

Tema 2 CAVA

Hirica Ioan Alexandru

Generearea exemplilor

Pentru a genera exemplele pozitive iau locatiile fetelor din fisierul de adnotari, le redimensionez la dimensiunea 36x36 si pe aceasta o transform intr-o histograma de gradienti orientati care va fi adaugata intr-un vector. Pentru a augmenta datele aceasta casuta de 36x36 ii fac flip si de asemenea o rotesc in intervalul $[-5, 5]$ grade folosind un pas de 0.5.

Pentru a genera exemplele negative redimensionez imaginea folosind factorii $[1, 0.9, 0.75, 0.5]$ si aleg la intamplare 13 ferestre de dimensiuni 36x36 din fiecare imagine redimensionata. Aceste 13 ferestre din fiecare imagine nu includ fetele adnotate. Pentru a determina daca fereastra aleasa nu contine fata adnotata folosesc functia din laborator `get_intersection_of_boxes` care trebuie sa aiba o valoare = 0.

Folosind aceste tehnici generez 166428 de exemple pozitive si aproximativ 142678 de exemple negative. (numarul de exemple negative poate diferi deoarece cele 13 ferestre alese aleator pot contine o fata adnotata, caz in care fereastra aleasa este ignorata).

Antrenarea algoritmilor:

Algoritmul de detectare al fetelor contine se face folosind o retea neuronală care contine un hidden layer avand 100 de perceptronii. Aceasta retea se antrenează folosind exemplele pozitive si negative enuntate mai sus folosindu-se de histograma de gradienti orientati pentru fiecare fereastra. Am observat ca folosirea unei retele care contine mai putine hidden layers pentru acest subpunct a dat rezultate mai bune decat folosirea unei retele mai adanci

Algoritmul de identificare al personajelor se face folosind o retea neuronală care contine 4 hidden layere avand (500, 250, 100, 50) de perceptronii . Aceasta se antrenează asemanator, numai ca label-urile au fost schimbate, reflectand personajul care este in fereastra respectiva. Fiecare nume a fost encodat astfel incat 'andy' primeste valoarea 1, 'ora' valoarea 2, 'louie' valoarea 3, 'tommy' valoarea 4.

Fiecare algoritm a fost rulat pana cand eroarea acestuia nu s-a mai imbunatatit cu un numar semnificativ timp de 10 epoci.

Rulare task1:

In primul pas fiecare imagine este redimensionata folosind factorii [1, 0.9, 0.85, 0.8, 0.75, 0.7, 0.65, 0.6, 0.5, 0.4, 0.3, 0.2]. Pentru fiecare imagine redimensionata aceasta va fi transformata intr-o histograma de gradienti orientati. In urmatorul pas vom glisa o fereastră si vom rula rețeaua aleasa pentru a face predictia. Vom rula folosind functia din sklearn `model.predict_log_proba` care pentru fiecare label ne va returna o valoare intre $(-\infty, 0)$. Vom selecta label-ul care are valoarea cea mai mare si in cazul in care aceasta valoare este mai mare decat -4, vom selecta acea fereastră si o vom redimensiona pentru a reflecta coordonatele din imaginea originala. Dupa ce facem aceste detectii vom folosi algoritmul NMS din laborator pentru a selecta ferestrele care au cele mai bune scoruri. Toate aceste detectii vor fi salvate in fisiere .npy conform exemplului oferit din folderul `evaluare/cod_evaluare/fisiere_solutie`.

Rulare task2:

Task 2 se ruleaza in acelasi mod cu task 1 si anume se foloseste o fereastră glisanta, numai ca a fost adaptat pentru a folosi modelul pentru recunoasterea personajelor. Se foloseste NMS pentru detectiile fiecaror personaje.

O alta incercare a fost de a rula pe detectiile faciale de la Task 1, folosind acelasi model de retea neuronală, rezultatele fiind inferioare, astfel implementarea acestui task asemanator cu task 1 fiind mai performant.