

Concepte și Aplicații în Vederea Artificială - Tema 2

Detectarea și recunoașterea facială a personajelor din serialul de desene animate Familia Flintstone

Obiectiv

Scopul acestei teme este implementarea unui sistem automat de detectare și recunoaștere facială a personajelor din serialul de desene animate Familia Flintstone folosind algoritmi de Vedere Artificială discutați la curs și implementați parțial la laborator.

Familia Flintstone

Familia Flintstone (titlul original în engleză *The Flintstones*) sau *Aventuri în epoca de piatră* este un desen animat clasic american creat de Hanna-Barbera Productions și care a rulat la începutul anilor 1960 la televiziunea ABC. Acțiunea se petrece în orașul preistoric Bedrock și se concentrează pe viața unei familii fictive din epoca de piatră. Personajele principale sunt Fred Flintstone, un operator la cariera de piatră, soția sa Wilma, vecinii lor Barney și Betty Rubble, precum și copiii lor, Pebbles și Bamm-Bamm. Serialul explorează umorul și satira preistorică, adesea parodiind viața cotidiană a anilor '60 prin prisma unei lumi primitive. Familia Flintstone a devenit rapid un fenomen cultural și a avut un impact semnificativ asupra industriei animației fiind cunoscut pentru umorul său inteligent și personajele memorabile.

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

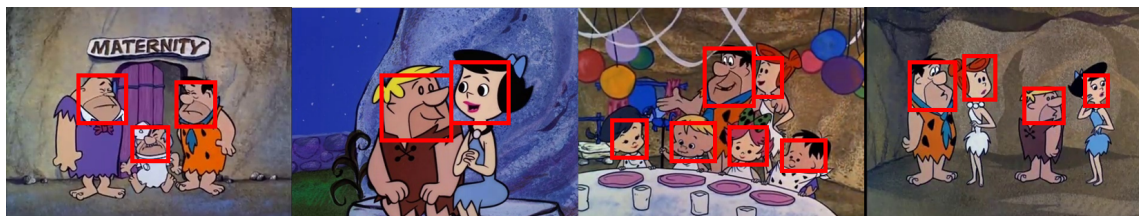


Figura 1: Detectare facială a personajelor din Familia Flintstone: fiecare față de interes este adnotată cu o fereastră dreptunghiulară de culoare roșie.



Figura 2: Recunoașterea facială a personajelor din Familia Flinstone: fiecare față de interes este adnotată cu o fereastră dreptunghiulară de culoare specifică clasei personajului (albastru - Fred, galben - Wilma, verde - Barney, violet - Betty).

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*.

Task 2 - recunoaștere facială

A doua problemă pe care o aveți de rezolvat constă în recunoașterea facială a numai anumitor personaje. Alături de Fred, personajul principal din serial, apar cu preponderență alte trei personaje: Wilma - soția lui Fred, Barney - prietenul și vecinul lui Fred, Betty - soția lui Barney. Vom considera problema recunoașterii faciale numai pentru aceste patru personaje, Fred, Wilma, Barney și Betty. 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 (Fred, Wilma, Barney, Betty) 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 Fred, Wilma, Barney și Betty). Fiecare detecție are o culoare specifică clasei personajului (albastru - Fred, galben - Wilma, verde - Barney, violet - Betty).

Descrierea datelor

Arhiva cu materiale (disponibilă aici: <https://tinyurl.com/CAVA-2023-TEMA2>) conține patru directoare: *antrenare*, *validare*, *testare* și *evaluare*.

Directorul *antrenare* conține datele de antrenare. Pentru fiecare din cele patru personaje (Fred, Wilma, Barney, Betty) 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;

1	0001.jpg	167	293	234	354	fred
2	0002.jpg	167	168	233	230	fred
3	0003.jpg	162	51	229	131	fred
4	0004.jpg	164	280	231	348	fred
5	0005.jpg	164	297	229	359	fred
6	0006.jpg	165	154	234	220	fred
7	0007.jpg	162	138	230	220	fred
8	0008.jpg	161	200	231	280	fred
9	0009.jpg	128	245	200	333	fred
10	0010.jpg	129	175	201	265	fred
11	0011.jpg	116	91	334	316	fred
12	0012.jpg	115	83	320	310	fred
13	0013.jpg	110	77	316	312	fred
14	0014.jpg	224	74	301	159	fred
15	0015.jpg	146	107	270	243	fred
16	0016.jpg	138	106	273	248	fred
17	0017.jpg	136	108	272	245	fred
18	0018.jpg	145	109	271	245	fred
19	0019.jpg	154	110	282	246	fred
20	0020.jpg	152	113	283	248	fred
21	0021.jpg	156	119	323	261	fred
22	0022.jpg	157	118	320	264	fred
23	0023.jpg	106	81	314	316	fred
24	0024.jpg	103	66	317	315	fred
25	0025.jpg	106	87	314	323	fred

1	0001.jpg	177	110	291	246	wilma
2	0002.jpg	102	110	218	239	unknown
3	0002.jpg	237	104	335	222	wilma
4	0003.jpg	103	113	217	239	unknown
5	0003.jpg	237	100	325	226	wilma
6	0004.jpg	103	116	221	247	unknown
7	0004.jpg	245	95	340	214	wilma
8	0005.jpg	116	114	227	230	unknown
9	0005.jpg	231	108	324	214	wilma
10	0006.jpg	102	113	216	234	unknown
11	0006.jpg	242	104	329	217	wilma
12	0007.jpg	316	89	357	149	wilma
13	0008.jpg	121	86	173	144	wilma
14	0009.jpg	278	129	316	168	unknown
15	0009.jpg	103	113	165	182	fred
16	0009.jpg	27	150	70	188	wilma
17	0009.jpg	17	189	49	218	unknown
18	0010.jpg	268	107	334	207	wilma
19	0011.jpg	89	198	200	292	fred
20	0011.jpg	262	98	351	205	wilma
21	0012.jpg	263	99	354	210	wilma
22	0013.jpg	267	110	355	213	wilma
23	0014.jpg	269	106	336	212	wilma
24	0015.jpg	322	131	354	167	unknown
25	0015.jpg	105	111	167	183	fred

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`.

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

În total sunt 4000 imagini de antrenare ce conțin 6977 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.

- *fișiere_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.

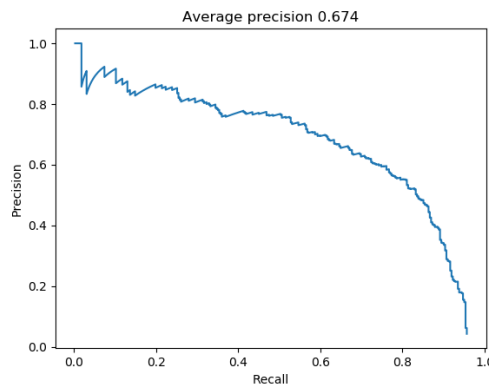


Figura 4: **Grafic precizie-recall pentru task-ul 1.** Performanța unei soluții realizate de noi pentru task-ul 1 pe datele de validare atinge un scor precizie-medie de 0.674.

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 (Fred, Wilma, Barney sau Betty) 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

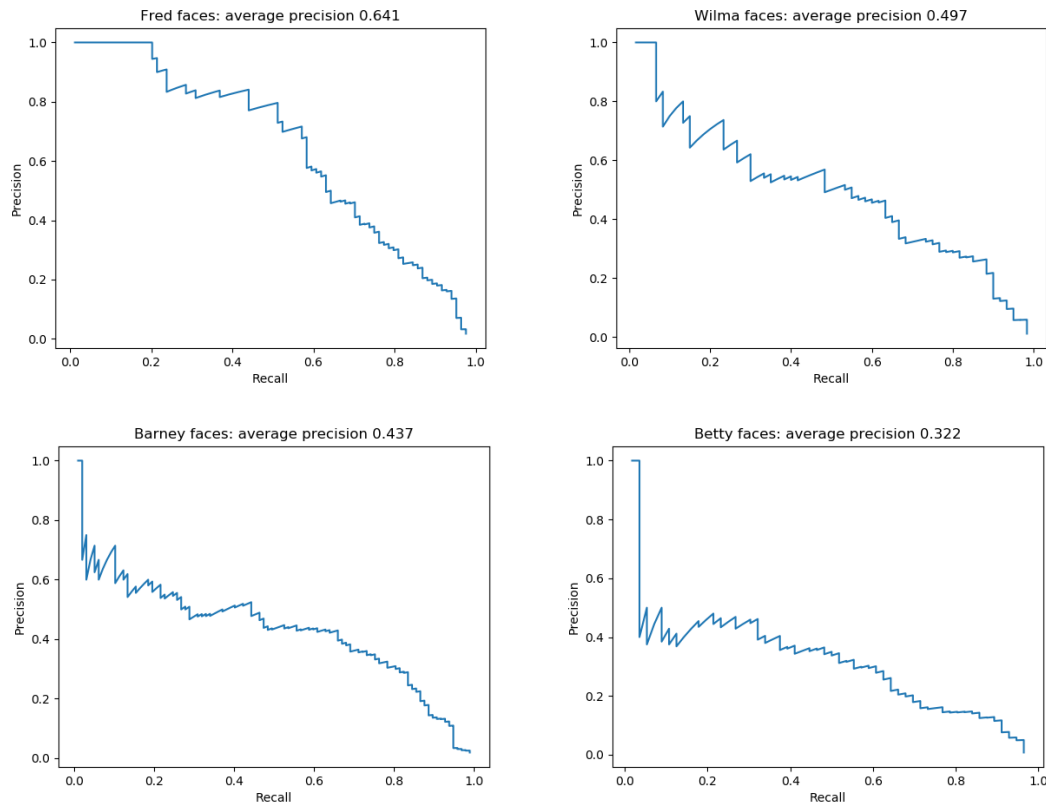


Figura 5: **Grafic precizie-recall pentru task-ul 2.** Performanța unei soluții realizate de noi pentru task-ul 2 pe datele de validare atinge un scor precizie-medie de 0.641 pentru Fred, 0.497 pentru Wilma, 0.437 pentru Barney și 0.322 pentru Betty.

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 4 ilustrează graficul precizie-recall și precizia medie = 0.674 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 pentru soluția de bază 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 pentru soluția de bază algoritmi de tipul Faster-RCNN sau YOLO sau rețele pre-antrenate pe fețe care v-ar ușura rezolvarea problemei. Îi puteți folosi pentru a lua puncte bonus. 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 $\frac{4 \cdot x}{75}$ puncte. Practic acordăm 4 puncte pentru soluțiile care ating un prag de 75% AP.
 - soluția propusă de noi (Figura 4) ar obține $\frac{4 \cdot 67.4}{75} = 3.58$ puncte.
 - o soluție care atinge un prag de 37.5% AP va obține 2 puncte.
 - o soluție care depășește pragul de 75% AP va obține 4 puncte, nu se iau puncte bonus aici.
- **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 ar obține un $mAP = (0.641 + 0.497 + 0.437 + 0.322) / 4 = 0.474$ (Figura 5) care ar însemna $\frac{4 \cdot 47.4}{60} = 3.16$ puncte.
 - o soluție care atinge un prag de 30% mAP va obține 2 puncte.
 - o soluție care depășește pragul de 60% mAP va obține 4 puncte, nu se iau puncte bonus aici.

- **BONUS - 25% din punctajul pentru fiecare task** - vă puteți crește punctajul cu 25% pentru fiecare task, dacă folosiți o soluție secundară (un detector *state-of-the-art* de obiecte/fețe) de tip Faster-RCNN sau YOLO, pe care să o adaptați la problema voastră și a cărei performanță să depășească pragul impus pentru fiecare task (75%AP pentru task-ul 1 și 60% mAP pentru task-ul 2). Astfel, dacă obțineți cu soluția de bază 2 puncte la un task, puteți crește punctajul la 2,5 puncte condiționat de faptul că adaptați un detector pentru task-ul vostru și reușiți să depășiți pragul menționat.
- **documentație - 1 punct** - descrieți într-un fișier pdf de minim două pagini soluția voastră de bază 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. Dacă folosiți și o soluție secundară pentru bonus includeți detalii despre aceasta în documentație.
- **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;

Termene limită

Prima fază - trimiterea codului. Încărcați o arhivă zip cu codul soluției voastre și un fișier pdf ce descrie soluția voastră până duminică, 21 ianuarie 2024, ora 23:59 la link-ul acesta <https://tinyurl.com/CAVA-2023-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 arhivă voastră pentru a putea fi încărcate direct.**

A doua fază - trimiterea rezultatelor. Luni, 22 ianuarie, vom publica datele de test în directorul *testare* de la adresa <https://tinyurl.com/CAVA-2023-TEMA2>. Veți rula soluția voastră pe imaginile de test și veți încărca rezultatele în aceeași zi ca o arhivă zip folosind următorul link <https://tinyurl.com/CAVA-2023-TEMA2-REZULTATE> în formatul recomandat.