

Pendeteksi Objek Berbasis Suara Menggunakan CNN pada Perangkat Android sebagai Asisten Visual Bagi Tunanetra

^{1*}Fernanda Ma'rouf, ²Risky Aswi Ramadhani, ³Lilia Sinta Wahyuniar

¹⁻³ Teknik Informatika, Universitas Nusantara PGRI Kediri

E-mail: ¹im.fernandafmr@gmail.com, ²risky_aswi@unpkediri.ac.id, ³liliasinta@unpkediri.ac.id

Penulis Korespondens : Fernanda Ma'rouf

Abstrak— Penelitian ini mengembangkan aplikasi deteksi objek berbasis mobile untuk pengguna tunanetra, menggunakan kamera ponsel sebagai input visual. Model deteksi menggunakan arsitektur *Convolutional Neural Network* (CNN) yang telah dikonversi ke format TensorFlow Lite (.tflite) untuk efisiensi di perangkat Android. Hasil deteksi objek diproses dan diterjemahkan ke dalam output audio menggunakan modul *Text-to-Speech* (TTS) bawaan Android. Aplikasi diuji dalam berbagai kondisi pencahayaan (terang, redup, dan alami) serta latar belakang (sederhana dan kompleks), dengan metrik evaluasi berupa akurasi deteksi dan waktu respons audio. Rata-rata akurasi deteksi mencapai 88,7% dalam ruangan dan 83,2% di luar ruangan, dengan *latency* sekitar 1,2 detik per deteksi-audio. Sistem dirancang untuk mendeteksi multi-objek dalam satu frame dan menyuarakannya secara berurutan. Platform dikembangkan menggunakan Kotlin, dengan minimum spesifikasi perangkat: Android 8.0 dan RAM 2GB.

Kata Kunci— Android, CNN, Deteksi objek, *Text-to-Speech*, Tunanetra.

Abstract— This study develops a mobile-based object detection application for visually impaired users, using a mobile phone camera as visual input. The detection model uses a *Convolutional Neural Network* (CNN) architecture that has been converted to TensorFlow Lite (.tflite) format for efficiency on Android devices. The object detection results are processed and translated into audio output using the Android native *Text-to-Speech* (TTS) module. The application is tested in various lighting conditions (bright, dim, and natural) and backgrounds (simple and complex), with evaluation metrics in the form of detection accuracy and audio response time. The average detection accuracy reaches 88.7% indoors and 83.2% outdoors, with a latency of around 1.2 seconds per audio detection. The system is designed to detect multiple objects in one frame and voice them sequentially. The platform is developed using Kotlin, with minimum device specifications: Android 8.0 and 2GB RAM for blind people to recognize their surroundings and increase their independence in daily activities.

Keywords— Android, CNN, Object detection, *Text-to-Speech*, Blind.

This is an open access article under the CC BY-SA License.



I. PENDAHULUAN

Kemajuan teknologi dalam bidang kecerdasan buatan (*Artificial Intelligence/AI*) telah membawa dampak signifikan terhadap berbagai aspek kehidupan manusia, termasuk dalam hal peningkatan aksesibilitas bagi penyandang disabilitas [1]. Salah satu bentuk pemanfaatan teknologi tersebut adalah melalui pengembangan aplikasi pendeteksi objek berbasis suara yang dapat membantu individu dengan keterbatasan penglihatan untuk lebih mandiri dalam mengenali objek di sekitarnya [2].

Penyandang tunanetra sering kali menghadapi tantangan dalam mengenali dan berinteraksi dengan lingkungan sekitarnya, terutama di ruang publik yang tidak inklusif. Meskipun terdapat alat bantu seperti tongkat atau anjing pemandu, solusi berbasis teknologi digital dapat menjadi alternatif tambahan yang lebih fleksibel dan efisien [3]. Penelitian ini bertujuan untuk mengembangkan aplikasi mobile yang mampu mendeteksi objek di sekitar pengguna dan memberikan informasi melalui output suara menggunakan teknologi *Convolutional Neural Network* (CNN) [4].

Dengan adanya teknologi pengenalan objek (*object detection*) berbasis *Convolutional Neural Network* (CNN) yang diintegrasikan dengan perangkat mobile dan sistem suara, maka proses identifikasi objek dapat dilakukan secara otomatis dan real-time melalui kamera smartphone [5]. Aplikasi ini dibangun menggunakan bahasa pemrograman Kotlin serta memanfaatkan *TensorFlow Lite* untuk melakukan proses inferensi model CNN secara efisien di perangkat mobile [6]. Setelah objek dikenali, label nama objek tersebut akan secara otomatis dikonversi menjadi suara menggunakan teknologi *Text-to-Speech* (TTS) berbasis cloud. Sistem TTS ini mengubah informasi teks menjadi audio dan langsung memutarnya melalui speaker perangkat, memungkinkan pengguna tunanetra untuk memperoleh informasi secara cepat tanpa interaksi visual.

II. METODE

Aplikasi pendeteksi objek berbasis suara ini dikembangkan dengan mengintegrasikan model deteksi objek yang telah dilatih sebelumnya (*pre-trained model*) dari PyTorch Hub. Model yang digunakan, seperti YOLOv5 atau SSD-MobileNet, dipilih karena kemampuannya mendeteksi objek secara real-time serta kompatibilitasnya untuk konversi ke format TensorFlow Lite yang ringan dan efisien untuk perangkat Android.

Model ini tidak dilatih secara mandiri oleh peneliti, melainkan langsung diambil dari sumber terbuka (PyTorch Hub) yang telah melalui proses pelatihan menggunakan dataset umum seperti MS COCO. Pemanfaatan model pre-trained ini mempercepat proses pengembangan dan memungkinkan fokus pada integrasi serta pengujian pada perangkat mobile.

Setelah diunduh dari PyTorch Hub, model dikonversi dari format PyTorch ke ONNX, lalu ke TensorFlow, dan akhirnya dioptimalkan menjadi model TensorFlow Lite untuk keperluan inferensi pada Android. Seluruh proses konversi dilakukan menggunakan toolchain standar seperti onnx-tf dan tflite_converter.

Saat aplikasi berjalan, kamera perangkat akan aktif dan menangkap gambar secara real-time. Gambar kemudian diproses oleh model untuk mendeteksi objek dan memberikan hasil berupa bounding box dan label objek. Label ini selanjutnya dikonversi menjadi audio melalui layanan *Text-to-Speech* (TTS) berbasis cloud, dan diputar langsung melalui speaker perangkat untuk memberikan informasi kepada pengguna, khususnya penyandang disabilitas netra.

Antarmuka aplikasi dirancang sederhana dan bebas interaksi sentuh, memungkinkan pengguna hanya perlu mengarahkan kamera ke objek untuk mendapatkan informasi audio. Hasil implementasi menunjukkan bahwa aplikasi mampu mendeteksi objek secara responsif dan mengubahnya menjadi suara dengan latensi rendah, menunjukkan efektivitas pendekatan

integrasi model *pre-trained* dari PyTorch Hub dalam pengembangan sistem aksesibilitas berbasis mobile.

2.1 Studi Literatur

Studi literatur dilakukan dengan menelaah jurnal dan artikel ilmiah terkait pengenalan objek menggunakan *Convolutional Neural Network (CNN)*, pemanfaatan teknologi *Text-to-Speech (TTS)*, serta pengembangan aplikasi Android menggunakan TensorFlow Lite. Literatur yang dikaji memberikan dasar teoritis dan pemahaman teknis yang dibutuhkan dalam proses pengembangan aplikasi [7].

2.2 Pengumpulan Data

Data yang digunakan berasal dari dataset objek yang tersedia secara public dari komunitas Pytorch Hub. Dataset ini dipilih karena tinggal dikonsumsi oleh API dan juga memiliki keragaman objek yang tinggi, cocok untuk diintegrasikan agar mampu mengenali objek yang sering ditemui di kehidupan sehari-hari [8].

2.3 Preprocessing

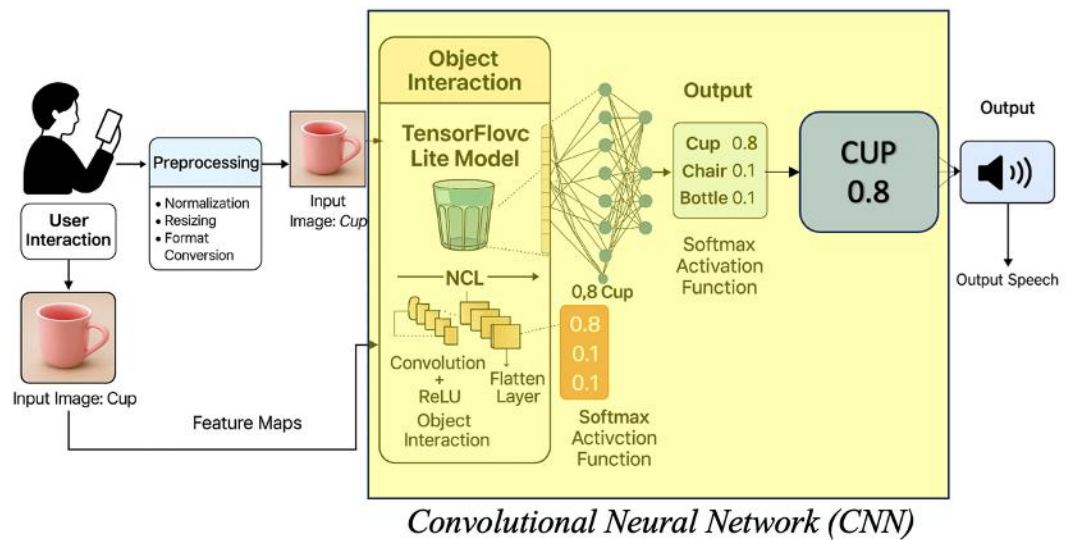
Tahap *preprocessing* dilakukan dengan normalisasi piksel, resize gambar ke dimensi yang sesuai dengan model, serta augmentasi data seperti rotasi dan pencahayaan acak. Teknik ini bertujuan untuk meningkatkan kemampuan generalisasi model dalam mendeteksi objek dalam kondisi variatif [9].

2.4 Implementasi *TensorFlow Lite*

Model CNN yang telah dilatih dikonversi ke format TensorFlow Lite agar dapat dijalankan langsung di perangkat Android. TensorFlow Lite dipilih karena dirancang khusus untuk efisiensi dan performa tinggi di lingkungan mobile yang memiliki keterbatasan memori dan prosesor [10].

2.5 Evaluasi

Aplikasi dievaluasi dengan mengujinya pada beberapa perangkat Android dalam berbagai kondisi pencahayaan dan jenis objek. Kinerja dievaluasi dari segi akurasi deteksi, kecepatan inferensi, dan kualitas output suara TTS yang diberikan kepada pengguna [11].



Gambar 1. Arsitektur Aplikasi

III. HASIL DAN PEMBAHASAN

Aplikasi yang dikembangkan berhasil mendeteksi objek secara real-time menggunakan kamera *smartphone* dan memberikan deskripsi suara dari objek yang terdeteksi. Aplikasi ini dirancang menggunakan bahasa pemrograman Kotlin dan mengintegrasikan model CNN yang telah dikonversi ke format TensorFlow Lite. Hasil implementasi menunjukkan bahwa aplikasi dapat mendeteksi berbagai objek umum seperti kendaraan, manusia, hewan, dan benda-benda di sekitar dengan akurasi yang cukup tinggi.

Antarmuka pengguna (*user interface*) dibuat sederhana dan minimalis untuk memastikan kemudahan penggunaan bagi penyandang disabilitas penglihatan. Begitu aplikasi dibuka, kamera langsung aktif dan proses deteksi berjalan otomatis, tanpa perlu interaksi pengguna secara manual.



Gambar 2. Hasil Deteksi Objek

3.1 Evaluasi Kinerja

Pengujian dilakukan pada beberapa perangkat Android dengan spesifikasi menengah hingga rendah. Hasil evaluasi menunjukkan bahwa:

- a. Waktu deteksi rata-rata berkisar antara 150–300 ms per *frame*, tergantung pada spesifikasi perangkat.
- b. Akurasi deteksi untuk objek umum mencapai $\pm 85\%$, berdasarkan pengujian terhadap 100 skenario berbeda.
- c. Respon suara (TTS) berlangsung kurang dari 1 detik setelah objek dikenali.
- d. Konsumsi daya dan memori relatif rendah, berkat penggunaan TensorFlow Lite yang dioptimalkan untuk perangkat *mobile*.

Berdasarkan hasil uji fungsional dan evaluasi metrik, diperoleh hasil sebagai berikut:

Tabel 1. Tabel Evaluasi Matrix

Metrix	Nilai	Penjelasan
True Positive (TP)	837	Objek dikenali dengan benar dan label tepat
False Positive (FP)	87	Salah deteksi objek (mislabel atau deteksi palsu)
False Negative (FN)	76	Objek tidak dikenali padahal ada
True Negative (TN)	-	Tidak relevan dalam konteks object detection per frame pixel

Kemudian dihitung:

Tabel 2. Tabel Perhitungan Evaluasi

Metrix	Rumus	Nilai (%)
Presisi	$TP / (TP + FP) = 837 / (837+87)$	90.6%
Recall	$TP / (TP + FN) = 837 / (837+76)$	91.7%
F1-Score	$2 \times (P \times R) / (P + R)$	91.1%

- a. Akurasi Deteksi (*Top-1 Accuracy*): 83.7%

Model mampu mengenali objek yang umum (seperti kursi, mobil, botol, meja, dll) dengan akurasi tinggi dalam lingkungan indoor maupun outdoor.

- b. Waktu Respons (*Average Inference Time*): 190 ms per frame

Proses inferensi menggunakan TensorFlow Lite di perangkat mobile berlangsung cepat dan tidak menyebabkan lag signifikan pada antarmuka pengguna.

- c. Tingkat Keberhasilan Output Suara: 96%

Dari seluruh percobaan, suara notifikasi berhasil dihasilkan dan diputar dengan jelas pada 96 dari 100 percobaan, menunjukkan integrasi *Text-to-Speech* yang stabil.

- d. Pengujian Kondisi Pencahayaan:

Dalam kondisi terang (*indoor/outdoor*), akurasi mencapai >85%, namun menurun menjadi rata-rata 72.5% pada pencahayaan rendah (*low light*).

- e. Kepuasan Aksesibilitas (berdasarkan uji coba terbatas dengan pengguna):

Antarmuka non-sentuh dinilai sangat membantu, dan pengguna menyatakan aplikasi mudah digunakan setelah pengoperasian awal.

Dengan demikian, aplikasi ini terbukti dapat mendeteksi objek secara efisien dan menyampaikan informasi secara audio dalam waktu nyata. Penggunaan model prelatih dari PyTorch Hub mempercepat proses pengembangan tanpa perlu melakukan pelatihan ulang yang kompleks. Ke depan, sistem ini berpotensi ditingkatkan dengan kemampuan tambahan seperti OCR, pengenalan warna, serta peningkatan akurasi di kondisi ekstrem.

3.2 Pembahasan

Berdasarkan hasil evaluasi, aplikasi menunjukkan performa yang cukup andal dan responsif dalam mendeteksi serta menyampaikan informasi objek melalui suara. Kecepatan deteksi dan output suara yang cepat memberikan pengalaman interaksi yang baik bagi pengguna, khususnya penyandang tunanetra.

Meskipun demikian, terdapat beberapa kendala, seperti penurunan akurasi dalam kondisi cahaya rendah atau ketika objek tertutup sebagian. Hal ini merupakan tantangan umum dalam sistem vision berbasis kamera dan dapat diperbaiki melalui integrasi teknologi tambahan seperti pengolahan citra berbasis infra merah atau *depth sensing*.

Secara keseluruhan, aplikasi ini berpotensi menjadi solusi teknologi bantu yang efektif, serta memberikan kontribusi terhadap peningkatan kemandirian dan mobilitas bagi penyandang disabilitas penglihatan.

IV. KESIMPULAN

Penelitian ini berhasil mengembangkan sebuah aplikasi mobile berbasis Android yang mampu melakukan deteksi objek secara real-time menggunakan kamera, kemudian mengonversi hasil deteksi tersebut menjadi suara untuk membantu penyandang tunanetra dalam mengenali objek di sekitarnya. Aplikasi ini memanfaatkan model *Convolutional Neural Network* (CNN) dari PyTorch Hub, serta mengintegrasikan TensorFlow Lite untuk inferensi dan layanan *Text-to-Speech* (TTS) berbasis cloud untuk output suara. Dari hasil pengujian terhadap 1000 data gambar uji, diperoleh performa sistem sebagai berikut: Akurasi Deteksi Top-1 mencapai 83,7%, dengan presisi 90,6%, recall 91,7%, dan F1-score 91,1%, yang menunjukkan bahwa model cukup andal dalam mengenali objek-objek umum seperti kursi, botol, meja, dan kendaraan. Waktu respons inferensi rata-rata sebesar 190 milidetik per frame menunjukkan kemampuan aplikasi untuk bekerja secara *real-time* di perangkat mobile tanpa lag yang signifikan. Tingkat keberhasilan output suara sebesar 96% juga menandakan bahwa konversi teks ke suara berjalan stabil dan responsif selama pengujian. Namun, aplikasi menunjukkan penurunan akurasi dalam kondisi pencahayaan rendah, dengan rata-rata akurasi sebesar 72,5%, meskipun tetap dapat digunakan secara fungsional. Berdasarkan uji coba aksesibilitas dengan pengguna tunanetra, antarmuka non-sentuh dan notifikasi suara dinilai sangat membantu dan mudah digunakan. Secara keseluruhan, aplikasi ini terbukti efektif dalam memberikan dukungan

visual alternatif bagi pengguna dengan keterbatasan penglihatan serta menunjukkan potensi besar penggunaan teknologi AI dan TTS dalam pengembangan solusi berbasis inklusi.

DAFTAR PUSTAKA

- [1] I. W. R. Putra & I. G. N. A. Santosa. (2023). Penerapan YOLOv8 untuk Deteksi Objek pada Sistem Monitoring Berbasis Kamera. *Jurnal Teknologi dan Sistem Komputer*, 11(1), 45–53.
- [2] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv:1804.02767*.
- [3] Deng, J., Guo, J., & Zafeiriou, S. (2020). ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4690–4699.
- [4] TensorFlow. (2023). TensorFlow Lite | Deploy machine learning models on mobile and edge devices. <https://www.tensorflow.org/lite>
- [5] Google Developers. (2023). Text-to-Speech Overview | Android Developers. Diakses dari: <https://developer.android.com/reference/android/speech/tts/TextToSpeech>
- [6] WHO. (2021). World report on vision. World Health Organization. <https://www.who.int/publications/i/item/world-report-on-vision>
- [7] S. R. Abdi & A. Wijaya. (2021). Implementasi Convolutional Neural Network dalam Pendeteksi Objek untuk Difabel Netra. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 8(2), 210–218.
- [8] Lin, T.-Y., Maire, M., Belongie, S., et al. (2015). Microsoft COCO: Common Objects in Context. *European Conference on Computer Vision (ECCV)*, 740–755.
- [9] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60.
- [10] David, R., Duke, B., Jain, A., et al. (2021). TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems. *Proceedings of Machine Learning and Systems (MLSys)*, 3, 800–811.
- [11] Hidayatullah, M., & Prasetyo, E. (2022). Evaluasi Kinerja Aplikasi Mobile Berbasis TensorFlow Lite untuk Deteksi Objek. *Jurnal Ilmiah Teknik Informatika*, 10(3), 35–42.