

BlindSight

Documentație tehnică

Pop Răzvan, Kovacs Norbert

20 iulie 2025

Rezumat

BlindSight propune un sistem software și hardware menit să faciliteze deplasarea în siguranță a persoanelor nevăzătoare prin mediul urban. Partea software rulează direct pe telefonul utilizatorului și execută inferențe locale, în timp real, cu ajutorul unui model YOLOv11l convertit în format *TensorFlow Lite*. Aplicația identifică semafoarele, trecerile de pietoni, stațiile de autobuz, farmaciile și semnele pentru traversare. De îndată ce un element relevant intră în câmpul camerei, aplicația transmite un mesaj audio preînregistrat și, simultan, descrie obiectul prin sinteză vocală.

Componenta hardware constă într-o centură imprimată 3D folosind o imprimantă 3D BambuLab P1S, în care sunt montați **3** senzori ultrasonici HC-SR04 controlați de un microcontroler ESP32. Dispozitivul comunică prin Bluetooth Serial cu telefonul și transmite continuu distanța față de obstacolele detectate la mai puțin de **50** de centimetri. La recepția unui prag critic, aplicația declanșează o vibrație de intensitate variabilă și repetă distanța aproximată.

Întregul proiect îmbină astfel **3** dimensiuni: percepție la distanță (vizuală), percepție de proximitate (ultrasunete) și interfață auditiv-haptică accesibilă. Rezultatul oferă utilizatorului o experiență fluidă, fără dependență de internet, cu timpi de răspuns sub **200** de milisecunde și cu un cost total al echipamentului adițional sub **30** de euro.

1 Introducere, motivație și scop

Raportările Organizației Mondiale a Sănătății indică peste **300** de milioane de persoane cu deficiențe severe de vedere, dintre care majoritatea trăiesc în zone urbane cu infrastructură marcată de trafic intens, aglomerată și obstacole imprevizibile. Pentru un nevăzător, traversarea unei intersecții sau recunoașterea unei stații de autobuz reprezintă încă un exercițiu de curaj susținut prin memorie, baston, însoțitori ori aplicații care descriu scene generale, însă cu latențe mari și adesea condiționate de conexiunea la rețea. Observarea acestei realități sociale a generat în cadrul echipei BinarySquad ambiția de a crea o soluție integral locală, ușoară, prietenoasă și ieftină, capabilă să ofere feedback instantaneu în situațiile critice ale deplasării urbane.

Motivația proiectului derivă în primul rând din faptul că smartphone-urile moderne dispun de procesoare suficient de puternice pentru a rula modele de detecție de obiecte fără a consuma date mobile sau a expune utilizatorul la lipsa de semnal. În al doilea rând, senzorii ultrasonici au ajuns la prețuri modice, iar combinația lor cu un ESP32 permite un canal bidirecțional de date cu un consum energetic redus. În al treilea rând, legislația accesibilității recomandă soluții care nu necesită intervenții majore în oraș, iar un dispozitiv portabil răspunde acestei cerințe.

Scopul BlindSight este, așadar, creșterea autonomiei prin furnizarea în timp real a celor mai relevante informații spațiale: culoarea semaforului, prezența unei treceri de pietoni, apropierea unui obstacol la nivelul trunchiului sau al picioarelor și identificarea unor repere precum stația de transport public ori farmacia. Funcțiile sunt concepute să lucreze simultan, iar mesajele sunt sintetizate într-o manieră minimală pentru a nu oferi prea multă informație în același timp.

1.1 Delimitare conceptuală

BlindSight nu este o aplicație de navigare *turn by turn* și nu concurează cu hărțile tradiționale. Proiectul se concentrează pe micro-orientare, adică pe acele detalii care fac diferența între un traseu reușit și un incident: culoarea reală a semaforului pietonal, bordura nevopsită, coșul de gunoi care obstrucționează trotuarul ori mașina parcată neregulamentar. În același timp, sistemul nu solicită un server extern; datele vizuale rămân pe telefon, protejând confidențialitatea utilizatorului.

1.2 Structura documentației

Prezentul material detaliază mai întâi contextul și fezabilitatea soluției comparativ cu alternativele existente; apoi descrie tehnologiile software și hardware utilizate; continuă cu o discuție asupra codului sursă și se încheie cu reflecții asupra limitărilor curente și a direcțiilor de dezvoltare.

2 Soluții similare

În ultimul deceniu au apărut mai multe produse și inițiative destinate orientării persoanelor nevăzătoare. Seeing AI, dezvoltat de Microsoft, folosește servicii în cloud pentru a descrie scene, texte și bancnote; avantajul său este recunoașterea vocală sofisticată, dar dezavantajul rămâne dependența de rețea și imposibilitatea de a estima distanțe exacte

până la obstacole. Be My Eyes, o aplicație susținută prin voluntariat, intermediază convorbiri video cu persoane văzătoare dispuse să ofere ajutor *ad-hoc*; soluția este socială și empatică, însă inefficientă atunci când utilizatorul are nevoie de feedback la fiecare secundă.

Pe segmentul hardware dedicat, OrCam MyEye integrează o cameră montată pe rama ochelarilor și un mini-difuzor direcționat către ureche; dispozitivul recunoaște text și fețe, dar costă peste **2 500** de euro și nu oferă integrare cu senzori de proximitate. Sisteme precum Wayfindr propun standarde de semnalizare audio pentru navigația indoor, bazate pe beaconi instalați în clădiri; ele depind de amenajarea spațiilor publice, proces costisitor și lent.

BlindSight se distinge de aceste proiecte prin două elemente. În primul rând, modelul de detecție rulează local, fără latența și consumul energetic asociate upload-ului de imagini. În al doilea rând, aplicația combină două modalități complementare: detectează repere, iar ultrasunetele confirmă obstacole imediat periculoase.

3 Tehnologii utilizate

Dezvoltarea BlindSight a necesitat o suită de instrumente software și platforme hardware. Codul aplicației este scris preponderent în limbajele Java și Kotlin, împreună cu fișiere de layout XML gestionate în Android Studio. Legarea modelului de detecție se realizează prin mecanismul oficial *ML Model Binding*, care generează automat o clasă de tip wrapper pentru invocarea inferențelor. Conversia din arhitectura YOLOv11l către *TensorFlow Lite* a fost efectuată în Python, folosind un notebook de Google Colab cu extindere pentru optimizarea GPU.

Datasetul, compus din aproximativ **7 000** de imagini etichetate, a fost organizat în Roboflow, unde au fost aplicate augmentări precum variații de luminozitate. Ulterior, modelul a fost antrenat pe o instanță Colab cu GPU T4; procesul a durat în jur de **6** ore pentru **200** de epoci, obținându-se o precizie mAP 50 de peste 88 %.

Din punct de vedere hardware, telefonul rulează Android 12, dispune de **4** gigaocteți de memorie RAM și un procesor ARM Cortex-A53; aceste specificații acoperă marea majoritate a dispozitivelor mid-range lansate după 2021. Elementele de pe centură au fost imprimate pe un BambuLab P1S folosind PETG negru pentru flexibilitate și rezistență la uzură. Piese au fost modelate în ThinkerCad, care oferă parametri simpli de extrudare și dimensiuni exacte pentru prinderea senzorilor. Microcontrolerul ales, ESP32 DevKit, furnizează **2** nuclee la **240** de megaherți și un modul Bluetooth Classic integrat, suficient pentru throughput-ul de câțiva octeți pe secundă cerut de senzorii HC-SR04. În fine, macheta semaforului, utilizată în faza de testare, este controlată de un Arduino UNO care alimentează două LED-uri și simulează schimbarea culorii la intervale predefinite pentru un semafor.

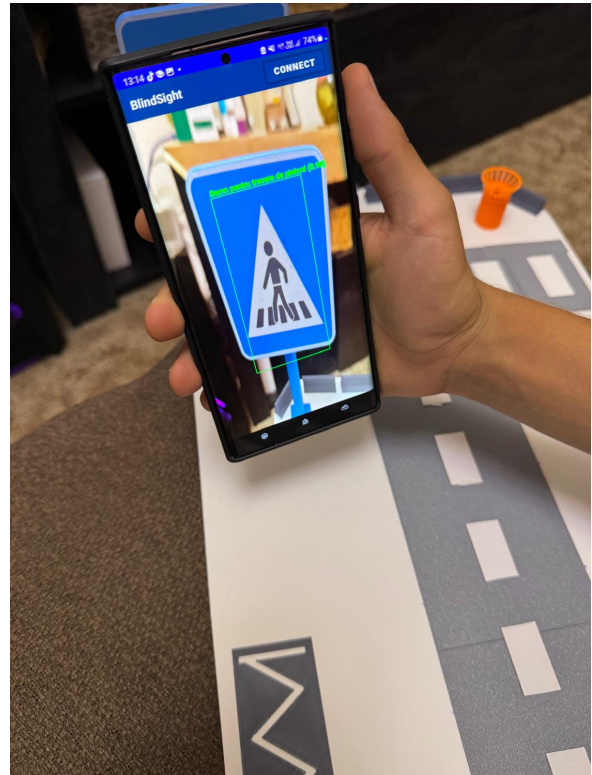
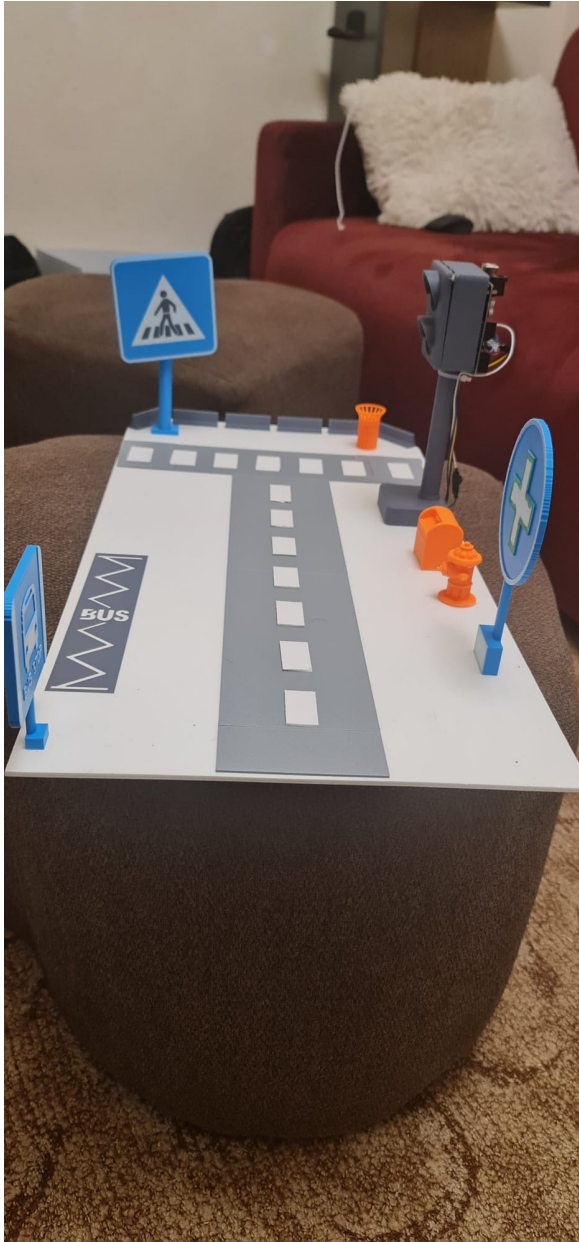


Figura 1: Macheta de test și interfața real-time a aplicației BlindSight.

4 Descrierea codului

4.1 Aplicația Android

Structura logică a aplicației începe cu *MainActivity*, componenta centrală responsabilă de inițializarea interfeței, stabilirea conexiunii Bluetooth și gestionarea ciclului de viață al detecției vizuale. La lansare, activitatea face un *binding* către camera dispozitivului, iar fiecare cadru este direcționat către modelul *TensorFlow Lite*. Rezultatul inferenței constă într-o listă de etichete și scoruri de încredere; dacă un scor depășește pragul de **0,90**, logica declanșează un fișier audio preînregistrat, de exemplu „trecere de pietoni în față”, și inițiază o scurtă sintagmă TTS care confirmă clasa atingând un ton ascendent pentru a semnaliza certitudinea.

Pentru interacțiunea cu nevăzătorii, echipa a optat pentru gesturi simple de tip *swipe*.

Fiecare glisare orizontală de la stânga la dreapta deschide un meniu cu setări, iar direcția inversă expune lista dispozitivelor Bluetooth detectate. O glisare verticală în sus afișează un tutorial audio, iar o glisare cu două degete revine în modul de cameră. Timpii aferenți gesturilor sunt delimitați de un detector de gesturi care filtrează viteza și distanța mișcării pentru a evita declanșările accidentale.

Conexiunea Bluetooth folosește profilul SPP (Serial Port Profile), iar adresa dispozitivului ESP32 este salvată într-un fișier pentru reconectare rapidă la pornirile ulterioare. Un fir de execuție secundar ascultă permanent fluxul de octeți venit de la centură. Mesajele sunt formate din perechi „*S:id distanță*” și se termină cu caracterul de linie nouă, ceea ce permite parsarea ușoară. Dacă distanța raportată scade sub **50** de centimetri, firul notifică firul principal, iar acesta comandă vibratorul sistemului să emită un impuls de **500** de milisecunde urmat de o pauză scurtă. Intensitatea este variată proporțional cu apropierea; mai puțin de **20** de centimetri produce o vibrație continuă de **2** secunde.

Pentru a menține consumul de energie în limite rezonabile, inferența vizuală este suspendată atunci când ecranul devine negru sau când utilizatorul blochează telefonul, iar firul de Bluetooth continuă să ruleze, oferind astfel protecție de proximitate chiar și cu ecranul stins.

4.2 Aplicație ESP32

Codul microcontrolerului este scris în C++ folosind framework-ul oficial Arduino pentru ESP32. Fiecare ciclu de execuție pornește prin trimiterea unui impuls de **10** microsecunde pe pinul TRIG al senzorului curent, după care măsoară durata impulsului de ecou. Durata se convertește în distanță prin înmulțirea cu viteza sunetului împărțită la 2; rezultatul este formatat cu o zecimală și transmis prin portul serial Bluetooth. Pentru a nu suprasolicita magistrala, ciclul se repetă la **5** secunde, valoare optimă determinată empiric: mai rapid de atât ar dubla consumul energetic, iar mai lent ar întârzia reacția la pericole. Cele **3** module HC-SR04 sunt orientate ușor divergent, unul drept înainte și celelalte două la unghiuri de aproximativ **45** de grade, acoperind astfel un câmp orizontal de aproape **120** de grade.



Figura 2: Semaforul de test controlat de Arduino, capturat în faza de lumină roșie.

4.3 Controlerul semaforului de test

Pentru a verifica detecția culorilor, a fost construită o machetă la scară redusă: un mic postament imprimat 3D pe care sunt montate două LED-uri de **5** milimetri, respectiv roșu și verde. Arduino-ul UNO alimentează pe rând câte un LED, cu verdele pornit timp de **5** secunde, pauză de **100** de milisecunde pentru tranziție, apoi roșul activ vreme de **10** secunde. Camera telefonului a fost poziționată la aproximativ 50 cm distanță, iar scorurile de detecție au depășit consistent pragul de **90** la sută chiar în condiții de iluminare neuniformă. Experimentul a confirmat corectitudinea metodelor de augmentare aplicate în Roboflow și a permis calibrarea balanței de alb a camerei pentru a reduce fals-pozitivele în lumina artificială din interior.



Figura 3: Semaforul de test controlat de Arduino, capturat în faza de lumină roșie.

4.4 Flux de date consolidat

În momentul în care utilizatorul intră într-o intersecție, aplicația primește simultan două fluxuri: unul video de la cameră și unul de date textuale de la senzorii ultrasonici. Pipeline-ul video este segmentat astfel: pre-procesare (redimensionare la 320×320 , normalizare), inferență, post-procesare (filtrare scor, non-max suppression), generare mesaj. Pipeline-ul ultrasonic însumează citirea bufferului, parsarea distanței și compararea cu pragul fix. Ambele ajung în modulul de feedback. Dacă cele două evenimente se suprapun (de pildă, semafor roșu și obstacol la **30** de centimetri), logica prioritizează mesajul referitor la semafor, apoi, după un semnal sonor scurt, citește distanța. Astfel se evită coliziunile audio și se păstrează contextul vital al traversării.

5 Concluzii și direcții viitoare

BlindSight validează ipoteza că algoritmi de detecție a obiectelor pot fi rulați local pe telefoane mid-range fără a sacrifica precizia, iar senzorii ultrasonici, deși simpli, completează lacunele inevitabile ale viziunii artificiale în unghiuri moarte sau pe timp de noapte. Experimentele pe machetă, urmate de teste în oraș, arată un timp de reacție de sub

200 de milisecunde și o rată de fals-pozitive sub **5** procente pentru semafoare și treceri de pietoni.

Provocările rămân semnificative. În primul rând, datasetul actual nu acoperă suficiente variații de anotimp și condiții de ploaie, iar plăcuțele de stație de autobuz din alte orașe pot avea altă cromatică. În al doilea rând, modelul YOLOv11l ocupă aproape **40** de megaocteți; deși rulează rapid, pachetul APK ajunge la limita permisă de anumite magazine de aplicații. În al treilea rând, feedback-ul audio simultan pentru două evenimente apropiate necesită o coregrafie fină, altfel utilizatorul poate pierde informația critică.

Pe termen scurt, echipa își propune extinderea datasetului printr-o campanie de *crowd-sourcing* în care voluntarii să fotografieze indicatoare și semafoare sub diverse unghiuri. În paralel, se va explora cuantizarea completă la **8** biți, ceea ce poate reduce dimensiunea modelului la mai puțin de **15** megaocteți, cu o penalizare minimă de acuratețe. Pe termen mediu, centura ar putea primi un modul IMU cu busolă digitală, oferind astfel indicații privind orientarea corpului față de traseul sugerat. Totodată, integrarea cu orarele publice de transport ar permite aplicației să anunțe sosirea autobuzului odată ce stația a fost detectată.

Rezultatul final urmărește un echilibru între simplitate, cost și funcționalitate, adresând direct nevoile persoanelor nevăzătoare și punând în practică principiul designului universal: tehnologia devine cu adevărat utilă doar atunci când poate fi folosită ușor de către oricine, indiferent de abilități.

Bibliografie

- [1] Microsoft. SeeingAI – Bringing the world to life for the blind and low-vision community. Disponibil online la <https://www.microsoft.com/en-us/ai/seeing-ai>.
- [2] Be My Eyes. The global network connecting blind and low-vision individuals with sighted volunteers. Disponibil la <https://www.bemyeyes.com/>.
- [3] OrCam. MyEye 3.0 Wearable Device. Documentație de produs, 2024. <https://www.orcam.com/>.
- [4] Wayfindr. Open Standard for Indoor Audio Navigation. White-paper, 2023. <https://www.wayfindr.net/>.
- [5] World Health Organization. World report on vision. Geneva, 2019.