

## Concepte și Aplicații în Vederea Artificială - Tema 2

### Detectarea și recunoașterea facială a personajelor din serialul de desene animate *Viața cu Louie*

#### Obiectiv

Scopul acestei teme este implementarea unui sistem automat de detectare și recunoaștere facială a personajelor din serialul de desene animate *Viața cu Louie* (*Life with Louie*) folosind algoritmi de Vedere Artificială discutați la curs și implementați parțial la laborator.

#### Viața cu Louie

*Viața cu Louie* este un serial de desene animate lansat în anul 1994 și care are 3 sezoane, a câte 13 episoade. Serialul prezintă întâmplări din viața lui Louie Anderson, un copil de 8 ani care trăiește în Cedar Knoll, Wisconsin (SUA). Multe episoade sunt inspirate din viața adevăratului Louie Anderson, comedian american care apare la începutul episoadelor și care le vorbește telespectatorilor, acesta regizând, de altfel, și toate episoadele.

#### Task 1 - detectarea facială

Prima problemă pe care o aveți de rezolvat constă în detectarea facială a *tuturor* fețelor personajelor care apar în imagini. Pentru fiecare imagine de intrare algoritmul vostru trebuie să returneze o mulțime de detecții asociate (fereastră dreptunghiulară și scor) ce localizează *toate* fețele dintr-o imagine. Figura 1 arată câteva exemple din mulțimea de antrenare și adnotările corespunzătoare, ce constau în ferestre dreptunghiulare de culoare roșie ce încadrează perfect *fiecare* față.

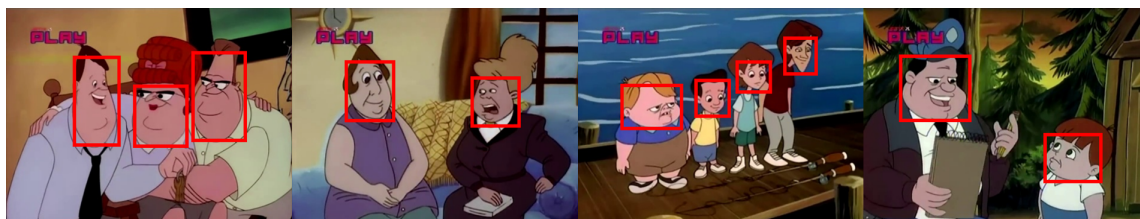


Figura 1: Detectare facială a personajelor din *Viața cu Louie*: fiecare față de interes este adnotată cu o fereastră dreptunghiulară de culoare roșie.

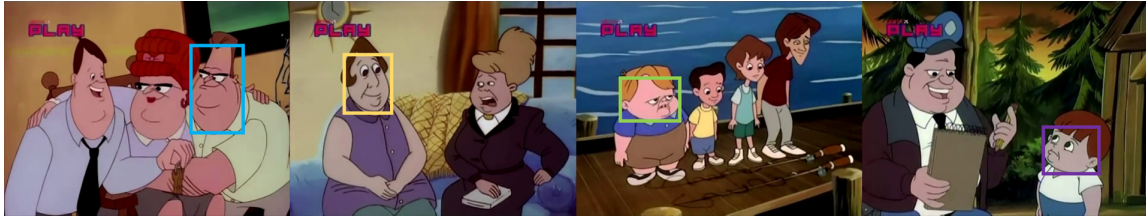


Figura 2: Recunoașterea facială a personajelor din *Viața cu Louie*: fiecare față de interes este adnotată cu o fereastră dreptunghiulară de culoare specifică clasei personajului (albastru - Andy, galben - Ora, verde - Louie, violet - Tommy).

## Task 2 - recunoaștere facială

A doua problemă pe care o aveți de rezolvat constă în recunoașterea facială a numai anumitor personaje. Alături de Louie, personajul principal din serial, apar cu preponderență alte trei personaje: Andy - tatăl său, Ora - mama sa, Tommy - fratele său. Vom considera problema recunoașterii faciale numai pentru aceste patru personaje, Andy, Ora, Louie și Tommy. Pentru fiecare imagine de intrare algoritmul vostru trebuie să returneze o mulțime de detecții asociate (numele personajului, fereastră dreptunghiulară și scor) ce localizează fețele celor patru personaje de interes (Andy, Ora, Louie, Tommy) din imagine. Figura 2 arată câteva exemple din mulțimea de antrenare și adnotările corespunzătoare, ce constau în ferestre dreptunghiulare ce încadrează perfect *fețele de interes* (pentru Andy, Ora, Louie și Tommy). Fiecare detecție are o culoare specifică clasei personajului (albastru - Andy, galben - Ora, verde - Louie, violet - Tommy).

## Descrierea datelor

Arhiva cu materiale (disponibilă aici: <https://tinyurl.com/CAVA-2022-TEMA2>) conține patru directoare: *antrenare*, *validare*, *testare* și *evaluare*.

Directorul *antrenare* conține datele de antrenare. Pentru fiecare din cele patru personaje (Andy, Ora, Louie, Tommy) există un director corespunzător cu 1000 imagini de antrenare. Fiecare imagine conține adnotate toate fețele personajelor care apar în imagine. Fiecărei fețe adnotate îi corespunde o linie din fișierul text corespunzător (Figura 3) și are formatul:

```
nume_imagine xmin ymin xmax ymax nume_personaj
```

unde:

- `nume_imagine` reprezintă numele imaginii din directorul corespunzător personajului;
- `xmin ymin xmax ymax` reprezintă coordonatele ferestrei dreptunghiulare ce încadrează fața în imagine. Colțul din stânga sus are coordonatele (`xmin`, `ymin`) iar colțul din dreapta jos are coordonatele (`xmax`, `ymax`).

andy.txt	louie.txt
1 0001.jpg 265 31 316 142 andy	1 0001.jpg 168 160 283 233 louie
2 0001.jpg 110 192 201 273 louie	2 0002.jpg 338 91 463 165 louie
3 0002.jpg 155 146 196 200 andy	3 0002.jpg 129 214 316 308 andy
4 0003.jpg 76 125 147 242 andy	4 0003.jpg 131 161 238 250 louie
5 0003.jpg 20 41 69 111 unknown	5 0004.jpg 58 158 203 277 louie
6 0003.jpg 340 74 403 146 unknown	6 0005.jpg 156 168 333 294 louie
7 0003.jpg 410 38 464 112 unknown	7 0006.jpg 127 92 331 250 louie
8 0003.jpg 384 132 432 208 unknown	8 0007.jpg 119 76 268 186 louie
9 0004.jpg 113 81 178 152 andy	9 0008.jpg 165 121 354 249 louie
10 0004.jpg 341 126 407 173 louie	10 0009.jpg 148 105 316 238 louie
11 0005.jpg 86 171 224 270 louie	11 0010.jpg 123 150 279 273 louie
12 0005.jpg 329 1 430 178 andy	12 0011.jpg 165 109 336 266 louie
13 0006.jpg 236 64 287 139 andy	13 0012.jpg 80 112 336 296 louie
14 0007.jpg 248 29 347 186 andy	14 0013.jpg 144 172 216 226 louie
15 0008.jpg 177 33 388 327 andy	15 0013.jpg 308 86 363 168 andy
16 0009.jpg 195 100 290 195 andy	16 0014.jpg 147 131 245 215 louie
17 0010.jpg 118 64 224 210 unknown	17 0014.jpg 6 71 69 188 ora
18 0010.jpg 397 68 478 259 andy	18 0015.jpg 306 254 406 326 louie
19 0011.jpg 212 143 348 308 andy	19 0015.jpg 144 85 241 192 unknown
20 0012.jpg 110 62 252 288 andy	20 0016.jpg 89 95 307 252 louie
21 0013.jpg 202 52 311 238 andy	21 0017.jpg 54 189 132 249 louie
22 0014.jpg 255 158 354 317 andy	22 0017.jpg 286 79 356 137 unknown

Figura 3: Fiecărei fețe adnotate din imaginile de antrenare îi corespunde o linie în fișierul text corespunzător cu adnotări ce urmează formatul `nume_imagine xmin ymin xmax ymax nume_personaj`.

- `nume_personaj` reprezintă numele personajului ce are fața adnotată. Folosim pentru adnotarea fețelor cinci clase posibile: *andy*, *ora*, *louie*, *tommy* și *unknown* (pentru toate fețele personajelor diferite de cele patru de interes).

În total sunt 4000 imagini de antrenare ce conțin 7236 fețe adnotate.

Directorul *validare* conține datele de validare. Acestea constau din 200 de imagini adnotate în formatul descris mai sus. Puteți să vă folosiți de aceste date de validare pentru a estima performanțele diverselor voastre soluții.

Directorul *testare* are aceeași structură ca directorul *validare*. Vom face publice datele de test după **prima fază** (detalii mai târziu). Vom evalua performanța algoritmului vostru pe 200 de imagini de test, similare ca distribuție a fețelor cu imaginile de validare.

Directorul *evaluare* vă indică cum să vă scrieți codul astfel încât să respectați formatul fișierelor cu rezultate impus pentru faza de evaluare pe datele de test (**faza a doua**) ce va avea loc după trimiterea codului cu soluția fiecărui student. Conține următoarele sub-directoare:

- *fake\_test* - acest director exemplifică cum vor arăta datele de testare, el păstrează aceeași structură ca cea descrisă pentru directorul *validare* descris anterior. Acest director va fi similar cu directorul *testare* în care vom pune imaginile de testare pentru faza a doua de evaluare.
- *fisiere\_solutie* - acest director exemplifică formatul fișierelor cu rezultatele pe care trebuie să le trimiteți în faza a doua. Veți trimite rezultatele voastre în acest format, încărcând o arhivă zip a unui director similar cu cel numit *331\_Alexe\_Bogdan*;

- *cod\_evaluare* - acest director conține codul care va fi folosit pentru evaluarea automată a rezultatelor voastre folosind adnotările soluțiilor corecte (ferestrele ground-truth). Asigurați-vă că acest cod rulează pe fișierele voastre. Puteți folosi în acest sens datele de validare. Adnotările soluțiilor corecte vor fi disponibile după faza a doua.

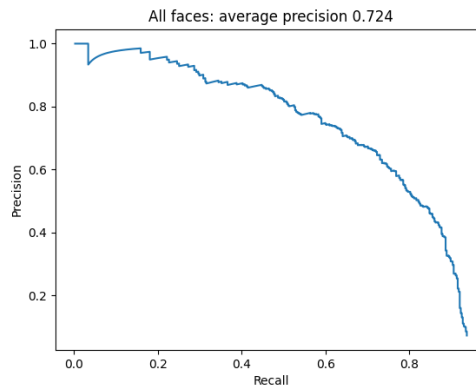


Figura 4: **Grafic precizie-recall pentru task-ul 1.** Performanța unei soluții realizate de noi pentru task-ul 1 pe datele de validare atinge un scor precizie-medie de 0.724.

### Protocolul de evaluare și măsura de performanță

Cuantificăm performanța algoritmilor voștri pentru task-ul 1 de detectare facială și task-ul 2 de recunoaștere facială în imagini test prin grafice de tip precizie-recall. Aceste grafice combină două valori:

- *precizia*: procentul de detecții returnate de algoritmul vostru ca fiind corecte (ele conțin o față de interes). În cazul ideal, algoritmul vostru are o precizie = 1 = 100%, adică fiecare detecție furnizată de algoritm reprezintă o față de interes. Pentru task-ul 1 toate fețele din imagine sunt fețe de interes, pentru task-ul 2 numai fețele unui personaj specific (Andy, Ora, Louie sau Tommy) sunt de interes.
- *recall* (= rată de detectare): procentul de fețe de interes din imaginile test localizate corect. În cazul ideal algoritmul vostru are un recall = 1 = 100%, adică localizează corect toate fețele de interes din imagine.

Fiecare punct de pe graficul precizie-recall reprezintă precizia și recall-ul algoritmului vostru obținute pentru toate detecțiile (ordonate descrescător după scor) care depășesc un anumit scor prag (threshold). De aceea, este foarte important ca atât pentru task-ul 1 cât și pentru task-ul 2 să aveți un scor asociat detecțiilor voastre. Sumarizăm întregul grafic prin *precizia medie* care reprezintă aria de sub grafic. Funcțiile *eval\_detections* (pentru task-ul 1) și *eval\_detections\_character* (pentru task-ul 2) realizează aceste grafice și calculează precizia medie. Este important de reținut că fără un scor asociat pentru detecții nu putem calcula aria de sub graficul precizie-recall întrucât în acest caz graficul va fi reprezentat de un singur punct.

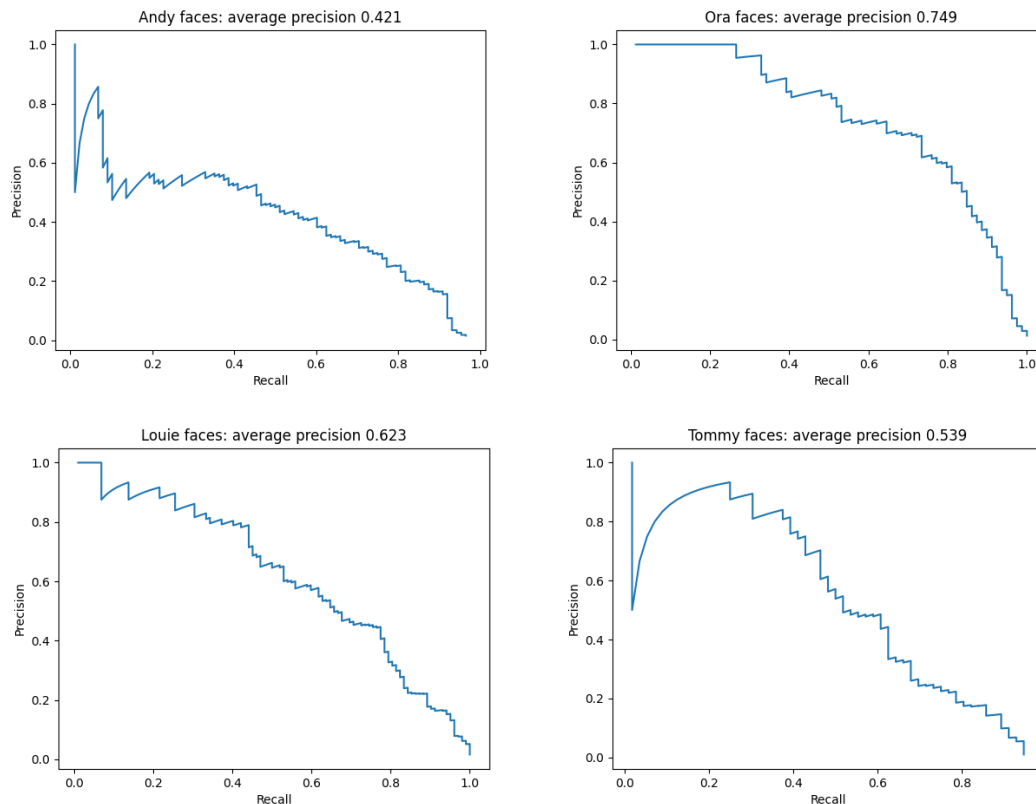


Figura 5: **Grafic precizie-recall pentru task-ul 2.** Performanța unei soluții realizate de noi pentru task-ul 2 pe datele de validare atinge un scor precizie-medie de 0.421 pentru Andy, 0.749 pentru Ora, 0.623 pentru Louie și 0.539 pentru Tommy.

Figura 4 ilustrează graficul precizie-recall și precizia medie = 0.724 pentru task-ul 1 pe setul de date de validare obținută de una din soluțiile noastre. Un detector facial perfect ar avea precizia medie = 1, cel mai din dreapta punct de pe graficul funcției aflându-se în acest caz în colțul din dreapta sus.

Figura 5 ilustrează graficele precizie-recall pentru fiecare personaj în parte pentru task-ul 2 pe setul de date de validare obținute de una din soluțiile noastre.

### Restricții în implementarea soluției

Implementarea voastră trebuie să urmeze paradigma de glisare a unei ferestre (sliding-window), extragerea de caracteristici pentru fiecare fereastră și folosirea unui clasificator. Puteți folosi o paletă întreagă de caracteristici: histograme de gradienti orientați, caracteristici obținute folosind rețele convoluționale, caracteristici bazate pe culoare, etc.

Nu puteți folosi algoritmi de tipul Faster-RCNN sau YOLO sau rețele pre-antrenate pe fețe care v-ar ușura rezolvarea problemei. Dacă există neclarități ne puteți întreba dacă aveți sau nu voie să folosiți anumite lucruri.

## Notare

Pentru această temă, vom folosi următoarele reguli de notare:

- **Task 1 - 4 puncte** - vom evalua performanța algoritmului vostru în problema detectării faciale pe o mulțime de 200 imagini de testare. Vom folosi ca măsură de performanță precizie-medie (Average Precision = AP) obținută pe baza graficului precizie-recall. Vom calcula punctajul corespunzător soluției voastre astfel:
  - o soluție care atinge un prag de  $x\%$  AP va obține  $\frac{4 \cdot x}{80}$  puncte. Practic acordăm 4 puncte pentru soluțiile care ating un prag de 80% AP.
  - soluția propusă de noi (Figura 4) ar obține  $\frac{4 \cdot 72.4}{80} = 3.62$  puncte.
  - o soluție care atinge un prag de 40% AP va obține 2 puncte.
  - o soluție care depășește pragul de 80% AP poate obține maxim 1 punct bonus în plus. Astfel, dacă ați avea o soluție perfectă cu 100% AP veți obține 5 puncte (1 punct bonus în plus).
- **Task 2 - 4 puncte** - vom evalua performanța algoritmului vostru în problema recunoașterii faciale pe aceeași mulțime de 200 imagini de testare. Vom folosi ca măsură de performanță media preciziilor-medii (mean Average Precision = mAP) obținută pe baza celor patru grafice precizie-recall. Vom calcula punctajul corespunzător soluției voastre astfel:
  - o soluție care atinge un prag de  $x\%$  mAP va obține  $\frac{4 \cdot x}{60}$  puncte. Practic acordăm 4 puncte pentru soluțiile care ating un prag de 60% mAP.
  - soluția propusă de noi (Figura 5) ar obține un  $mAP = (0.421 + 0.749 + 0.623 + 0.539) / 4 = 0.583$  care ar însemna  $\frac{4 \cdot 58.3}{60} = 3.88$  puncte.
  - o soluție care atinge un prag de 30% mAP va obține 2 puncte.
  - o soluție care depășește pragul de 60% mAP poate obține maxim 1 punct bonus în plus. Astfel, dacă ați avea o soluție cu mAP de cel puțin 75% mAP veți obține 5 puncte (1 punct bonus în plus).
- **documentație - 1 punct** - descrieți într-un fișier pdf de minim două pagini soluția voastră pentru rezolvarea celor două task-uri. Puteți ilustra aspecte cheie ale soluției voastre adăugând secvențe de cod și vizualizări ale imaginilor pentru soluția voastră. Acest fișier ar trebui să conțină suficientă informație astfel încât un student de nivel mediu de la cursul nostru să poată reimplementa soluția descrisă de voi.
- **oficiu - 1 punct** - primiți acest punct dacă formatul fișierelor voastre urmează formatul impus iar codul nostru de evaluare rulează pe datele primite de la voi fără a face modificări în fișierele voastre;
- **BONUS - 2 puncte** - puteți primi adițional 1 punct în plus pentru fiecare task, dacă depășiți pragurile de 80% AP și 60% mAP descrise mai înainte.

### **Termene limită**

**Prima fază - trimiterea codului.** Încărcați o arhiva zip cu codul soluției voastre și un fișier pdf ce descrie soluția voastră până duminică, 22 ianuarie, ora 23:59 la link-ul acesta <https://tinyurl.com/CAVA-2022-TEMA2-SOLUTII>. Codul vostru ar trebui să includă un fișier README (vedeți exemplul din materiale) cu următoarele informații: (i) librăriile folosite de voi și necesare pentru rularea soluției voastre; (ii) indicații despre cum ar trebui rulat codul pentru fiecare task. Studenții care nu încarcă un fișier pdf cu descrierea soluției lor vor avea nota scăzută cu 1 punct. Termenul limită este strict, nu vom accepta soluții primite mai târziu. **Implementarea voastră trebuie să fie gata de rulat pe setul de testare, astfel includeți toate fișierele ce conțin modele, descriptori etc. în arhiva voastră pentru a putea fi încărcate direct.**

**A doua fază - trimiterea rezultatelor.** Luni, 23 ianuarie, vom publica datele de test în directorul *testare* de la adresa <https://tinyurl.com/CAVA-2022-TEMA2>. Veți rula soluția voastră pe imaginile de test și veți încărca rezultatele în aceeași zi ca o arhiva zip folosind următorul link <https://tinyurl.com/CAVA-2022-TEMA2-REZULTATE> în formatul recomandat.