# REVIEW OF RELATED LITERATURE

This chapter discusses the literature and studies related to EyeDentify, a voice-interactive mobile assistant designed to enhance the daily activities of visually impaired individuals. The review covers technologies such as the YOLO algorithm, Tesseract OCR, TensorFlow Lite, and voice interaction systems. This literature and study review aided the researchers in gaining familiarity with and understanding the topic.

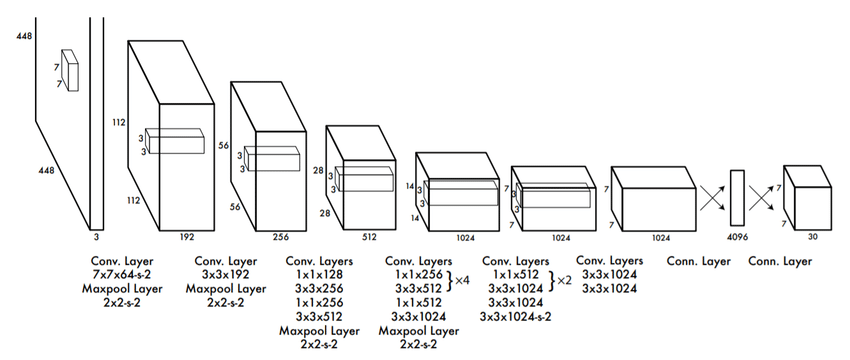
## Related Literature

**Object Detection Using YOLO Algorithm.** YOLO algorithm is a computer vision model that is commonly used for object detection these days (Chai et al., 2021). YOLO divides the attached image into an S x S grid, and it will predict the bounding boxes and class probabilities for each of these grid cells, perfectly suited for real-time object detection within one pass in any neural network (Ayachi et al., 2025). This unified architecture significantly improves speed and efficiency compared to traditional region-based methods. The version trained in Pytorch and is most widely used is YOLOv5, which has several model sizes, YOLOv5m, YOLOv5s, YOLOv5x, and YOLOv5l, with increasing balance between accuracy and speed for different hardware capabilities (Ayachi et al., 2025).

According to Zhu et al. (2025), the YOLO target detection technique is based on the model's small size and quick computation speed. The structure of YOLO is clear. The neural network may immediately output the bounding box's location and categorization. YOLO's speed is fast because it needs to attach the image to the network to get the final recognized output. Therefore, it can also detect video in real time (Jiang et al., 2022). YOLO recognizes objects directly from the global picture, encoding global information while reducing the inaccuracy of viewing the background as the object. YOLO has excellent generalization potential since it can acquire highly generalized characteristics and transfer them to various fields (Chen et al., 2024).

Jiang et al. (2022) state that despite its advancements over time, tiny item detection is still a challenge. The accuracy of small item identification can still be improved, even though Yolo V5's Mosaic enhancement helps to some extent. Although YOLO's speed advantages were covered in the study, there is sometimes a trade-off between detection rapidity and precision that isn't thoroughly examined in all situations (S et al., 2020).

*Yolo Architecture.* From an architectural perspective, YOLO resembles GoogleNet. It has a total of 24 convolutional layers, 4 max-pooling layers, and 2 fully connected layers, as illustrated below (Kelta, 2022).



**Fig 1. YOLO Architecture Diagram**

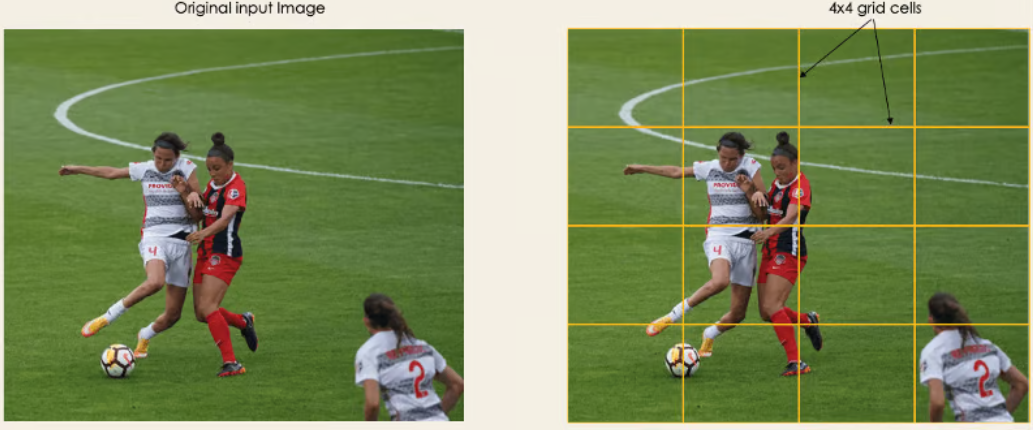
This is how the architecture operates:

* Before applying any convolutions, it scales the incoming image to a size of 448 \* 448 pixels.
* The first step includes applying a 1x1 convolution to reduce the number of channels. This is followed by a 3x3 convolution that produces a cuboid-shaped output.
* The last layer's linear activation function is the only exception to this rule, meaning all other layers have ReLU as the activation function.
* Other techniques, such as batch normalization and dropout, assist in regularizing the model and preventing overfitting.

*How does YOLO Object Detection work*?According to S et al. (2020), four approaches are commonly taken when identifying an item using the YOLO algorithm:

1. Residual Blocks
2. Bounding Box Regression
3. Intersection Over Unions, or IOU for short
4. Non-Maximum Suppression

*Residual Blocks.* As seen in the graphic on the right, the first step begins by splitting the whole image into grids of N × N equal-sized cells. In this case, N equals 4. The class of the item it maps, as well as the confidence probability value, must be determined and estimated by each grid cell.

**

**Fig 2. Sample Picture**

*Bounding Box Regression.* The next stage is to identify the bounding boxes for each rectangle or item in the picture; the number of bounding boxes may be as large as the number of objects in the picture.

YOLO uses a single regression module to calculate the bounding box parameters of a collection of predicted outputs, as seen below, where Y stands for the vector output of each bounding box.

***Y = [pc, bx, by, bh, bw, c1, c2]***

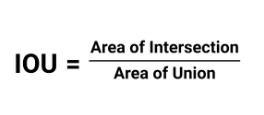
This is greatly useful in the training stage of the model.

* **pc:** is the probability score of the grid that has an object.
* **bx, by:** are the coordinates of the center of the bounding box relative to the grid cell.
* **bh, bw:** the height and width of the bounding box relative to the grid cell.
* **c1, c2:** relate to the two classes: player and ball. There is no limit to the number of classes that can be defined depending on the situation.

*Intersection Over Unions or IOU.* IOU filters eliminate duplicated or low-quality detections:

* It assesses how closely a predicted box resembles the real item.

Formula:



*Non-Max Suppression or NMS.* This last stage removes repeated detections of the same item.

* Sorts all bounding boxes according to confidence score.
* Keeps the highest-scoring box while removing any that strongly overlap with it.
* Continues the procedure until all superfluous boxes are removed.

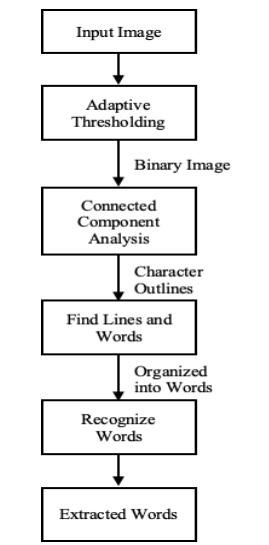
**Tesseract OCR for image recognition**. OCR is a method that enables a machine to identify scripts or alphabets written during a user's vocal communication without human engagement (Sahu & Sonkusare, 2017). Tesseract, an OCR engine built by Google, is freely accessible for anybody to use. It supports several languages and can handle different font sizes and styles (Francis & Sangeetha, 2025).

In the study of Mubeen et al. (2022), unveiled a novel approach that leverages the strengths of both established and deep learning techniques. Their key contribution is the creation of a hybrid OCR system incorporated into an Android application. This program highlights user experience by providing an ad-free and straightforward interface, making it simple to use for a variety of text extraction tasks.

Tesseract OCR, while strong, can have performance concerns, notably in terms of speed and accuracy (Dawood, 2024) Large amounts of documentation or highly detailed photographs may slow down the procedure. The quality and resolution of the input photos have a significant impact on performance (Christian & Kusuma, 2023).

In the study of Joshi (2024), research covers over 100 languages, the testing appears to be done just in English text, with no comparison analysis across multiple languages to determine correctness. Studies on Tesseract, such as those done by Smith (2021) have consistently highlighted its versatility and accuracy while analyzing printed text. However, its performance is predicted to be strongly influenced by the input image quality. The dependence on picture quality is a significant barrier, particularly in handling real-world challenges, which frequently entail detecting messages in noisy or low-resolution images.

*How Tesseract Engine Works.* While offline optical character recognition (OCR) allows for a picture to be scanned from a single page, online OCR allows points to be constantly stated as a function of time in a stream of images (Fauzan & Wibowo, 2021). Tesseract OCR is a tool for identifying character images based on the analysis of pixel distribution and character training (Lestari & Mulyana, 2022). The following figure demonstrates how Tesseract OCR works:



**Fig 2. How Tesseract Engine Works**

Tesseract transforms the input image into a binary format, which is the first step in recognizing components, and it corresponds to a stored component template outline. At this stage, all outlines are collected and organized as blobs and are grouped as text lines. The areas and lines undergo examination and modification to yield text which is proportional. Words are formed by parsing the text based on the gaps between characters, which include definitive gaps and vague spaces (Kevin Wiguna et al., 2020).

High-confidence letter shape recognition (first pass) is used in adaptive recognition, which comes after the recognition stage. In addition, it will be simpler to identify the remaining characters from the previous stage in the next step (second pass). Tesseract uses libraries to enhance accuracy during the character segmentation stage. According to Patience et al. (2024), the results showed that Tesseract struggled to correctly capture the text from the picture. Despite its comprehensive capabilities, the flyer's intricate design components and text formatting presented issues, resulting in a substantial decrease in text recognition accuracy. This conclusion demonstrates OCR's limits when dealing with highly styled or graphically rich materials.

**Mobile AI Application Using TensorFlow Lite.** TensorFlow Lite was chosen since it is specifically designed for running deep learning models on mobile devices. While employing resources like as compute and memory is an efficient execution method, the models are stored in a specific reduced file format (Adi & Casson, 2021).TensorFlow Lite also supports a variety of languages, such as Java, Python, C++, Objective-C, and Swift. The library is known for its great performance, which is achieved through the use of hardware acceleration and model optimization (Google, 2024)

*Why Use TensorFlow Lite?* TensorFlow Lite models are substantially smaller and more efficient than standard TensorFlow models, making them suitable for low-power devices like mobile. TensorFlow Lite models are substantially smaller and more efficient than standard TensorFlow models, making them suitable for low-power devices. During this conversion, improvements such as quantization can be implemented to minimize model size and latency with low or no loss in accuracy (Zhen, 2025).

According to Li et al. (2025), ML/DL models may operate natively on a mobile device thanks to on-device inference. However, the limitation is that mobile devices have limited processing power, which makes it difficult to manage complex models.

This primary limitation, in contrast with cloud-based solutions, limits the range of complexity and capability of models that can be deployed using Tensorflow Lite.

**Voice Interaction Technologies.** Voice interaction technologies have emerged as powerful tools, especially to enhance daily activities for visually impaired individuals (Tharun et al., 2025). Text-to-Speech (TTS) and Speech-to-Text (STT) are two strong voice interaction technologies that are essential for bridging the gap between spoken and written communication, particularly in modern applications (B et al., 2025)

*Text-to-Speech (TTS)*. Text-to-speech (TTS) systems translate digital text into spoken words, allowing the application to react audibly to users (Raffoul & Jaber, 2023). A lightweight, multilingual TTS engine specifically made for mobile devices is Google Text-to-Speech. This is made for visually impaired individuals by reading the text, item, or anything surrounding it out loud so that they may understand it (Le Phuong et al., 2021).

Text-To-Speech (TTS) technology improves accessibility and cognitive skills for the blind by delivering auditory feedback. TTS combined with OCR and object detection technologies allows for more efficient access to both physical and digital information (Gawande et al., 2023)

*Speech-to-Text (STT).* An STT system transcribes speech to text written in words, employing Automatic Speech Recognition (ASR) systems (Alharbi et al., 2021).The popularity of Google's Speech Recognition API stems from its accuracy, supports multiple languages, and can be integrated with mobile devices (IANCU, 2019).For users with vision impairment, STT technology interprets users’ voices, enabling system navigation without the use of screens.

In the study of Xu et al. (2021), the use of cloud-based STT services in accessibility applications is particularly useful in real-time voice command interfaces. Yet, latency and reliance on the Internet are still challenges in offline situations or remote areas.

## Review of Related System

Raghavan et al. (2021) implement a virtual AI assistant for people with partial vision impairment. This research has a great foundation in terms of helping the daily activities of partially sighted people. EyeDentify and this virtual AI assistant have a lot of things in common, like text recognition, object recognition, the ability to command the application itself, and the ability to detect human faces in the camera frame. If you see any sides, they are equal in terms of features. However, EyeDentify overcame this limitation. EyeDentify can do more, such as discover his position, although existing studies can only do basic commands such as time, weather, online browsing, and so on. In terms of accuracy, EyeDentify outperformed it since it employed YOLOv5 for higher accuracy than the existing assistant, which used MobileNet v1 for substantially lower accuracy in terms of object detection.

In the study of More et al. (2022), it implements an Android application for visually impaired people based on AI technology. This is somewhat similar to EyeDentify because it was designed for visually impaired people, like the swipe accessibility, but it has limitations because it only does basic things like read the text, calculator, weather, location, battery, time and date, and object (object detection). EyeDentify features an innovative feature in which you may command the system itself and request that it serve as a navigation tool for visually impaired individuals, and the technologies used are far faster than it.

The study of Nayak & Chandrakala (2020), a smartphone app for visually impaired individuals, is extremely useful, particularly for those with visual impairment, since it includes an email helper, SMS reader, navigation guide, and calendar assistant. EyeDentify provides all of these functions and can do more, including recognition of objects, text reading, voice interaction systems, and assistive features. This is a really innovative version of current research, and it will greatly benefit the visually impaired in every aspect.

While Kumar & Jain (2022) worked on a hardware-based dual system solution, consisting of a wearable mask and a smart cane that solely employed YOLOv5 for path detection. EyeDentify takes the concept a step further by including multimodal aid technologies. Implementing Tesseract OCR in EyeDentify offers an innovation that was lacking from previous research, allowing my system to detect and interpret text around the user. The voice-interactive component also improves the user experience because it is creative, as opposed to previous research, which just provides alerts or directions. In conclusion, EyeDentify much exceeded and outperformed it.

In the study of Hüseyin YILDIRIM & Rovshenov (2022), they use a volunteer reader software for those with visual impairments. The application operates on a digital model of social assistance based entirely on volunteerism, linking visually impaired users with volunteer readers who transform requested textbooks into audiobooks. The key innovation in EyeDentify is the combination of computer vision and machine learning technologies (YOLO algorithm, Tesseract OCR, and TensorFlow Lite) with voice interaction capabilities. On one hand, the volunteer reader app relies on the human volunteers to produce the audio content, whereas EyeDentify appears to actually assess and interpret the visual world in real-time on behalf of the visually impaired. The inclusion of YOLO (You Only Look Once) suggests it can detect objects, Tesseract OCR suggests it can recognize visually-impaired text from an image, and TensorFlow Lite suggests on-device machine learning capacity It seems that your application allows for immediate assistance in performing daily living tasks that do not involve reading audiobooks, and potentially gives users the ability to recognize objects, read text in their environment, navigate spaces in their environment, and engage with their world utilizing voice commands and voice feedback. This offers a much wider scope and higher level of technology than the basic audiobook platform presented in the document.

## Synthesis

As discussed in related studies, it appears that assistive technologies, particularly for the visually impaired, have developed over time. However, there is still much room for innovation. The synthesis of crucial technologies such as YOLO algorithm, Tesseract OCR, TensorFlow Lite, and voice interaction systems provides a good platform for further developing ideas such as EyeDentify.

The YOLO technique excels over traditional region-based methods for real-time object detection. The YOLO algorithm architecture, which involves single-pass processing of images through a neural network, offers the speed necessary for mobile applications. Ayachi et al. (2025) and Zhu et al. (2025) have highlighted some of YOLO's advantages with speed and global feature extraction. However, some studies, especially Jiang et al. (2022), have shown ongoing issues with small object detection. EyeDentify might rely on YOLO's potential while addressing its limits through optimized implementation.

Tesseract OCR is an open-source, multi-language supporting text recognition software, but it struggles with imaging and formatting. Mubeen et al. (2022) and Francis & Sangeetha (2025) showcase its strengths, while Dawood (2024) alongside Patience et al. (2024) focus on its struggles, particularly with speed and accuracy for complex images. Incorporating image pre-processing techniques to improve quality before OCR use will allow EyeDentify to bypass these concerns.

According to Adi & Casson (2021), TensorFlow Lite is the best framework for running AI models on mobile devices. The issue raised by Li et al. (2025) is the limited computational resources of mobile devices. However, TensorFlow Lite's optimization techniques will allow EyeDentify to offload resources without ruining its quality.

Speech interaction technologies like Text-to-Speech (TTS) and Speech-to-Text (STT) serve as the fundamental interface between blind users and computers. As noted by Tharun et al. (2025) and Gawande et al. (2023), TTS improves accessibility by implementing feedback mechanisms, while Xu et al. (2021) highlights the importance of STT for voice command interfaces, even when there are issues with responsiveness and reliance on the internet. The combination of these technologies by EyeDentify facilitates an effortless voice-interactive experience with offline features to solve the problems of connectivity issues.

Assistive technologies designed by Raghavan et al. (2021) and More et al. (2022) incorporate valuable remaining features, but often possess an accuracy technology gap or limited functionality scope. The approach taken by Kumar & Jain (2022) and Hüseyin YILDIRIM & Rovshenov (2022) in their volunteer reader application offers aid in different ways but lacks thorough real-time support that advanced computer vision combined with voice interaction can provide.

EyeDentify meets the gaps through a combination of object detection and YOLO, text recognition with Tesseract OCR, optimized deployment capabilities utilizing TensorFlow Lite, innovative voice interaction, and intelligent mobile assistant capabilities into one system. This integration provides a more comprehensive service that is not just limited to identifying objects and reading texts. It is also able to provide navigational instructions and respond to verbal requests almost instantaneously, and will enhance the everyday tasks and independence of individuals with blindness and visual impairment.

## REFERENCES

Adi, S. E., & Casson, A. J. (2021). Design and optimization of a TensorFlow Lite deep learning neural network for human activity recognition on a smartphone. *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 7028–7031. <https://doi.org/10.1109/EMBC46164.2021.9629549>

Alharbi, S., Alrazgan, M., Alrashed, A., Alnomasi, T., Almojel, R., Alharbi, R., Alharbi, S., Alturki, S., Alshehri, F., & Almojil, M. (2021). Automatic Speech Recognition: Systematic Literature Review. *IEEE Access*, *9*, 131858–131876. <https://doi.org/10.1109/ACCESS.2021.3112535>

Ayachi, R., Said, Y., Afif, M., Alshammari, A., Hleili, M., & Ben Abdelali, A. (2025). Assessing YOLO models for real-time object detection in urban environments for advanced driver-assistance systems (ADAS). *Alexandria Engineering Journal*, *123*, 530–549. <https://doi.org/10.1016/j.aej.2025.03.077>

B, D. A., Salian, S. H., & C, T. D. (2025). Voice Assistive System for Visually Impaired: Development, Applications, Challenges, and Future Trends. In *Int. J. Sci. R. Tech* (Vol. 2, Issue 3). [www.ijsrtjournal.com](http://www.ijsrtjournal.com)

Chai, J., Zeng, H., Li, A., & Ngai, E. W. T. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, *6*, 100134. <https://doi.org/10.1016/j.mlwa.2021.100134>

Chen, L., Li, G., Zhang, S., Mao, W., & Zhang, M. (2024). YOLO-SAG: An improved wildlife object detection algorithm based on YOLOv8n. *Ecological Informatics*, *83*, 102791. <https://doi.org/10.1016/j.ecoinf.2024.102791>

Christian, I., & Kusuma, G. P. (2023). Improving OCR Performance on Low-Quality Image Using Pre-processing and Post-processing Methods. *International Journal of Engineering Trends and Technology*, *71*(6), 396–405. <https://doi.org/10.14445/22315381/IJETT-V71I6P239>

Dawood, K. B. (2024). Tesseract OCR: Features, Applications, and Limitations. *Folio3AI Blog*.

Fauzan, M. R., & Wibowo, A. P. W. (2021). PENDETEKSIAN PLAT NOMOR KENDARAAN MENGGUNAKAN ALGORITMA YOU ONLY LOOK ONCE V3 DAN TESSERACT. *Jurnal Ilmiah Teknologi Infomasi Terapan*, *8*(1), 57–62. <https://doi.org/10.33197/jitter.vol8.iss1.2021.718>

Francis, Sofi. A., & Sangeetha, M. (2025). A comparison study on optical character recognition models in mathematical equations and in any language. *Results in Control and Optimization*, *18*, 100532. <https://doi.org/10.1016/j.rico.2025.100532>

Gawande, U., Rathod, N., Bodkhe, P., Kolhe, P., Amlani, H., & Thaokar, C. (2023). Novel Machine Learning based Text-To-Speech Device for Visually Impaired People. *2023 2nd International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN)*, 1–5. <https://doi.org/10.1109/ICSTSN57873.2023.10151637>

Google. (2024). LiteRT overview. *Google AI for Developers*.

Hüseyin YILDIRIM, H., & Rovshenov, A. (2022). *Designing of Volunteer Reader Mobile Application for Visually Impaired Individuals*. [www.serialbox.com](http://www.serialbox.com)

IANCU, B. (2019). Evaluating Google Speech-to-Text API’s Performance for Romanian e-Learning Resources. *Informatica Economica*, *23*(1/2019), 17–25. <https://doi.org/10.12948/issn14531305/23.1.2019.02>

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo Algorithm Developments. *Procedia Computer Science*, *199*, 1066–1073. <https://doi.org/10.1016/j.procs.2022.01.135>

Joshi, K. (2024). Study of Tesseract OCR. *GLS KALP: Journal of Multidisciplinary Studies*, *1*(2), 41–50. <https://doi.org/10.69974/glskalp.01.02.54>

Kelta, Z. (2022, September). *YOLO Object Detection Explained: A Beginner’s Guide*.

Kevin Wiguna, A., Suciati, N., & Khotimah, W. N. (2020). Aplikasi Penerjemah Gambar Teks Berbahasa Inggris Menggunakan Teknologi Realitas Tertambah pada Perangkat Berbasis Android. *Jurnal Teknik ITS*, *8*(1). <https://doi.org/10.12962/j23373539.v8i1.40070>

Kumar, N., & Jain, A. (2022). A DEEP LEARNING BASED MODEL TO ASSIST BLIND PEOPLE IN THEIR NAVIGATION. *Journal of Information Technology Education: Innovations in Practice*, *21*, 95–114. <https://doi.org/10.28945/5006>

Le Phuong, N., Vu, L., Mai, H., & Minh, T. (2021). Improving EFL Students’ Intonation In-Text Using Shadowing Technique with the Implementation of Google Text-to-Speech. *AsiaCALL Online Journal*, *13*(1), 93–121. <https://doi.org/10.11251/acoj.13.01.006>

Lestari, I. N. T., & Mulyana, D. I. (2022). Implementation of OCR (Optical Character Recognition) Using Tesseract in Detecting Character in Quotes Text Images. *Journal of Applied Engineering and Technological Science (JAETS)*, *4*(1), 58–63. <https://doi.org/10.37385/jaets.v4i1.905>

Li, Y., Dang, X., Tian, H., Sun, T., Wang, Z., Ma, L., Klein, J., & Bissyandé, T. F. (2025). An empirical study of AI techniques in mobile applications. *Journal of Systems and Software*, *219*, 112233. <https://doi.org/10.1016/j.jss.2024.112233>

More, A., Gayakwad, T., Suryawanshi, M., Kshirsagar, S., & Deole, P. (2022). ANDROID APPLICATION FOR VISUALLY IMPAIRED PEOPLE BASED ON AI TECHNOLOGY. *International Research Journal of Modernization in Engineering Technology and Science*, *4*(4). [www.irjmets.com](http://www.irjmets.com)

Mubeen, Dr. S., Brahmani, J., Kalyan, D. P., Jagirdar, A., & Kumar, A. P. (2022). Optical Character Recognition Using Tesseract. *International Journal for Research in Applied Science and Engineering Technology*, *10*(11), 672–675. <https://doi.org/10.22214/ijraset.2022.47414>

Nayak, S., & Chandrakala, C. B. (2020). Assistive mobile application for visually impaired people. *International Journal of Interactive Mobile Technologies*, *14*(16), 52–69. <https://doi.org/10.3991/ijim.v14i16.15295>

Patience, O. O., Amaechi, E. M., George, O., & Isaac, O. N. (2024). Enhanced Text Recognition in Images Using Tesseract OCR within the Laravel Framework. *Asian Journal of Research in Computer Science*, *17*(9), 58–69. <https://doi.org/10.9734/ajrcos/2024/v17i9499>

Raffoul, S., & Jaber, L. (2023). Text-to-Speech Software and Reading Comprehension: The Impact for Students with Learning Disabilities. *Canadian Journal of Learning and Technology*, *49*(2), 1–18. <https://doi.org/10.21432/cjlt28296>

Raghavan, R., Krishnan, V., Nishad, H., & Shaikh, B. (2021). Virtual AI Assistant for Person with Partial Vision Impairment. *ITM Web of Conferences*, *37*, 01019. https://doi.org/10.1051/itmconf/20213701019

S, R. B., Marium, A., Srinivasan, G. N., & Shetty, S. A. (2020). Literature Survey on Object Detection using YOLO. *International Research Journal of Engineering and Technology*. [www.irjet.net](http://www.irjet.net)

Sahu, N., & Sonkusare, M. (2017). A Study on Optical Character Recognition Techniques. *International Journal of Computational Science, Information Technology and Control Engineering*, *4*(1), 01–15. <https://doi.org/10.5121/ijcsitce.2017.4101>

Smith, R. (2021). An Overview of the Tesseract OCR Engine. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, 629–633. <https://doi.org/10.1109/ICDAR.2007.4376991>

Tharun, D. C., Sandeep, K., Diganth, A. B., Salian, S. H., & Ganesh. (2025). Voice Assistive System for Visually Impaired: Development, Applications, Challenges, and Future Trends. *International Journal of Scientific Research and Technology*, *2*(3).

Xu, B., Tao, C., Feng, Z., Raqui, Y., & Ranwez, S. (2021). *A Benchmarking on Cloud based Speech-To-Text Services for French Speech and Background Noise Effect*. <http://arxiv.org/abs/2105.03409>

Zhen, T. (2025). Optimization Strategies for Low-Power AI Models on Embedded Devices. *Applied and Computational Engineering*, *133*(1), 38–45. <https://doi.org/10.54254/2755-2721/2025.20598>

Zhu, J., Qin, C., & Choi, D. (2025). YOLO-SDLUWD: YOLOv7-based small target detection network for infrared images in complex backgrounds. *Digital Communications and Networks*, *11*(2), 269–279. <https://doi.org/10.1016/j.dcan.2023.11.001>