

## SISTEM IOT DISTRIBUIT PENTRU PROCESAREA IMAGINILOR

Andrei CHIPER<sup>1\*</sup>, Mihai-Petre OPROIU<sup>2</sup>

<sup>1</sup>Inginerie electronică, telecomunicații și tehnologii informaționale, grupa 4LF602, Facultatea de Inginerie electrică și știința calculatoarelor, Universitatea Transilvania, Brașov, România

<sup>2</sup>Facultatea de Inginerie electrică și știința calculatoarelor, Universitatea Transilvania, Brașov, România

\*Autorul corespondent: Andrei Chiper, [andrei.chiper@student.unitbv.ro](mailto:andrei.chiper@student.unitbv.ro)

**Rezumat.** Scopul acestei lucrări constă în utilizarea unui kit de dezvoltare cu un procesor grafic care să realizeze detecție de obiecte cu răspuns audio în favoarea persoanelor nevăzătoare și implementarea unui program eficient din punct de vedere a resurselor, toate acestea la un buget cât mai mic. Se propune folosirea unei camere web, a unui set de căști și a modului NVIDIA Jetson Nano, capabil să îndeplinească cerințele aplicației, și se caută integrarea unor metode folosite de a efectua detecția, fără a fi nevoie de achiziționarea unor echipamente în plus.

**Cuvinte cheie:** inteligență artificială, învățare automată, rețea neurală, NVIDIA Jetson Nano, redare audio

### Introducere

Tehnologia a devenit din ce în ce mai avansată cu ajutorul inteligenței artificiale, dar totodată și costul investițiilor a crescut datorită acestei evoluții, lucru care este observabil în toate domeniile de lucru, ele fiind afectate și de alți factori cum ar fi înaintarea în vârstă a populației cu efecte asupra forței de muncă, schimbarea preferințelor datorată cererilor mai mari de la diferiți furnizori [1].

Inteligența Artificială reprezintă abilitatea unui computer de a efectua sarcini în mod asemănător unei ființe înzestrată cu inteligență naturală, inteligență umană. Un sistem cu inteligență artificială trebuie să fie capabil să rezolve problemele pentru care a fost pregătit și să învețe din greșelile experiențelor anterioare.

Detecția de obiecte are anumite aspecte de luat în considerare, cum ar fi: acuratețea modelului folosit, resursele limitate, timpul de răspuns al sistemului când aplicația este executată, rezoluția camerei folosite, mediul de testare a aplicației. În acest sens, prin eficientizarea modului de lucru și utilizarea unor resurse limitate, companiile pot beneficia de progresul tehnologic.

### Kitul de dezvoltare NVIDIA Jetson Nano și învățarea automată

NVIDIA Jetson Nano este un sistem pe modul (SoM) integrat și kit de dezvoltare din familia NVIDIA Jetson, care are în componența sa un procesor grafic cu 128 de nuclee Maxwell, procesor ARM A57 64-bit și memorie RAM de 4GB LPDDR4. Kitul permite rularea a mai multor rețele neurale în paralel pentru aplicații precum clasificare de imagini, detectare obiecte, segmentarea și procesarea vorbirii și oferă o putere de calcul de 472 GFLOPS cu un consum de putere între 5 și 10 wați [2]. Jetson Nano dispune de un sistem de operare bazat pe Linux Ubuntu, special dezvoltat pentru kit, care vine cu utilități și pachete folosite de dezvoltatorii de programe.

Învățarea automată (Machine Learning, ML) este o ramură a informaticii în continuă evoluție care are ca obiectiv dezvoltarea unor algoritmi informatici pentru a simula inteligența umană, algoritmi care folosesc mediul înconjurător pentru a învăța. Are la bază inteligența artificială, dar se bazează și pe idei din diferite discipline cum ar fi, probabilitate și statistică, psihologie, filozofie.

Algoritmii de învățare automată folosesc un set de date de intrare pentru a realiza o sarcină dorită fără a mai fi programați să producă doar un anumit rezultat. În schimb, aceștia sunt programați să se adapteze, modeleze automat prin repetare (numite experiențe) în așa fel încât să devină din ce în ce mai buni la atingerea rezultatului dorit [3].

## Implementarea codului folosind limbajul de programare Python

Alegerea limbajului de programare Python este datorat utilizării ușoare pentru începători, dar și experți. Versatil și puternic, este foarte bun pentru machine learning și inteligență artificială, deoarece algoritmi folosiți sunt lizibili.

În realizarea programului au fost folosite mai multe biblioteci, printre cele mai importante fiind modulele jetson.inference și jetson.utils din repository-ul de pe GitHub numit jetson-inference, responsabile cu recunoașterea de imagini și încărcarea imaginilor, biblioteca Numpy care conține o colecție mare de funcții matematice de nivel înalt, OpenCV pentru vedere computerizată în timp real (afișare video) și gTTS (Google Text-to-Speech) pentru interfața cu API-ul text to speech al Google Translate. (ieșire, răspuns audio)

Codul a fost implementat cu ajutorul repository-ului menționat mai sus. Informațiile preluate au fost analizate, studiate, mai apoi urmând implementate.

Modelul de rețea neurală folosită în această lucrare este SSD-MobileNet V2 antrenat pe setul de date COCO (Common Objects in Context) [4], model ales pentru eficiența acestuia (Fig. 1), deoarece poate rula detecția cu 27 de cadre pe secundă.

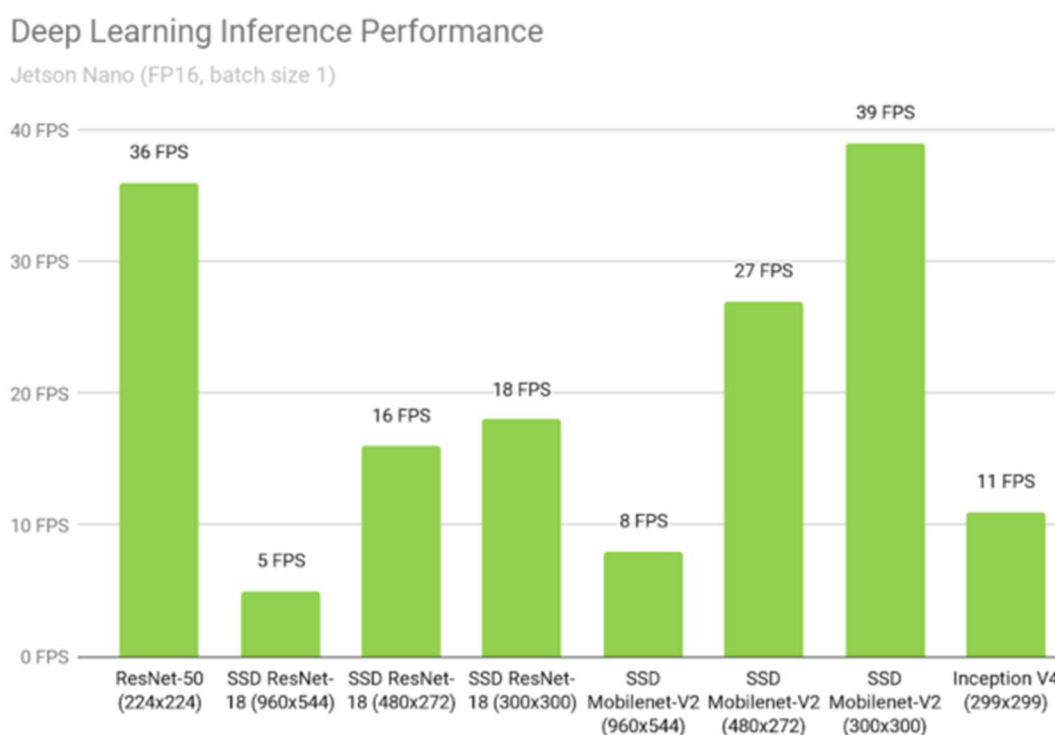


Figura 1. Comparație între performanța atinsă de diferite modele

Pentru a defini funcția din Fig. 2, care redă numele obiectului, este nevoie de un thread care să ruleze concomitent cu detecția de obiecte, înainte de a porni camera. Funcția trebuie să ia la cunoștință funcționalitatea programului, să împartă variabilele cu restul programului, prin urmare și definirea celor două variabile globale speak și item.

Thread-ul rulează funcția doar o dată apoi se încheie. Folosind instrucțiunea repetitivă while, thread-ul rămâne în funcționare până la întreruperea programului. Instrucțiunea if are rol de a întrerupe repetarea aceluiași cuvânt în cazul în care obiectul detectat este asemănător celui anterior.

```
def sayItem():
    global speak
    global item
    while True:
        if speak == True:
            output = gTTS(text = item, lang = 'en', slow = False)
            output.save('output.mp3')
            os.system('mpg123 output.mp3')
            speak = False

x = threading.Thread(target = sayItem, daemon = True)
x.start()
```

Figura 2. Funcția pentru ieșirea audio gTTS

După definirea funcției pentru ieșirea audio, programul trebuie să recunoască obiectele, când vorbirea este încheiată acesta să treacă la faza de identificare a unui obiect nou. Apare mențiunea că nu se dorește ca programul să redea audio ce detectează dacă nu are o certitudine stabilită. În Fig. 3 sunt reprezentate instrucțiunile folosite, iar programul va reda sunetul doar când numele ultimului obiect se schimbă așa cum se observă în ultima instrucțiune if.

```
if speak == False:
    confidence = round(d.Confidence, 3)
    if confidence >= .9:
        item = self.net.GetClassDesc(d.ClassID)
        if item != itemAux:
            speak = True
    else:
        item = ""
    itemAux = item
```

Figura 3. Faza de recunoaștere a obiectelor după vorbire

### Acces la distanță, protocolul RTSP și rezultate obținute

Cu ajutorul programului VNC Viewer modulul se poate conecta prin protocolul Secure Shell (SSH) la distanță de pe un alt dispozitiv pentru a lucra pe acesta, fără a utiliza prea multe conexiuni cablate în plus și pentru un aspect mai curat în zona de lucru. S-au realizat două teste pentru a verifica detecția de obiecte: local prin intermediul unei camere web conectate prin interfața USB la modul și conectat cu ajutorul protocolului RTSP de la un dispozitiv extern la modul.

Protocolul RTSP a fost folosit pentru a transmite video informația de la camera video a dispozitivului la modul, aceasta urmând să fie prelucrată, însă rezultatele nu au fost prea mulțumitoare, întrucât conexiunea nu putea fi stabilită sau doar întâmpinând erori, latențe mari între conexiunea dintre dispozitive de câteva secunde.

Prin intermediul camerei web rezultatele obținute au fost cele așteptate, mulțumitoare, se poate calibra rata de detecție cu mici reglaje a certitudinii în cod. Numărul de cadre pe secundă (CPS) înregistrat folosind echipamentele puse la dispoziție variază între de 23 și 25 CPS.

## **Concluzii**

Printr-o îmbunătățire a datelor transmise prin intermediul protocolului RTSP, se pot obține aplicații în timp real folosite pentru diferite lucruri cum ar fi detecție de obiecte pentru persoane nevăzătoare sau aproape oriunde în domeniul medical, dar și din alte diferite domenii, geo locație, entertainment, securitate.

## **Referințe**

1. Niek Stadhouders, Florian Kruse, Marit Tanke, Xander Koolman, Patrick Jeurissen, Effective healthcare cost-containment policies: A systematic review [online]. 2019, 123, pp. 71-79. [accesat 12.02.2022]. Disponibil: <https://doi.org/10.1016/j.healthpol.2018.10.015>
2. Jetson Nano [online]. [accesat 12.02.2022]. Disponibil: [https://elinux.org/Jetson\\_Nano](https://elinux.org/Jetson_Nano)
3. El Naqa, Issam and Murphy, Martin J. , Machine Learning in Radiation Oncology: Theory and Applications [online]. 2015, pp. 3-11. [accesat 12.02.2022]. Disponibil: [https://doi.org/10.1007/978-3-319-18305-3\\_1](https://doi.org/10.1007/978-3-319-18305-3_1)
4. Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár, Microsoft COCO: Common Objects in Context [online]. 2014, pp. 740-755. [accesat 16.02.2022]. Disponibil: <https://arxiv.org/abs/1405.0312>