

Csapattagok:

Atkári György

Bata Kristóf

Filep Tamás

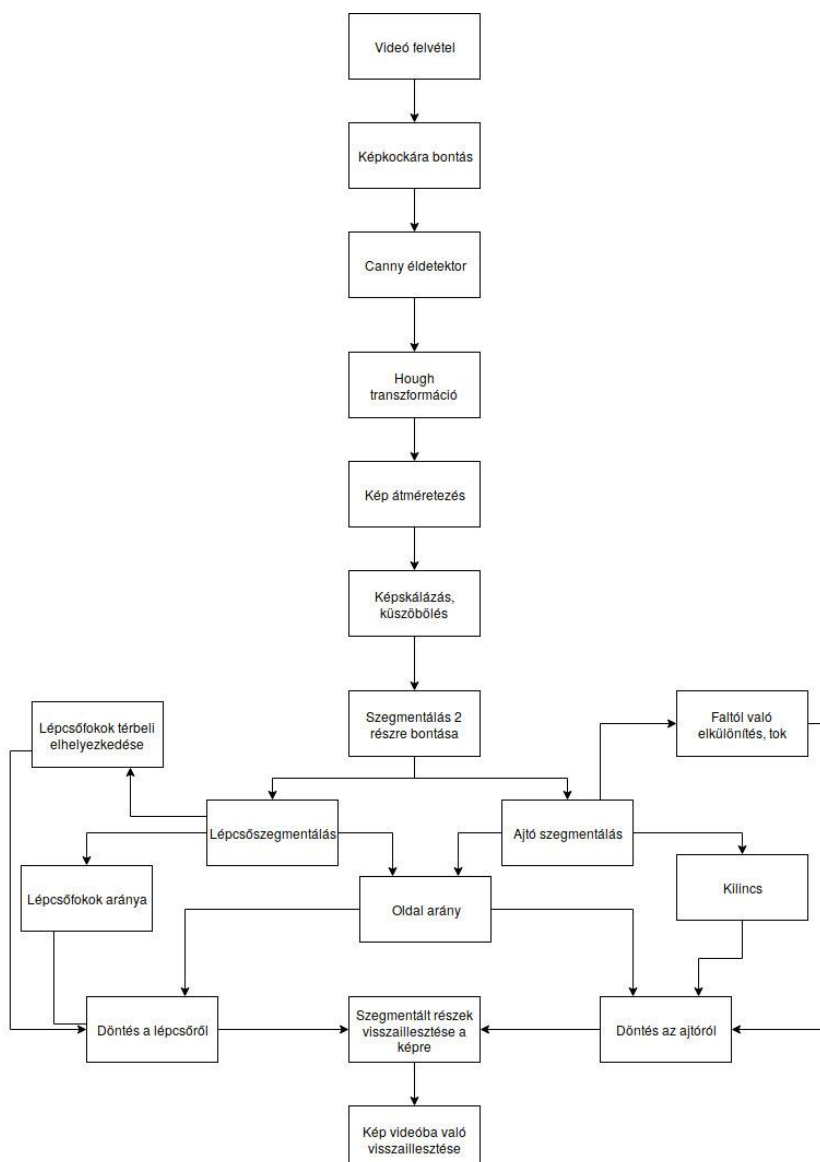
Rafai Norbert

LÉPCSŐ- ÉS AJTÓFELISMERÉS VIDEÓFELVÉTELEKEN

2. MÉRFÖLDKŐ

1. IMPLEMENTÁCIÓ OPENCV-BEN

Az áttekintett szakirodalom alapján az egyik lehetséges választott módszer folyamatábrája a következő:



2. MEGOLDÁS MATLAB-BAN, CNN SEGÍTSÉGÉVEL

A hatékonyabb gépi tanulás elősegítésére az eredeti RGB képen elvégzünk bizonyos előfeldolgozó műveleteket, előkészítve ezzel a szegmentálásra. Ezek egyszerű képfeldolgozó műveletek, szűrők. Fontos, hogy a képen található fölösleges információkat eldobjuk, ezzel is hatékonyabbá téve az algoritmusunkat.

A színes képet minden esetben szürkeárnyalatossá alakítjuk, mivel a képpontok színe nem hordoz számunkra lényegi információt. A szürkeárnyalatos képen alkalmaznunk kell továbbá hisztogram-széthúzást is. Az egy ponton átmenő egyeneseket legegyszerűbben a Hough térben a trigonometrikus görbék metszéspontjából nyerhetjük ki, ami tulajdonképpen nem is egy pont, hanem egy adott sugarú kör.

Az általunk végzett előfeldolgozás algoritmus:

```
I = rgb2gray(imread('door.6.jpg'));
rotI = imrotate(I,0,'crop');

% bináris kép előállítása
BW = edge(rotI,'prewitt');
st1 = strel('diamond',5);
st2 = strel('diamond',3);
BW = imdilate(BW, st1);
BW = imerode(BW, st2);
figure;
imshow(BW);
figure;

%%
% Hough-transzformáció elvégzése a bináris képen
[H,T,R] = hough(BW);
imshow(H,[],'XData',T,'YData',R,...
        'InitialMagnification','fit');
xlabel('\theta'), ylabel('\rho');
axis on, axis normal, hold on;

%%
% sarokpontok detektálása a kép Hough-transzformáltján
P = houghpeaks(H,4,'threshold',ceil(0.3*max(H(:))));
x = T(P(:,2)); y = R(P(:,1));
plot(x,y,'s','color','white');

%%
% Élek detektálása és kirajzolása
lines = houghlines(BW,T,R,P,'FillGap',1000,'MinLength',1100);
figure, imshow(rotI), hold on
max_len = 0;
for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');

    plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
    plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');

    len = norm(lines(k).point1 - lines(k).point2);
    if ( len > max_len)
        max_len = len;
        xy_long = xy;
    end
end
```

Az algoritmus végrehajtva tesztképeken az alábbi ábrán látható:



1. ábra - Az előfeldolgozás eredménye 2 tesztképen

Az előfeldolgozott képet már átadhatjuk egy előre betanított neurális hálónak.

Megoldásunk egyik alapját az Object Detection Using Deep Learning MATLAB-os példa adta [13]. Ebben be van mutatva, hogyan lehet egy konvolúciós neurális hálót megtanítani stop tábla felismerésére, ezt próbáltuk úgy átalakítani, hogy ajtó és lépcső detektálására is alkalmas legyen.

A tanításhoz R-CNN-t (Regions with Convolutional Neural Network) használ, amivel meg tudja tanulni, hogy a kép egyes régióin milyen objektum van. Ehhez olyan képek szükségesek, melyeken előre bejelölték, hogy az adott régión milyen objektum szerepel.

A példa úgynevezett transfer learning módszert használ, amelynek lényege, hogy a neurális hálót egy sok képet tartalmazó adatbázison tanítják, majd ezt finomhangolják a kívánt objektum megtalálásához. Ezzel a módszerrel sokkal kevesebb tanítóképre van szükség, mintha nem lenne egy előre betanított hálónk. Az előre betanítás a CIFAR-10 kép-gyűjteménnyel történik [14]. Ezután saját képekkel tanítva lehet finomhangolni a hálót.

Esetünkben 20, ajtót tartalmazó képpel finomhangoltuk a hálót, mely képekhez megadtuk az ajtót befoglaló téglalapok pozícióját és méretét. A tesztelés során gyenge eredmények születtek, nem igazán sikerült megtalálnia a képen levő ajtókat. Ennek miertjére egyelőre még nem sikerült rájöttünk, de dolgozunk a probléma kiküszöbölésén.

A tanuló algoritmus alkalmazása az előfeldolgozott „door.jpg” képen:

```
clc
load('DoorsRcnn.mat','rcnn');
testImage = imread('door.jpg');

[bboxes, score, label] = detect(rcnn, testImage, 'MiniBatchSize', 32);

annotation = cell(size(label,1),1);
for i=1:size(label,1)
    annotation{i} = sprintf('%s: (Confidence = %f)', label(i), score(i));
end

bbox = bboxes(:, :);

outputImage = insertObjectAnnotation(testImage, 'rectangle', bbox, annotation,
'LineWidth', 30, 'FontSize', 72);

figure;
imshow(outputImage);
```

A neurális háló által elért eredmények:



2. ábra - A neurális háló által megállapított címkék

LÉPCSŐK DETEKTÁLÁSA

A lépcsők detektálásához nem szükséges teljes 3D rekonstrukciót készítenünk.

A legegyszerűbb módja a mélységi információ kinyerésének az optical flow alkalmazása, kihasználva, hogy több frame-ből álló videófelvételen kell detektálnunk a lépcsőket. Ennek lényege az egymást követő frameken történő pontmegfeleltetésen alapszik.

SZAKIRODALOM

[1] *SSIP 2016 TEAM-C dokumentáció*

<https://SSIP2016teamcblog.wordpress.com/>

[2] *E. Jauregi, E. Lazkano and B. Sierra - Approaches to door identification for robot navigation*

http://cdn.intechopen.com/pdfs/10237/InTechApproaches_to_door_identification_for_robot_navigation.pdf

[3] *F. Mahmood, F. Kuwar - A Self-Organizing Neural Scheme for Door Detection in Different Environments*

<https://arxiv.org/pdf/1301.0432.pdf>

[4] *S. Cloix, G. Bologna, V. Weiss, T. Pun, D. Hasler - Descending Stairs Detection with Low-Power Sensors*

<https://pdfs.semanticscholar.org/2d70/ab62aa5378efe9354d8e6091cdb641ee9830.pdf>

[5] *Jens Hensler, Michael Blaich, and Oliver Bittel - Real-time Door Detection based on AdaBoost learning algorithm*

<https://pdfs.semanticscholar.org/8369/06b2ca9109760d370ee567ba4cd8b7ce9f33.pdf>

[6] *R. Sekkal, F. Pasteau, M. Babel, B. Brun, I. Leplumey - SIMPLE MONOCULAR DOOR DETECTION AND TRACKING*

http://www.irisa.fr/lagadic/pdf/2013_icip_sekkal.pdf

[7] *Xiaodong Yang and Yingli Tian - Robust Door Detection in Unfamiliar Environments by Combining Edge and Corner Features*

http://www-ee.ccny.cuny.edu/wwwn/yltian/Publications/CVAVI2010_Yang.pdf

[8] *Hannes Harms, Eike Rehder, Tobias Schwarze, Martin Lauer - Detection of Ascending Stairs using Stereo Vision*

http://www.mrt.kit.edu/z/publ/download/harms/StairDetection/2015_Harms_StairDetection.pdf

[9] *Rai Munoz, Xuejian Rong, Yingli Tian - DEPTH-AWARE INDOOR STAIRCASE DETECTION AND RECOGNITION FOR THE VISUALLY IMPAIRED*

<http://www-ee.ccny.cuny.edu/wwwn/yltian/Publications/ICME16-W155.pdf>

[10] *Mukul Anand Pathak, Kshitij Kamlakar, Shwetant Mohapatra, Prof. Uma Nagaraj - DEVELOPMENT OF CONTROL SOFTWARE FOR STAIR DETECTION IN A MOBILE ROBOT USING ARTIFICIAL INTELLIGENCE AND IMAGE PROCESSING*

http://www.iaeme.com/MasterAdmin/uploadfolder/IJCET_07_03_008/IJCET_07_03_008.pdf

[11] *Deep Learning for Computer Vision with MATLAB Manual*

<https://www.mathworks.com/company/newsletters/articles/deep-learning-for-computer-vision-with-matlab.html>

[12] *Object Detection Using Deep Learning – MATLAB manual*

<https://www.mathworks.com/help/vision/examples/object-detection-using-deep-learning.html>

[13] *MatConvNet: Convolutional Neural Networks for MATLAB by Andrea Vedaldi and Karel Lenc*

<https://arxiv.org/pdf/1412.4564.pdf>

[14] *Alex Krizhevsky - Learning Multiple Layers of Features from Tiny Images*

<https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>