Gradus Vol 10, No 2 (2023)
ISSN 2064-8014

# KÉPFELDOLGOZÁS ALAPÚ SZEMÉLY-SZÁMLÁLÁS BEÁGYAZOTT RENDSZEREN

# IMAGE-PROCESSING BASED PERSON-COUNTING IN EMBEDDED SYSTEMS

*Kovács Tamás*[0000-0003-3947-86961*], *Pásztor Attila*[0000-0001-7354-5114]

Informatika Tanszék, GAMF Műszaki és Informatikai Kar, Neumann János Egyetem, Magyarország
https://doi.org/10.47833/2023.2.CSC.012

***Kulcsszavak:***
mesterséges intelligencia;
képfeldolgozás;
vizuális objektum azonosítás

***Keywords:***
artificial intelligence, image processing,
visual object identification

***Cikktörténet:***
Beérkezett 2023. szeptember 10.
Átdolgozva 2023. október 31.
Elfogadva 2023. november 5.

***Összefoglalás***

*Ebben a munkában egy olyan képfeldolgozáson alapuló személy-számlálási megoldást mutatunk be, amely alkalmazható akár kis memóriával és számítási kapacitással rendelkező beágyazott rendszeren is. Az itt bemutatott módszer elsősorban beltérben, illetve nem túl gyorsan változó vizuális információ esetén alkalmazható, és azon alapul, hogy mozgás-detekciós eljárással elválasztjuk egymástól a lassan változó hátteret és a gyorsabban változó előtér objektumokat. A jelen munkában az eljárás használhatóságát felmérő első tesztek eredményeit mutatjuk be.*

***Abstract***

*In the present work a camera-picture based method is introduced for visual people counting, that is capable for implementation on cheap microcontroller boards with limited memory and CPU capacity. The method is based upon separating the slowly varying background from the faster moving foreground objects by motion-detection. In this work only a brief test result is presented, which may justify the usability of the method.*

## 1. Bevezetés

A vizuális, azaz kamera kép alapú emberi jelenlét érzékelés és a jelenlévő személyek számának becslése gyakran előforduló feladat az automatizálás vagy általában az intelligens rendszerek fejlesztése terültén. A személyek számának becslése nagyobb kihívást jelent, emellett ilyen esetben a terület, amelyen a becslést kell végezni rendszerint jóval nagyobb is, mint egy közeli személy jelenlétének detektálásánál. A személy szám becslés fontos lehet például a tömegközlekedésben a járatra várakozó vagy a járműben utazó utaslétszám meghatározásánál, vagy intelligens épületgépészeti alkalmazások fejlesztésénél. Ez a cikk alapvetően azokra a kutatási és fejlesztési eredményeket tárgyalja, amely Neumann János Egyetem és az Airvent Légtechnikai Zrt. Közös pályázati célkitűzései nyomán indult el. A pályázat ezen szegmensének konkrét célkitűzése: a létszám becslése egy iroda vagy előadó méretű zárt térben, ahol egy légfertőtlenítő berendezés üzemel. Az ezzel a becsléssel nyert információ segíti a berendezés teljesítményének optimális megválasztását. Lényeges feltétel a feladat megoldásánál, hogy az alkalmazott hardver bekerülési költsége nem haladhatja meg a 40-50 Eurós nagyságrendet, mivel a kész épületgépészeti termék árát nem emelheti meg jelentősen a beépített intelligencia.

* Kapcsolattartó szerző.
E-mail cím: kovacs.tamas@nje.hu

Közel húsz évvel ezelőtt publikált Kim és munkatársai [1] egy személy-azonosító és követő rendszert, mely egy biztonsági kapura rögzített kamera képét dolgozta fel, az akkori technikai háttér mellett kb. 10 fps (frames per second) sebességgel, ami valós idejűnek tekinthető. A cél a kapu előterében áthaladó emberek számlálása irány szerint is elkülönítve.

Ez az alkalmazott séma azóta sok hasonló, nem „deep learning" alapú megoldásnak lett a kiinduló pontja. Hsieh és munkatársai [2] is ezt a sémát alkalmazták több javítással: a háttérbecslést egy statisztikai módszerrel végezték (Gaussian Mixture Pixel Probability, kevert Gauss eloszlás módszere) így az jóval robusztusabb lett. Az előtér meghatározásnál pedig egy hiba-javító algoritmust alkalmaztak, amely kiszűri a kamera kisebb rezgéseiből adódó zavart. Emellett a háttérkivonás után a háttérkép változó árnyékait is megbecsülték és eliminálták. Az ezt követő mintázat illesztés alapú objektum azonosításnál is több gyorsító módosítást vezettek be, amely sokat segített a valós idejű alkalmazásoknál.

Egy 2010-ben közzétett munkájában Hou és Grantham [3] ugyanúgy a kevert Gauss eloszlás módszerét alkalmazta a háttér kivonásnál. Ebben a munkában az azonosítandó objektumok a képen meglehetősen kisméretű ember-alakok voltak (kb. 10-15 pixel átmérőjű objektumok), azaz az objektum azonosítás jelentette a legnagyobb kihívást. Hou és Grantham a Kanade-Lucas-Tomasi (KLT) sarokpont keresési eljárást alkalmazásával jelölte ki az alakok jellegzetes pontjait, majd a klaszterezési eljárással rendelte őket egy objektumhoz. Ezen klaszterek geometriai jellemzői alapján azonosította a humán alakokat a képen (többek közt a burkoló ellipszis kis- és nagytengely méretei alapján).

2017-ben Anshari és Shim [4] a fenti feldolgozási sémát alkalmazta, de a háttérkivonás és binárissá transzformálás után egyszerű folt detektáló eljárással azonosították a humán jelenlétet, és ezzel az egyszerűsítéssel alkalmassá tették az algoritmust arra, hogy olyan kisebb teljesítményű beágyazott hardveren is működjön, mint a Raspberry Pi 3.

Ezek mellett az egyszerűbb heurisztikus megoldások mellett kidolgozásra kerültek betanítást igénylő (deep learning) osztályozási módszerek is. Ilyen például Al-Zaydi és munkatársai megoldása [5], ahol egy tanítható regressziós modell alapján állapítják meg a képen látható személyek számát, amelynél a tanítást a Gauss Process Regression eljárás alapján végezték. Wahyuni és munkatársai [6] pedig szintén egy tanítható Support Vector Machine osztályozót használtak az objektum azonosításhoz.

Kanatov [7] és egy évvel később Nogueira [8] a manapság igen elterjedten alkalmazott konvolúciós neurális háló (CNN) alapján történő objektum felismerési technikát használták, amely a betanítási folyamatban nagy számítási kapacitást igényel ugyan, de az ismert CNN együtthatók alapján a rendszer implementálható egyszerűbb hardver eszközökre is.

A továbbiakban ezeknek a módszereknek olyan implementációit mutatjuk be, amelyek működőképesek lehetnek egy olyan beágyazott rendszeren, amely ára a már említett 40-50 Eurós nagyságrendet nem haladja meg.

## 2. A feladat részletei

Ahhoz, hogy a légfertőtlenítő berendezés megfelelő információt kaphasson, valós idejű, és 1 frame per másodpercnél nem ritkább robosztus mérést kell megvalósítani. A használható szenzor: egy darab fixen felszerelt és fix tengelyű RGB szenzor és képrögzítő (kamera), amely lehetőleg a falon elhelyezett berendezésen van telepítve. A képfeldolgozás során a kamera képet már szürkeárnyalatos formátumban olvassuk ki a kamera pufferéből a memóriával és a CPU idővel való takarékoskodás céljából.

A célkitűzés megfogalmazásában a megoldások pontosságát tekintve 10-20%-os mérési hibát még elfogadhatónak tekintünk. Ebben a cikkben a pontosság szisztematikus mérésével nem foglalkozunk, mert ehhez egy nagyságrenddel nagyobb számú tesztre lenne szükség, mint ami itt bemutatásra kerül. A bemutatott tesztek eredményei csupán előzetes információt szolgáltatnak a megoldások használhatóságát és a fejlesztési irány helyességét illetően.

A konkrét eszköz, amellyel az itt bemutatott teszteket végeztük egy ESP-EYE nevű mikrovezérlő kártya (lásd az 1. ábrán), amely egy OV2640 2 megapixeles kamerát, egy ESP32 chipet, 8 MB PSRAM-ot és 4 MB Flash memóriát tartalmaz. Ez a kis költségű kártya sajnos nem alkalmas arra, hogy olyan robosztusan működő, általános célú objektum-felismerő rendszereket

futtasson, mint például a YOLO típusú rendszerek [9] vagy Kanatov rendszere [7, 8]. Készíthető rá ugyan neurális háló alapú osztályozó alkalmazás, de ennek képességei messze elmaradnak a PC-re vagy Raspberry-re szánt modellekétől. Tesztelési céllal készítettünk egy ilyen (háttér vagy személy) osztályozó alkalmazást, amelyet a COCO128 kép adatbázis segítségével tanítottunk be. Ezek a kezdeti teszteredmények mind a pontosság, mind a gyorsaság tekintetében sokkal gyengébbnek bizonyultak, mint a YOLO modell tesztjei. További fejlesztéssel és optimalizálással javíthatóak ezek az eredmények, de valószínű, hogy a deep-learning módszerek ilyen hardveren csak egyéb algoritmusokkal kombinálva működnek elfogadhatóan. Ennek a kutatási vonalnak a bemutatása túllépné a jelen cikk kereteit, így itt csak ennyit írunk róla.

A továbbiakban egy olyan algoritmust mutatunk be részletesen, amely alapvetően Kim és munkatársai [1] algoritmusát követi abban, hogy mozgás-detekció alapján különíti el a hátteret és az előtér objektumokat, és azzal a feltevéssel élünk, hogy a gyorsan változó, azaz mozgó objektumok egy iroda-helyiségben többnyire mind személyek. Emellett az algoritmus a szerény hardver miatt még a mozgás-detekciós részben is számos ponton eltér az eredeti algoritmustól.


*1. ábra: Az ESP-EYE kamerás beágyazott rendszer*

## 3. A mozgás-detekciós módszer

Ahogy fentebb is írtuk, a jelen probléma megoldásához választott algoritmus első lépése megegyezik Kim módszeréve [1]. Ebben a lépésben két kamera-kép, $P_t$ és $P_{t+\Delta t}$, különbségének abszolút értékét vesszük, amelyek egymás után fix $\Delta t$ idővel lettek rögzítve (az indexek az időt jelölik). A küszöbölés során egy rögzített globális $K$ küszöb alatti értékek helyére 0-t, a többi érték helyére 255-öt írunk. Ezután a két transzformáció után a kapott bináris differencia-kép:

$$D_t(x,y) = \begin{cases} 0, & ha \ |P_{t+\Delta t}(x,y) - P_t(x,y)| < K \\ 255, & egyébként \end{cases} \quad (1)$$

Az $x$ és $y$ a pixelek koordinátáit jelölik. A $\Delta t$ idő-eltolás paraméterrel a mozgás sebességére való érzékenységet állíthatjuk: nagyobb értékekre a módszer lassabb mozgásokra is érzékeny, míg a kisebb értékeknél csak a gyors mozgások látszanak. A $K$ küszöb a módszer általános érzékenységét állíthatjuk: kisebb küszöbnél több fehér (255-ös értékű) pixelt kapunk, de a kapott zaj is nagyobb lesz, amit majd szűrnünk kell.

A zaj és a kisebb jelentéktelen fehér (255) klaszterek szűrését egy morfológiai nyitással (erózió majd dilatáció) oldjuk meg, ahol a struktúra-elem egy 3x3-as méretű pixel négyzet. A morfológiai zárás szintén hasznos lenne az algoritmusban, de ennek szerepét jórészt átveszi a következő, felbontás-csökkentéses lépés, így költség-takarékossági okból elhagytuk.

A felbontás csökkentés másik fontos szerepe az, hogy az eztán következő klaszterezési eljárás számítási igényét jelentősen csökkenti. Ez egy beágyazott rendszer esetében lényeges szempont lehet. Az eljárás során az eredeti *(W, H)* méretű képből egy *(W/B, H/B)* méretű képet kapunk úgy, hogy az eddig feldolgozott képet *BxB* méretű diszjunkt pixel-blokkokra osztjuk, majd az egyes blokkokban található fehér pixelek száma alapján, egy egyszerű küszöböléssel eldöntjük, hogy a blokkhoz rendelt pixel érték 0 vagy 255 lesz. Matematikai formulával leírva az eredmény kép:

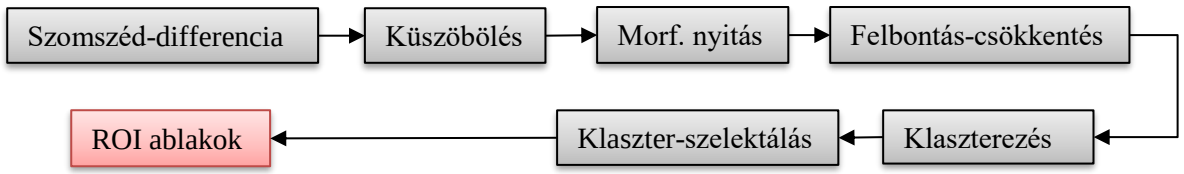
$$S_t\left(\left[\frac{x}{B}\right],\left[\frac{y}{B}\right]\right)=\begin{cases}0, & ha\ \sum_{\left[\frac{x}{B}\right]\le\frac{x}{B}<\left[\frac{x}{B}\right]+1,\ \left[\frac{y}{B}\right]\le\frac{y}{B}<\left[\frac{y}{B}\right]+1}\frac{D_t(x,y)}{255}<K\\ 255, & egyébként\end{cases} \quad (2)$$

A klaszterezést a fehér pixelekre végezzük el, a szomszéd-távolság módszerrel, azaz: azok a pixelek tartoznak azonos klaszterhez, amelyek közelebb vannak egymáshoz, mint egy fix R távolság-érték. Itt az R-t szigorúan választjuk: R=1.01, azaz az oldalban szomszédos pixelek tartoznak azonos klaszterhez. Ez az érték-választás azért célszerű, mert a felbontás-csökkentés alaposan csökkentette a klaszter pixelek számát és a köztük lévő távolságot is.

Ezt követően el kell döntenünk, hogy a kapott klaszter mérete és alakja alapján egy mozgó személyt jelöl-e. A jelen munkában egyelőre az egyszerűségre törekszünk, így a klaszterek leírására csupán a klaszter tömegét (azaz a benne lévő fehér pixelek számát) ($C_M$) továbbá a körülírt téglalap szélességét és magasságát ($C_w$, $C_H$) használjuk. Mivel elsősorban álló vagy ülő személyekre számítunk, feltesszük, hogy $C_w \le C_H$ , emellett a kis tömegű klasztereket figyelmen kívül hagyjuk, azaz feltételként szabjuk, hogy $C_M \ge 2$. Ezeket a feltételeket természetesen mindig az adott helyszín és kamera-beállítás figyelembe vételével kell beállítani.

A szűrés után megmaradt klaszterek körülírt téglalapjai adják a ROI (Region Of Interest) téglalap halmazt, amelyek potenciálisan személyt tartalmaznak.

A teljes eljárás összefoglalása az 2. ábrán látható.




*2. ábra: Egyszerű beágyazott rendszeren is implementálható mozgás-detektáló módszer algoritmusának blokk ábrája*

## 4. Implementáció és az első gyorstesztek

Az előző fejezetben tárgyalt algoritmus valójában egy több ciklusból álló iteratív fejlesztési munka eredménye: először egy több algoritmikus elemet tartalmazó és robusztusabban működést ígérő algoritmus készült el, majd minden ciklusban egy-egy elem kikerült az algoritmusból, vagy egyszerűbb alakban integrálódott egy másik lépésbe (pl.: a morfológiai zárás a felbontás-csökkentés lépésébe). Ezt az egyszerűsítési ciklust ismételtük, míg el nem jutottunk egy olyan alkalmazásig, amely telepíthető az ESP-EYE eszközre, működéséhez elegendő a rendelkezésre álló memória és a sebessége sem túlságosan lassú. Az implementációs során figyelembe kellett azt is venni, hogy a lefordított kód feltöltése után csak korlátozott számú kamera-kép tárolható a memóriában. Ha az előző fejezett végén említett neuronháló-modell kódja is feltöltésre kerül, akkor csupán két kamera kép fér el egyszerre a memóriában. Mi ezt a verziót implementáltuk, mivel a neuronháló alapú osztályozó modell a későbbi komplexebb verzióknál szükséges lehet. Ilyen módon a kép-transzformációs eljárások mindig ugyanazt a két kép-puffert használták, és az egyikből a másikba transzformálták a képet, úgy hogy transzformáció iránya oda-vissza változott a két puffer között, így azok egymást váltva tartalmazták a legfrissebb transzformációs eredményt.

Az érzékelő rendszer első tesztjei egy kisméretű helyiségben történtek 1-2 személy jelenléte mellett. A két egymásból kivont kamera-képet szürkeárnyalatos módban, 1024x768-as felbontással rögzítettük 0.5 másodperc eltéréssel. Az ESP-EYE-on az érzékelő alkalmazás mellett egy web-szerver alkalmazást is futtattunk, amely http csatornán elküldte a PC web-böngészőjének a legutóbbi kép-transzformáció eredményét ellenőrzés céljából. Emellett a (32x24 pixeles) felbontás-csökkentett képet kiírattuk a soros portra bekötött konzolra is. Egy ilyen teszt eredmény látható a 3. ábrán. A klaszterező eljárás ennél az esetnél két klasztert azonosít (lásd karakteres felbontású kép), azaz a

rendszer sikeresen határozza meg a személyek számát, a klaszter-szelekciós lépés itt nem szűr ki hamis pozitív klasztert. Ennél tesztnél a teszt-alanyok a felvétel pillanatában tudatosan végeztek kisebb (5-10 cm) kiterjedésű mozgást. Amennyiben ez a tudatosság nincs meg, a felvételek több mint felén nem található értékelhető klaszter. Az utolsó fejezetben erre a problémára még visszatérünk.

*3. ábra: Az mozgás-detekciós módszer demonstrációja ESP-EYE beágyazott rendszeren egy kisméretű helyiség esetén*. Balról jobbra az első az eredeti kép, a második a küszöbölt és morfológiai nyitáson átesett kép, míg a harmadik a 32x24-es felbontásra csökkentett kép a konzolon karakteresen megjelenítve. Az eredeti kép mérete 1024x768. Az alkalmazott paraméterek: *időeltolás* $\Delta$t=500ms,

## 5. Konklúzió és a további fejlesztési munka

A kutatási-fejlesztési munka jelenlegi fázisában egyelőre csak ez a néhány gyorsteszt áll még rendelkezésre, amelyet az előző fejezetben bemutattunk. A munka következő lépése lesz, hogy sokféle környezetben, személy-számmal és kamera beállítással teszt-sorozatokat készítünk és értékeljük a pontosságot és a szisztematikus hibákat. Továbbá meg kell még határozni néhány kulcsfontosságú idő-paramétert: milyen gyakran végez egy irodában dolgozó személy a rendszerünk számára is detektálható mozgást, és az mennyi ideig tart átlagosan. Ezek a paraméterek fontosak ahhoz, hogy a mintavételezés gyakoriságát és a kiértékelés statisztikai módszerét beállíthassuk.

Mindemellett az első tesztekből látható, hogy a módszer működőképes lehet egy olyan alacsony költségű mikrovezérlő kártyán is, mint az ESP-EYE. Ha a bemutatott mozgás-detekció alapú módszer önmagában nem hozna az adott feladatban megkövetelt pontosságot, akkor lehetőség van még a neuronháló alapú személy-háttér osztályozó modell felhasználására is, ám ez a megoldás sebességét jelentősen csökkentheti.

Az is jól látszik ezekből a tesztekből, hogy az alacsony költségű, gyenge optikájú illetve felbontású beépített kép-szenzorok nem lesznek alkalmasak egy nagyobb teremben (pl. előadó-teremben vagy nagyobb tanteremben) lévő nagyobb létszám meghatározására, mivel az ember-alakok gyenge pixel-felbontása ezt lehetetlenné teszi.

## Köszönetnyilvánítás

A szerzők köszönetet mondanak a projektben résztvevő intézmények - AIRVENT ZRT és a Neumann János Egyetem GAMF Műszaki és Informatikai Kar - kollégáinak. Köszönettel tartozunk a kutatás támogatásáért, amely a “ Széles körben használható levegő sterilizáló megoldások kifejlesztése intelligens működés optimalizáló vezérléssel 2020-1.1.2-PIACI-KFI-2021-00294 " pályázat keretében valósult meg. A projekt a Magyar Állam és az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával, a Széchenyi 2020 program keretében valósul meg.

## Irodalomjegyzék

[1] Kim, Jae-Won, et al. "Real-time vision-based people counting system for the security door." Proceedings of the IEEK Conference. The Institute of Electronics and Information Engineers, 2002.

[2] Hsieh, Jun-Wei, Cheng-Shuang Peng, and Kao-Chin Fan. "Grid-based template matching for people counting." 2007 IEEE 9th Workshop on Multimedia Signal Processing. IEEE, 2007. DOI: 10.1109/MMSP.2007.4412881

[3] Hou, Ya-Li, and Grantham KH Pang. "People counting and human detection in a challenging situation." IEEE Transactions on Systems, Man, and Cybernetics-part A: Systems and Humans 41.1 (2010): 24-33. DOI: 10.1109/TSMCA.2010.2064299

[4] Ansari, Md Israfil, and Jaechang Shim. "People Counting System using Raspberry Pi." Journal of Multimedia Information System 4.4 (2017): 239-242.

[5] Al-Zaydi, Zeyad, Branislav Vuksanovic, and Imad Habeeb. "Image processing based ambient context-aware people detection and counting." International Journal of Machine Learning and Computing 8.3 (2018): 268-273. DOI: 10.18178/ijmlc.2018.8.3.698

[6] Wahyuni, Elvira Sukma, Rizqi Renafasih Alinra, and Hendra Setiawan. "People counting for indoor monitoring." 2017 International Conference on Computing, Engineering, and Design (ICCED). IEEE, 2017. DOI: 10.1109/CED.2017.8308112

[7] Kanatov, Maksat, and Lyazzat Atymtayeva. "Deep convolutional neural network based person detection and people counting system." Advanced Engineering Technology and Application 7.3 (2018): 9-16. DOI 10.21608/aeta.2018.200330

[8] Nogueira, Valério, et al. "RetailNet: A deep learning approach for people counting and hot spots detection in retail stores." 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). IEEE, 2019. DOI: 10.1109/SIBGRAPI.2019.00029

[9] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018). https://github.com/ultralytics/yolov5. DOI 10.48550/arXiv.1804.02767

[10] Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer International Publishing, 2014., https://doi.org/10.48550/arXiv.1405.0312