

Rozpoznawanie twarzy w terminalach mobilnych

Katedra Systemów Multimedialnych

1

Cechy biometrii twarzy

	Bezpieczeństwo		Praktyczność				
	Odporność na fałszerstwo	Dokładność	Szybkość	Rejestracja	Wygoda	Koszt	Rozmiar
Odcisk palca							
Tęcza							
Geometria twarzy							
Weryfikacja głosowa							
Biometria naczyniowa palca							

2

Rozpoznawanie twarzy - zastosowania

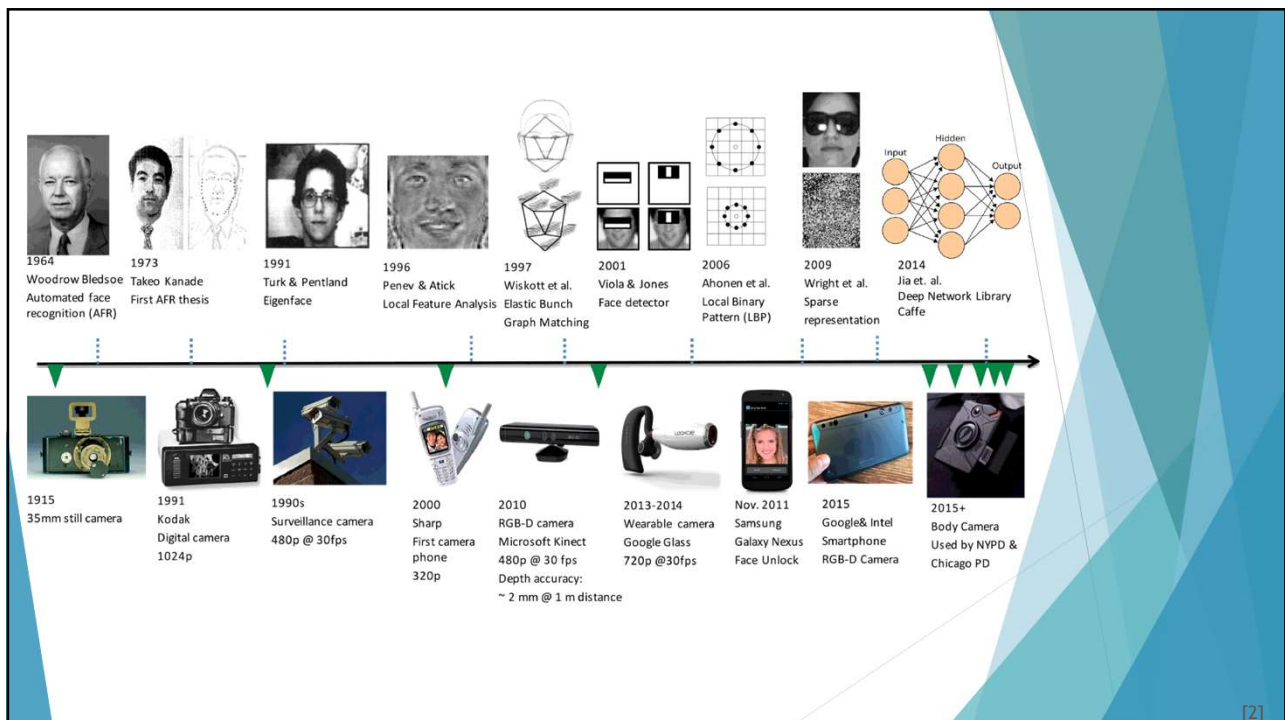
- Weryfikacja - polega na przedstawieniu się użytkownika, a następnie porównaniu wskazanego identyfikatora ze wzorcem zapisanym w bazie (porównanie 1:1)



- Identyfikacja - polega na analizie cech użytkownika, a następnie porównaniu ich ze wszystkim dostępnymi wzorcami w bazie (porównanie 1:N)

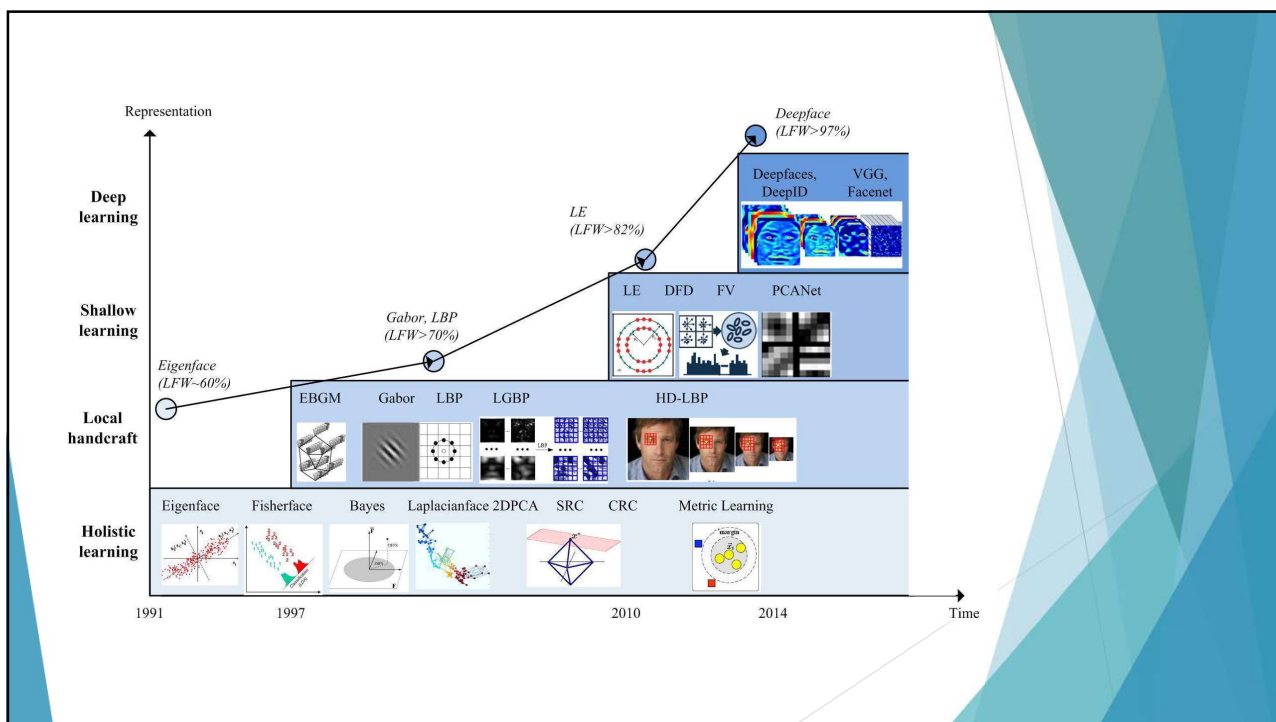


3



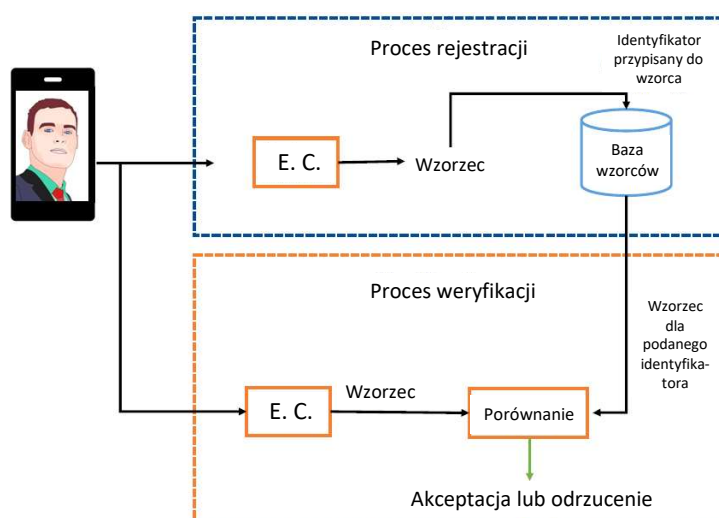
[2]

4

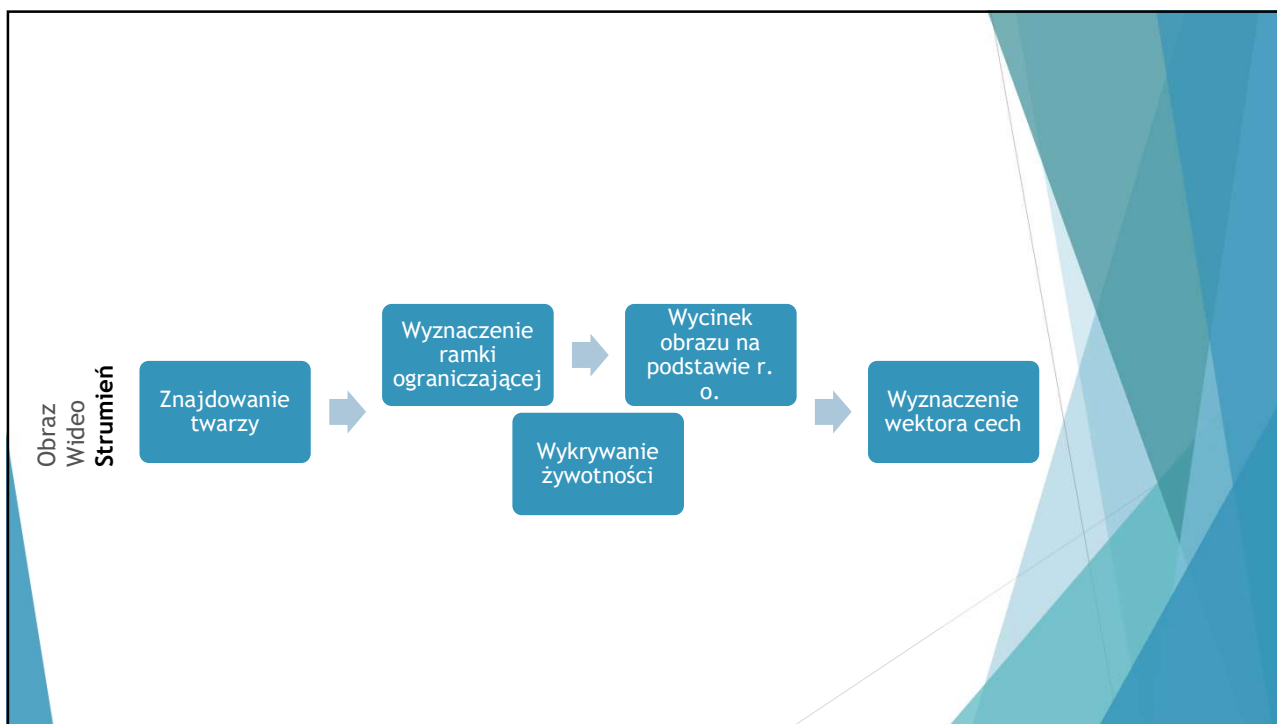


5

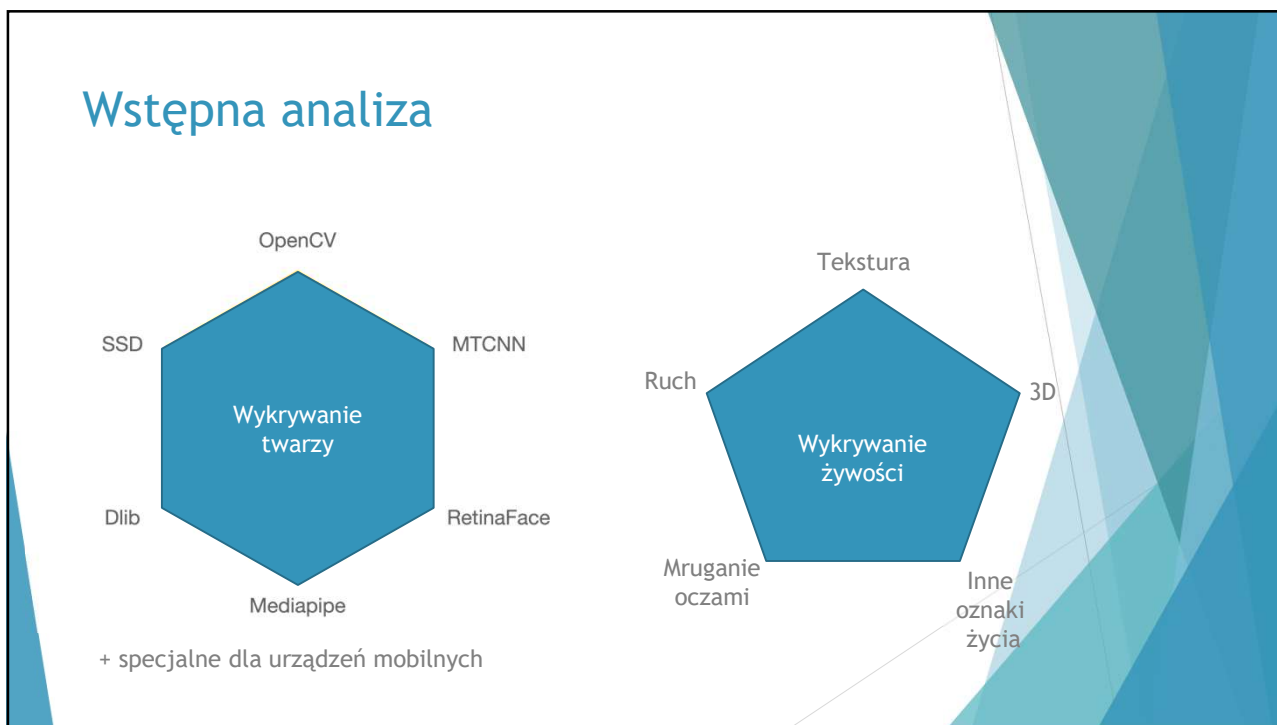
Typowy schemat procesów



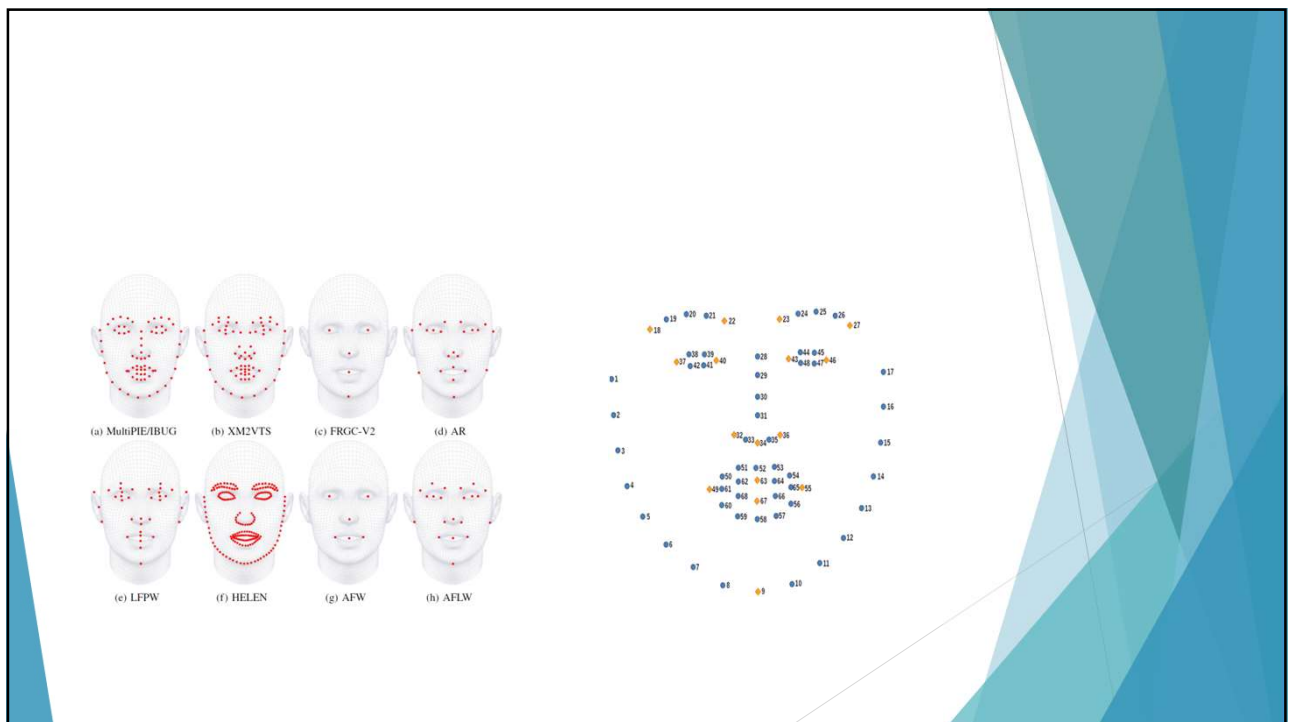
6



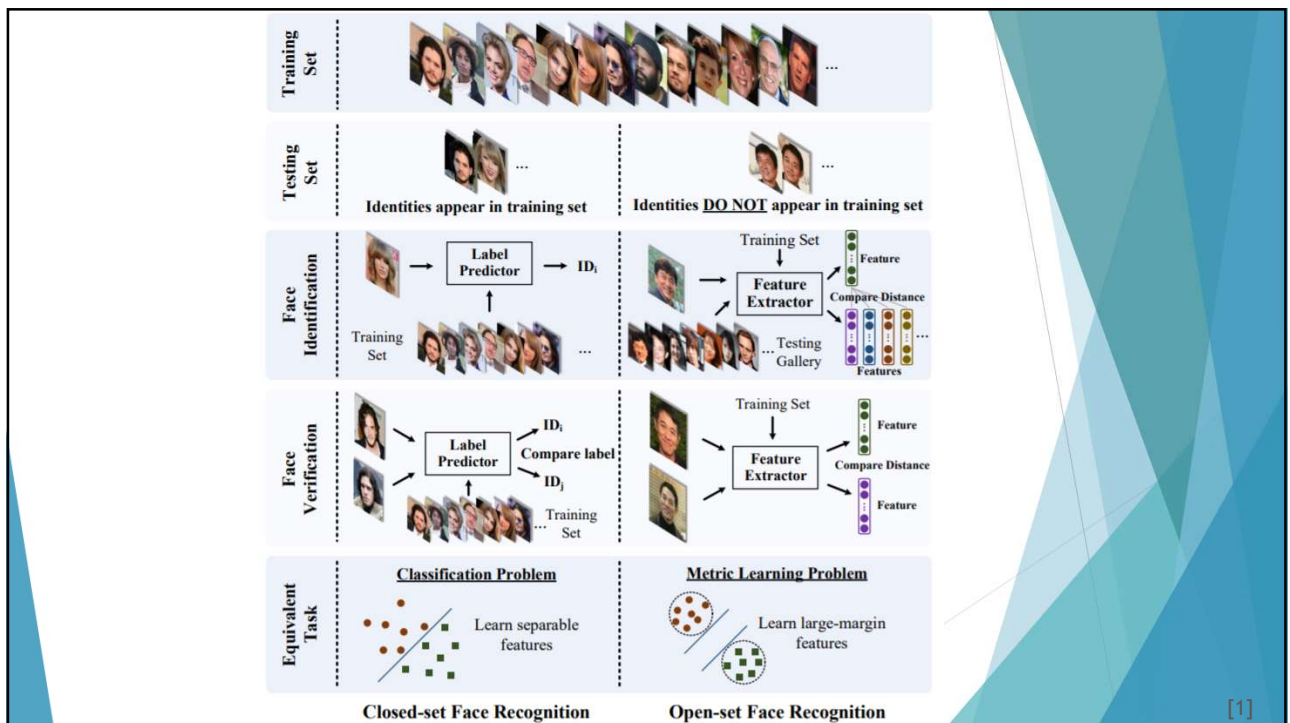
7



8



9



10

Wyznaczanie parametrów

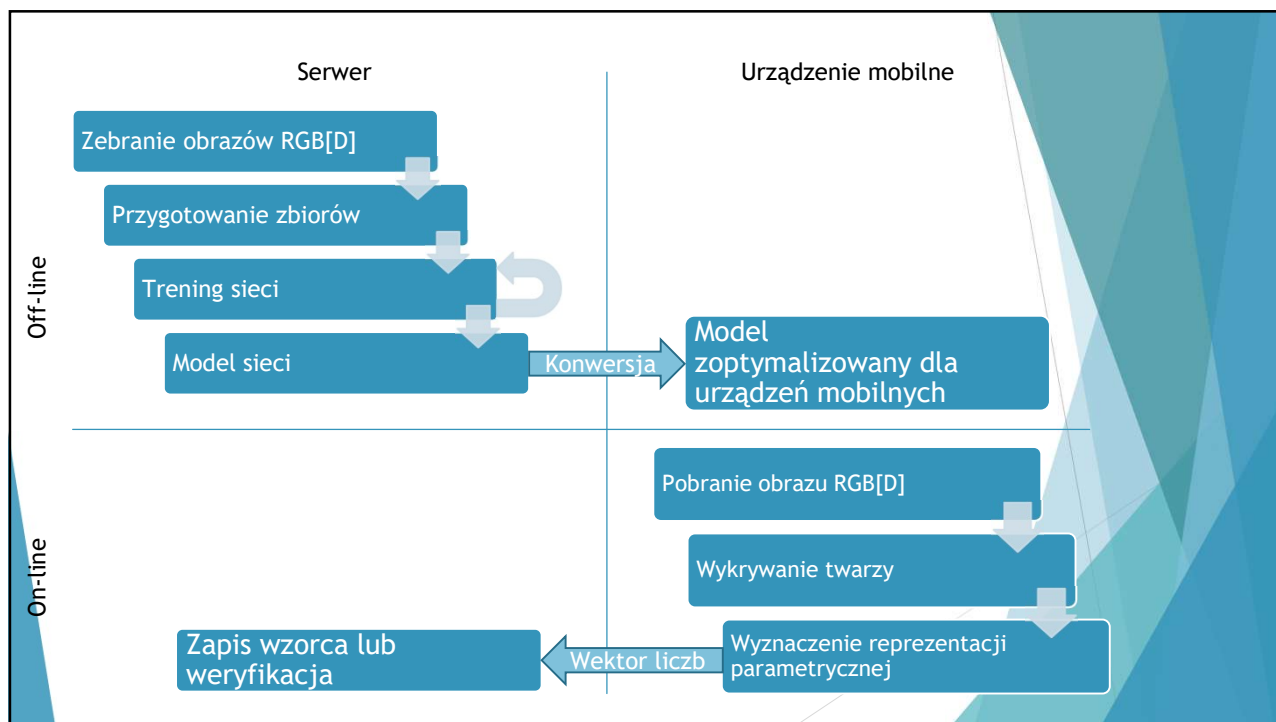
Modele/metody

Model/metoda	LFW Score
Facenet512	99.65%
SFace	99.60%
ArcFace	99.41%
Dlib	99.38 %
Facenet	99.20%
VGG-Face	98.78%
Human-beings	97.53%
OpenFace	93.80%

Architektury sieci

- ResNet-18
- ResNet-27
- ResNet-50
- ResNet-64
- ResNet-100
- AlexNet
- VGGNet-10
- GoogLeNet-24

11



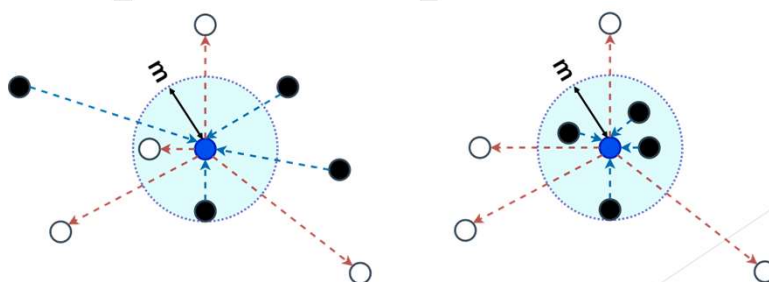
12

Funkcje straty

- Contrastive loss

- Minimalizacja odległości w przestrzeni wektora cech dla obrazów należących do tej samej osoby, i maksymalizacja dystansu dla obrazów należących do różnych osób

$$(1 - Y) \frac{1}{2} (D_W)^2 + (Y) \frac{1}{2} \{ \max(0, m - D_W) \}^2$$



13

Funkcje straty

- Hinge loss

$$Y \cdot \max(0, D_w - p + m) + (1 - Y) \cdot \max(0, p - D_w + m)$$

- Triplet loss

$$\mathcal{L}(A, P, N) = \max(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha, 0)$$

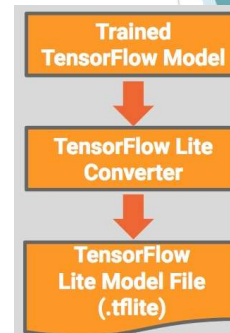


Więcej funkcji strat na <https://www.mdpi.com/2079-9292/9/8/1188/htm>

14

System Android

- Język programowania
 - Java
 - Kotlin
- Wykrywanie twarzy
 - `android.media.FaceDetector`
 - Google ML Kit **FaceDetector**
 - ...
- Zastosowanie uczenia maszynowego
 - Google ML Kit (Object Detection & Tracking API, możliwe konwersje)
 - Tensor Flow (`org.tensorflow:tensorflow-lite`)
 - ...



Więcej na androidrepo.com

15

System Android - wykrywanie twarzy

A lightweight model (224KB in size) for detecting one or multiple faces within an image captured by a smartphone camera or webcam, primarily targeting front-facing camera images. Runs super-real-time (~275FPS on Pixel 2 single-core CPU with [XNNPACK](#) inference).

For each detected face, returns:

- Facial bounding box coordinates
- 6 approximate facial [keypoint](#) coordinates:
 - Left eye (from the observer's point of view)
 - Right eye
 - Nose tip
 - Mouth
 - Left eye tragion
 - Right eye tragion
- Detection confidence score

Model Type

Convolutional Neural Network

Model Architecture

Convolutional Neural Network: [SSD](#)-like with a custom encoder.

Input(s)

RGB image (possibly a video frame) resized to 128x128 pixels, represented as a 128x128x3 array of float values in the range [-1.0, 1.0].

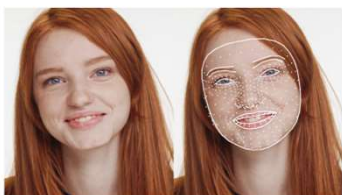
Output(s)

Tensor of predicted embeddings representing anchors transformation which are further used in Non Maximum Suppression algorithm.

16

System Android - oznaczanie punktów szczególnych twarzy

A lightweight model to predict 3D facial surface landmarks from monocular video captured by a front-facing camera on a smartphone in real time.



Left: Input frame. Right: Output face landmarks

Model Type

- Convolutional Neural Network

Model Architecture

- Convolutional Neural Network: [MobileNetV2](#)-like with customized blocks for real-time performance.

Inputs

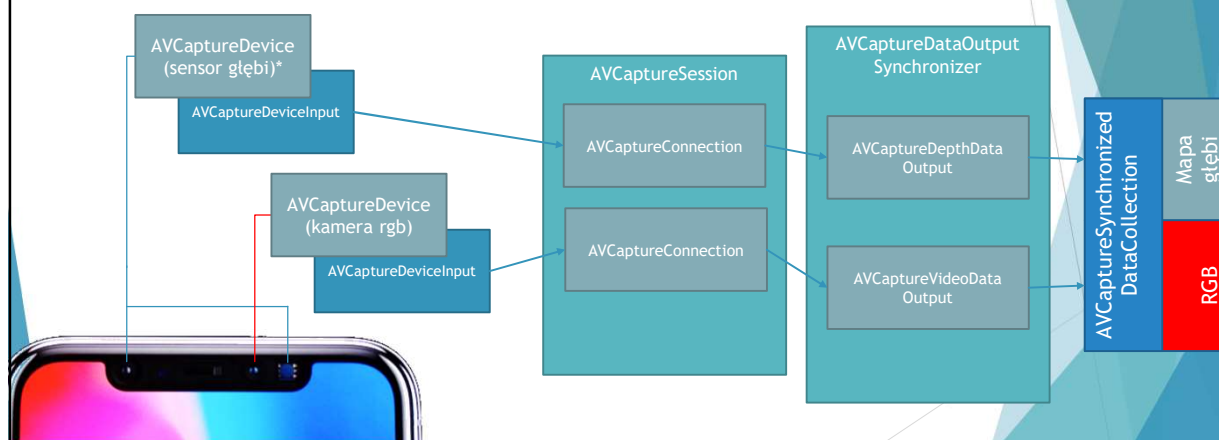
- Image of cropped face with 25% margin on each side, and resized to 256x256.

Output(s)

- Facial surface represented** as 478 3D landmarks flattened into a 1D tensor: (x1, y1, z1), (x2, y2, z2), ... x- and y-coordinates follow the image pixel coordinates; z-coordinates are relative to the face center of mass and are scaled proportionally to the face width.
- Face flag** indicating the likelihood of the face being present in the input image. Used in tracking mode to detect that the face was lost and the face detector should be applied to obtain a new face position. Face probability threshold is set at 0.5 by default and can be adjusted.

17

System iOS

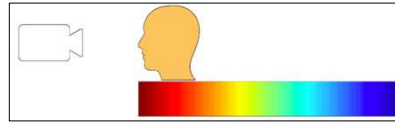


*W przypadku urządzeń z czujnikiem głębi

18

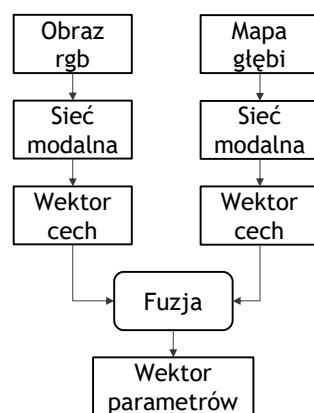
Mapa głębi

Zobrazowanie odległości paletą syntetycznych barw



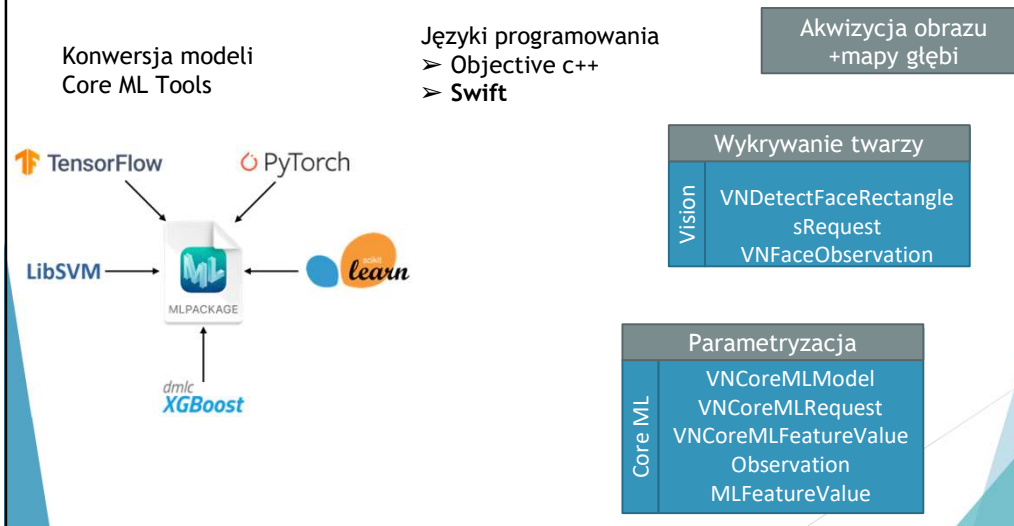
19

Przetwarzanie wielomodalne



20

Komponenty aplikacji



21

Dziękuję za uwagę

➤ Źródła

- [1] Liu, Weiyang, et al. "Sphereface: Deep hypersphere embedding for face recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017
- [2] A. K. Jain, K. Nandakumar, A. Ross, 50 years of biometric research: Accomplishments, challenges, and opportunities, *Pattern Recognition Letters* 79, 2016, 80-105.
- [https://storage.googleapis.com/mediapipe-assets/MediaPipe%20BlazeFace%20Model%20Card%20\(Short%20Range\).pdf](https://storage.googleapis.com/mediapipe-assets/MediaPipe%20BlazeFace%20Model%20Card%20(Short%20Range).pdf)
- [+ Model%20Card%20MediaPipe%20Face%20Mesh%20V2.pdf](#)

22