

# WYKORZYSTANIE MECHANIZMÓW AI DO WYKRYWANIA ODLEGŁOŚCI OD EKRANU

**Kacper Ciosek, Łukasz Apiecionek**

Uniwersytet Kazimierza Wielkiego  
Wydział Informatyki  
ul. Kopernika 1, 85-074 Bydgoszcz  
e-mail: kacper.ciosek@student.ukw.edu.pl

**Streszczenie:** Praca dotyczy wykorzystania mechanizmów sztucznej inteligencji do wykrywania odległości twarzy użytkownika od ekranu. Opracowano aplikację mobilną na system Android, która analizuje obraz z kamery przedniej i określa dystans użytkownika od urządzenia. W projekcie wykorzystano model FaceLandmarker z biblioteki MediaPipe, który identyfikuje kluczowe punkty twarzy. Następnie, na podstawie analizy statystycznej i kalibracji, aplikacja estymuje odległość użytkownika. System może znaleźć zastosowanie w aplikacjach edukacyjnych i zdrowotnych, np. informując dzieci o nieprawidłowej odległości od ekranu. Testy wykazały, że rozwiązanie działa poprawnie w określonych warunkach, ale jego skuteczność spada przy słabym oświetleniu lub większych dystansach.

**SŁOWA KLUCZOWE:** Sztuczna inteligencja, Sieci neuronowe, Konwolucyjne sieci neuronowe, Wykrywanie twarzy, Analiza obrazu, MediaPipe, Android Studio

## USE OF AI MECHANISMS FOR DETECTING SCREEN DISTANCE

**Abstrakt:** The thesis explores the use of artificial intelligence mechanisms to detect the user's facial distance from the screen. A mobile application for Android was developed, analyzing the front camera image to estimate the user's distance from the device. The project utilizes the FaceLandmarker model from the MediaPipe library, which identifies key facial landmarks. Using statistical analysis and calibration, the application estimates the distance. This system can be applied in educational and health-related applications, such as alerting children about improper screen proximity. Tests demonstrated that the solution performs well under certain conditions but loses accuracy in low-light environments or at greater distances.

**Keywords:** Artificial intelligence, Neural networks, Convolutional neural networks, Face detection, Image analysis, MediaPipe, Android Studio

### 1. WSTĘP

W ostatnich latach świat informatyki doświadcza dynamicznego rozwoju w obszarze sztucznych sieci neuronowych. Pomimo silnej koncentracji badań i wdrożeń na dużych modelach językowych (LLM) oraz różnego rodzaju chatbotach, istnieje wiele innych podejść i rozwiązań opartych na sieciach neuronowych, które są równie istotne z punktu widzenia rozwoju nauki i technologii [1], lecz pozostają relatywnie słabiej eksponowane w literaturze oraz debacie publicznej [2].

Jednym z tych rozwiązań są sieci konwolucyjne. Pozwalają one na rozpoznawanie i lokalizację obiektów na obrazach.

Działają poprzez zastosowanie warstw konwolucyjnych, które przekształcają dane wielowymiarowe mniejsze wymiary. Pozwala to na zmniejszenie ilości przetwarzanych danych z równoczesnym zachowaniem cech tych wielowymiarowych danych [3].

Dzięki temu powstały między innymi technologie pozwalające odczytywanie tekstu z obrazu i rozpoznawanie twarzy [4].

Jednym zagadnieniem jest rozpoznawanie cech charakterystycznych twarzy, gdzie model sieci neuronowej przetwarza obraz, gdzie została wykryta twarz, w celu wykrycia lokalizacji oczu, nosa, ust i obrysu twarzy [5 - 6].

Celem pracy jest implementacja modelu wykrywającego cechy charakterystyczne twarzy w aplikacji mobilnej w celu

wykrycia odległości twarzy od ekranu. Aplikacja ma mieć możliwość kalibracji mechanizmu wykrywania tejże odległości.

## 2. OPIS ROZWIĄZANIA

Modelami użytymi do wykrycia i identyfikacji cech charakterystycznych są MediaPipe FaceLandmarker. Na tę grupę modeli składa się:

- BlazeFace – wykrywa twarz na ekranie i zwraca obszar gdzie jest twarz, 6 punktów charakterystycznych i pewność wykrycia
- FaceMesh – bierze dane wyjściowe z BlazeFace i wykrywa 478 kolejnych punktów charakterystycznych twarzy, kilka flag oznaczających mimikę twarzy i pewność, że twarz jest twarzą
- Blendshape V2 – wykrywa 52 stanów w jakim jest twarz z prawdopodobieństwem 0-1. Np. mrugnięcie okiem, otwarcie ust, podniesienie brwi.

W projekcie zostały wykorzystane BlazeFace i FaceMesh. Pierwszy do znalezienia twarzy, drugi do znalezienia pozycji oczu, które zostaną użyte w obliczeniach twarzy od ekranu [7].

Obliczanie odległości twarzy od ekranu zostało rozwiązane poprzez zapisanie odległości pomiędzy oczami użytkownika przy znanej odległości. Następnie przy zmianie odległości między oczami w wykrytej twarzy (przez zbliżanie bądź oddalanie się twarzy) odległość od ekranu jest szacowana poniższym algorytmem:

$$d_2 = \frac{w_1 * d_1}{w_2}$$

gdzie:

- $d_1$  – odległość twarzy od ekranu przy kalibracji
- $w_2$  – odległość pomiędzy oczami przy kalibracji
- $d_2$  – szukana odległość twarzy od ekranu
- $w_1$  – odległość pomiędzy oczami przy nieznannej odległości twarzy od ekranu

Cały projekt został wykonany jako aplikacja mobilna na systemy Android. Po uruchomieniu aplikacji pokazany jest widok z przedniej kamery urządzenia. Na tym panelu użytkownik widzi odległość swojej twarzy od ekranu i jest w

stanie także skalibrować algorytm obliczeniowy wykorzystując predefiniowane progi odległości od 20cm do 80cm, w odstępach co 10 cm.

Aplikacja zawiera panel boczny, gdzie ma możliwość zresetowania parametry algorytmu do domyślnych.

Skalibrowany już algorytm zapisuje się w pamięci telefonu, dzięki czemu użytkownik nie musi robić tego ponownie po wyłączeniu i włączeniu aplikacji.

Testy były ukierunkowane na sprawdzenie dokładności wykrywania odległości twarzy od ekranu. Wartość faktyczna została określona przez pomiar odległości umiejscowienia trójnożu z telefonem od miejsca, gdzie umieszczona będzie twarz. Test przeprowadzono na odległości 30cm i 70cm. Błąd względny w testach przy różnej kalibracji waha się od 0.5% do 4.5%.

Przetestowane zostały też możliwości samego modelu pod względem wykrycia i utrzymania wykrycia twarzy. Jest on w stanie znaleźć twarz na odległości mniejszej bądź równej 1.2m od kamery. Jeżeli twarz jest już wykryta, jest w stanie utrzymać wykrycie do 4m. Jednakże na większych odległościach model łatwiej traci wykrycie, np. przez zakrycie twarzy książką lub szybki ruch głową.

## 3. PODSUMOWANIE I WNIOSKI

Autor reasumuje, że możliwe jest wykonanie aplikacji zdolnej do obliczenia odległości twarzy od ekranu, a następnie podaje możliwe zastosowania takiej technologii, takie jak analityka czytelności interfejsu użytkownika, a także promowanie zdrowszych nawyków użytkownika urządzeń mobilnych.

Samą technologię wykorzystaną w projekcie ocenia za adekwatną, ale zostawiającą pole do rozwoju.

W całości prace stanowi prostą implementację modelu w aplikacji mobilnej. Proporcja części teoretycznej do praktycznej jest zbyt duża, zwłaszcza jeśli by odjąć zrzuty ekranu. Brakuje informacji o sieciach konwolucyjnych wykorzystujących transformery, nowe podejście w sieciach konwolucyjnych, które może być alternatywą dla tradycyjnych CNN [8].

## **Literatura**

1. Overcoming classic challenges for artificial neural networks by providing incentives and practice; Kazuki Irie, Brenden M. Lake; arXiv:2410.10596 [cs.AI]
2. <https://www.forbes.com/sites/janakirammsv/2024/02/12/is-generative-ai-overshadowing-the-proven-workhorses-of-modern-tech/> (dostęp 2026.01.13)
3. Zhao, X., Wang, L., Zhang, Y. et al. A review of convolutional neural networks in computer vision. *Artif Intell Rev* 57, 99 (2024). <https://doi.org/10.1007/s10462-024-10721-6> (dostęp 2026.01.18)
4. Deep Learning; Yann LeCun, Yoshua Bengio, Geoffrey Hinton
5. Facial Landmark Detection Using Deep Learning: A Comprehensive Approach with Resnet18; Sk. Mastan Sharif, N. Sri Harsha2, Sk. Abdul Subhan, Y. Threshal; <http://dx.doi.org/10.4108/eai.28-4-2025.2358028> (dostęp 20226.01.18)
6. Fast Facial Landmark Detection and Applications: A Survey; Kostiantyn Khabarлак, Larysa Koriashkina; arXiv:2101.10808 [cs.CV]
7. [https://ai.google.dev/edge/mediapipe/solutions/vision/face\\_landmarker](https://ai.google.dev/edge/mediapipe/solutions/vision/face_landmarker) (dostęp 2026.01.12)
8. A survey of the Vision Transformers and their CNN-Transformer based Variants; Asifullah Khan, Zunaira Rauf, Anabia Sohail, Abdul Rehman, Hifsa Asif, Aqsa Asif, Umair Farooq; arXiv:2305.09880.