UNIVERSITATEA POLITEHNICA BUCUREŞTI

Facultatea de Electronică, Telecomunicaţii şi Tehnologia Informaţiei

~ Proiect APDSV ~

**Estimarea distanței / vitezei unei mașini față de semnul de circulație de "Stop" / "Trecere de pietoni" din secvențe video filmate din mașină**

Studenţi: Bojan Vlad-Andrei

Lipan George-Cristian

Văduva Marian-Cătălin

Grupa: 411-CMM

Bucureşti 2018

1. Introducere

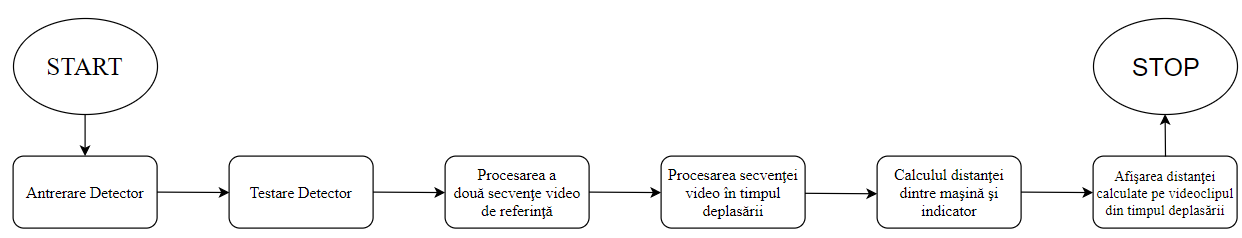
Orice conducător auto trebuie să cunoască semnificaţia indicatoarelor rutiere pentru a putea circula în siguranţă pe drumurile publice şi pentru a nu pune în pericol participanţii la trafic. De asemenea, trebuie ca acesta să fie atent să le observe în timp util pentru evitarea pericolelor.

Datorită dezvoltării noilor tehnologii prezente în maşinile eletrice, există mai multe metode de a atenţiona conducătorul auto în ceea ce priveşte traficul, indicatoarele rutiere şi nu numai.

Una din metode este detecţia indicatoarelor rutiere prezente în faţa automobilului şi estimarea distanţei dintre automobil şi indicatoare în timpul deplasării. Am folosit această metodă pentru realizarea prezentului proiect în MATLAB folosind secvenţe video filmate din maşină.

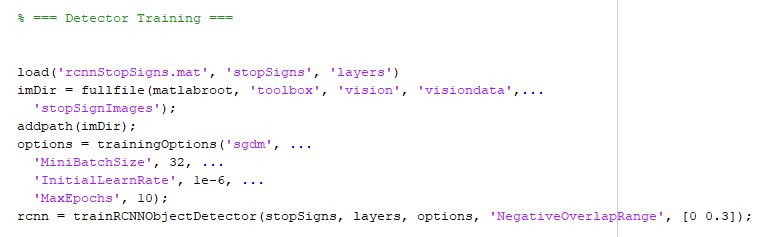
1. Descrierea proiectului

Schema bloc a proiectului

****

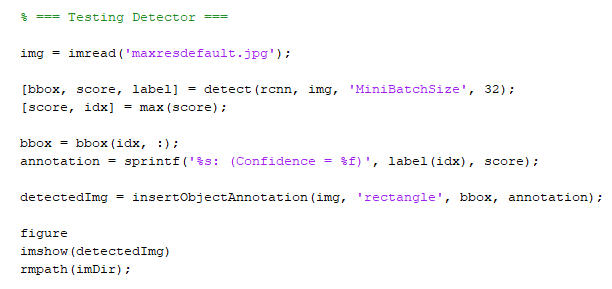
Blocul “Antrenare Detector” reprezintă un algoritm de detecţie a indicatorului rutier “STOP” din mai multe imagini ce fac parte dintr-o bază de date inclusă în MATLAB.

Codul din MATLAB:



Blocul “Testare Detector” are rolul de a testa detectorul de indicatoare de “STOP” folosind o imagine aleasă la întâmplare în care să fie prezent un astfel de indicator rutier.

Codul din MATLAB:

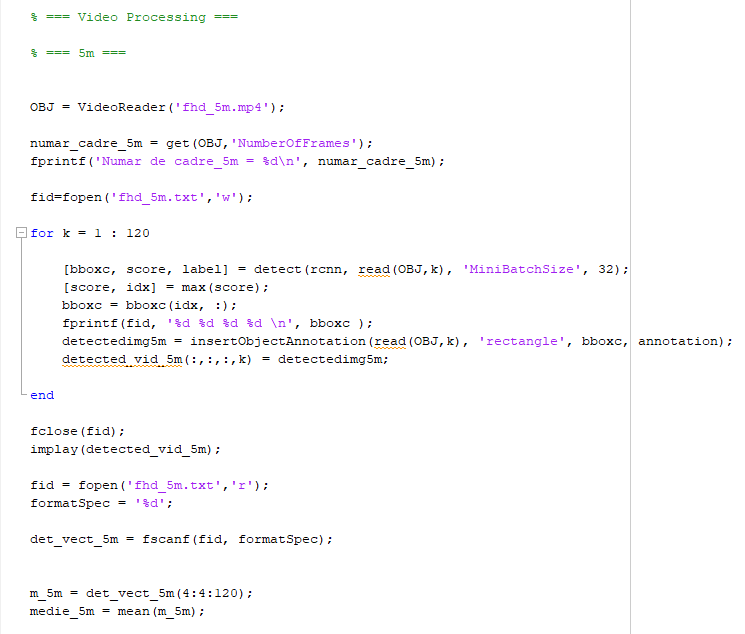


Rezultatul codului este imaginea aleasă de noi, ‘maxresdefaul.jpg’, la care se adaugă un chenar galben ce înconjoară indicatorul rutier prezent în imagine.



Blocul “Procesarea a două secvenţe video de referinţă” este un bloc în care sunt trecute prin detector două secvenţe video filmate din poziţii cunoscute ca şi distanţă până la un indicator rutier “STOP”. Acestea sunt filmate de la distanţele de 5.05m şi 7.54m, distanţe măsurate cu ajutorul unui aparat de măsură cu fascicul laser. Cele două blocuri de cod sunt asemănătoare. Pentru primul test, am ales ca secvenţele video să aibe rezoluţie FullHD (1920x1080) în speranţa ca detectorul să funcţioneze bine la imagini de calitate înaltă.

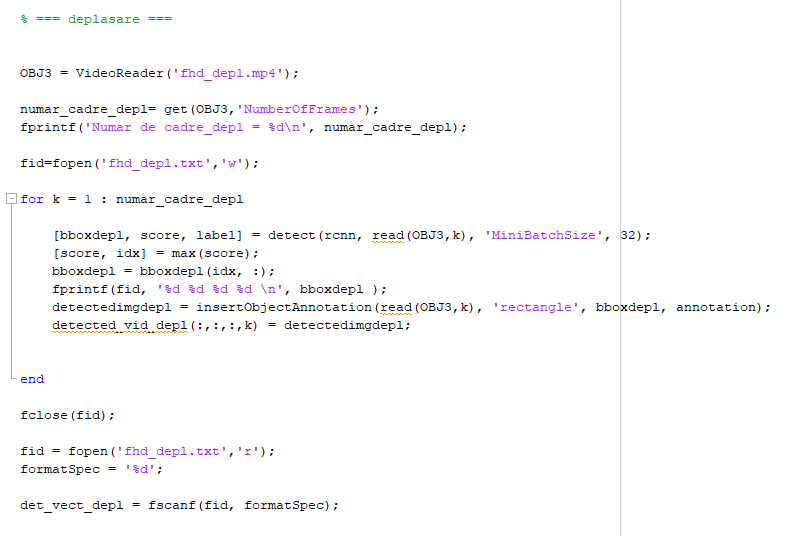
Codul din MATLAB pentru distanţa de 5.05m:



Am decis că 120 de cadre sunt suficiente pentru procesarea secvenţelor video de referinţă (se stă pe loc şi se filmează). Pentru a reţine dimensiunile pătratelor galbene ce înconjoară indicatorul “STOP” am ales să le scriem pe un fişier text, apoi, pentru a le prelucra, să le extragem din fişierul text şi să le introducem într-un vector în MATLAB. Informaţiile pătratelor galbele sunt: coordonatele x şi y ale colţului din stânga-sus al pătratului şi indecşii cu care se adună aceste coordonate pentru a încadra indicatorul rutier. Pentru observarea rezultatului detectorului, am folosit funcţia “implay” pentru a vedea cum se comportă detecţia indicatorului rutier din fiecare cadru. Am realizat media indecşilor de pe axa y pentru prelucrări ulterioare (calculul distanţei). Durata de procesare a acestui bloc este de aproximativ 20-25 min (în cazul în care se rulează pe un PC performant).

Blocul “Procesarea secvenţei video în timpul deplasării” reprezintă algoritmul de detecţie al indicatorului rutier “STOP” dintr-o secvenţă video care s-a filmat ­în timpul deplasării spre indicator.

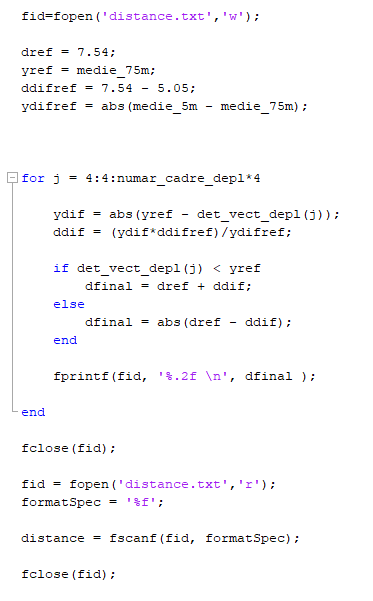
Codul din MATLAB:



După cum se poate observa, codul este asemănător cu cel de la blocul “Procesarea a două secvenţe video de referinţă”. Se foloseşte acelaşi principiu doar că de această dată ne folosim de toate cadrele secvenţei video. Calitatea este aceeaşi, FullHD. Durata de procesare, ­însă, este dublă faţă de cea de la blocul anterior (40-45 min). Cu cât rezoluţia secvenţei video este mai mare, cu atât procesarea durează mai mult.

Blocul “Calculul distanţei dintre maşină şi indicator” este un algoritm ce calculează distanţa de la automobil până la indicatorul rutier “STOP” din secvenţa video care s-a filmat în timpul deplasării.

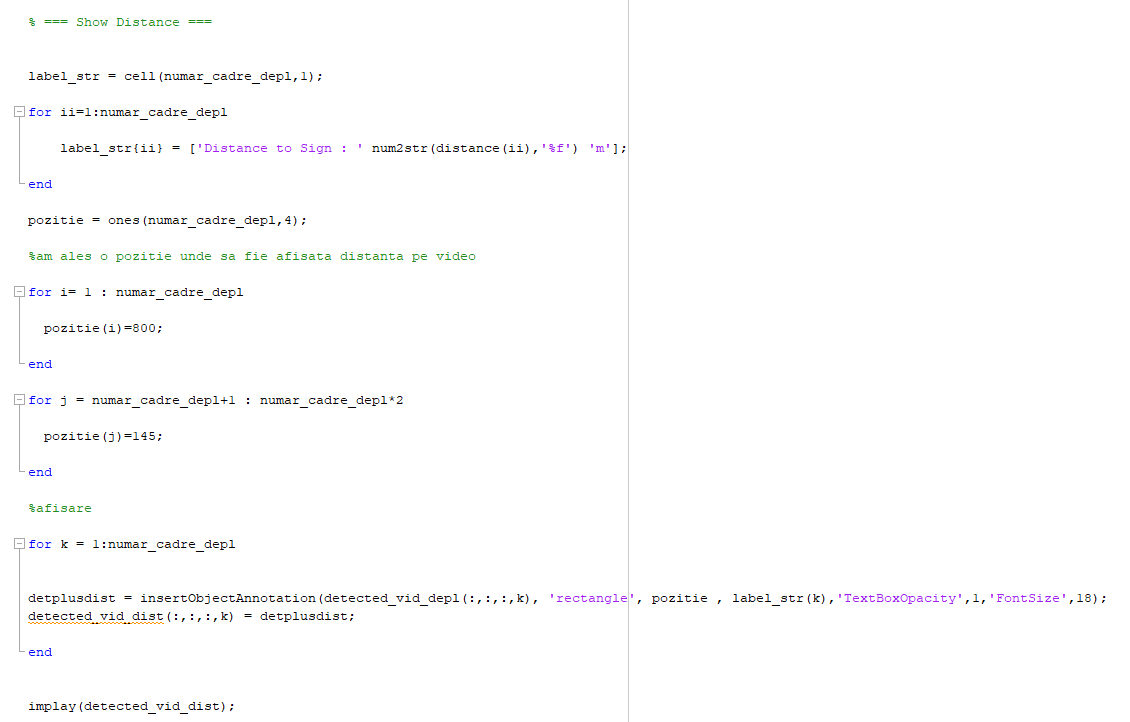
Codul din MATLAB:



Algoritmul începe prin alegerea uneia din cele două distanţe de referinţă, în cazul nostru fiind aleasă cea de 7.54m. De asemenea, alegem indexul mediu pe y corespunzător distanţei de referinţă alese. Se face diferenţa dintre cele două distanţe de referinţă şi, de asemenea, diferenţa dintre indecşii medii pe y ale celor două distanţe.

Calculul se realizează în bucla for: se face diferenţa în modul dintre indexul pe y de referinţă şi indexul pe y citit din blocul anterior (vectorul “det\_vect\_depl”) apoi, prin regula de 3 simplă, se determină cât s-a deplasat vehiculul faţă de distanţa de referinţă de 7.54m. După acest calcul, intervine o condiţie: dacă indexul citit este mai mic decât indexul de referinţă, rezultă că pătratul galbel se îngustează (indicatorul ocupă mai puţini pixeli în cadru), deci automobilul se află la o distanţă mult mai mare de 7.54m; în caz contrar, dimensiunea pătratului creşte, deci automobilul se apropie de indicator. Distanţele calculate la fiecare cadru se salvează în vectorul “distance” cu ajutorul unui fişier text, metodă similară cu cea folosită în blocurile anterioare.

Blocul “Afişarea distanţei calculate pe videoclipul din timpul deplasării” are rolul de a afişa, pe fiecare cadru al secvenţei video procesate anterior, distanţa dintre automobil şi indicator calculată.

Codul din MATLAB:

În prima parte a blocului se realizează aşa numitul “label” ce este defapt o etichetă cu un şir de caractere ce se doreşte afişat pe cadrele video. Pentru aceasta se foloseşte un ciclu for pentru a citi toate distanţele calculate anterior şi pentru a le introduce în etichete.

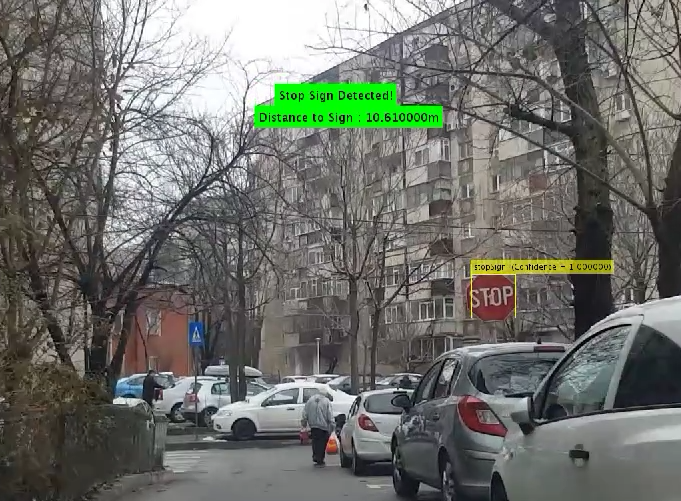
În a doua parte a blocului se formează un vector constant numit “pozitie” ce are rolul de a afişa eticheta în poziţia dorită pe cadrele video. Acest vector se construieşte simplu în funcţie de preferinţele utilizatorului şi de rezoluţia secvenţei video procesate.

Ultima parte a blocului este rezervată afişării propriu-zise a etichetelor în cadrele video folosind aceeaşi funcţie ce se foloseşte la afişarea pătratelor galbene: “insertObjectAnnotation”. Rezultatul final se afişează cu funcţia “implay”. Durata de procesare a blocului este aproximativ egală cu a blocului “Procesarea secvenţei video în timpul deplasării” (40-45 min).

1. Concluzii

Rezultatele finale ale proiectului sunt plauzibile, însă există mici probleme ce ar putea fi rezolvate ulterior.

Una din probleme este de detecţie în cazul în care indicatorul rutier iese din secvenţa video. O posibilitate de rezolvare ar putea fi implementând o condiţie (să se oprească detecţia/să se atenţioneze în cazul în care nu există coordonate x şi y în colţul din stanga-sus al pătratului galben). Pentru asta, am realizat o îmbunătăţire a proiectului din punctul de vedere al afişării: atunci când nu se detectează semnul, se va afişa pe ecran cu roşu mesajul “Stop Sign Undetected!”. În cazul în care semnul este detectat, se va afişa pe ecran cu verde mesajul “Stop Sign Detected!” şi distanţa de la automobil până la indicator. Codul îmbunătăţit este ataşat în anexă.



O altă problemă este variaţia rapidă şi continuă a dimensiunii pătratului galben de detecţie chiar şi în cazul în care vehiculul nu se deplasează.

Anexă

1. Variantă fără îmbunătăţiri

(nume fişier: “ProiectAPDSVSTOP.m”)

clear

close all

clc

% === Detector Training ===

load('rcnnStopSigns.mat', 'stopSigns', 'layers')

imDir = fullfile(matlabroot, 'toolbox', 'vision', 'visiondata',...

'stopSignImages');

addpath(imDir);

options = trainingOptions('sgdm', ...

'MiniBatchSize', 32, ...

'InitialLearnRate', 1e-6, ...

'MaxEpochs', 10);

rcnn = trainRCNNObjectDetector(stopSigns, layers, options, 'NegativeOverlapRange', [0 0.3]);

% === Testing Detector ===

img = imread('maxresdefault.jpg');

[bbox, score, label] = detect(rcnn, img, 'MiniBatchSize', 32);

[score, idx] = max(score);

bbox = bbox(idx, :);

annotation = sprintf('%s: (Confidence = %f)', label(idx), score);

detectedImg = insertObjectAnnotation(img, 'rectangle', bbox, annotation);

figure

imshow(detectedImg)

rmpath(imDir);

% === Video Processing ===

% === 5m ===

OBJ = VideoReader('fhd\_5m.mp4');

numar\_cadre\_5m = get(OBJ,'NumberOfFrames');

fprintf('Numar de cadre\_5m = %d\n', numar\_cadre\_5m);

fid=fopen('fhd\_5m.txt','w');

for k = 1 : 120

[bboxc, score, label] = detect(rcnn, read(OBJ,k), 'MiniBatchSize', 32);

[score, idx] = max(score);

bboxc = bboxc(idx, :);

fprintf(fid, '%d %d %d %d \n', bboxc );

detectedimg5m = insertObjectAnnotation(read(OBJ,k), 'rectangle', bboxc, annotation);

detected\_vid\_5m(:,:,:,k) = detectedimg5m;

end

fclose(fid);

implay(detected\_vid\_5m);

fid = fopen('fhd\_5m.txt','r');

formatSpec = '%d';

det\_vect\_5m = fscanf(fid, formatSpec);

m\_5m = det\_vect\_5m(4:4:120);

medie\_5m = mean(m\_5m);

% === 7.5m ===

OBJ2 = VideoReader('fhd\_7.5m.mp4');

numar\_cadre\_75m = get(OBJ2,'NumberOfFrames');

fprintf('Numar de cadre\_7.5m = %d\n', numar\_cadre\_75m);

fid=fopen('fhd\_7.5m.txt','w');

for k = 1 : 120

[bbox75m, score, label] = detect(rcnn, read(OBJ2,k), 'MiniBatchSize', 32);

[score, idx] = max(score);

bbox75m = bbox75m(idx, :);

fprintf(fid, '%d %d %d %d \n', bbox75m );

detectedimg75m = insertObjectAnnotation(read(OBJ2,k), 'rectangle', bbox75m, annotation);

detected\_vid\_75m(:,:,:,k) = detectedimg75m;

end

fclose(fid);

implay(detected\_vid\_75m);

fid = fopen('fhd\_7.5m.txt','r');

formatSpec = '%d';

det\_vect\_75m = fscanf(fid, formatSpec);

m\_75m = det\_vect\_75m(4:4:120);

medie\_75m = mean(m\_75m);

% === deplasare ===

OBJ3 = VideoReader('fhd\_depl.mp4');

numar\_cadre\_depl= get(OBJ3,'NumberOfFrames');

fprintf('Numar de cadre\_depl = %d\n', numar\_cadre\_depl);

fid=fopen('fhd\_depl.txt','w');

for k = 1 : numar\_cadre\_depl

[bboxdepl, score, label] = detect(rcnn, read(OBJ3,k), 'MiniBatchSize', 32);

[score, idx] = max(score);

bboxdepl = bboxdepl(idx, :);

fprintf(fid, '%d %d %d %d \n', bboxdepl );

detectedimgdepl = insertObjectAnnotation(read(OBJ3,k), 'rectangle', bboxdepl, annotation);

detected\_vid\_depl(:,:,:,k) = detectedimgdepl;

end

fclose(fid);

fid = fopen('fhd\_depl.txt','r');

formatSpec = '%d';

det\_vect\_depl = fscanf(fid, formatSpec);

% === Distance Calculation ===

fid=fopen('distance.txt','w');

dref = 7.54;

yref = medie\_75m;

ddifref = 7.54 - 5.05;

ydifref = abs(medie\_5m - medie\_75m);

for j = 4:4:numar\_cadre\_depl\*4

ydif = abs(yref - det\_vect\_depl(j));

ddif = (ydif\*ddifref)/ydifref;

if det\_vect\_depl(j) < yref

dfinal = dref + ddif;

else

dfinal = abs(dref - ddif);

end

fprintf(fid, '%.2f \n', dfinal );

end

fclose(fid);

fid = fopen('distance.txt','r');

formatSpec = '%f';

distance = fscanf(fid, formatSpec);

fclose(fid);

% === Show Distance ===

label\_str = cell(numar\_cadre\_depl,1);

for ii=1:numar\_cadre\_depl

label\_str{ii} = ['Distance to Sign : ' num2str(distance(ii),'%f') 'm'];

end

pozitie = ones(numar\_cadre\_depl,4);

%am ales o pozitie unde sa fie afisata distanta pe video

for i= 1 : numar\_cadre\_depl

pozitie(i)=800;

end

for j = numar\_cadre\_depl+1 : numar\_cadre\_depl\*2

pozitie(j)=145;

end

%afisare

for k = 1:numar\_cadre\_depl

detplusdist = insertObjectAnnotation(detected\_vid\_depl(:,:,:,k), 'rectangle', pozitie , label\_str(k),'TextBoxOpacity',1,'FontSize',18);

detected\_vid\_dist(:,:,:,k) = detplusdist;

end

implay(detected\_vid\_dist);

1. Variantă cu îmbunătăţiri pentru semnale video FullHD (1920 x 1080)

(nume fişier: “ProiectAPDSVSTOP\_1080p\_update.m”)

clear

close all

clc

% === Detector Training ===

load('rcnnStopSigns.mat', 'stopSigns', 'layers')

imDir = fullfile(matlabroot, 'toolbox', 'vision', 'visiondata',...

'stopSignImages');

addpath(imDir);

options = trainingOptions('sgdm', ...

'MiniBatchSize', 32, ...

'InitialLearnRate', 1e-6, ...

'MaxEpochs', 10);

rcnn = trainRCNNObjectDetector(stopSigns, layers, options, 'NegativeOverlapRange', [0 0.3]);

% === Testing Detector ===

img = imread('maxresdefault.jpg');

[bbox, score, label] = detect(rcnn, img, 'MiniBatchSize', 32);

[score, idx] = max(score);

bbox = bbox(idx, :);

annotation = sprintf('%s: (Confidence = %f)', label(idx), score);

detectedImg = insertObjectAnnotation(img, 'rectangle', bbox, annotation);

figure

imshow(detectedImg)

rmpath(imDir);

% === Video Processing ===

% === 5m ===

OBJ = VideoReader('ref\_5m\_1080.mp4');

numar\_cadre\_5m = get(OBJ,'NumberOfFrames');

fprintf('Numar de cadre\_5m = %d\n', numar\_cadre\_5m);

fid=fopen('ref\_5m\_1080.txt','w');

for k = 1 : 120

[bboxc, score, label] = detect(rcnn, read(OBJ,k), 'MiniBatchSize', 32);

[score, idx] = max(score);

bboxc = bboxc(idx, :);

fprintf(fid, '%d %d %d %d \n', bboxc );

detectedimg5m = insertObjectAnnotation(read(OBJ,k), 'rectangle', bboxc, annotation);

detected\_vid\_5m(:,:,:,k) = detectedimg5m;

end

fclose(fid);

implay(detected\_vid\_5m);

fid = fopen('ref\_5m\_1080.txt','r');

formatSpec = '%d';

det\_vect\_5m = fscanf(fid, formatSpec);

m\_5m = det\_vect\_5m(4:4:120);

medie\_5m = mean(m\_5m);

% === 7.5m ===

OBJ2 = VideoReader('ref\_75m\_1080.mp4');

numar\_cadre\_75m = get(OBJ2,'NumberOfFrames');

fprintf('Numar de cadre\_7.5m = %d\n', numar\_cadre\_75m);

fid=fopen('ref\_75m\_1080.txt','w');

for k = 1 : 120

[bbox75m, score, label] = detect(rcnn, read(OBJ2,k), 'MiniBatchSize', 32);

[score, idx] = max(score);

bbox75m = bbox75m(idx, :);

fprintf(fid, '%d %d %d %d \n', bbox75m );

detectedimg75m = insertObjectAnnotation(read(OBJ2,k), 'rectangle', bbox75m, annotation);

detected\_vid\_75m(:,:,:,k) = detectedimg75m;

end

fclose(fid);

implay(detected\_vid\_75m);

fid = fopen('ref\_75m\_1080.txt','r');

formatSpec = '%d';

det\_vect\_75m = fscanf(fid, formatSpec);

m\_75m = det\_vect\_75m(4:4:120);

medie\_75m = mean(m\_75m);

% === deplasare ===

OBJ3 = VideoReader(inainte\_vitezamedie\_1080.mp4');

numar\_cadre\_depl= get(OBJ3,'NumberOfFrames');

fprintf('Numar de cadre\_depl = %d\n', numar\_cadre\_depl);

label\_no\_det = cell(numar\_cadre\_depl,1);

for ii=1:numar\_cadre\_depl

label\_no\_det{ii} = ['Stop Sign Undetected!'];

end

label2 = cell(numar\_cadre\_depl,1);

for ii=1:numar\_cadre\_depl

label2{ii} = ['Stop Sign Detected!'];

end

pozitiehd = ones(numar\_cadre\_depl,4);

%am ales o pozitie unde sa fie afisata distanta pe video

for i= 1 : numar\_cadre\_depl

pozitiehd(i)=870;

end

for j = numar\_cadre\_depl+1 : numar\_cadre\_depl\*2

pozitiehd(j)=270;

end

fid=fopen(‘inainte\_vitezamedie\_1080.txt','w');

fid2=fopen('cadre\_fara\_detectie\_ inainte\_vitezamedie\_1080.txt','w');

for k = 1 : numar\_cadre\_depl

[bboxdepl, score, label] = detect(rcnn, read(OBJ3,k), 'MiniBatchSize', 32);

[score, idx] = max(score);

bboxdepl = bboxdepl(idx, :);

if isempty(bboxdepl) == 1

fprintf(fid2, '%d \n', k );

detectedimgdepl = insertObjectAnnotation(read(OBJ3,k), 'rectangle',pozitiehd, label\_no\_det(k),'TextBoxOpacity',1,'FontSize',18,'Color','red');

detected\_vid\_depl(:,:,:,k) = detectedimgdepl;

else

fprintf(fid, '%d %d %d %d \n', bboxdepl );

detectedimgdepl = insertObjectAnnotation(read(OBJ3,k), 'rectangle', bboxdepl, annotation);

detected\_vid\_depl(:,:,:,k) = detectedimgdepl;

detectedimgdepl = insertObjectAnnotation(detected\_vid\_depl(:,:,:,k), 'rectangle', pozitiehd, label2(k),'TextBoxOpacity',1,'FontSize',18,'Color','green');

detected\_vid\_depl(:,:,:,k) = detectedimgdepl;

end

end

fclose(fid);

fclose(fid2);

fid = fopen(‘inainte\_vitezamedie\_1080.txt','r');

formatSpec = '%d';

det\_vect\_depl = fscanf(fid, formatSpec);

fclose(fid);

implay(detected\_vid\_depl);

fid2 = fopen('cadre\_fara\_detectie\_inainte\_vitezamedie\_1080.txt','r');

formatSpec = '%d';

no\_detection = fscanf(fid2, formatSpec);

fclose(fid2);

% === Distance Calculation ===

fid=fopen('distance\_inainte\_vitezamedie\_1080.txt','w');

dref = 7.54;

yref = medie\_75m;

ddifref = 7.54 - 5.05;

ydifref = abs(medie\_5m - medie\_75m);

numar\_cadre\_cu\_detectie = numar\_cadre\_depl - length(no\_detection);

for j = 4:4:numar\_cadre\_cu\_detectie\*4

ydif = abs(yref - det\_vect\_depl(j));

ddif = (ydif\*ddifref)/ydifref;

if det\_vect\_depl(j) < yref

dfinal = dref + ddif;

else

dfinal = abs(dref - ddif);

end

fprintf(fid, '%.2f \n', dfinal );

end

fclose(fid);

fid = fopen('distance\_inainte\_vitezamedie\_1080.txt','r');

formatSpec = '%f';

distance = fscanf(fid, formatSpec);

fclose(fid);

% === Show Distance ===

label\_str = cell(numar\_cadre\_cu\_detectie,1);

for ii=1:numar\_cadre\_cu\_detectie

label\_str{ii} = ['Distance to Sign : ' num2str(distance(ii),'%f') 'm'];

end

pozitie = ones(numar\_cadre\_depl,4);

%am ales o pozitie unde sa fie afisata distanta pe video

for i= 1 : numar\_cadre\_depl

pozitie(i)=840;

end

for j = numar\_cadre\_depl+1 : numar\_cadre\_depl\*2

pozitie(j)=303;

end

%afisare

index = 0;

OBJ4 = VideoReader('inainte\_vitezamedie\_1080.mp4');

for k = 1:numar\_cadre\_depl

[bboxdepl, score, label] = detect(rcnn, read(OBJ4,k), 'MiniBatchSize', 32);

[score, idx] = max(score);

bboxdepl = bboxdepl(idx, :);

if isempty(bboxdepl) == 1

index = index + 1;

detplusdist = insertObjectAnnotation(detected\_vid\_depl(:,:,:,k), 'rectangle',pozitiehd, label\_no\_det(k),'TextBoxOpacity',1,'FontSize',18,'Color','red');

detected\_vid\_dist(:,:,:,k) = detplusdist;

else

detplusdist = insertObjectAnnotation(detected\_vid\_depl(:,:,:,k), 'rectangle', pozitie , label\_str(abs(k-index)),'TextBoxOpacity',1,'FontSize',18,'Color','green');

detected\_vid\_dist(:,:,:,k) = detplusdist;

end

end

implay(detected\_vid\_dist,50);

1. Variantă cu ­îmbunătăţiri pentru semnale video HD (1280 x 720)

(nume fişier: “ProiectAPDSVSTOP\_720p\_update.m”)

clear

close all

clc

% === Detector Training ===

load('rcnnStopSigns.mat', 'stopSigns', 'layers')

imDir = fullfile(matlabroot, 'toolbox', 'vision', 'visiondata',...

'stopSignImages');

addpath(imDir);

options = trainingOptions('sgdm', ...

'MiniBatchSize', 32, ...

'InitialLearnRate', 1e-6, ...

'MaxEpochs', 10);

rcnn = trainRCNNObjectDetector(stopSigns, layers, options, 'NegativeOverlapRange', [0 0.3]);

% === Testing Detector ===

img = imread('maxresdefault.jpg');

[bbox, score, label] = detect(rcnn, img, 'MiniBatchSize', 32);

[score, idx] = max(score);

bbox = bbox(idx, :);

annotation = sprintf('%s: (Confidence = %f)', label(idx), score);

detectedImg = insertObjectAnnotation(img, 'rectangle', bbox, annotation);

figure

imshow(detectedImg)

rmpath(imDir);

% === Video Processing ===

% === 5m ===

OBJ = VideoReader('ref\_5m\_720\_2.mp4');

numar\_cadre\_5m = get(OBJ,'NumberOfFrames');

fprintf('Numar de cadre\_5m = %d\n', numar\_cadre\_5m);

fid=fopen('ref\_5m\_720.txt','w');

for k = 1 : 70

[bboxc, score, label] = detect(rcnn, read(OBJ,k), 'MiniBatchSize', 32);

[score, idx] = max(score);

bboxc = bboxc(idx, :);

fprintf(fid, '%d %d %d %d \n', bboxc );

detectedimg5m = insertObjectAnnotation(read(OBJ,k), 'rectangle', bboxc, annotation);

detected\_vid\_5m(:,:,:,k) = detectedimg5m;

end

fclose(fid);

implay(detected\_vid\_5m);

fid = fopen('ref\_5m\_720.txt','r');

formatSpec = '%d';

det\_vect\_5m = fscanf(fid, formatSpec);

%

m\_5m = det\_vect\_5m(4:4:120);

medie\_5m = mean(m\_5m);

% === 7.5m ===

OBJ2 = VideoReader('ref\_75m\_720.mp4');

numar\_cadre\_75m = get(OBJ2,'NumberOfFrames');

fprintf('Numar de cadre\_7.5m = %d\n', numar\_cadre\_75m);

fid=fopen('ref\_75m\_720.txt','w');

for k = 1 : 70

[bbox75m, score, label] = detect(rcnn, read(OBJ2,k), 'MiniBatchSize', 32);

[score, idx] = max(score);

bbox75m = bbox75m(idx, :);

fprintf(fid, '%d %d %d %d \n', bbox75m );

detectedimg75m = insertObjectAnnotation(read(OBJ2,k), 'rectangle', bbox75m, annotation);

detected\_vid\_75m(:,:,:,k) = detectedimg75m;

end

fclose(fid);

implay(detected\_vid\_75m);

fid = fopen('ref\_75m\_720.txt','r');

formatSpec = '%d';

det\_vect\_75m = fscanf(fid, formatSpec);

m\_75m = det\_vect\_75m(4:4:120);

medie\_75m = mean(m\_75m);

% === deplasare ===

OBJ3 = VideoReader('inainte\_vitezamica\_720.mp4');

numar\_cadre\_depl= get(OBJ3,'NumberOfFrames');

fprintf('Numar de cadre\_depl = %d\n', numar\_cadre\_depl);

label\_no\_det = cell(numar\_cadre\_depl,1);

for ii=1:numar\_cadre\_depl

label\_no\_det{ii} = ['Stop Sign Undetected!'];

end

label2 = cell(numar\_cadre\_depl,1);

for ii=1:numar\_cadre\_depl

label2{ii} = ['Stop Sign Detected!'];

end

pozitiehd = ones(numar\_cadre\_depl,4);

%am ales o pozitie unde sa fie afisata distanta pe video

for i= 1 : numar\_cadre\_depl

pozitiehd(i)=600;

end

for j = numar\_cadre\_depl+1 : numar\_cadre\_depl\*2

pozitiehd(j)=200;

end

fid=fopen('inainte\_vitezamica\_720.txt','w');

fid2=fopen('cadre\_fara\_detectie\_inainte\_vitezamica\_720.txt','w');

for k = 1 : numar\_cadre\_depl

[bboxdepl, score, label] = detect(rcnn, read(OBJ3,k), 'MiniBatchSize', 32);

[score, idx] = max(score);

bboxdepl = bboxdepl(idx, :);

if isempty(bboxdepl) == 1

fprintf(fid2, '%d \n', k );

detectedimgdepl = insertObjectAnnotation(read(OBJ3,k), 'rectangle',pozitiehd, label\_no\_det(k),'TextBoxOpacity',1,'FontSize',18,'Color','red');

detected\_vid\_depl(:,:,:,k) = detectedimgdepl;

else

fprintf(fid, '%d %d %d %d \n', bboxdepl );

detectedimgdepl = insertObjectAnnotation(read(OBJ3,k), 'rectangle', bboxdepl, annotation);

detected\_vid\_depl(:,:,:,k) = detectedimgdepl;

detectedimgdepl = insertObjectAnnotation(detected\_vid\_depl(:,:,:,k), 'rectangle', pozitiehd, label2(k),'TextBoxOpacity',1,'FontSize',18,'Color','green');

detected\_vid\_depl(:,:,:,k) = detectedimgdepl;

end

end

fclose(fid);

fclose(fid2);

fid = fopen('inainte\_vitezamica\_720.txt','r');

formatSpec = '%d';

det\_vect\_depl = fscanf(fid, formatSpec);

fclose(fid);

implay(detected\_vid\_depl);

fid2 = fopen('cadre\_fara\_detectie\_inainte\_vitezamica\_720.txt','r');

formatSpec = '%d';

no\_detection = fscanf(fid2, formatSpec);

fclose(fid2);

% === Distance Calculation ===

fid=fopen('distance\_inainte\_vitezamica\_720.txt','w');

dref = 7.54;

yref = medie\_75m;

ddifref = 7.54 - 5.05;

ydifref = abs(medie\_5m - medie\_75m);

numar\_cadre\_cu\_detectie = numar\_cadre\_depl - length(no\_detection);

for j = 4:4:numar\_cadre\_cu\_detectie\*4

ydif = abs(yref - det\_vect\_depl(j));

ddif = (ydif\*ddifref)/ydifref;

if det\_vect\_depl(j) < yref

dfinal = dref + ddif;

else

dfinal = abs(dref - ddif);

end

fprintf(fid, '%.2f \n', dfinal );

end

fclose(fid);

fid = fopen('distance\_inainte\_vitezamica\_720.txt','r');

formatSpec = '%f';

distance = fscanf(fid, formatSpec);

fclose(fid);

% === Show Distance ===

label\_str = cell(numar\_cadre\_cu\_detectie,1);

for ii=1:numar\_cadre\_cu\_detectie

label\_str{ii} = ['Distance to Sign : ' num2str(distance(ii),'%f') 'm'];

end

pozitie = ones(numar\_cadre\_depl,4);

%am ales o pozitie unde sa fie afisata distanta pe video

for i= 1 : numar\_cadre\_depl

pozitie(i)=530;

end

for j = numar\_cadre\_depl+1 : numar\_cadre\_depl\*2

pozitie(j)=233;

end

%afisare

index = 0;

OBJ4 = VideoReader('inainte\_vitezamica\_720.mp4');

for k = 1:numar\_cadre\_depl

[bboxdepl, score, label] = detect(rcnn, read(OBJ4,k), 'MiniBatchSize', 32);

[score, idx] = max(score);

bboxdepl = bboxdepl(idx, :);

if isempty(bboxdepl) == 1

index = index + 1;

detplusdist = insertObjectAnnotation(detected\_vid\_depl(:,:,:,k), 'rectangle',pozitiehd, label\_no\_det(k),'TextBoxOpacity',1,'FontSize',18,...

'Color','red');

detected\_vid\_dist(:,:,:,k) = detplusdist;

else

detplusdist = insertObjectAnnotation(detected\_vid\_depl(:,:,:,k), 'rectangle', pozitie , label\_str(abs(k-index)),'TextBoxOpacity',1,'FontSize',18,...

'Color','green');

detected\_vid\_dist(:,:,:,k) = detplusdist;

end

end

implay(detected\_vid\_dist,30);