ROBOTYKA AFEKTYWNA

Wykład 2: Percepcja wizualna — rozpoznawanie stanu afektywnego

Automatyka i Robotyka, studia II stopnia

dr inż. Mateusz Pomianek • Politechnika Rzeszowska

Po tym wykładzie będziesz umiał/a:

- Zbudować pipeline detekcji twarzy i landmarków w czasie rzeczywistym
- Wyjaśnić FACS i Action Units jako pomost między obrazem a emocją
- Opisać, jak CNN klasyfikuje ekspresję (konwolucja $\rightarrow$ softmax)
- Ocenić jakość modelu metrykami accuracy/precision/recall/F1
- Zintegrować detektor emocji z pipeline'em ROS2 z W1



SZYBKIE PRZYWOŁANIE Z W1

Wymień 5 etapów pipeline'u afektywnego. Którymi dwoma zajmiemy się dzisiaj?



Dziś: kanał wizyjny etapów 2-3 — od piksela do etykiety emocji.

Dlaczego to jest trudne?

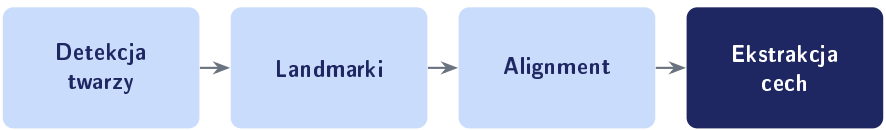
Kamera robota widzi twarz operatora pod kątem 30°, w słabym oświetleniu hali, częściowo zasłoniętą maską ochronną.

Człowiek rozpozna emocję intuicyjnie w ułamku sekundy.

Algorytm musi to zrobić z surowej macierzy pikseli, w $<$50 ms, na sprzęcie edge.



Pipeline percepcji wizualnej



- Detekcja twarzy: lokalizacja bounding-boxa (np. RetinaFace, SSD)
- Landmarki: punkty charakterystyczne (oczy, brwi, usta, kontur)
- Alignment: normalizacja pozy i skali twarzy przed ekstrakcją cech
- Ekstrakcja cech: to jest **etap 2** z pipeline'u W1 — wejście do klasyfikatora

MediaPipe Face Mesh

- 468 punktów 3D na twarzy (x, y, z) — gęsta siatka geometrii twarzy
- Działa w czasie rzeczywistym na CPU: $\sim$10-15 ms na klatkę
- 52 współczynniki blendshape (np. `mouthSmile`, `browDown`) — gotowe cechy ekspresji
- Open-source, biblioteka Google — jeden import w Pythonie

Dlaczego to ważne dla robotyka?

Blendshapes można użyć jako cech wejściowych do prostego klasyfikatora (np. drzewa decyzyjnego) — bez trenowania własnej sieci CNN. To najszybsza droga do działającego proto-

MediaPipe vs Dlib vs OpenFace

Cecha	MediaPipe	Dlib	OpenFace
Liczba punktów	468 (3D)	68 (2D)	68 + AU
Szybkość (CPU)	∼10-15 ms	∼15-25 ms	∼30-50 ms
Wyjście AU	Brak (blendshapes)	Brak	Tak — natywnie
Zastosowanie	Prototypy real-time	Klasyczne CV	Badania nad AU

FACS i Action Units (Ekman, Friesen)

- Facial Action Coding System: opisuje ekspresję jako kombinację niezależnych ruchów mięśni twarzy
- Każdy ruch = Action Unit (AU) z numerem i intensywnością (A–E)
- Emocje podstawowe to konkretne kombinacje AU

AU	Mięsień / ruch	Przykład kombinacji
AU1+AU2	Unoszenie brwi (wewn./zewn.)	Zaskoczenie: AU1+2+5+26
AU4	Marszczenie brwi	Złość: AU4+5+7+23
AU6+AU12	Policzki + kąciki ust w górę	Radość: AU6+12
AU15	Kąciki ust w dół	Smutek: AU1+4+15

Cechy geometryczne vs teksturowe

Geometryczne

Odległości i kąty między punktami landmarków.

$$d(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Np. odległość kącików ust rośnie przy uśmiechu.

Teksturowe

Wzorce jasności pikseli wokół punktów (zmarszczki, fałdy).

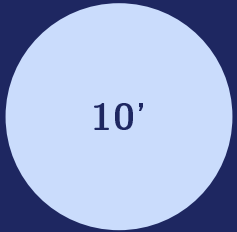
LBP: każdy piksel porównany z sąsiadami → binarny kod wzorca lokalnego

HOG: histogram kie-

SZYBKIE PRZYWOŁANIE

Jaka kombinacja Action Units odpowiada radości? A jaka złości?

30 sekund indywidualnie → porównaj z sąsiadem



PRZERWA

10 minut — wracamy o [uzupełnić godzinę]

Szybkie przypomnienie

- Cechy (geometryczne + teksturowe) $\rightarrow$ wejście klasyfikatora
- FACS/AU to interpretowalny pomost między obrazem a emocją

Teraz: jak sieć CNN sama uczy się takich cech i zamienia je na etykietę emocji?

Architektura CNN — od piksela do etykiety



To jest wnętrze etapu 3 (Klasyfikacja) z pipeline'u W1 — szczegóły matematyczne na następnym slajdzie.

Matematyka: konwolucja, softmax, strata

Konwolucja 2D (jądro K o rozmiarze $m \times n$):

$$(I * K)(x, y) = \sum_i \sum_j I(x + i,\, y + j) \cdot K(i, j)$$

Softmax (zamiana wyjść sieci na rozkład prawdopodobieństwa nad K emocjami):

$$\mathrm{softmax}(z_i) = \frac{e^{z_i}}{\sum_k e^{z_k}}$$

Funkcja straty — cross-entropy (y = etykieta prawdziwa, $\hat{y}$ = predykcja):

$$\mathcal{L} = -\sum_k y_k \cdot \log(\hat{y}_k)$$

Trening = szukanie wag minimalizujących $\mathcal{L}$ na zbiorze treningowym (np. FER2013 — slajd 16).

ResNet vs EfficientNet

ResNet — połączenia skrótowe

$$y = F(x) + x$$

Skip connection omija warstwy konwolucyjne — rozwiązuje problem zanikającego gradientu w głębokich sieciach.

EfficientNet — compound scaling

głębokość, szerokość i rozdzielczość skalowane razem współczynnikiem φ.

Lepszy stosunek dokładności do liczby FLOPS — praktyczny

Zbiory danych do treningu

Zbiór	Rozmiar	Charakterystyka	Wyzwanie
FER2013	∼36 tys., 48×48, szary	7 klas, dane z internetu	Duży szum etykiet (∼10%)
AffectNet	∼1 mln, RGB	8 klas + valence/arousal	Silny bias demograficzny
RAF-DB	∼30 tys., RGB	Etykiety wieloosobowe (crowd)	Mniejsza skala niż AffectNet

Krytyczna refleksja: model wytrenowany na jednym zbiorze często słabo generalizuje na inne oświetlenie, kamerę czy populację operatorów.

Jak ocenić jakość klasyfikatora?

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Dlaczego nie sama accuracy? Klasy emocji są niezbalansowane (np. *disgust* to <2% zbioru FER2013) — model może ignorować rzadkie klasy i wciąż mieć wysoką accuracy. F1 per klasa i macierz pomyłek ujawniają ten problem.

Kod: detekcja twarzy i landmarki (MediaPipe)

```
import cv2
import mediapipe as mp

mp_face = mp.solutions.face_mesh
face_mesh = mp_face.FaceMesh(
    max_num_faces=1,
    refine_landmarks=True)

cap = cv2.VideoCapture(0)
ret, frame = cap.read()
rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
result = face_mesh.process(rgb)
landmarks = result.multi_face_landmarks[0]
```

Kod: inferencja emocji + publikacja w ROS2

```
features = extract_geometric_features(landmarks)
valence, arousal, conf = emotion_model.predict(features)

msg = AffectState(
    valence=valence, arousal=arousal,
    confidence=conf)
self.pub.publish(msg) # -> /affect/state, patrz W1
```

`emotion_model` to lekki klasyfikator (np. drzewo decyzyjne lub MobileNetV3) — wytrenowany offline, uruchamiany w węźle z W1.

Ograniczenia praktyczne

- Oświetlenie: silny kontrast/podświetlenie tylne psuje detekcję landmarków
- Okluzja: maska ochronna, ręka, kask — częściowa utrata cech twarzy
- Kąt kamery: modele trenowane głównie na twarzach en face
- Budżet czasu z W1: ekstrakcja cech musi zmieścić się w $\sim$12 ms marginesu

Kluczowe pojęcia dzisiejszego wykładu

Landmarki (MediaPipe, Dlib)	**FACS / Action Units**	Cechy geometryczne vs teksturowe
Konwolucja i softmax	**ResNet / EfficientNet**	Accuracy, precision, recall, F1

Quiz kontrolny (3 pytania)

1. Jaka jest różnica między cechą geometryczną a teksturową?
2. Model ma 95% accuracy, ale F1 dla klasy *disgust* = 0,1. Co to oznacza?
3. Dlaczego EfficientNet może być lepszym wyborem niż ResNet na Jetsonie z W1?

Wykład 3: Percepcja audio i fuzja multimodalna

- Cechy akustyczne: MFCC, pitch, energia, jitter/shimmer
- Klasyfikatory głosu: SVM na cechach vs wav2vec2 end-to-end
- Fuzja early/late — łączenie kanału wizyjnego (dziś) z audio

Praca własna przed W3:

Nagraj 10-sekundową próbkę głosu i policz jej podstawowe cechy w `librosa` (`pip install librosa`).

Dziękuję za uwagę

Pytania?