

UNIVERSITATEA POLITEHNICA BUCUREȘTI

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

~ Proiect APDSV ~

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

Studenti: Bojan Vlad-Andrei

Lipan George-Cristian

Văduva Marian-Cătălin

Grupa: 411-CMM

București 2018

## 1. Introducere

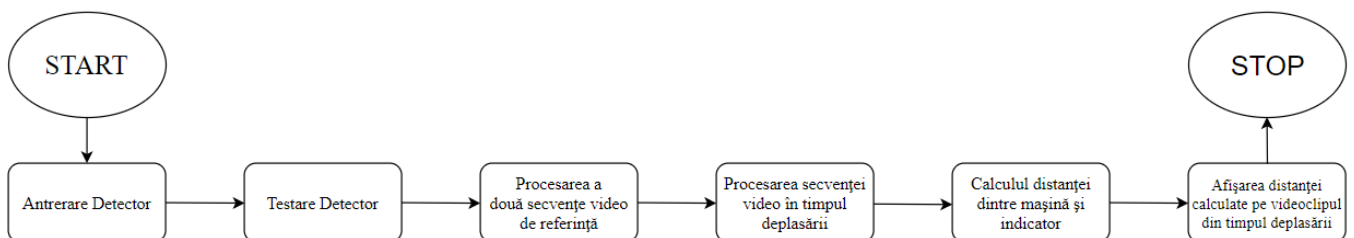
Orice conducător auto trebuie să cunoască semnificația indicatoarelor rutiere pentru a putea circula în siguranță pe drumurile publice și pentru a nu pune în pericol participanții la trafic. De asemenea, trebuie ca acesta să fie atent să le observe în timp util pentru evitarea pericolelor.

Datorită dezvoltării noilor tehnologii prezente în mașinile electrice, există mai multe metode de a atenționa conducătorul auto în ceea ce privește traficul, indicatoarele rutiere și nu numai.

Una din metode este detecția indicatoarelor rutiere prezente în fața automobilului și estimarea distanței dintre automobil și indicatoare în timpul deplasării. Am folosit această metodă pentru realizarea prezentului proiect în MATLAB folosind secvențe video filmate din mașină.

## 2. Descrierea proiectului

Schema bloc a proiectului



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

Codul din MATLAB:

```
% === 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]);
```

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:

```
% === 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);
```

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



Blocul “Procesarea a două secvențe video de referință” este un bloc în care sunt trecute prin detector două secvențe video filmate din poziții cunoscute ca și distanță până la un indicator rutier “STOP”. Acestea sunt filmate de la distanțele de 5.05m și 7.54m, distanțe măsurate cu ajutorul unui aparat de măsură cu fascicul laser. Cele două blocuri de cod sunt asemănătoare. Pentru primul test, am ales ca secvențele video să aibe rezoluție FullHD (1920x1080) în speranța ca detectorul să funcționeze bine la imagini de calitate înaltă.

Codul din MATLAB pentru distanța de 5.05m:

```
% === 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);
imshow(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);
```

Am decis că 120 de cadre sunt suficiente pentru procesarea secvențelor video de referință (se stă pe loc și se filmează). Pentru a reține dimensiunile pătratelor galbene ce înconjoară indicatorul “STOP” am ales să le scriem pe un fișier text, apoi, pentru a le prelucra, să le extragem din fișierul text și să le introducem într-un vector în MATLAB. Informațiile pătratelor galbe sunt: coordonatele x și y ale colțului din stânga-sus al pătratului și indecșii cu care se adună aceste coordonate pentru a încadra indicatorul rutier. Pentru observarea rezultatului detectorului, am folosit funcția “imshow” pentru a vedea cum se comportă detecția indicatorului rutier din fiecare cadru. Am realizat media indecșilor de pe axa y pentru prelucrări ulterioare (calculul distanței). Durata de procesare a acestui bloc este de aproximativ 20-25 min (în cazul în care se rulează pe un PC performant).

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

Codul din MATLAB:

```
% === 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);
```

După cum se poate observa, codul este asemănător cu cel de la blocul “Procesarea a două secvențe video de referință”. Se folosește același principiu doar că de această dată ne folosim de toate cadrele secvenței video. Calitatea este aceeași, FullHD. Durata de procesare, însă, este dublă față de cea de la blocul anterior (40-45 min). Cu cât rezoluția secvenței video este mai mare, cu atât procesarea durează mai mult.

Blocul “Calculul distanței dintre mașină și indicator” este un algoritm ce calculează distanța de la automobil până la indicatorul rutier “STOP” din secvența video care s-a filmat în timpul deplasării.

Codul din MATLAB:

```
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);
```

Algoritmul începe prin alegerea uneia din cele două distanțe de referință, în cazul nostru fiind aleasă cea de 7.54m. De asemenea, alegem indexul mediu pe y corespunzător distanței de referință alese. Se face diferența dintre cele două distanțe de referință și, de asemenea, diferența dintre indecșii medii pe y ale celor două distanțe.

Calculul se realizează în bucla for: se face diferența în modul dintre indexul pe y de referință și indexul pe y citit din blocul anterior (vectorul “det\_vect\_depl”) apoi, prin regula de 3 simplă, se determină cât s-a deplasat vehiculul față de distanța de referință de 7.54m. După acest calcul, intervine o condiție: dacă indexul citit este mai mic decât indexul de referință, rezultă că pătratul galbel se îngustează (indicatorul ocupă mai puțini pixeli în cadru), deci automobilul se află la o distanță mult mai mare de 7.54m; în caz contrar,

dimensiunea pătratului crește, deci automobilul se apropie de indicator. Distanțele calculate la fiecare cadru se salvează în vectorul “distance” cu ajutorul unui fișier text, metodă similară cu cea folosită în blocurile anterioare.

Blocul “Afișarea distanței calculate pe videoclipul din timpul deplasării” are rolul de a afișa, pe fiecare cadru al secvenței video procesate anterior, distanța dintre automobil și indicator calculată.

Codul din MATLAB:

```
% === 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

imshow(detected_vid_dist);
```

În prima parte a blocului se realizează așa numitul “label” ce este defapt o etichetă cu un șir de caractere ce se dorește afișat pe cadrele video. Pentru aceasta se folosește un ciclu for pentru a citi toate distanțele calculate anterior și pentru a le introduce în etichete.

În a doua parte a blocului se formează un vector constant numit “pozitie” ce are rolul de a afișa eticheta în poziția dorită pe cadrele video. Acest vector se construiește simplu în funcție de preferințele utilizatorului și de rezoluția secvenței video procesate.

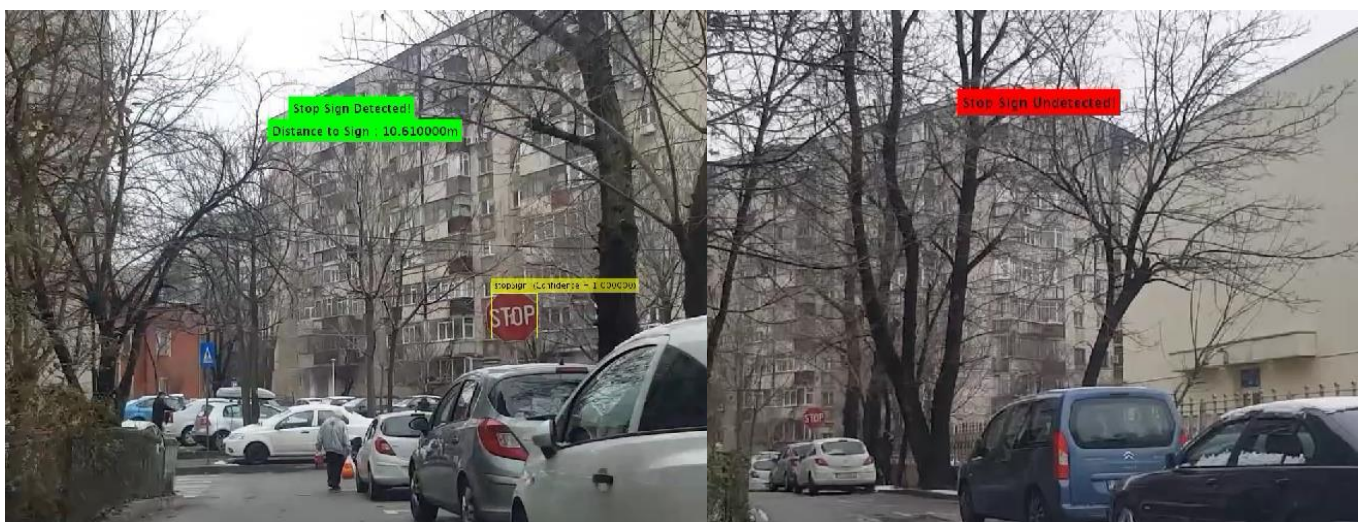
Ultima parte a blocului este rezervată afișării propriu-zise a etichetelor în cadrele video folosind aceeași funcție ce se folosește la afișarea pătratelor galbene: “insertObjectAnnotation”. Rezultatul final se afișează cu funcția “imshow”. Durata de procesare a blocului este aproximativ egală cu a blocului “Procesarea secvenței video în timpul deplasării” (40-45 min).



### 3. Concluzii

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

Una din probleme este de detecție în cazul în care indicatorul rutier iese din secvența video. O posibilitate de rezolvare ar putea fi implementând o condiție (să se oprească detecția/să se atenționeze în cazul în care nu există coordonate x și y în colțul din stanga-sus al pătratului galben). Pentru asta, am realizat o îmbunătățire a proiectului din punctul de vedere al afișării: atunci când nu se detectează semnul, se va afișa pe ecran cu roșu mesajul “Stop Sign Undetected!”. În cazul în care semnul este detectat, se va afișa pe ecran cu verde mesajul “Stop Sign Detected!” și distanța de la automobil până la indicator. Codul îmbunătățit este atașat în anexă.



O altă problemă este variația rapidă și continuă a dimensiunii pătratului galben de detecție chiar și în cazul în care vehiculul nu se deplasează.



## Anexă

1. Variantă fără îmbunătățiri  
(nume fișier: “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);
```

2. Variantă cu îmbunătățiri pentru semnale video FullHD (1920 x 1080)  
(nume fișier: “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);
```

```
imshow(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
```

```

[bbbox75m, score, label] = detect(rcnn, read(OBJ2,k), 'MiniBatchSize', 32);
[score, idx] = max(score);
bbbox75m = bbbox75m(idx, :);
fprintf(fid, '%d %d %d %d \n', bbbox75m );
detectedimg75m = insertObjectAnnotation(read(OBJ2,k), 'rectangle', bbbox75m,
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

pozitieh = ones(numar_cadre_depl,4);

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

for i= 1 : numar_cadre_depl

    pozitieh(i)=870;

end

for j = numar_cadre_depl+1 : numar_cadre_depl*2

    pozitieh(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',pozitieh,
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', pozitieh, label2(k),'TextBoxOpacity',1,'FontSize',18,'Color','green');
        detected_vid_depl(:, :, k) = detectedimgdepl;
    end
end

```

```

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);

```

3. Variantă cu îmbunătățiri pentru semnale video HD (1280 x 720)  
(nume fișier: “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);
imshow(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

pozitieh = ones(numar_cadre_depl,4);

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

for i= 1 : numar_cadre_depl

    pozitieh(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

imshow(detected_vid_dist,30);

```