**Hand localization using Yolo on Hand database**

1. **Definirea problemei**

“Detectarea mâinilor este o procedură esențială de pre-procesare pentru multe sarcini de computer vision legate de mâini, cum ar fi estimarea poziției mâinii, recunoașterea gesturilor mâinii, analiza activității umane și așa mai departe. Cu toate acestea, detectarea în mod fiabil a mai multor mâini din scene aglomerate rămâne a fi o sarcină dificilă din cauza diversității aspectului complex al mâinilor (de exemplu, diferite forme ale mâinii, culori ale pielii, lumini, orientări și scale etc.) în imagini.”[1]

1. **Specificul bazei de date**

“Este un set de date cuprinzător de imagini cu mâinile colectate din diferite surse de seturi de date de imagini publice. Un total de 13050 de instanțe de mână sunt adnotate. Instanțele de mână mai mari decât o zonă fixă ​​a casetei de delimitare (1500 de pixeli pătrați) sunt considerate suficient de „mari” pentru detectări și sunt folosite pentru evaluare. Acest lucru oferă aproximativ 4170 de cazuri de mână de înaltă calitate. În timpul colectării datelor, nu a fost impusă nici o restricție asupra poziției sau vizibilității oamenilor și nici nu a fost impusă o constrângere asupra mediului. În fiecare imagine sunt adnotate toate mâinile care pot fi percepute clar de oameni. Adnotările constau într-un dreptunghi de delimitare, care nu trebuie să fie aliniat pe axă, orientat în raport cu încheietura mâinii.”[2]

1. **Arhitectura**

“YOLO este alcătuit din trei componente cheie: head, neck și backbone. Backbone este partea rețelei formată din straturi convoluționale pentru a detecta caracteristicile cheie ale unei imagini și a le procesa. Backbone este mai întâi antrenat pe un set de date de clasificare, și de obicei antrenat la o rezoluție mai mică decât modelul final de detectare, deoarece detectarea necesită detalii mai fine decât clasificarea. Neck folosește caracteristicile straturilor de convoluție din backbone cu straturi fully connected pentru a face predicții privind probabilitățile și coordonatele căsuței de delimitare. Head este stratul final de ieșire al rețelei care poate fi interschimbat cu alte straturi cu aceeași formă de intrare pentru învățarea prin transfer.”[3]

1. **Librăria**

Librăria utilizată este PyTorch.

1. **Descrierea aplicației**

Codul a fost scris folosind limbajul de programare Python și mediul de lucru Google Colab, iar pentru YOLO s-a folosit ultima versiune, și anume versiunea 5.

Setul de date a fost prelucrat cu ajutorul platformei Roboflow [4] care poate oferi la nevoie pre-procesări cum ar fi redimensionarea imaginilor, auto-orientarea, croparea, ajustarea contrastului, precum și augmentare, adică rotație, aplicare de blur etc. atât la nivel de imagine cât și la nivelul dreptunghiului de delimitare.

Unul dintre obstacolele întâlnite a fost faptul că adnotările imaginilor sunt implicit în format .mat (Matlab), iar YOLO acceptă format .txt(clasa, x,y,w,h), unde clasa = obiectul de interes(în cazul acesta mâna), x și y reprezintă coordonatele centrului dreptunghiului care încadrează mâna, w și h reprezintă lățimea, respectiv înălțimea dreptunghiului. Soluția la această problemă se găsește la referința [5], care în plus face în așa fel încât dreptunghiurile care încadrează mâinile să fie paralele cu axa Ox, și nu înclinate deoarece YOLO nu acceptă acest lucru. În unele cazuri dreptunghiurile au și fost mărite ca să se evite situația ca dacă dreptunghiul a fost rotit să fie paralel cu Ox acesta încă să încadreze mâna.

Ulterior pentru realizarea codului s-au folosit materialele puse la dispoziție de Roboflow de la referința [6], în special notebook-ul pentru Google Colab care face legătura cu setul de date Hand Dataset pre-procesat, după ce a fost încărcat pe platformă.

1. **Performanța atinsă**

“Performanța este evaluată folosind precizia medie (AP) (zona sub curba Precision Recall).”[7]

Precizia medie este prezentă în articolul [7], care introduce și baza de date.

Parametrii care au fost variați în cadrul antrenării sunt numărul de epoci, precum și dimensiunea imaginilor. Pentru al doilea tip de parametri variați valorile s-au ales conform [8], și anume 320x320 și 416x416. Pentru numărul de epoci s-au folosit doar valorile 100 și 200 deoarece timpul de antrenare a fost lung, mai exact 3h 8min 37s pentru 100 epoci la 416x416, 6h 38min 35s pentru 200 epoci la 416x416, 2h 15min 38s pentru 100 epoci la 320x320 și 4h 29min 59s pentru 200 epoci la 320x320.

Pentru precizia medie se pot atribui următoarele valori: aproximativ 40 pentru 100 epoci la 416x416, 42.4 pentru 200 epoci la 416x416, 36.5 pentru 100 epoci la 320x320 și 38.8 pentru 200 epoci la 320x320.

Pentru recall(câte valori de True Positive au fost găsite) se poate spune în felul următor: aproximativ 72 pentru 100 epoci la 416x416, 73 pentru 200 epoci la 416x416, 72 pentru 100 epoci la 320x320 și 73 pentru 200 epoci la 320x320.

1. **Comparație cu rezultate din literatură**

Conform [7] pentru variația “({h, 2w}, {2h, 2w}, {h, 3w} și {2h, 3w}), unde „h” și „w” sunt înălțimea și lățimea casetei de delimitare a mâinii” s-au obținut următoarele valori pentru AP: 46.13, 38.75, 44.04, 40.97. Pentru următoarea figură din articol rezultatele autorilor au furnizat o valoare de 48.2 pentru AP comparativ cu cele de referință care au oferit o valoare de 33.57.

Graphical user interface, chart, line chart

Description automatically generated

Figura 7.1: Curba Precision Recall [7]

1. **Concluzie**

În final se poate spune că detecția mâinii folosind YOLO pentru parametrii aleși să fie variați a furnizat rezultate decente în condițiile limitărilor resurselor oferite de Google Colab. Se poate presupune că dacă într-un context cu mai multe resurse la dispoziție dacă numărul epocilor ar fi fost crescut, performanța ar fi fost mai ridicată.

1. **Bibliografie**

[1] Chi Xu , Wendi Cai, Yongbo Li , Jun Zhou, Longsheng Wei, *Accurate Hand Detection from Single-Color Images by Reconstructing Hand Appearances*, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6982909/>, accesat la data: 06.01.2022

[2] *Hand Dataset*, <https://www.robots.ox.ac.uk/~vgg/data/hands/>, accesat la data: 06.01.2022

[3] *YOLO Explained*, <https://medium.com/analytics-vidhya/yolo-explained-5b6f4564f31>, accesat la data: 06.01.2022

[4] *Give your software the sense of sight*, <https://roboflow.com/>, accesat la data: 06.01.2022

[5] *Convert Oxford Hand Dataset for YOLO*, <https://github.com/SignusRobotics/oxford-hands-to-yolo>, accesat la data: 06.01.2022

[6] *YOLOv5*, <https://models.roboflow.com/object-detection/yolov5>, accesat la data: 06.01.2022

[7] A. Mittal, A. Zisserman, P. H. S. Torr, *Hand detection using multiple proposals*, <https://www.robots.ox.ac.uk/~vgg/publications/2011/Mittal11/mittal11.pdf>, accesat la data: 06.01.2022

[8] *What should be the size of input image for training a YOLOv3 Model Architecture CNN.?*, <https://stackoverflow.com/questions/63160524/what-should-be-the-size-of-input-image-for-training-a-yolov3-model-architecture>, accesat la data: 06.01.2022