Documentatie Tema 2 - Detectarea si recunoasterea faciala a personajelor din Scooby-Doo

Mădălin Ioana

Ianuarie 2026

1 Introducere

Aceasta documentatie descrie solutia implementata pentru Tema 2 din cadrul cursului *Concepte si Aplicatii in Vederea Artificiala*, avand ca obiectiv detectarea si recunoasterea faciala a personajelor din serialul animat *Scooby-Doo*.

Problema este impartita in doua task-uri:

- **Task 1**: Detectarea tuturor fetelor din imagini
- **Task 2**: Recunoasterea faciala pentru personajele Fred, Daphne, Shaggy si Velma

Solutia propusa utilizeaza o abordare bazata pe **sliding window**, extragerea de trasaturi folosind o retea neuronala convolutionala si clasificare realizata cu PyTorch.

2 Structura proiectului

Proiectul este implementat intr-un singur fisier principal:

- `cnn.py` – antrenare, inferenta si salvarea rezultatelor

Codul este conceput pentru a rula pe platforma Kaggle, folosind GPU.

3 Descrierea datelor

Datele sunt organizate conform structurii impuse in tema:

- `antrenare/` – imagini si fisiere de adnotari pentru fiecare personaj
- `validare/` – imagini pentru evaluare

Adnotarile au formatul:

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

Clasele considerate sunt:

- Fred
- Daphne
- Shaggy
- Velma

4 Arhitectura retelei neuronale

Modelul folosit este o retea neuronala multi-task construita peste un backbone `ResNet18` (antrenat de la zero).

4.1 Structura generala

Reteaua contine:

- Backbone convolutional (ResNet18 fara stratul fully-connected)
- Straturi comune fully-connected
- Trei capete de iesire:
  - clasificare fata / non-fata
  - clasificare personaj
  - regresie bounding box

```
self.cls_fata = nn.Linear(512, 1)
self.cls_personaj = nn.Linear(512, 4)
self.reg_cutie = nn.Linear(512, 4)
```

5 Pregatirea datelor

5.1 Exemple pozitive

Pentru fiecare fata adnotata:

- se extrage un crop cu jitter aleator
- se calculeaza coordonatele relative fata de fereastra

5.2 Exemple negative

Ferestre negative sunt extrase aleator din fundalul imaginilor, folosind dimensiuni predefinite (ancore).

Raportul pozitiv : negativ este aproximativ **1 : 3**.

6 Augmentari

Pentru a creste robustetea modelului, sunt aplicate urmatoarele augmentari:

- flip orizontal
- rotatii mici
- variatii de luminozitate, contrast si saturatie

```
T.RandomHorizontalFlip(),
T.RandomRotation(10),
T.ColorJitter(...)
```

7 Functiile de pierdere

Functia de pierdere totala este compusa din:

- **Binary Cross-Entropy** pentru detectarea fetei
- **Cross-Entropy** pentru clasificarea personajului
- **Smooth L1 Loss** pentru regresia bounding box-ului

Regresia este aplicata doar pentru ferestrele etichetate ca fata.

8 Antrenarea modelului

Modelul este antrenat timp de **25 epoci** folosind:

- Optimizer: Adam
- Learning rate initial: 0.0005
- Scheduler: StepLR

9 Inferenta – Sliding Window

Pentru fiecare imagine:

- se folosesc mai multe scale
- se aplica ferestre glisante cu pas de 16 pixeli
- fiecare fereastra este clasificata de retea

Ferestrele cu scor de fata peste pragul 0.75 sunt pastrate.

10 Non-Maximum Suppression

Pentru eliminarea detectiilor redundante se aplica **NMS** cu prag IoU = 0.3:

```
pastreaza = nms(t_boxes, t_scores, 0.3)
```

11 Task 1 – Detectarea faciala

Pentru Task 1 sunt salvate:

- bounding box-uri
- scoruri de detectie
- numele imaginilor

Fisierele rezultate:

```
detections_all_faces.npy
scores_all_faces.npy
file_names_all_faces.npy
```

12 Task 2 – Recunoasterea faciala

Pentru fiecare personaj se genereaza fisiere separate. Scorul final este produsul dintre:

- probabilitatea de fata
- probabilitatea personajului

13 Rulare

Scriptul poate fi rulat direct pe Kaggle:

```
python kaggle_solutin.py
```

Rezultatele sunt salvate automat in directoarele `task1/` si `task2/`.

14 Bonus – Implementare YOLO

Pentru obtinerea punctajului bonus, a fost implementata si o solutie secundara utilizand modelul **YOLOv8** (biblioteca `ultralytics`). Aceasta foloseste transfer learning pe modelul `yolov8n.pt` si obtine performante superioare fata de solutia de baza, fiind antrenata pe datele convertite in formatul specific YOLO. Datele au fost transformate din format `(xmin, ymin, xmax, ymax)` in format YOLO `(x_center, y_center, width, height)` normalizat, iar antrenarea a fost realizata pe 15 epoci cu dimensiune de imagine 640x640. Solutia elimina necesitatea sliding window-ului si NMS manual, acestea fiind integrate in arhitectura, rezultand un pipeline mai eficient.

15 Concluzii

Solutia propusa respecta cerintele temei si implementeaza o abordare completa de detectare si recunoastere faciala folosind sliding window si CNN. Abordarea este robusta si generalizeaza bine pe datele de validare.