

Detectarea si recunoasterea faciala a personajelor din serialul de desene animate Familia Flinstone

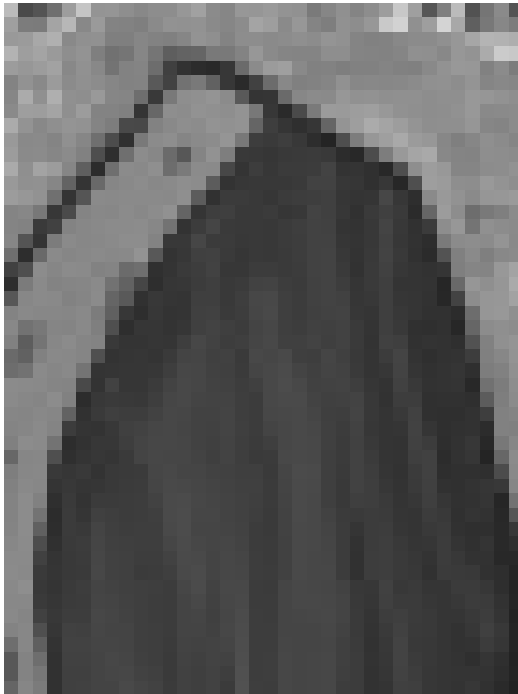
Task 1(identificarea fetelor)

Extragerea exemplelor

Pentru extragerea exemplelor pozitive am parcurs imaginile de antrenare si am extras fetele personajelor folosindu-ma de anotari.

Pentru exemplele negative am extras imagini random care sa aiba intersectia cu un bounding box mai mica ca 0.05. La acestea am mai adaugat exemple obtinute folosind hard mining. Am testat modelul antrenat pe imaginile de antrenare si am selectat detectiile care aveau un scor mai mare ca 3, intersection over union mai mic ca 0.15 cu orice bounding box si care nu se aflau in interiorul unui bounding box.

Exemplu imagine fals pozitiva:



Extragerea de caracteristici

Ca si caracteristici am folosit histograme de gradienti orientati(hog). Pentru a extrage aceste caracteristici am folosit functia hog din OpenCV cu urmatoorii parametri:

```
features = hog(img, pixels_per_cell=(self.params.dim_hog_cell, self.params.dim_hog_cell),  
cells_per_block=(2, 2), feature_vector=True)
```

Dimensiunea celulelor hog a fost de 6. Pentru exemplele pozitive am extras caracteristici si din imaginea oglindita.

Dimensiunile finale:

Positive Features Shape: (13954, 900)

Negative Features Shape: (72283, 900)

Antrenare

Pentru antrenare am folosit un SVM liniar. Pentru a determina parametrul C verificam ce clasificator are cel mai bun scor pe datele de antrenare.

Lista valori incercate:

Cs = [10 ** -5, 10 ** -4, 10 ** -3, 10 ** -2, 10 ** -1,1]

Testare

Pentru testare am incarcat cel mai bun calsificator pe care l-am salvat la pasul de antrenare. Apoi am redimensionat imaginea(width=scale*width,height=sclae*height).

Lista valori:

scales=[1. ,0.87055, 0.75785828, 0.66069345, 0.57622779, 0.50270779, 0.43855988, 0.38239679, 0.33287077, 0.28877435, 0.2499512, 0.21516408, 0.18401213, 0.15607649, 0.13190697]

Ulterior am extras toate patch-urile de 36X36 si am verificat scorul acestora.

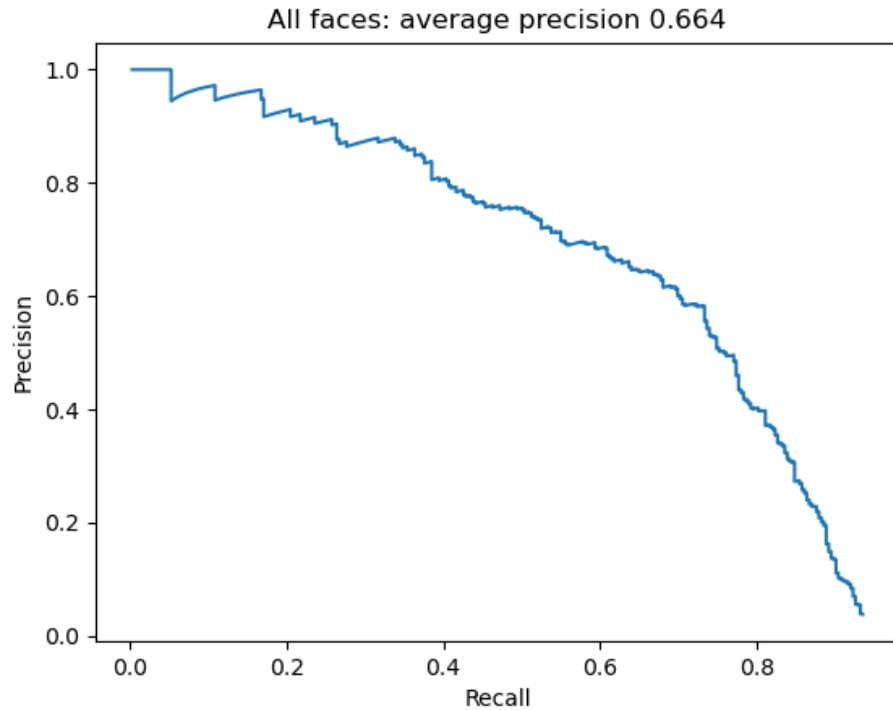
Calcularea scorului:

```
w = self.best_model.coef_.T
bias = self.best_model.intercept_[0]

for y in range(0, num_rows - num_cell_in_template_height):
    for x in range(0, num_cols - num_cell_in_template_width):
        descr = hog_descriptors[y:y + num_cell_in_template_height, x:x + num_cell_in_template_width].flatten()
        score = np.dot(descr, w)[0] + bias
```

Dupa ce am obtinut bounding box-ul pentru dimensiunea respectiva, l-am adus la dimensiunea din imaginea originala. Dupa ce am obtinut detectii pentru fiecare scale, am aplicat non maximal suppression.

Am considerat toate detectiile cu un scor mai mare de 0 ca fiind pozitive si am obtinut urmatorul rezultat:



Task 2(identificarea personajelor)

Extragerea exemplilor

Pentru Barney am extras imagini de dimensiunea 36X36, iar pentru restul 36X48. Am luat aceste valori deoarece cei 3 din urma au fetele mai dreptunghiulare.

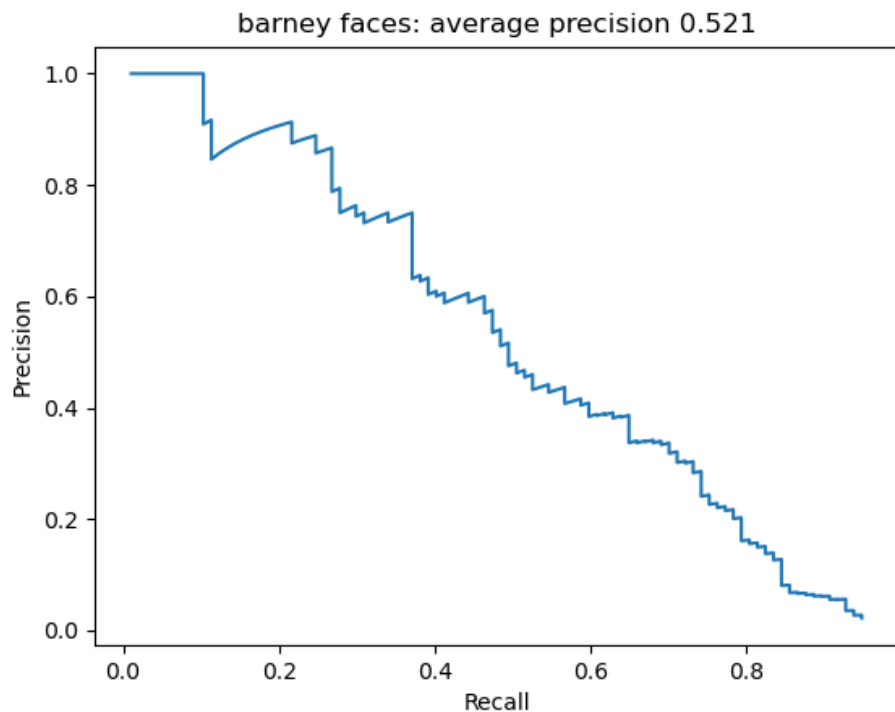
Pentru exemplele negative am adaugat la cele extrase la prima etapa si imagini cu personaje considerate negative pentru clasificatorul respectiv. Pentru personajele cu dimensiuni noi am reluat partea de extragere de exemple negative de la primul task.

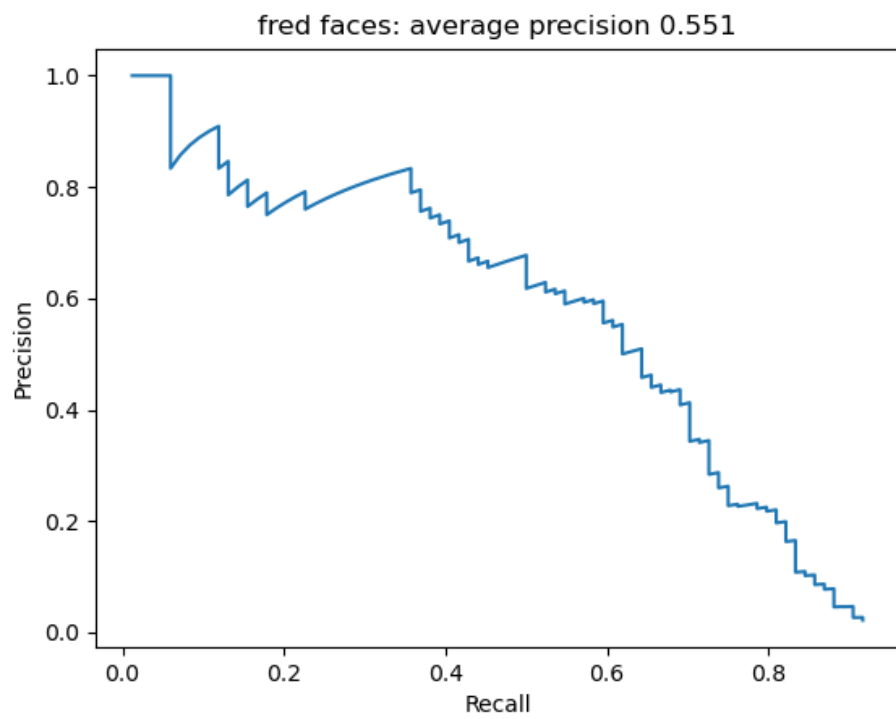
Extragerea de caracteristici

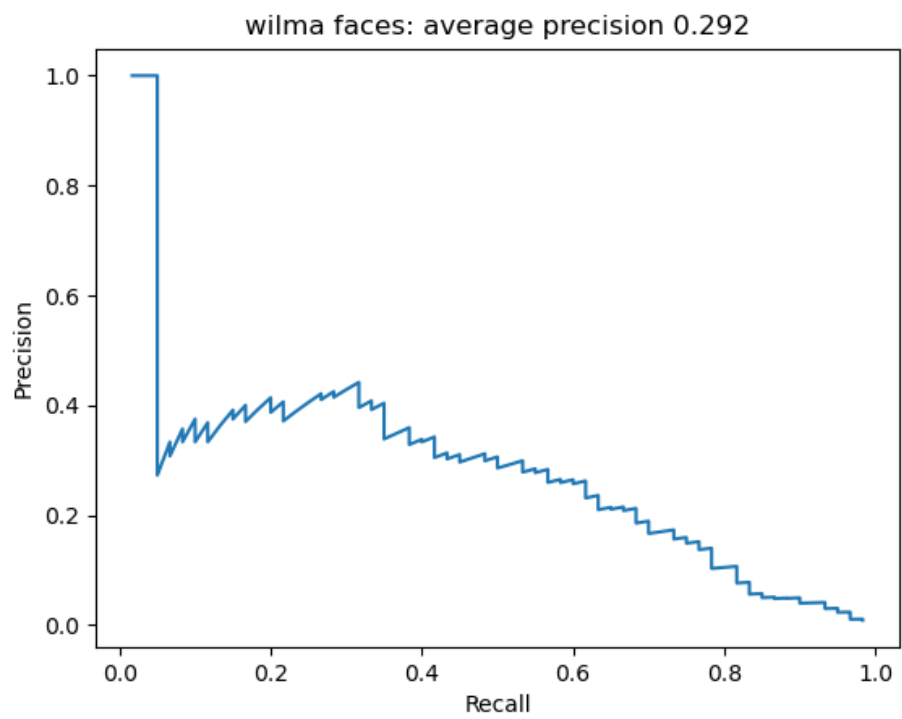
Am folosit aceiasi parametri ca la primul task, doar ca la Wilma si Betty am pus dimensiunea celulei hog valoarea 4 pentru a captura mai bine detaliile.

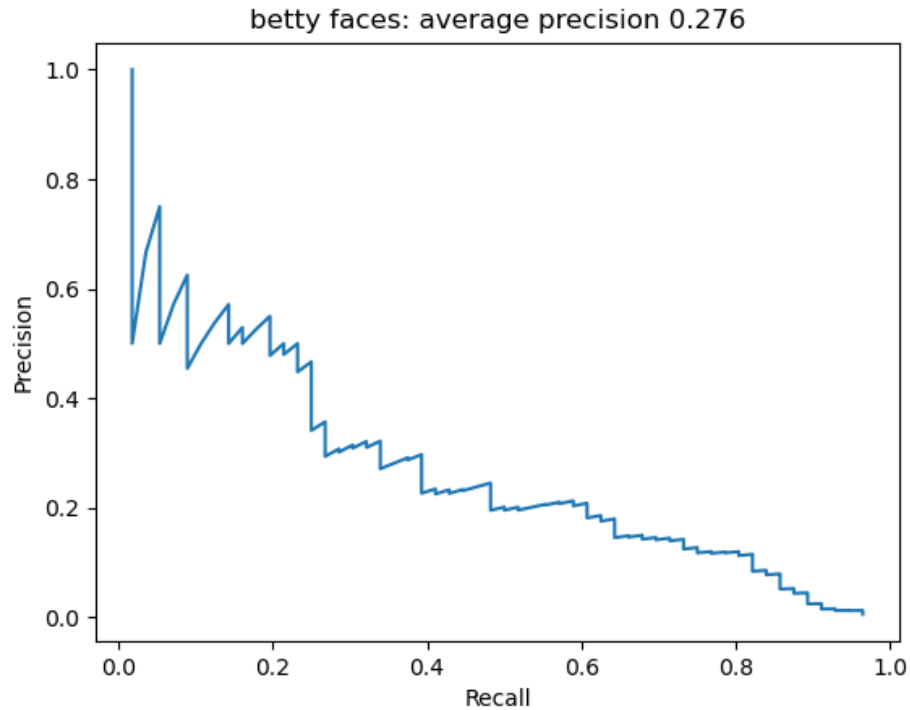
Antrenare si Testare

Fiecare personaj are clasificatorul ei dedicat antrenat si testat ca la prima cerinta. Am obtinut urmatoarele rezultate:









Bonus

Antrenare

Am folosit YOLOv4 pentru a completa cele 2 task-uri. Am urmat pasii de pe urmatorul github pentru instalare si antrenare :github.

Pentru task-ul 2 am antrenat modelul apoximativ 1700 de iteratii pana la un average loss de 0.8 pe setul de antrenare.

Pentru task-ul 1 am folosit weight-urile obtinute mai devreme si am antrenat apoximativ 1200 de iteratii pana la un average loss de 1.3 pe setul de antrenare.

Am obtinut 100% pentru ambele cerinte.

Am antrenat acest model folosind google colab pentru un GPU mai puternic.

Testare

Am incarcat modelul YOLO cu urmatoarea functie:

```
net = cv.dnn.readNet("C:/Users/rares/Downloads/yolov4-flintstone-1_1000.weights",  
"C:/Users/rares/darknet/yolov4-flintstone-1.cfg")
```

Am obtinut straturile de iesire:

```
layer_names = net.getUnconnectedOutLayersNames()
```

Am creat structura de date care reprezinta imaginea si o trimitem ca input in retea:

```
blob = cv.dnn.blobFromImage(img, 0.00392, (416, 416), (0, 0, 0), True,  
crop=False)
```

```
net.setInput(blob)
```

```
outs = net.forward(layer_names)
```