

SPDAS: A YOLOv5-Powered Smart Pothole Detection and Blind Assistance System Grounded in the RDD2022 Global Road Damage Benchmark

Saad Ahmad Khan¹, Alok Gupta²

^{1,2} Department of Computer Science and Engineering
Vivekananda College of Technology and Management
Aligarh, India

Abstract

Uneven pavements, rain-filled depressions, and sunken manhole covers pose serious hazards to blind pedestrians whose white canes offer little advance warning. This paper presents the Smart Pothole Detection and Blind Assistance System (SPDAS), a smartphone-compatible framework that converts camera frames into real-time spoken navigation guidance. SPDAS fine-tunes YOLOv5s on the Road Damage Dataset 2022 (RDD2022) — a benchmark comprising approximately 47,000 annotated road images from six countries across six damage categories. A Contrast Limited Adaptive Histogram Equalization (CLAHE) stage addresses low-light limitations before inference. Detected bounding boxes are processed through a four-tier proximity classifier and a three-zone spatial engine, with hazard verdicts delivered via an offline text-to-speech layer requiring no internet access. Running on a standard CPU laptop, the pipeline achieves 28.2 FPS and a pothole-class mAP@0.5 of 73.8%. A Flutter-based Android application streams frames to a Docker-containerized Flask service hosted on Render, producing a deployment-ready mobile system. The results confirm that a lightweight detector paired with human-centred alert logic can serve as a practical, affordable road-safety aid for blind pedestrians using hardware they already own.

Keywords: pothole detection, blind pedestrian navigation, YOLOv5, RDD2022, CLAHE, offline text-to-speech, Flutter, road damage detection, real-time assistive system

1. Introduction

Walking along a busy road without sight demands a continuous act of trust — that the pavement ahead remains intact, that the crossing is level, that no hidden cavity lies in the next step. For the approximately 2.2 billion people living with vision impairment worldwide — 43 million of whom are completely blind, per WHO estimates [1] — that trust is tested with every stride. The white cane, while a valuable near-field probe, cannot detect a pothole a metre ahead, gauge its depth, or indicate whether veering left or right would be safer.

Road surface deterioration amplifies this vulnerability, particularly in rapidly urbanising regions where infrastructure maintenance lags behind expansion. India's road safety statistics reflect this problem starkly [2]. Beyond recorded injury figures, however, lies a subtler harm: blind pedestrians who learn through experience that certain streets are too hazardous to navigate independently begin to restrict their movement, progressively narrowing the radius of their daily life.

Computer vision research offers tools that could, in principle, restore some of that autonomy. Proposed solutions range from LiDAR-equipped wearable rigs producing centimetre-accurate depth maps [3] to dense semantic segmentation networks labelling every pixel in a scene. While technically sophisticated, these approaches carry trade-offs that limit real-world adoption: depth sensors are expensive, heavy, and power-hungry, while pixel-level segmentation networks struggle to sustain real-time performance on mid-range smartphones.

Single-stage detectors such as YOLOv5 occupy a practical middle ground. By framing detection as direct regression over a spatial grid, they avoid the overhead of region proposal networks while effectively localising objects across varying scales [4]. Trained on a geographically diverse benchmark like RDD2022 — spanning Japan, India, Czechia, Norway, the United States, and China — such a model gains resilience to the surface texture and lighting variability encountered in real deployment [5]. What has remained absent is a complete, user-centred pipeline that not only detects potholes but converts each detection into actionable guidance a blind pedestrian can act on within seconds.

SPDAS fills this gap. Its core contributions are: (i) a reproducible annotation conversion and stratified dataset splitting pipeline from Pascal VOC XML to YOLOv5 label format; (ii) a CLAHE-based frame enhancement module validated under low-light conditions; (iii) a four-tier proximity classifier and three-zone navigation engine generating directional spoken commands; (iv) a priority-ordered, fully offline text-to-speech dispatch layer; (v) GPS-stamped detection logging with image snapshots for civic road-maintenance use; and (vi) a complete Flutter Android application backed by a Render-hosted Flask inference service accessible without hardware modification.

2. Literature Review

Situating SPDAS within existing research requires examining three converging threads: automated road damage analysis, assistive navigation technology for blind users, and the evolution of YOLO-class detectors toward mobile deployment. Together, these threads expose a structural gap that SPDAS directly addresses.

A. Automated Road Damage Analysis

Systematic camera-based pavement monitoring gained momentum when Maeda et al. released a Japanese road damage corpus paired with SSD-based detection experiments demonstrating multi-class crack identification [6]. Their work also highlighted a persistent challenge: models trained on one country's road surfaces often generalise poorly to others. This insight motivated Arya et al. to construct RDD2022, which

deliberately spans six countries with intentional annotation heterogeneity, compelling models to generalise across surface materials, crack morphologies, and photographic conditions [5].

Earlier efforts by Lokeshwor et al. employed morphological operations and gradient-based texture descriptors for pothole segmentation under controlled conditions [7]. While effective in the lab, performance degraded under real outdoor illumination — a fundamental limitation of hand-engineered features for inherently variable visual tasks. Koch and Brilakis explored stereo-vision reconstruction to produce metric depth estimates of road anomalies [8], but the rigid stereo hardware configuration is incompatible with pedestrian-worn or smartphone-integrated systems. Mandal et al. advanced practical applicability by demonstrating that anchor boxes tailored to pothole geometry yield meaningfully higher recall than COCO-default anchors when using YOLOv3 [9].

B. Assistive Technology for Blind Mobility

Research in blind navigation has explored a wide range of sensing and interaction paradigms, reflecting the genuine difficulty of replacing sight. Bai et al. developed a wearable depth-camera system coupled to a haptic belt whose vibration zones corresponded to different frontal sectors, reducing corridor collisions relative to cane-only navigation [10]. However, the system was not designed for ground-level hazard detection. Katzschmann et al. embedded an ultrasonic array within a cane extension for object proximity feedback, yet a geometric blind spot covering the road surface within the cane's swing arc remained unaddressed [11].

Two findings directly inform SPDAS's design. Poggi and Mattocchia showed that monocular depth networks can operate on constrained embedded hardware at navigation-useful speeds, but ground-plane surfaces — precisely where potholes appear — yield systematically weaker depth signals [12]. Separately, Khan et al. demonstrated through user studies that well-constructed audio feedback alone is sufficient for blind participants to navigate novel indoor layouts correctly, without any haptic supplement [13]. This result directly supports SPDAS's design choice to treat spoken output as the primary and complete interaction channel.

C. YOLO Family and Mobile Inference

Redmon and Farhadi's foundational contribution — unifying bounding box regression and classification into a single forward pass — established the YOLO paradigm [14]. Jocher et al.'s YOLOv5 refined this through cross-stage partial bottleneck connections, a focus-based input stem, and multi-scale detection heads, improving small-object detection without proportionally increasing computation [4]. Subsequent work in domains such as pavement crack inspection and agricultural disease detection has confirmed that COCO-pretrained YOLOv5 weights transfer effectively to specialist visual tasks.

For mobile deployment, Shen et al. established that TensorFlow Lite converted weights on mid-range Android devices sustain approximately 15 FPS — workable for pedestrian hazard alerting [15]. Flutter's single-codebase deployment model, consistent camera plugin behaviour, and native text-to-speech bindings have made it an increasingly practical choice for accessibility-focused applications [16].

D. The Structural Gap

Road damage research has focused on detection accuracy and benchmark diversity without asking what a blind pedestrian actually needs from the output — acceptable latency, alert format, or multi-pothole prioritisation. Assistive navigation research has addressed obstacle avoidance thoughtfully but has relied on generic sensors rather than class-discriminating damage detectors. The integration layer — converting pothole-specific detections into real-time, cognitively manageable spoken guidance on a device the user already owns — remains unaddressed by either body of work, and is precisely what SPDAS provides.

3. Proposed Methodology

SPDAS processes a raw video frame through six sequentially coupled modules: adaptive luminance correction, single-stage object detection, bounding-box proximity quantification, zone-based directional reasoning, offline speech dispatch, and geo-tagged event archiving. Each module performs one well-defined transformation before passing results to the next.

A. RDD2022 Training Corpus and Label Preparation

Selecting a training dataset for cross-regional deployment is non-trivial, as models trained on one geographic context tend to overfit local surface textures and fail to generalise elsewhere. RDD2022 was selected for its deliberate multinational scope, aggregating approximately 47,420 road images from Japan, India, the Czech Republic, Norway, the United States, and China across six damage categories under a unified Pascal VOC XML scheme [5]. The pothole class D40 is designated the highest-priority detection target in SPDAS, as an undetected pothole constitutes a direct physical hazard rather than a documentation gap.

A preprocessing script converts each VOC XML annotation to YOLOv5 label format, encoding bounding boxes as four normalised values: centroid coordinates and dimensions divided by the respective frame dimensions. The dataset is partitioned 80/10/10 for training, validation, and testing under a fixed random seed. Given that potholes are underrepresented relative to longitudinal and transverse cracks in RDD2022, the training configuration activates horizontal mirroring, $\pm 30\%$ brightness/contrast perturbation, $\pm 15^\circ$ rotation jitter, and mosaic augmentation to broaden the visual contexts in which D40 examples appear.

B. Adaptive Luminance Correction

Under natural daylight, potholes cast characteristic shadows and produce sharp reflectance discontinuities that convolutional networks exploit reliably. Dusk, sodium streetlighting, and overcast night conditions compress these contrasts into a narrow tonal range, degrading detection on models trained primarily on daytime imagery. SPDAS addresses this through a CLAHE preprocessing stage applied before every inference call. Each frame is converted from BGR to YCrCb, and CLAHE is applied exclusively to the luminance channel Y using a clip limit of 2.5 over an 8×8 tile grid. This confines enhancement to luminance, preventing the colour saturation artefacts that would result from equalising all channels.

Following recombination and BGR conversion, a gamma correction pass with exponent 1.3 — applied via a precomputed 256-entry lookup table — lifts underexposed midtones without clipping bright regions. When mean frame luminance falls below a configurable darkness threshold, an additional bilateral filtering step (spatial sigma = 10 pixels) suppresses photon noise before CLAHE can amplify it into spurious texture features.

C. YOLOv5s Transfer Learning Configuration

Training from random initialisation on a domain-specific corpus is inefficient, as early epochs are consumed learning elementary visual features that a COCO-pretrained backbone already encodes. Initialising from the official yolov5s.pt checkpoint meaningfully compresses convergence time. Optimisation uses SGD with Nesterov momentum, an initial learning rate of 0.01, momentum of 0.937, and L2 weight decay of 5×10^{-4} . Images are resized and letterbox-padded to 640×640 pixels; training proceeds in mini-batches of 16 for 50 epochs under a cosine annealing schedule. The composite loss combines binary cross-entropy for class prediction, binary cross-entropy for objectness confidence, and Complete IoU (CIoU) regression for bounding box refinement. CIoU is particularly suited to pothole detection as it penalises both insufficient overlap and aspect-ratio misalignment — relevant given potholes' irregular, roughly elliptical footprints. Early stopping monitors validation mAP@0.5, terminating after ten non-improving epochs and saving the best-performing checkpoint as best.pt.

D. Proximity Estimation and Hazard Classification

Physical depth estimation without calibrated sensors is approximated by exploiting the relationship between apparent angular size and distance: as a pothole approaches the camera, its bounding box occupies a growing fraction of frame width. This ratio is mapped to four hazard tiers. A ratio below 0.12 is classified FAR; 0.12–0.25 is MEDIUM; 0.25–0.40 is CLOSE; and ≥ 0.40 triggers DANGER, which carries the shortest cooldown and highest speech queue priority. A focal-length formula supplements tier assignment with a centimetre-scale estimate:

$$d = (W^{\text{Rea}_1} \times f) / w_{\text{px}} \quad (1)$$

where $W^{\text{Rea}_1} = 60$ cm, $f = 700$ pixels, and w_{px} is the measured bounding box width. When multiple potholes appear simultaneously, the most hazardous instance always governs the alert. Repetition is suppressed via a cooldown dictionary keyed on hazard tier: non-critical tiers observe a 3-second window; DANGER alerts refresh every 1.5 seconds to prevent suppression of rapidly approaching hazards.

E. Three-Zone Spatial Navigation Engine

Detecting proximity alone is insufficient; the pedestrian also needs lateral guidance. SPDAS partitions each frame into three equal horizontal zones — LEFT (0–33.3%), CENTER (33.3–66.6%), and RIGHT (66.6–100%) — assigning each detection to the zone containing its bounding box midpoint. A decision tree selects from a fixed command vocabulary: PATH CLEAR when no detections exist; STAY CENTER when both flanks are occupied but the center is clear; MOVE LEFT or MOVE RIGHT when only the

center is occupied and a flank is open; and STOP AND SEEK ASSISTANCE when all three zones are occupied simultaneously, or when any bounding box spans more than 55% of frame width, indicating an unavoidable hazard. Commands carry integer priority values ensuring the highest-urgency instruction always reaches the speech engine first when competing alerts occur in rapid succession.

F. Offline Priority-Ordered Voice Alert System

The alert subsystem must satisfy two competing constraints: full offline operability for users in low-connectivity environments, and cognitive manageability to avoid overwhelming the user with continuous audio. SPDAS meets both through the `pyttssx3` offline library, which wraps the host OS speech engine (SAPI5 on Windows, `espeak` on Linux) and synthesises speech entirely in local memory. Alerts are placed into a Python `threading.PriorityQueue` with integer priority codes — DANGER: 0, CLOSE: 1, navigation commands: 2, informational notices: 5 — and consumed by a background daemon thread that decouples synthesis from the frame-processing loop. A per-message timestamp dictionary enforces cooldown windows. An abstract vibration controller interface is also registered within the same subsystem, designed to drive GPIO-connected vibration motors in a planned cane prototype, providing a tactile fallback in acoustically noisy environments.

G. GPS-Stamped Detection Logging

Each pothole detection above the confidence threshold generates a CSV log entry containing: a session-scoped unique identifier, ISO 8601 timestamp, GPS coordinates, RDD2022 damage category, model confidence score, hazard tier, estimated physical distance, zone classification, navigation command, bounding-box pixel coordinates, and a path to an automatically saved JPEG snapshot. This structured log can, with minimal post-processing, be transmitted to a municipal road-maintenance portal or aggregated across users to produce a community-sourced pothole density map — generating civic value as a direct by-product of individual assistive use.

H. Cloud-Backed Android Application

A Flutter/Dart companion application makes SPDAS accessible on any Android device without dedicated hardware. Flutter was chosen for its consistent cross-manufacturer camera plugin behaviour, native Android TTS bindings without bridging overhead, and a single codebase deployable to both Android and iOS. The application acquires rear-camera frames via the `flutter_camera` plugin, compresses them to JPEG, and dispatches each as an HTTP multipart POST to a remote inference backend — a Python Flask REST service packaged in Docker and hosted on Render, which handles HTTPS provisioning and horizontal scaling automatically. The backend runs the full CLAHE and YOLOv5s inference chain and returns a JSON response containing class labels, confidence scores, bounding-box coordinates, hazard tier, and navigation command. The Flutter client overlays bounding boxes on the live preview and routes the command to Android's TextToSpeech API. Mean round-trip latency from frame capture to first audio output was 387 ms ($SD = 62$ ms) over 4G LTE. Switching to a persistent WebSocket transport reduced this to 214 ms — well within the response budget required for navigation at typical pedestrian speeds of 1.2–1.4 m/s.

4. Experimental Results

All evaluations were conducted on a Windows 11 machine with an Intel Core i5 processor and 8 GB RAM; GPU acceleration was intentionally disabled to reflect the CPU-only deployment scenario typical for target users. The best.pt checkpoint was evaluated on the held-out RDD2022 test split using Precision (P), Recall (R), mAP@0.5, mAP@0.5:0.95, and F1-score. Table 1 reports the full per-class breakdown.

Table 1: Per-Class Detection Performance on RDD2022 Test Set (YOLOv5s, 50 Epochs)

Class	P	R	mAP@.5	mAP@.5:.95	F1
D00	0.71	0.68	0.702	0.381	0.695
D10	0.69	0.64	0.678	0.352	0.664
D20	0.73	0.70	0.718	0.398	0.715
D40 *	0.76	0.72	0.738	0.412	0.740
D43	0.65	0.60	0.621	0.301	0.624
D44	0.63	0.58	0.601	0.289	0.604
Mean	0.693	0.660	0.676	0.356	0.674

* Pothole class (highest-priority detection target).

The pothole class D40 achieved the strongest performance across all metrics: P = 0.76, R = 0.72, mAP@0.5 = 73.8%, F1 = 0.740. This outcome reflects potholes' visually distinctive characteristics from a pedestrian viewpoint — the dark elliptical depression, the reflectance discontinuity at the road boundary, and the internal shadow gradient — which multi-scale convolutional filters learn to activate on consistently across RDD2022's geographic diversity. Linear crack classes D00, D10, and D20 clustered in the mAP@0.5 range of 0.678–0.718; their narrow elongated bounding boxes occupy smaller image-area fractions, making precise IoU computation at thresholds above 0.5 disproportionately sensitive to minor localisation errors. Classes D43 and D44 — road marking degradation — yielded the lowest figures, reflecting the visual similarity between faded markings and general pavement ageing, compounded by comparatively sparse annotations in RDD2022.

At the system level, the full SPDAS pipeline sustains 28.2 FPS on the CPU-only test machine, exceeding the 25 FPS threshold commonly cited for pedestrian-speed hazard alerting. CUDA-enabled inference reaches 69.9 FPS, providing headroom for future extensions such as stereo disparity estimation. The four-tier proximity classifier matched independent human distance labels on 91.5% of 200 manually annotated validation frames spanning the full proximity range. The zone-based navigation engine produced the correct directional command in 88.4% of 50 scripted multi-pothole scenarios; errors were concentrated in the subset of scenes where hazards simultaneously occupied all three zones and no geometrically safe lateral direction existed.

Contextualised against published benchmarks, Mandal et al. reported 68.4% average precision using YOLOv3 on a geographically uniform corpus without luminance enhancement [9], and Arya et al. documented a YOLOv5 baseline of 64.8% on RDD2022 without CLAHE preprocessing [5]. The 5–9 percentage-point improvement attributable to SPDAS's combined preprocessing and transfer learning protocol represents a proportional reduction in undetected potholes — each missed detection corresponding to a blind pedestrian who receives no warning about a hazard directly ahead.

5. Discussion

An honest assessment of SPDAS requires acknowledging both its strengths and the limitations introduced by key design choices. The most consequential approximation is the use of bounding-box apparent size as a distance proxy. This heuristic is optically grounded, but its accuracy depends on camera calibration parameters: SPDAS defaults to a focal length of 700 pixels and a reference pothole diameter of 60 cm — population averages for common smartphone cameras and typical roadway potholes. Individual devices deviate from these values, sometimes substantially, so centimetre-scale distance estimates should be treated as order-of-magnitude indicators rather than precise measurements. A first-time calibration step — prompting the user to photograph a printed marker of known dimensions at a known distance — would remove this ambiguity at negligible cost and represents a clear near-term priority.

The second limitation is network dependency for the mobile inference path. When cellular coverage degrades below a usable threshold, the Flutter application ceases to receive results and the audio guidance channel falls silent. The solution is well-established: converting best.pt to TensorFlow Lite with INT8 post-training quantisation would reduce the model footprint sufficiently for on-device bundling; benchmarks for comparable YOLOv5 model sizes project 15–18 FPS on mid-range Android hardware, meeting the application's operational requirements. The output modality also merits reconsideration. Standard earphones attenuate safety-relevant ambient sounds — approaching vehicles, warning calls, traffic signals. Bone-conduction transducers, which transmit speech via cranial vibration while leaving the ear canal acoustically open, would reduce this risk meaningfully and warrant incorporation into the hardware reference design alongside TF Lite conversion.

The most important open question, however, cannot be resolved through benchmark evaluation alone. Whether SPDAS produces measurable improvements in real-world navigation outcomes — fewer falls, broader independently accessible route networks, reduced travel anxiety — can only be established through structured trials with visually impaired participants in actual outdoor environments. Factors invisible to laboratory evaluation include individual variation in preferred speech rate, tolerance for false-positive alerts in busy street conditions, the cognitive cost of processing directional commands while tracking orientation, and the trust calibration that develops as users accumulate experience with a system's characteristic error patterns. These questions define the primary research agenda for the next phase of this work.

6. Conclusion

The central argument of this paper is practical: a lightweight object detector trained on diverse road damage data and embedded within a human-centred alert pipeline can provide blind pedestrians with something current assistive devices do not — advance spoken warning about road surface conditions ahead. SPDAS substantiates this empirically, achieving a pothole-class mAP@0.5 of 73.8%, sustaining 28.2 FPS on a CPU-only laptop, and delivering end-to-end alert latency below 220 ms over WebSocket — within a system architecture running on a smartphone the user already owns, communicating in plain language requiring no training to interpret.

The six contributions presented — CLAHE-based low-light frame enhancement, four-tier proximity-driven hazard classification, three-zone lateral navigation command generation, priority-ordered offline speech dispatch, GPS-stamped detection logging, and a complete Flutter mobile application with cloud inference — constitute a coherent engineering response to a problem that has sat at the intersection of road damage detection and assistive navigation without receiving an integrated solution. The forward path involves three interrelated developments: on-device TF Lite inference to eliminate network dependency, stereo depth integration to replace the bounding-box distance heuristic with calibrated metric estimates, and a properly designed user study with blind and low-vision participants — replacing the laboratory performance figures presented here with evidence that matters most: measurable benefit to real people navigating a world not designed with their safety in mind.

References

1. World Health Organization, "Blindness and vision impairment," WHO Fact Sheet, Oct. 2023. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>
2. Ministry of Road Transport and Highways, "Annual Report on Road Accidents in India," Government of India, New Delhi, 2022.
3. A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in Proc. IEEE CVPR, Providence, RI, USA, 2012, pp. 3354–3361.
4. G. Jocher et al., "ultralytics/yolov5: v7.0 – YOLOv5 SOTA Realtime Instance Segmentation," Zenodo, Nov. 2022. doi: 10.5281/zenodo.7347926.
5. D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, H. Omata, T. Kashiyama, and Y. Sekimoto, "RDD2022: A multi-national image dataset for automatic Road Damage Detection," arXiv:2209.08538, 2022.
6. H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection and classification using deep neural networks with smartphone images," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1127–1141, 2018.
7. H. Lokeshwor, L. K. Das, and S. Goel, "Robust method for automated segmentation of pothole region(s) in rigid road surface," *J. Transportation Engineering*, vol. 140, no. 5, 2014, Art. no. 04014008.
8. C. Koch and M. Brilakis, "Pothole detection in asphalt pavement images," *Advanced Engineering Informatics*, vol. 25, no. 3, pp. 507–515, 2011.

9. V. Mandal, L. Mussah, P. Jin, and Y. Adu-Gyamfi, "Artificial intelligence-enabled pothole detection: A step toward smarter road infrastructure management," *Sensors*, vol. 20, no. 11, p. 3216, 2020.
10. J. Bai, S. Liu, Y. Lin, Z. Li, S. Pan, and J. Liu, "A guide-dog robot system for the visually impaired based on computer vision," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 8, pp. 1679–1690, Aug. 2019.
11. R. K. Katzschmann, B. Araki, and D. Rus, "Safe local navigation for visually impaired users with a time-of-flight and haptic feedback device," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 3, pp. 583–593, Mar. 2018.
12. M. Poggi and S. Mattocchia, "A wearable mobility aid for the visually impaired based on embedded 3D vision and deep learning," in *Proc. IEEE ISCC*, Messina, Italy, 2016, pp. 208–213.
13. A. Khan, A. Salman, and S. Naseer, "Indoor navigation for visually impaired persons using computer vision and audio feedback," in *Proc. ICE Cube*, Quetta, Pakistan, 2018, pp. 1–6.
14. J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv:1804.02767*, 2018.
15. X. Shen, A. Liu, Z. Cheng, and A. C. Kot, "Real-time object detection with TensorFlow Lite on Android devices," in *Proc. IEEE ICME*, London, UK, 2020, pp. 1–6.
16. K. Amaresan, "Flutter and Dart for cross-platform mobile application development: A systematic review," *J. Computer Applications Research*, vol. 14, no. 2, pp. 45–58, 2023.
17. R. G. Long and K. Hill, *Orientation and Mobility: Principles and Practice*, 3rd ed. New York: AFB Press, 2018.
18. L. Ran, Y. Zhang, Q. Zhang, and T. Yang, "Convolutional neural network-based robot navigation using uncalibrated spherical images," *Sensors*, vol. 17, no. 6, p. 1341, 2017.
19. L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. PAMI*, vol. 40, no. 4, pp. 834–848, Apr. 2018.