości i problemy ich wykorzystania do detekcji autentyczności twarzy.

Odrębną grupę metod stanowią metody wykorzystujące specjalne kamery lub zestawy kamer umożliwiające dobrą ocenę głębi obrazu oraz wykorzystanie ujęć pod różnymi kątami. W pracy [23] przedstawiono wykorzystanie kamer stereo do tego zagadnienia, gdzie umożliwia to dokonanie ujęć twarzy z różnych perspektyw, a nie tylko z jednego kąta. Kamery na podczerwień dają dodatkowe możliwości w tym zakresie. W pracy [24] przedstawiono superpozycję zdjęć wykonanych w podczerwieni z dwóch stron celem oceny autentyczności twarzy. Znane było też rozwiązanie stosowane przez pewną dużą firmę do autentykacji użytkowników przy logowaniu przy pomocy kamery z podpodczerwienią oraz filmy na YouTube demonstrujące jak to zabezpieczenie łatwo obejść.

3. Proponowana metoda wykrywania autentyczności twarzy

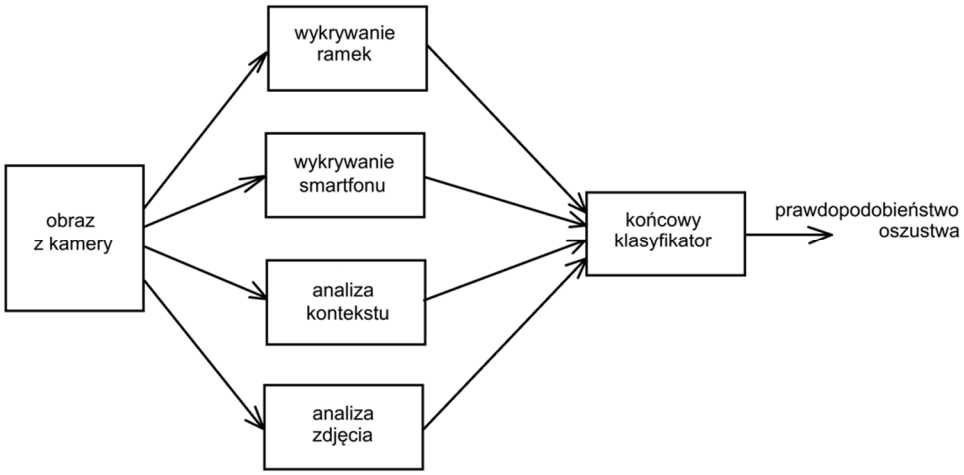
Zdjęcie wraz z wykrytą twarzą oraz pozycją tej twarzy jest do modułu wykrywania autentyczności twarzy dostarczane przez poprzedni moduł oparty na algorytmie BlazeFace [2]. Moduł wykrywania autentyczności twarzy nie otrzymuje w ogóle zdjęć, na których nie wykryto twarzy.

Proponowana metoda wykrywania autentyczności twarzy została zaimplementowana jako komitet typu stacking, jak pokazano na Rys. 2. Jako moduły pierwszego poziomu zostały użyte następujące algorytmy:

1. wykrywanie ramek,
2. wykrywanie smartfonu przy pomocy konwolucyjnej sieci neuronowej,
3. analiza kontekstu otoczenia twarzy,
4. analiza zdjęcia przy pomocy konwolucyjnej sieci neuronowej.

Ze względu na przyjętą architekturę proponowanej metody istnieje łatwa możliwość jej modyfikacji poprzez dodanie kolejnych modeli (algorytmów), ewentualnie wymianę jednego z istniejących modeli na inny.

Najpierw każdy z model przewiduje, czy twarz widoczna na zdjęciu jest autentyczna czy podstawiona na jakimś nośniku. Następnie przewidywania poszczególnych modeli są wysyłane do wejść sieci neuronowej, która działa jako klasyfikator zbiorczy i podejmuje ostateczną decyzję o autentyczności twarzy. Poszczególne modele są przedstawione w poniższych sekcjach.

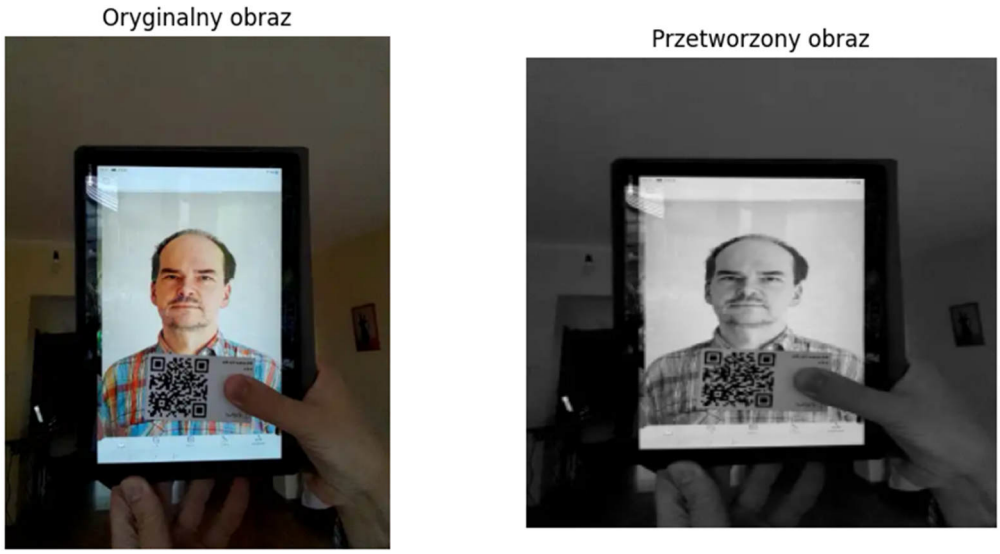


Rysunek 2. Schemat komitetu wykorzystywanego do oceny prawdopodobieństwa oszustwa podstawienia twarzy

3.1. Wykrywanie ramek

Pierwszym z opracowanych modeli jest algorytm wykrywania ramek, którego celem jest określenie, czy wykryta twarz jest wyświetlona na urządzeniu elektronicznym, takim jak smartfon lub tablet. Zasadą działania algorytmu jest wykrywanie krawędzi o wysokim kontraście wokół twarzy. Jeżeli wykryta zostanie wystarczająca ilość krawędzi o odpowiednim rozmiarze oraz jasności, wówczas algorytm ocenia twarz jako podstawioną.

W pierwszym kroku wykonywane jest przetwarzanie wstępne obrazu w celu jego normalizacji i dostosowania go do różnych rozmiarów obrazów wejściowych oraz pozycji twarzy na obrazie. Zdjęcie jest konwertowane do rozmiaru (256, 256). Po wykryciu twarzy na obrazie o takim rozmiarze, konwertowany jest on na skalę szarości.



Rysunek 3. Przetwarzanie wstępne obrazu w module wykrywania ramek.

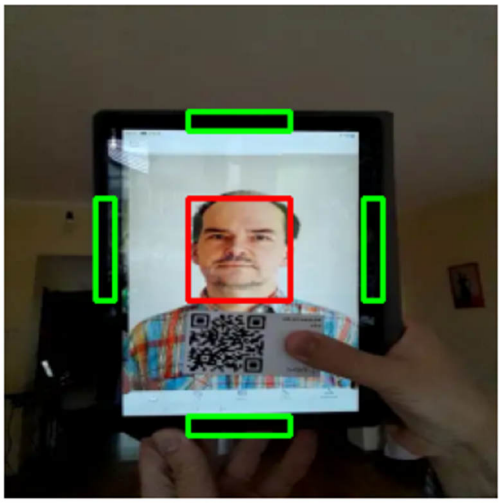
W celu wykrycia ramki wokół twarzy, na początku wybierany jest punkt środkowy twarzy, a następnie obraz jest analizowany w czterech kierunkach w stronę do krawędzi obrazu. W sposób iteracyjny wybierane są coraz to mniejsze fragmenty obrazu, stopniowo przesuwane w stronę zewnętrzną i obliczana jest ich średnia jasność. Jeżeli uzyskana wartość jest poniżej określonego progu, wówczas można określić, że znaleziona została ramka. Próg został ustalony w sposób eksperymentalny na zbiorze danych zawierających zarówno twarze wyświetlane na urządzeniach, jak i autentyczne. Jeżeli ramki zostaną wykryte w co najmniej dwóch kierunkach, wówczas algorytm określa twarz jako podstawioną. W większości przypadków zapewnia to, że nawet jeżeli zostanie wykryta pojedyncza ramka (na przykład

z powodu ciemnego koloru tła lub elementów twarzy, takich jak włosy), twarz nie zostanie błędnie oznaczona jako podstawiona.

```
while current_bezel_size >= min_bezel_size:
    # Wybieramy "pasek" o szerokości current_bezel_size
    bezel = gray[y_start:y_start + current_bezel_size, x_start:x_end]
    # Sprawdzamy, czy średnia wartość pikseli w pasku jest mniejsza niż próg
    average = np.mean(bezel)
    if average <= gray_threshold:
        candidates.append((x_start, y_start, x_end, y_start + current_bezel_size, average))

    # Przesunięcie paska
    y_start += 1

    if y_start + current_bezel_size >= y_end:
        current_bezel_size -= 1
        y_start = 0
```



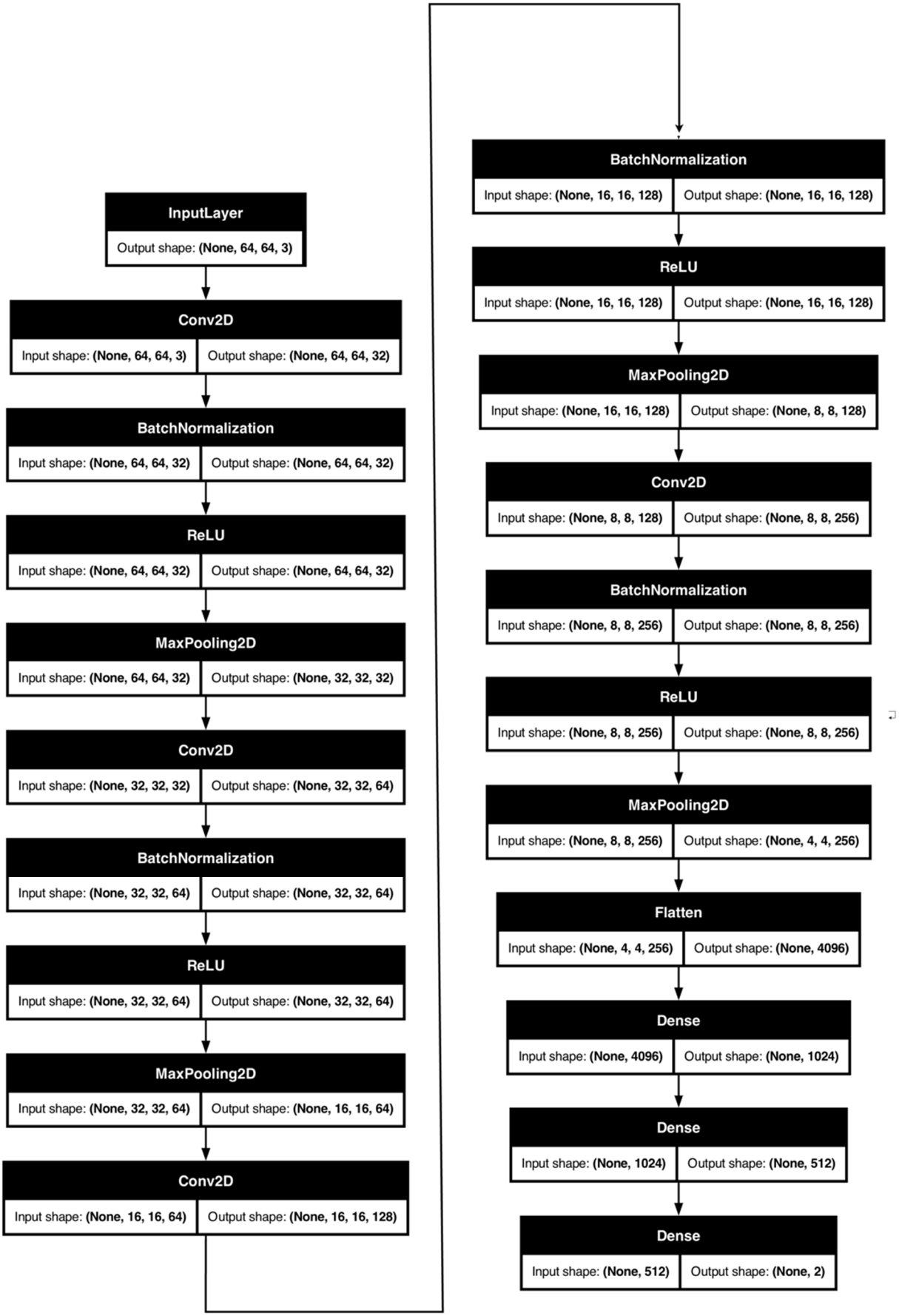
Rysunek 4. Przykładowy wynik algorytmu wykrywania ramek dla twarzy przedstawionej na Rysunku 3 – algorytm ten określa taki przypadek jako twarz podstawioną

Algorytm ten, podobnie jak pozostałe opisane modele, został zoptymalizowany pod kątem wydajności obliczeniowej, aby zminimalizować czas przetwarzania oraz umożliwić równoległe przetwarzanie danych przez każdy model wchodzący w skład proponowanej metody wykrywania autentyczności twarzy. Takie podejście znacznie skraca czas potrzebny na wykonanie analizy.

3.2. Wykrywanie smartfonu przy pomocy konwolucyjnej sieci neuronowej

Najpierw próbowano zastosować YOLOv4. Nie działało dobrze, bo było uczone rozpoznawać całe urządzenia, a tu często w widoku kamery jest tylko fragment urządzenia. Powstała więc konieczność stworzenia modelu nauczonego na własnych danych. Jako rozszerzenie modelu wykrywania ramek urządzeń elektronicznych zaimplementowano model wykrywania smartfonów z wykorzystaniem konwolucyjnej sieci neuronowej. Model ten ściśle współpracuje z pozostałymi z zaimplementowanych algorytmów i pozwala wykryć, czy twarz jest wyświetlana na urządzeniu takim jak smartfon lub tablet, dzięki cechom takim jak kształt, ramka lub części ekranu. Jednak model ten wykrywa również urządzenia elektroniczne, które nie wyświetlają twarzy, na przykład takie, które są trzymane przez osobę znajdującą się w kadrze. Problem ten zostanie rozwiązany w następnej wersji systemu poprzez analizę położenia twarzy względem smartfonu.

DOI: https://doi.org/10.53052/9788367652292.23



Rysunek 5. Proponowana architektura sieci neuronowej do wykrywania smartfonów.

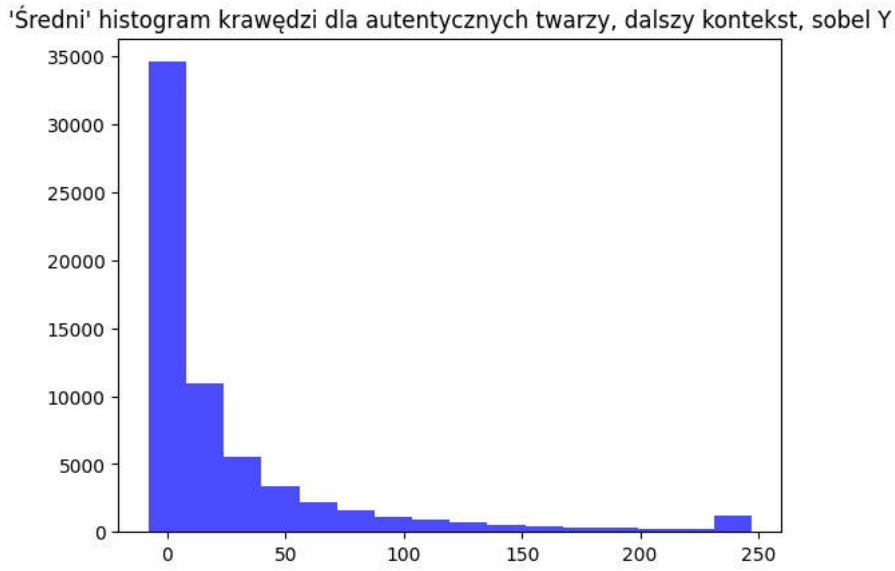
Sieć neuronowa została wytrenowana na podstawie zbioru danych zawierającego 275 obrazów, w tym zdjęć smartfonów, osób trzymających je (wyraźnie widocznych na zdjęciu) oraz podstawionych twarzy wyświetlanych na smartfonach (z dobrze widocznymi elementami takimi jak ramki). Przed wytrenowaniem sieci, obrazy zostały przetworzone poprzez zmianę ich rozmiarów do wartości 128x128 pikseli. Nie był to ten sam zbiór, na którym zostały przeprowadzone końcowe testy dla całej metody wykrywania autentyczności twarzy.

3.3. Analiza kontekstu otoczenia twarzy

Kolejnym modelem wchodzącym w skład komitetu jest analiza kontekstu otoczenia twarzy. Algorytm ten ma na celu wykrycie nieścisłości w obrazie wejściowym, takich jak ostre krawędzie lub obiekty, które znajdują się w otoczeniu twarzy, poprzez porównanie obszaru bliższego wykrytej twarzy do obszaru znajdującego się dalej od niej. Pozwala to na ulepszenie metody wykrywania oszustw poprzez analizę nie tylko twarzy, ale również jej bliskiego otoczenia oraz umożliwia dokładniejsze wykrywanie twarzy wyświetlanych na urządzeniach elektronicznych lub wydrukowanych.

W celu przeprowadzenia analizy, zdefiniowane są dwa obszary - obszar bliższy twarzy, który obejmuje 40% jej szerokości i wysokości we wszystkich kierunkach od prostokąta reprezentującego wykrytą twarz, oraz obszar dalszy twarzy, który jest dwukrotnie większy, czyli obejmuje kolejne 40% wysokości i szerokości twarzy poza obszarem bliższym. Dla obu obszarów przeprowadzane jest wykrywanie krawędzi przy użyciu operatora Sobela. Następnie dla wykrytych krawędzi obliczana jest średnia intensywność pikseli, oddzielnie dla osi X oraz Y, reprezentujących krawędzie poziome i pionowe.

Aby prawidłowo porównać obrazy wejściowe, przy pomocy opracowanej metody przeanalizowano zbiór danych składający się zarówno z twarzy autentycznych jak i z podstawionych. W wyniku utworzone zostały dwa histogramy ze średnią intensywnością wykrytych krawędzi, oddzielnie dla osi X i Y. Dla każdego nowego obrazu, rozkład krawędzi uzyskanych dla niego niniejszą metodą może być porównany do rozkładów średnich. Jeżeli rozkład w znaczący sposób się różni (o wystarczająco wiele odchyleń standardowych), wówczas można stwierdzić, że otoczenie twarzy nie jest spójne.



Rysunek 6. Histogram wygenerowany na podstawie zbioru danych dla dalszego otoczenia twarzy w osi Y.

Każdy przedział utworzonych dla obrazu wejściowego histogramów jest porównywany z rozkładami średnimi przy pomocy standaryzacji Z. Jeśli dany przedział różni się o więcej niż p=1,5 odchylenia standardowego w dowolnym kierunku, wówczas przedział ten jest zliczany. Jeśli n przedziałów różni się o ponad p odchyleń standardowych, wówczas otoczenie twarzy oceniane jest jako niespójne. Optymalne wartości odchyleń standardowych p i liczby przedziałów n zostały ustalone w sposób eksperymentalny, gdzie kryterium była dokładność algorytmu.

DOI: https://doi.org/10.53052/9788367652292.23

```
# Dla X
z_score_X = (hist_X - data_X['hist_mean']) / data_X['hist_std']
num_deviations += np.sum(np.abs(z_score_X) > max_deviation)

# Dla Y
z_score_Y = (hist_Y - data_Y['hist_mean']) / data_Y['hist_std']
num_deviations += np.sum(np.abs(z_score_Y) > max_deviation)

# Zwrócenie wyniku
probability = 1.0 if num_deviations >= min_num_deviations else 0.0
```

Algorytm został przetestowany na zbiorze zawierającym 500 zdjęć przedstawiających twarze autentyczne różnych osób na różnych tłach. Przetwarzanie wstępne każdego z obrazów uwzględniało zmianę rozmiaru zdjęcia do wartości 256x256 pikseli. Zdjęcia były przetwarzane w przestrzeni kolorów RGB.

Algorytm ten wzmacnia zdolność proponowanej metody do wykrywania niespójności na obrazie poprzez skupienie się nie tylko na samej twarzy, ale również jej szerszym kontekście i otoczeniu. Porównywanie intensywności krawędzi w dwóch oddzielnych obszarach wokół twarzy ma na celu wykrycie nagłych zmian czy krawędzi, w szczególności na pierwszym planie obrazu.

Rysunek 7. Przykład krawędzi uzyskanych dla twarzy wyświetlonej na smartfonie znajdującym się na pierwszym planie zdjęcia. System ocenia taki przypadek jako twarz podstawioną.

3.4. Analiza zdjęcia przy pomocy konwolucyjnej sieci neuronowej

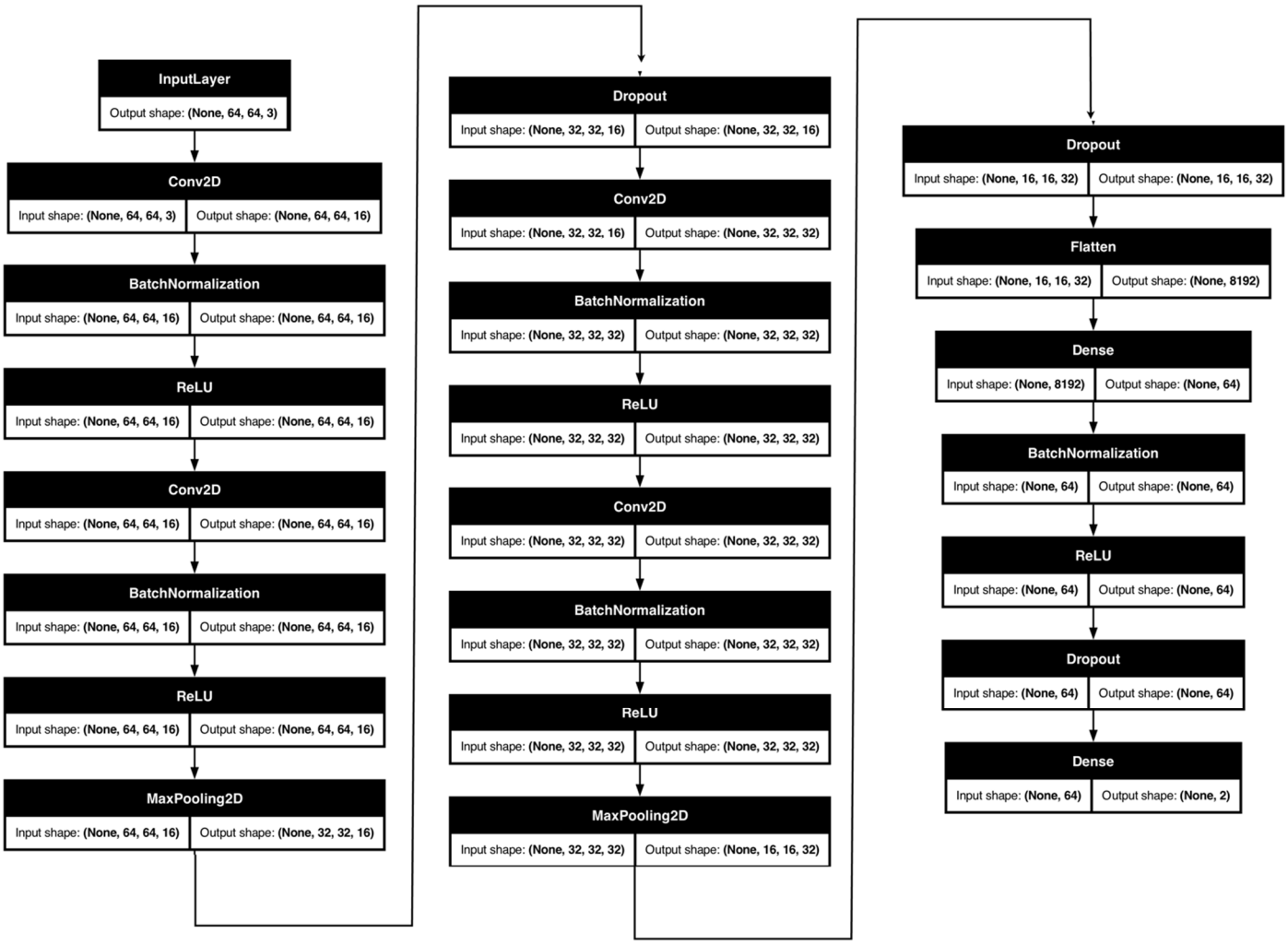
Ostatnim z proponowanych algorytmów jest analiza zdjęcia przy pomocy konwolucyjnej sieci neuronowej. Celem tego algorytmu jest analiza elementów obrazu, które w przeciwnym razie byłyby trudne do przetworzenia za pomocą algorytmów, takich jak jakość twarzy, wygląd obiektów i krawędzi w jej pobliżu czy spójność obrazu.

Architektura sieci składa się z dwóch zestawów warstw konwolucyjnych, po których następują warstwy normalizacji, funkcje aktywacji ReLU oraz warstwy łączącej i warstwy porzucenia (CONV => RELU => CONV => RELU => POOL => DROPOUT). W dalszej części sieci zaimplementowana jest warstwa w pełni połączona oraz warstwa wyjściowa z funkcją aktywacji Softmax.

Sieć neuronowa została wytrenowana na zbiorze 210 twarzy oznaczonych jako autentyczne lub podstawione, w tym zdjęć osób znajdujących się przed kamerą, wyświetlonych na smartfonie lub tablecie oraz wydrukowanych. Obrazy zostały wstępnie przetworzone, co uwzględnia wyśrodkowanie ich wokół wykrytej twarzy i zmianę ich rozmiaru do wartości 64x64 pikseli przed wykorzystaniem danych do trenowania sieci. Wyśrodkowanie obrazów wokół wykrytej twarzy ma na celu zapewnienie, że sieć skupi się na analizie bezpośredniego otoczenia twarzy zamiast elementów tła i jest w szczególności przydatne, gdy twarz jest niewielką częścią zdjęcia. Aby lepiej dostosować zbiór uczący do warunków rzeczywistej analizy, zastosowano augmentację danych poprzez powiększenie liczby przypadków które obejmowało: losowe odbicie lustrzane, losowy obrót o maksymalnie 5 stopni oraz losową zmianę jasności i nasycenia obrazu o maksymalnie 10%. Parametry zostały dobrane w taki sposób, aby wprowadzone zmiany były niewielkie i odzwierciedlały rzeczywiste warunki. Nie był to ten sam zbiór, na którym zostały przeprowadzone końcowe testy dla całej metody wykrywania autentyczności twarzy.

Warstwa wyjściowa sieci neuronowej ma dwa wyjścia z funkcją aktywacji Softmax, i zwraca dwie wartości, interpretowane jako prawdopodobieństwo przynależności obrazu do klasy "autentyczna" i "podstawiona". Dane

wyjściowe w formie prawdopodobieństw można w prosty sposób zintegrować z pozostałymi algorytmami w celu obliczenia ostatecznego wyniku i prawdopodobieństwa podstawienia twarzy.



Rysunek 8. Proponowana architektura sieci neuronowej.

3.5. Obliczanie ostatecznej oceny twarzy oraz prawdopodobieństwa

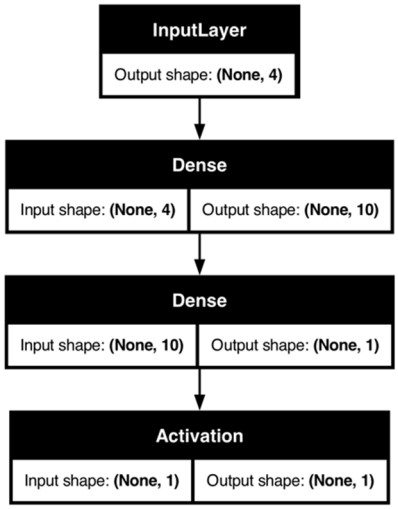
Jak wykazały nasze doświadczenia, wykorzystanie komitetu w formie stackingu (rysunek 1) jest tu lepszym rozwiązaniem, niż odgórne zdefiniowanie wag dla każdego ze składowych algorytmów (modeli). Sieć neuronowa będąca końcowym klasyfikatorem (rysunek 8) pobiera na wejściu wynik cząstkowy uzyskany z każdego z modeli, reprezentowany jako prawdopodobieństwo podstawienia, a następnie klasyfikuje twarz jako autentyczną lub podstawioną wraz dostarczeniem informacji o prawdopodobieństwie podstawienia.

Proponowana architektura sieci składa się z dwóch warstw w pełni połączonych z funkcjami aktywacji ReLU oraz warstwy wyjściowej wykorzystującej sigmoidalną funkcję aktywacji. Sieć ta przyjmuje jako wejścia cztery wartości pochodzące z omówionych powyżej modeli. Wartościami tymi są liczby rzeczywiste które reprezentują prawdopodobieństwa, że twarz jest podstawiona. Dla przykładu, jeżeli twarz jest oceniona przez dany algorytm jako podstawiona z prawdopodobieństwem 80% (czyli autentyczna z prawdopodobieństwem 20%), to wówczas liczba na wejściu do sieci będzie równa 0,20. Na wyjściu z sieci otrzymywana jest pojedyncza wartość z zakresu [0; 1], która oznacza prawdopodobieństwo, że twarz jest podstawiona. Jeżeli prawdopodobieństwo to jest większe niż pewien próg, który w testach przyjęto jako 50%, wówczas twarz jest oceniana jako podstawiona; w przeciwnym wypadku, twarz jest uznawana za autentyczną.

DOI: https://doi.org/10.53052/9788367652292.23

Sieć została wytrenowana dla uzyskanych z poszczególnych algorytmów wyników wraz z oczekiwanym wynikiem końcowym, zarówno dla twarzy autentycznych, jak i podstawionych. Trenowanie sieci odbyło się dla następujących parametrów: współczynnik uczenia: 0,01; ilość epok: 100; optymalizator: Adam; funkcja straty: Binary Cross Entropy.

Zaimplementowanie niniejszego sposobu oceny twarzy było kluczowym krokiem do poprawy wydajności systemu i pozwoliło na znacznie bardziej elastyczne podejście, które można w łatwy sposób rozszerzyć dla większych zbiorów danych lub poprzez implementację kolejnych modeli.



Rysunek 9. Proponowana architektura sieci wyznaczającej ostateczne prawdopodobieństwo podstawienia twarzy.

4. Implementacja metody wykrywania autentyczności twarzy

Opisana metoda została zrealizowana przy użyciu języka Python.

Użyte w opisywanej metodzie sieci neuronowe zostały stworzone przy pomocy biblioteki PyTorch, przy czym przetwarzanie wstępne danych wejściowych zostało wykonane przy pomocy funkcji z biblioteki Torchvision. Kod źródłowy realizujący proponowaną metodę dostępny jest pod adresem: https://github.com/Stukeley/FaceAuthenticityDetection

W celu przetwarzania obrazów zastosowano bibliotekę OpenCV. Ze względu na prostotę implementacji, tymczasowo wykorzystane zostały zapewnione przez OpenCV funkcje oparte o HaarCascade FrontalFace, które dostarczają współrzędne twarzy do metody wykrywania autentyczności twarzy. W rzeczywistym systemie, zamiast modelu HaarCascade działa algorytm BlazeFace. Pracujemy nad zamianą tego algorytmu na BlazeFace również w aplikacji testowej, aby generowane współrzędne twarzy na zdjęciu były identyczne z otrzymywanymi w rzeczywistym systemie. Nie jest planowane wykorzystanie algorytmu RetinaFace, ponieważ zalecane jest by w pełni odzwierciedlić warunki rzeczywistego systemu opartego o algorytm BlazeFace, a autorzy artykułu nie mają możliwości dokonywania zmian w już działających modułach.

Celem odzwierciedlenia rzeczywistych warunków, oceniana jest tylko dokładność metody wykrywania autentyczności twarzy, a nie metod dostarczających do niej dane i lokalizujących twarz na zdjęciu, bowiem w rzeczywistym systemie do tej metody będą dostarczane tylko zdjęcia z poprawnie wykrytą i zlokalizowaną twarzą, i tylko na takich zdjęciach metoda wykrywania autentyczności twarzy będzie działać.

5. Wyniki

Przetestowane zostały zarówno poszczególne modele wchodzące w skład komitetu metody wykrywania autentyczności twarzy, jak i cała metoda. Do testów wykorzystany został zbiór danych zawierających 185 zdjęć, 100 z nich przedstawiało osoby znajdujące się przed kamerą, a pozostałe 85 twarze podstawione na urządzeniach elektronicznych lub wydrukowane na papierze. Przygotowany został początkowo zbiór 200 zdjęć, jednak na 15 z nich

nie została wykryta twarz przez algorytm dostarczający dane do proponowanej metody wykrywania autentyczności twarzy. Aby odzwiedlić warunki rzeczywistego systemu testy zostały przeprowadzone na 185 zdjęciach zawierających wykryte i zlokalizowane twarze. Zbiór ten różnił się od zbiorów danych wykorzystanych do trenowania poszczególnych modeli wchodzących w skład komitetu, jak zostało opisane w podrozdziałach 3.2., 3.3 oraz 3.4.

Tabela 1. Uzyskane wyniki cząstkowe dla poszczególnych modeli (algorytmów). Zastosowane metryki zakładają, że wartość pozytywna oznacza twarz podstawioną.

Moduł	Dokładność	Precyzja	Czułość
Wykrywanie ramek	86,3%	85,2%	76,7%
Wykrywanie smartfonu	67,6%	65,6%	79,7%
Analiza kontekstu	68,5%	72,6%	63,1%
Analiza zdjęcia	85,9%	92,4%	80,2%

Dla sieci neuronowej oceniającej prawdopodobieństwo podstawienia twarzy na podstawie ocen poszczególnych modułów, przeprowadzona została walidacja krzyżowa - sieć neuronowa ta została wytrenowana pięciokrotnie, za każdym razem wykorzystując do trenowania cztery części zbioru zawierającego wyniki cząstkowe, a pozostałą część wykorzystując do walidacji tego modelu.

Tabela 2. Prawdopodobieństwo oszustwa wygenerowane przez poszczególne modele (algorytmy) w odpowiedzi na cztery przykładowe zdjęcia z kamery.

wykr. ramek	wykr. smartfonu	anal. kontekstu	anal. zdjęcia	końcowy wynik
0,0	0,934	1,0	0,001	0,05
0,0	0,518	1,0	0,001	0,04
1,0	0,999	0,0	0,989	0,99
0,0	1,0	1,0	0,389	0,67

Ostatnia kolumna w Tabeli 2 oznacza wyjście sieci neuronowej będącej ostatecznym klasyfikatorem w tym komitecie. Dzięki sigmoidalnej funkcji transferu w ostatniej warstwie tej sieci można interpretować jej wyjście jako prawdopodobieństwo, że dana twarz jest prawdziwa. Aby to było możliwe należy ograniczyć jej długość uczenia, zanim w końcowej fazie uczenia przewidywane wartości zaczną być bardzo bliskie zeru lub jedności, ale jednocześnie uczyć ją na tyle długo, by osiągnąć możliwe wysokie dokładności klasyfikacji.

W przeprowadzonej tu analizie za próg określający twarz podstawioną uznano wynik 0,50. Jeżeli prawdopodobieństwo zwrócone przez sieć jest większe od 50%, wówczas system określa twarz za podstawioną. W przypadku strojenia systemu, aby zminimalizować ilość twarzy autentycznych niepoprawnie oznaczonych jako podstawione (nawet kosztem zwiększonego oznaczania twarzy podstawionych jako autentyczne), możliwa jest zmiana tego progu na wyższą wartość. I na odwrót, jeśli ważniejsze jest skuteczne wykrycie twarzy podstawionych.

Tabela 3. Macierz pomyłek (błędów) uzyskana dla końcowej predykcji stworzonego modelu przy 5-krotnej walidacji krzyżowej.

		Przewidziane	
		Podstawione	**Autentyczne**
Rzeczywiste	**Podstawione**	79	6
	Autentyczne	12	88

Dla opisanego w niniejszym rozdziale zbioru 185 zdjęć, średnia dokładność całego systemu wyniosła 90,3%, średnia precyzja – 93,0%, a średnia czułość – 86,7%. Odchylenia standardowe tych wielkości wynosiły ok. 2%.

Tabela 4. Macierz pomyłek (błędów) uzyskana dla omawianego we wstępie rozwiązania komercyjnego uzasadniające konieczność stworzenia własnego rozwiązania (tabela zawiera tylko 24 zdjęcia z rejestracji czasu pracy ze względu na ograniczenia konta testowego). Problemem jest tu dominujące rozpoznawanie zdjęć autentycznych jako podstawione.

		Przewidziane	
		Podstawione	**Autentyczne**
Rzeczywiste	**Podstawione**	9	1
	Autentyczne	10	4

6. Wnioski

Przedstawiona została metoda wykrywania autentyczności twarzy w zastosowaniu w systiemie rejestracji czasu pracy. Przy często niskiej jakości zdjęć oraz przy ograniczeniach narzuconych autorom tego artykułu na proponową metodę, istniejące rozwiązania detekcji autentyczności twarzy się nie sprawdzały, wykrywając znaczną część twarzy autentycznych jako podstawione. Te ograniczenia to w szczególności dostarczane zdjęcia często wykonane niestaranne i z różnej perspektywy oraz w różnych warunkach oświetleniowych oraz skompresowanych do rozmiaru 6 kB w formacie webp, brak możliwości pozyskania lepszych zdjęć, użycia kamer z podczerwienią i zastosowania innych algorytmów na poprzednich etapach przetwarzania zdjęć. W związku z tym przystąpiono do opracowania własnej metody, która została opisana w tym artykule. Metoda ta jest oparta o komitet typu stacking, gdzie każdy składowy model ocenia inne aspekty zdjęcia. Wykorzystanie takiego komitetu umożliwia łatwą rozbudowę przedstawionego rozwiązania o kolejne modele. Ważnym, lecz niezwykle pracochłonnym zagadnieniem jest także wygenerowanie większego zbioru uczącego o odpowiednim zróżnicowaniu sytuacji oszustw polegających na podstawieniu do kamery twarzy danej osoby na zdjęciu, wideo lub w inny sposób. Opracowanie nowych algorytmów i dodanie ich do komitetu metody wykrywania autentyczności twarzy wraz z powiększeniem zbioru uczącego powinno pozwolić na uzyskanie większej dokładności, niż aktualne 90,3%, co będzie tematem przyszłej pracy autorów artykułu.

Podziękowania

Praca została dofinansowana przez NCBiR, projekt nr POIR.01.01.01-00-1144/19.

Literatura

1. Apple. “About Face ID advanced technology”, (2024, January 10), https://support.apple.com/en-us/102381
2. Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, Matthias Grundmann. BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs, CVPR 2019
3. Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, Stefanos Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition, CVRP 2018
4. Jiankang Deng et. al., RetinaFace: Single-shot Multi-level Face Localisation in the Wild, CVRP 2020

5. T. de Freitas Pereira, A. Anjos, J. M. De Martino, S. Marcel. LBP - TOP based countermeasure against face spoofing attacks. Proc. Asian Conf. Comput. Vis., 2012, pp. 121–132.
6. J. Komulainen, A. Hadid, M. Pietikainen. Context based face anti-spoofing. Proc. IEEE 6th Int. Conf. Biometrics: Theory Appl. Syst., 2013, pp. 1–8.
7. K. Patel, H. Han, A. K. Jain. Secure face unlock: Spoof detection on smartphones. IEEE Trans. Inf. Forensics Security, vol. 11, no. 10, pp. 2268–2283, Oct. 2016.
8. Z. Boulkenafet, J. Komulainen, A. Hadid. Face anti-spoofing based on color texture analysis. Proc. IEEE Int. Conf. Image Process., 2015, pp. 2636–2640.
9. Z. Boulkenafet, J. Komulainen, A. Hadid. Face antispoofing using speeded-up robust features and fisher vector encoding. IEEE Signal Process. Lett., vol. 24, no. 2, pp. 141–145, Feb. 2017.
10. Yahya-Zoubir Bahia, Fedila Meriem, Bengherabi Messaoud. Face spoofing detection using Heterogeneous Auto-Similarities of Characteristics. Engineering Applications of Artificial Intelligence, Vol. 130, April 2024, 107788
11. X. Song, X. Zhao, L. Fang, T. Lin. Discriminative representation combinations for accurate face spoofing detection. Pattern Recognit., vol. 85, pp. 220–231, 2019.
12. M. Asim, Z. Ming, M. Y. Javed. CNN based spatio-temporal feature extraction for face anti-spoofing. in Proc. IEEE 2nd Int. Conf. Image Vis. Comput., 2017, pp. 234–238.
13. Y. A. U. Rehman, L.-M. Po, J. Komulainen. Enhancing deep discriminative feature maps via perturbation for face presentation attack detection. Image Vis. Comput., vol. 94, 2020, Art. no. 103858.
14. M. Khammari. Robust face anti-spoofing using CNN with LBP and WLD. IET Image Process., vol. 13, pp. 1880–1884, 2019.
15. Z. Yu et al. Searching central difference convolutional networks for face anti-spoofing. in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 5294–5304.
16. Z. Yu, X. Li, X. Niu, J. Shi, and G. Zhao. Face anti-spoofing with human material perception. in Proc. Eur. Conf. Comput. Vis., 2020, pp. 557–575.
17. X. Yang et al. Face anti-spoofing: Model matters, so does data. in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 3502–3511.
18. Z. Yu et al. Multi-modal face anti-spoofing based on central difference networks. in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, 2020, pp. 2766–2774.
19. S. Zhang et al. CASIA-SURF: A large-scale multi-modal benchmark for face anti-spoofing. IEEE Trans. Biometrics Behav. Identity Sci., vol. 2, no. 2, pp. 182–193, Apr. 2020.
A. George, S. Marcel. Deep pixel-wise binary supervision for face presentation attack detection. in Proc. IEEE Int. Conf. Biometrics, 2019, pp. 1–8.
20. Ajian Liu, Yanyan Liang. MA-ViT: Modality-Agnostic Vision Transformers for Face Anti-Spoofing. Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)
21. Yu Zitong, et. al. Rethinking vision transformer and masked autoencoder in multimodal face anti-spoofing. https://doi.org/10.1007/s11263-024-02055-1. DR-NTU, Singapore 2024.
22. Muhamad Amirul Haq, Le Nam Quoc Huy, Muhammad Ridlwan. Multi-Angle Facial Recognition: Enhancing Biometric Security with a Broadly Positioned Stereo-Camera System. E3S Web of Conferences, 03032, 2024
23. Zhishan Li, Jiayan Yuan, Baozhi Jia, Yifan He, Lei Xie. An Effective Face Anti-Spoofing Method via Stereo Matching. IEEE Signal Processing Letters Volume: 28, pp: 847 – 851, 2021