**Bulletin of Electrical Engineering and Informatics**
Vol. 14, No. 1, February 2025, pp. 485∼504
ISSN: 2302-9285, DOI: 10.11591/eei.v14i1.7904 

# SeeAround: an offline mobile live support system for the visually impaired

**Othmane Sebban[1], Ahmed Azough[2], Mohamed Lamrini[1]**
[1]Laboratory of Applied Physics, Informatics and Statistics (LPAIS), Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez, Morocco
[2]Léonard de Vinci Pôle Universitaire, Research Center, Paris La Défense, France

**Article Info**

***Article history:***

Received Nov 19, 2023
Revised Aug 16, 2024
Accepted Aug 25, 2024

***Keywords:***

Google Translate API
Image captioning
Optical character recognition
Scale-invariant feature transform
Text-to-speech API
Visually impairment

**ABSTRACT**

The inability of blind or partially-sighted people to understand visual content and real-life situations reduces their standard of living, especially in a world mainly tailored for sighted individuals. Despite the progress made by certain devices to assist them in using touch, sound, or other senses, these solutions often fall short of bridging the comprehension gap. Our work proposes an intuitive, user-friendly mobile-based framework named "SeeAround" that is capable of automatically providing real-time audio descriptions of the user's immediate visual surroundings. Our solution addresses this challenge by leveraging key point detection, image captioning, text-to-speech (TTS), optical character recognition (OCR), and translation algorithms to offer comprehensive support for visually impaired individuals. Our system architecture relies on convolutional neural networks (CNNs) such as Inception-V3, Inception-V4, and ResNet152-V2 to extract detailed features from images and employs a multi-gated recurrent unit (GRU) decoder to generate word-by-word natural language descriptions. Our framework was integrated into mobile applications and optimized with TensorFlow lite pre-trained models for easy integration on the Android platform.



***Corresponding Author:***

Othmane Sebban
Laboratory of Applied Physics, Informatics and Statistics (LPAIS), Faculty of Sciences Dhar El Mahraz
Sidi Mohamed Ben Abdellah University
Fez 30003, Morocco
Email: othmane.sebban.usmba.ac.ma

## 1. INTRODUCTION

Visual impairment represents a major challenge for individuals, affecting their independence and ability to interpret and interact with their environment. To address these issues, technology is playing an increasingly crucial role in providing solutions designed to improve the quality of life of visually impaired people [1]. Although significant advances have been made through the use of assistive technologies, such as screen readers and Braille displays, these innovations have often not been sufficient to fully meet the needs of visually impaired people. In particular, they do not offer real-time assistance or an in-depth understanding of the environment that these individuals require to navigate their daily lives independently.

Existing solutions, despite their value, have significant shortcomings that limit their effectiveness. For example, image description systems that rely on predefined rules often fail to capture the nuance and fluidity required for accurate interpretation. Besides, navigation applications struggle to provide real-time, rich, and detailed contextual information about the environment. These limitations underline an urgent need

for development and innovation in assistive technologies for visually impaired people [1]. More intuitive and adapted tools are needed to truly enrich their perception of the world and simplify their everyday lives. By linking these aspects, it becomes clear that although progress has been made, much remains to be done to meet the global needs of visually impaired people, requiring a more holistic and integrated approach to the development of assistive technologies.

Previous works [2] often suffer from a lack of integration, accessibility, and usability while failing to fully exploit the potential of advanced technologies such as deep learning algorithms or real-time image processing. These shortcomings persist despite advances in assistive technologies [3], underscoring the need for new approaches to overcome these deficiencies. Our study aims to address these issues by introducing a new real-time speech assistance system that combines several advanced technologies, including image captioning, optical character recognition (OCR), translation, and text-to-speech (TTS) [3]-[5]. Our contribution lies in integrating these technologies into a unified platform, improving accessibility, usability, and autonomy for visually impaired users. Our innovative approach is based on the use of image captioning as a first step in helping visually impaired users understand their environment in real-time, in conjunction with text-to-speech and photo-based OCR [4]. We have conducted a study to assess the effectiveness of this approach, examining the different strategies and difficulties encountered when using the system.

The development of the "SeeAround" smartphone application, designed to help blind and partially-sighted people [6] understand their environment, is the focus of this article. The application transmits photographs captured live to the image/OCR captioning system, which processes them and produces audio in several languages, including English, Arabic, Spanish, Italian, and Indian languages. This translation module aims to provide visually impaired users with descriptions of their environment in their native language, facilitating better understanding and interaction with the world around them.

The diagram in Figure 1 illustrates the overall workflow of our system. This diagram details the integration of advanced technologies such as OCR, image captioning, translation, and text-to-speech. Starting with image capture by the user, the system processes the image through these modules to generate an audio description of the environment in real time. This innovative approach aims to enhance users' autonomy and interaction with their environment, underscoring the importance of user-centered design and accessibility in the development of assistive technologies.

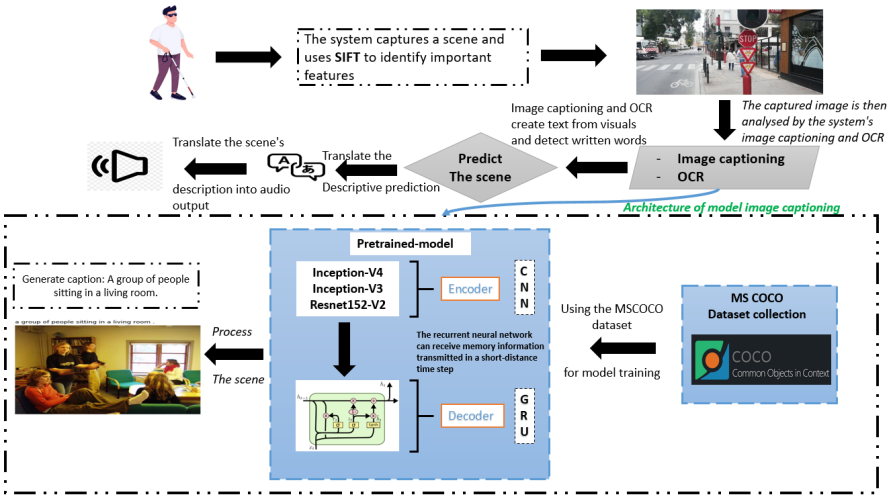



Figure 1. Flow diagram for the entire system

This paper is divided into six main sections, each covering an essential aspect of the subject at hand. In section 2, we provide a brief overview of previous work in the field of mobile applications applied to aiding the visually impaired. We present their flaws and make our proposal for improving the system. This proposal includes a specific algorithm for image processing as well as more advanced modules for enhanced functionality. Section 3 sets out the approach we have adopted for the key modules of our mobile application, "SeeAround." Here, we detail the choices and features of the modules to deliver an optimal user experience. In section 4, we

describe in detail the development process for the mobile system integrated into the Android studio environment, which is necessary for the creation of our mobile application. The results of the experiments carried out are presented in section 5, together with an in-depth analysis of these results. This section provides an insight into the impact and effectiveness of the technical choices and functionalities implemented. Finally, section 6 offers final thoughts and summarizes the main conclusions drawn from our study. We conclude the paper by highlighting the implications of our work and suggesting potential directions for future research in this field.

## 2. RELATED WORK

In recent years, the field of assistive technologies for the visually impaired has seen significant advances. These innovations are designed to improve navigation, provide crucial information about the environment, and enhance the usability of mobile devices for people with visual impairments. With the widespread adoption of smartphones and the evolution of mobile operating systems such as Android and iOS, there has been a notable increase in the development of features specifically designed to support this demographic. This literature review covers the main areas of these technological advances, highlighting our essential role in improving our system against older systems that additionally help visually impaired people's quality of life.

Kılıç [6] introduced a live image captioning application utilizing long short-term memory (LSTM) for speech synthesis integrated with an encoder-decoder structure. However, the system's reliance on a client-server architecture for image registration in the firebase database renders it unsuitable for real-time applications. In a related study [7], emphasis was placed on object detection, banknote recognition, and OCR. While these systems address specific needs, they may fall short in terms of real-time responsiveness and holistic functionality.

In addition, a system evaluated on the MS-COCO dataset demonstrated significant advances over existing methodologies [8]. The system they propose stands out in particular for its seamless integration with the custom Android app named "IMECA," which enables captions to be generated when the device is online. However, this dependence on an internet connection may pose practical limitations for some users, hampering the supposed real-time assistance capabilities.

Another solution introduces an Android application named "Eye of Horus," which generates textual captions and descriptions for images captured by the smartphone's camera [9]. Images are transferred to a remote server via a cloud system, where they are processed for caption generation, which is then returned and displayed on the application. Nevertheless, the system we have developed goes beyond the capabilities of this platform without using a server connected to a firebase database, guaranteeing immediate, and real-time assistance for visually impaired users.

In a separate endeavor, researchers devised a "narrator" system tailored for visually impaired individuals, incorporating convolutional neural networks (CNN), recurrent neural networks (RNN), LSTM, OCR, and text-to-speech conversion technologies [10]. This system's simplicity and scalability offer promising prospects, extending to the generation of continuous subtitles for videos. However, its reliance on still images extracted from video sequences for caption generation diverges from the live captioning functionality of our system. In conclusion, although the literature review shows a constant evolution of assistive technologies for the visually impaired, further progress is needed to meet the standards for a mobile-based assistive system that is both effective and reliable. This is what will be addressed in the next section.

## 3. METHOD

Our innovative "SeeAround" system is designed specifically for visually impaired individuals, providing a real-time experience through phone camera technology. It offers instant interaction with the surroundings and improved object recognition accuracy through the integration of the scale-invariant feature transform (SIFT) algorithm. The "SeeAround" mobile application aims to improve accessibility for visually impaired people by providing audio descriptions of images through automatic image captioning on a mobile device that can be used as illustrated in Figure 2. Our framework consists of a pipeline comprising six modules, each executing a specific task sequentially. Figure 3 illustrates an integrated system combining several modules for processing and understanding images and text. The OCR module extracts text from images, while the subtitling module generates text descriptions. These texts can be translated and converted into audio speech using the translation and text-to-speech modules. The system extracts key live images from mobile cameras and detects important features using SIFT, improving the accessibility of visual information for the visually impaired.

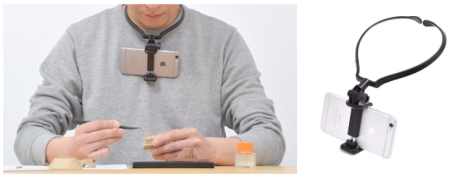


Figure 2. User interaction with the SeeAround mobile application

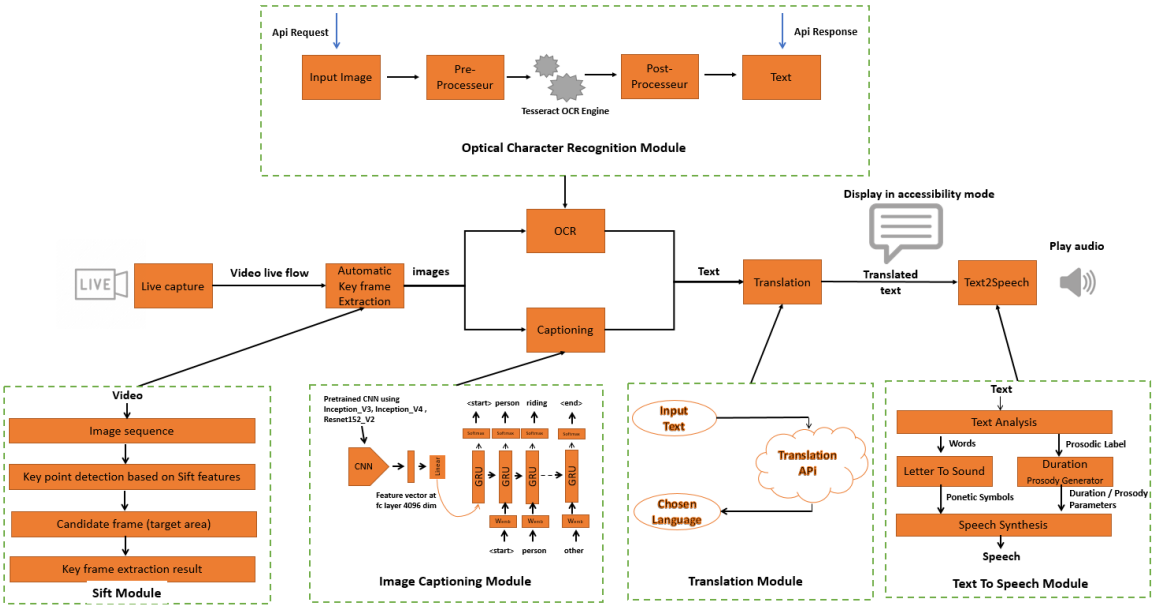



Figure 3. The SeeAround system's architecture

The initial module, discussed in subsection 3.1, focuses on keyframe extraction and extracts the main-frames from the live video mobile camera. Subsection 3.2 elaborates on the image captioning module, which employs an encoder-decoder learning approach optimized using TensorFlow lite for image subtitling. A multi-gated recurrent unit (GRU) automated image captioning model is used, guaranteeing accurate, context-sensitive descriptions. Following this, subsection 3.3 introduces the OCR module, which encompasses two text detection methods for detecting the text contained in the surrounding environment of the user. Subsection 3.4 details the translation module, converting languages into those preferred by the visually impaired. Finally, speech synthesis is facilitated by the speech-to-text module, elucidated in subsection 3.5, whether from OCR-detected text or image subtitles. In summary, our method uses advanced neural network architectures and optimization techniques to provide an accessible and effective solution for visually impaired people. It enables accurate text recognition, the generation of text descriptions from images, the translation of extracted text, and the conversion to audio in real-time.

### 3.1. Keyframe extraction module

The first module of our system searches for keyframes. The SIFT algorithm, which Lowe [11] first introduced in 2004 as a method for locating key points and their descriptions, is an effective keyframe extraction technique. Keypoints possess several properties, including their (x,y) coordinates, neighborhood size, orientation angle, and response strength. Utilizing the SIFT technique for keyframe extraction involves selecting optimal frames for each shot. Initially, a portion of these frames is gathered into an assortment known as the keyframes candidates set (KCS), ensuring each shot has at least one keyframe [12].

We have implemented a key frame extraction technique based on the SIFT descriptor. To select key images representative of each shot, we adopted an approach whereby certain images are grouped in a set known as a KCS. This is illustrated in Figure 3.

For the selection of key images, we adopt an approach in which certain images are grouped in a set called the key image candidate set (KICS). The first image of each shot is automatically included to ensure minimal representation. Subsequent images are selected according to a windowing rule, which consists of choosing images at regular positions defined by the size of a window. We experimented with a window size of n=25 [12], corresponding to the capture/exposure rate of most images, given that significant variations in content between consecutive images do not usually occur within a one-second interval.

Next, SIFT features are extracted from the images in KCS, generating 128-dimensional feature vectors for each image [12]. The number of vectors varies according to the image content, which justifies the use of the windowing rule. Despite the higher computational cost associated with extracting SIFT vectors compared to commonly used color histograms, SIFT features enable accurate image identification by preserving points of interest invariant to variations in illumination, rotation, and scale [11], [13], [14]. Figure 4 illustrates the key frame extraction process using the SIFT algorithm on mobile devices. Figure 4(a) shows the key points detected in the first image, totaling 200 points. Figure 4(b) shows the key points identified in the second image, with a total of 230 points. Similarly, Figure 4(c) shows 225 key points detected in the third image. Figure 4(d) shows 215 key points in the fourth image, while Figure 4(e) presents 191 key points in the fifth image. Finally, Figure 4(f) shows the 208 key points detected in the last image. Each sub-figure illustrates the results of keyframe extraction according to their position during real-time capture.


Figure 4. Keyframe extraction process using SIFT on mobile devices; (a) key points detected in the first image: 200 key points, (b) key points detected in the second image: 230 key points, (c) key points detected in the third image: 225 key points, (d) key points detected in the fourth image: 215 key points, (e) key points detected in the fifth image: 191 key points, and (f) key points detected in the final image: 208 key points

### 3.2. Image captioning module

Image captioning, which involves the automatic description of an image, is a challenge for machines despite its simplicity for humans, as shown in Figure 5. This task requires the exploitation of techniques from computer vision, artificial intelligence, and natural language processing (NLP). Image captioning has numerous potential applications, ranging from social media and autonomous vehicles to automatic NLP, image retrieval, and assistive devices for the visually impaired.

This section presents the theoretical underpinnings of the proposed image captioning technique, which is based on an encoder-decoder framework. We will explore the CNN architectures used to extract visual elements and image features, as well as the GRU-based decoders required to generate image captions. This approach will be illustrated by Figure 5(a) showing an example captioning of "a computer screen and keyboard on a desk," and Figure 5(b) illustrating a captioning of "a man sitting at a desk with a laptop computer".

To generate the captions, we use two neural network architectures in the "SeeAround" application. First, CNNs serve as encoders, using pre-trained models including Inception-V3 [8], Inception-V4 [15], and ResNet152-V2 [16]. These models extract image features, which are then fed into our second neural network architecture, the GRU. The GRU acts as a decoder, generating captions based on the extracted image features.


(a) (b)

Figure 5. A sample caption for two images; (a) a computer monitor and a keyboard on a desk and (b) a man sitting at a desk with a laptop computer

#### 3.2.1. Encoder

CNNs serve as the foundation of conventional encoder designs due to their remarkable feature extraction capabilities and ability to handle high-dimensional data effectively. The process of image encoding involves converting the image data into a feature vector. Fully connected (FC), pooling, and convolutional layers within the CNN architecture produce this vector, which encapsulates the crucial elements of the image.

Sophisticated computer vision algorithms are necessary for feature extraction and visual interpretation in image captioning tasks. Deep CNN architectures, such as Inception-V3 [8], Inception-V4 [15], and ResNet152-V2 [16], are particularly effective in meeting these requirements. In addition to Inception-V4, our work utilizes pre-trained deep CNN architectures, including ResNet152-V2 and Inception-V3, in the encoder section.

Inception-V3, a deep CNN with 42 convolutional, pooling, and FC layers achieved second place in the 2015 ImageNet large scale visual recognition challenge (ILSVRC) competition. Deep separable convolutions have been added to this architecture to create Xception, a version that outperforms Inception-V3 on the ImageNet dataset [8]. The structure of Inception-V3 is depicted in Figure 6.

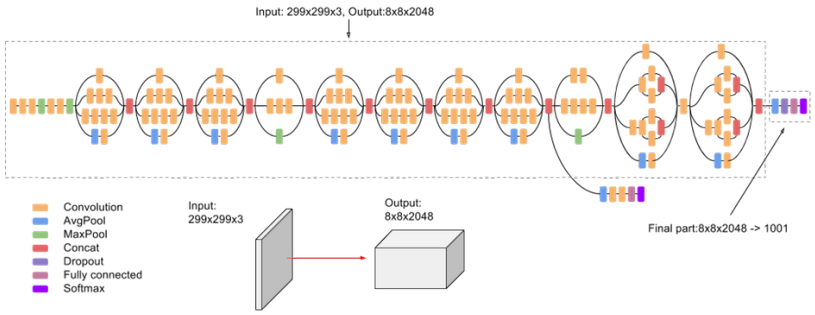



Figure 6. Inception-V3 global architecture

Similarly, Inception-V4 improves upon earlier versions of the Inception family by incorporating more inception modules and streamlining the design compared to Inception-V3. The structure of Inception-V4 is depicted in Figure 7. Another noteworthy architecture is ResNet152-V2, which boasts additional layers compared to the widely used ResNet50 model. This increased depth comes at the cost of higher memory requirements, making it suitable for large-scale machine-learning systems. The structure of ResNet152-V2 is depicted in Figure 8. In summary, the encoder module utilizes the convolutional and pooling layers of CNN architectures to extract high-level features from the input image. These features [17] are then passed to the decoder for caption synthesis.

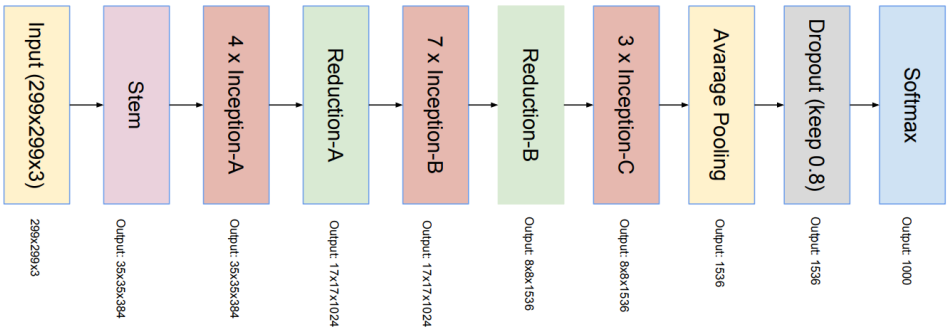



Figure 7. Inception-V4 global architecture

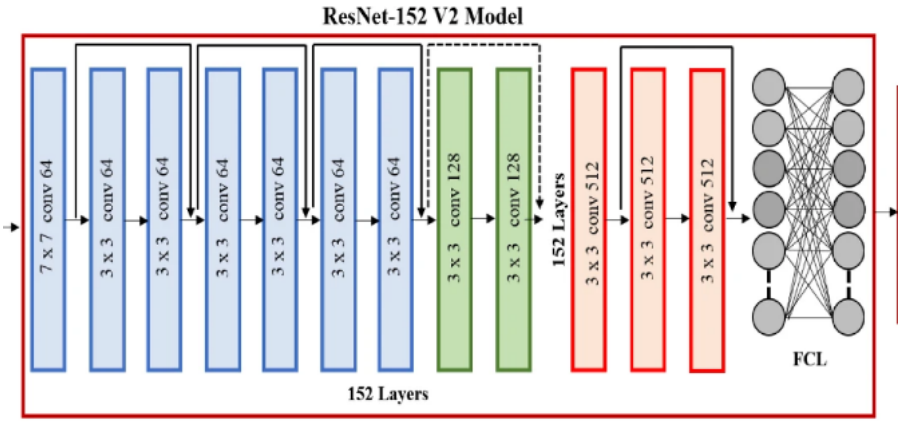



Figure 8. Resnet152-V2 global architecture

### 3.2.2. Decoder

A decoder constructs sentences to describe an image by utilizing feature representation and generating semantically appropriate sentences. Decoder construction commonly relies on RNNs, which can retain sections of inputs and generate meaningful captions from them [18]. RNNs are deep networks that utilize their internal state [19], [20] to analyze input sequences for various sequential applications, including speech recognition and image captioning. To compute the current hidden state, the RNN combines the hidden state from the previous time step with the current input using a nonlinear activation function that alters the output at each time step. Figure 9 shows the architecture of the GRU, which uses triggering mechanisms to avoid the evanescent gradient problem. The GRU combines information from previous states and current inputs to produce relevant outputs, essential for generating accurate image descriptions.

The integration layer takes tokens representing digital elements and produces an integration vector, also known as a word integration vector, containing linguistic features. This vector is then used to feed the GRU, a type of RNN equipped with mechanisms for controlling the flow of information through its cells. The GRU comprises a hidden state vector along with update and reset gates. The work by Keskin *et al.* [21] clarifies how information flows through the GRU.

$$z_t\& = \sigma(W_z x_t + W_z h_{t-1}) \tag{1}$$

$$r_t\& = \sigma(W_r x_t + W_r h_{t-1}) \tag{2}$$

$$u_t\& = \tanh(W_h x_t + W_h(r_t \odot h_{t-1})) \tag{3}$$

$$h_t\& = (1 - z_t)h_{t-1} + z_t u_t \tag{4}$$

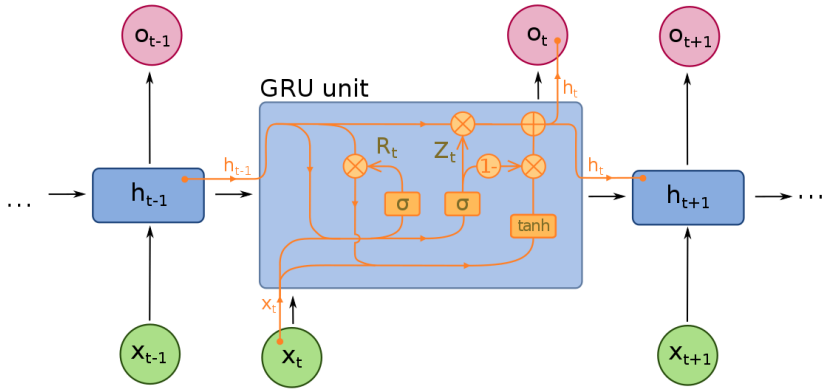



Figure 9. GRU architecture

Where $x_t$ and $h_t$ represent the input and hidden state vectors, and $r_t$, $z_t$, and $u_t$ correspond to the reset gate, update gate, and candidate hidden vector, respectively. $W$ denotes weight matrices; $\sigma$ and $\tanh$ denote sigmoid and hyperbolic tangent functions, respectively. $\odot$ denotes the element-wise multiplication operator. The multi-layer GRU is a combination of $K$-GRU for $k = 1, \ldots, K$. The first GRU layer takes the embedding vector, generated using a start-token from the embedding layer [21]. The output vector of the first layer feeds into the next GRU layer, and this process is repeated $K$ times, reaching the last output, which becomes the input for the FC layer. The embedding layer computes the first token, which the FC layer generates in the following time step.

The triggering mechanism in GRU addresses the vanishing gradient problem, while a gradient clipping strategy in GRU mitigates the exploding gradient problem [22]. GRU obtains input from the previous layer at each time step, allowing for the configuration of multiple layers. GRU demonstrates exceptional performance compared to conventional RNN-based architectures in numerous NLP tasks, including language modeling [18], [21].

### 3.2.3. Suggested pre-trained models for image captioning

We introduce a novel, pre-trained model designed to enhance image captions. Following the description of this proposed pre-trained model, we demonstrate its integration into our custom mobile application, 'SeeAround,' which utilizes an intuitive interface to implement the suggested pre-trained model. Specifically, we employ the deep GRU for image captioning. The encoder and decoder components of the encoder-decoder framework leverage NLP and OpenCV algorithms, respectively. RNN-based decoders convert the feature representation of an image, which CNN-based encoders have extracted, into captions in natural language. Various architectures, including init-inject, par-inject, pre-inject, and merge architectures, can be utilized to incorporate image and language features into RNNs [23]. In the init-inject architecture, the RNN's initial hidden state vector includes the image feature vector. This makes sure that the two are the same size, which makes it easier to set up an early binding architecture. Research indicates that the interject architecture outperforms others in terms of recovery and generation measures [23].

Research by Uslu *et al.* [17], propose a unique deep GRU design for generating natural language descriptions of images. An RNN-based decoder, operating under an init-inject architecture, implements this design. The decoder processes the features of the input image to generate a caption. It incorporates a multilayer GRU with an integration layer, GRUs, and a FC layer. The GRU learns to interpret the image's characteristics and vectors to produce meaningful attributions. The embedding layer represents words as significant vectors,

and the FC layer predicts the most relevant word corresponding to the attributions [17]. Converting words into vectors is necessary to enable RNNs to process them, as CNNs cannot handle word sequences. A commonly used pre-trained model for word integration creates vectors with the semantics of related words. The integration layer indexes words into integer tokens and converts them into 32-bit float arrays. Training the integration layer in conjunction with the network [17] captures the more compact aspects of words. We use GRU to model word relationships in captions, treating them as time-series data. Combining three separate RNNs creates a three-layer GRU [17], [21]. Encoder features create three vectors of the same size, added successively to the initial state of GRU layers. GRU layers receive time-series input from the integration layer's output.

Figure 10 illustrates the encoder-decoder framework using GRUs for the generation of natural language descriptions. CNNs such as Inception-V3, Inception-V4, and ResNet152-V2 are used as encoders to extract visual features from images, which are then transformed into linguistic vectors. GRU-based decoders use these vectors to model word relationships and generate textual captions.

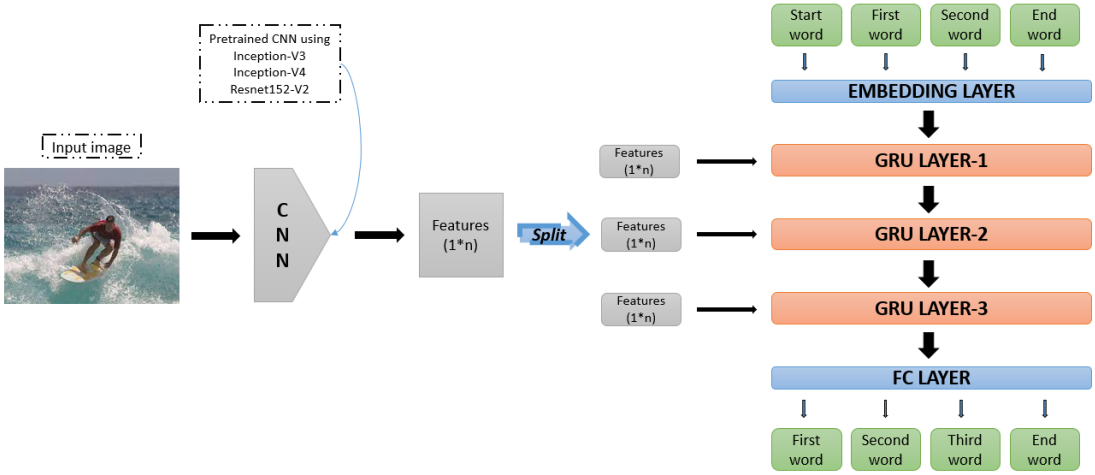



Figure 10. Encoder-decoder framework: leveraging GRU for natural language description

### 3.3. Optical character recognition module

OCR is the most direct method of converting printed and handwritten text into machine-encoded text, facilitating text mining, language translation, compact storage, online display, and editing [24]. Marosi in 2007 describes Tesseract as an OCR engine that processes input data as binary images with optionally defined polygonal text regions. During initial processing, Tesseract stores component outlines using connected component analysis. Nesting is the current method of grouping outlines into blobs, with text lines formed by the arrangement of these blobs. Character spacing determines variations in word breaks within text lines, with fixed-height text instantly segmented into character cells [25].

In this section, we present two approaches we have implemented in our system for the OCR module. We start with the first method, which is OCR pattern recognition for text extraction. The second method involves OCR feature detection, which complements the pattern recognition approach by focusing on the identification of unique text features to enhance accuracy and efficiency in text extraction.

#### 3.3.1. Pattern recognition optical character recognition

OCR is an electronic device that converts typed or handwritten letters into digital text. In other words, OCR is a system designed to scan documents and transform them into searchable text. Users can access this functionality [26]. A typical OCR system comprises several essential elements: scanning the image, initial preprocessing, identifying specific attributes, sorting and categorizing, and final processing. Figure 3 illustrates the basic components of OCR in a flowchart. To recognize characters in a scanned image, programs analyze characteristics such as the curvature, intersections, or slant of lines in a character, along with specific letter-related rules [24]. Figure 11 demonstrates the representation of detected text in the image: "Tirdiness can kill take a break". These samples are then used for character identification in the scanned image [27].


Figure 11. Illustration of OCR process flow in the SeeAround system

The initial image undergoes optical digitization [28]. Pre-processing involves noise reduction and various mathematical operations such as binarization, thinning, and edge detection. This process extracts distinctive features like edges, corners, and imperfections. In the classification phase, individual letters and characters are sorted and categorized. The final stage, post-processing, includes operations like grouping, error detection, and correction. The system captures the image and extracts textual or numerical patterns. To avoid redundancy, the image is cropped based on the text within it. The cropped image is then separated into individual letters or numbers, which are then interpreted for visually impaired users.

### 3.3.2. Feature detection optical character recognition

Using Tesseract OCR, OCR features are extracted from images, achieving a high data recognition rate through character correction. However, testing a large number of image files generates a considerable number of OCR features, as each image file produces a separate set of training data [28], [29]. This leads to an increased processing load. The functionalities present in OCR can be leveraged to identify data in other images, enabling the detection of personal information [29].

Detecting text from camera images of natural scenes is crucial for visually impaired individuals navigating complex environments. Text search is employed to identify and retrieve text within specific regions of the image [30]. This process involves classifying characters as they appear in the original image. To display the text in white on a black background, multiply the result by a new binary-transformed image. Text characters consist of strokes of constant or variable orientation, forming their basic structure. As depicted in Figure 12, programs are provided with numeric, alphabetic, or both types of samples in various fonts and formats. The final results of this sub-section.


Figure 12. Feature detection OCR: enhancing text recognition in images

Figure 13 shows the general results of OCR in our system. For OCR feature detection, Figure 13(a) shows the characters “10V” detected in the first image, Figure 13(b) identifies the character “P” in the second image, and Figure 13(c) detects “6V” in the third image. For OCR pattern recognition, Figure 13(d) shows the

printed text “GRAND TAXI,” Figure 13(e) shows the handwritten text “RESERVE AUX CONVOYEURS DE FONDS,” and Figure 13(f) displays the stylized text ”PRALINE.”.

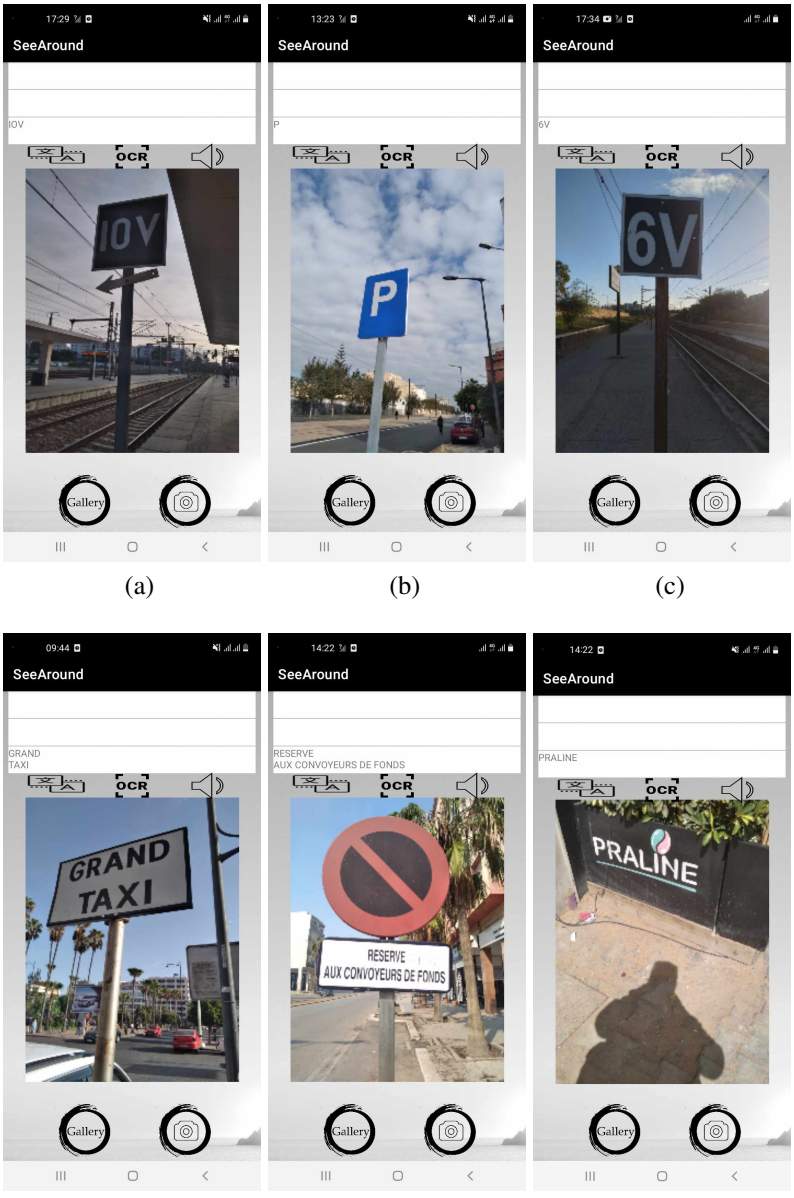



Figure 13. Sample OCR results demonstrating text detection capabilities; (a) result 1 of OCR feature detection, (b) result 2 of OCR feature detection, (c) result 3 of OCR feature detection, (d) result 1 of OCR pattern recognition, (e) result 2 of OCR pattern recognition, and (f) result 3 of OCR pattern recognition

### 3.4. Translation module

The translation module on our mobile device facilitates the conversion of text from English into various dialects, operating seamlessly on a mobile platform. The challenge of linguistic disparities has increasingly hindered effective information exchange over time, particularly in transnational communication. Interpreters are required to possess proficiency and knowledge in both the source and target languages, yet conventional methods for addressing linguistic barriers have proven ineffective and costly [31]. Moreover, teaching multilingualism in such diverse linguistic environments can be daunting, with tutoring sessions often presenting

financial and practical challenges.

In response to these issues, our study developed an Android app for language translation, aiming to facilitate stress-free communication, support language learning, and enhance translation capabilities. Leveraging Google's real-time translation API-iammannan along with NLP techniques, the application enables translation into widely spoken languages such as Spanish, Arabic, Hindi, French, and Italian [31]. Figure 14 illustrates the translation process, demonstrating how sentences inputted in English are translated into these languages, representing some of the most prevalent languages globally. The resulting outputs are provided in Arabic, Hindi, Italian, French, and Spanish.

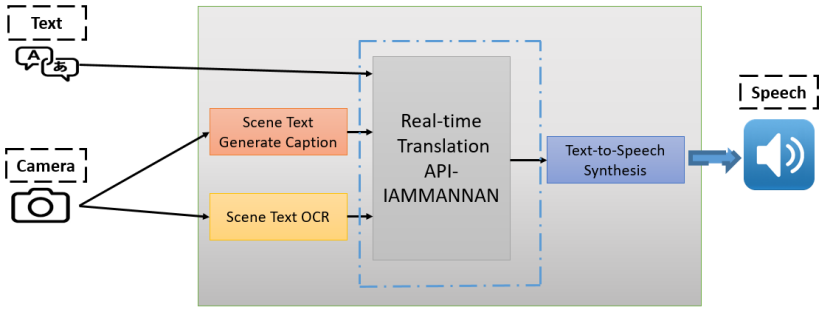



Figure 14. Translation module workflow: from text detection to multilingual support

### 3.5. Text to speech module

The fourth module, TTS, addresses the needs of visually impaired individuals through an Android application development solution. TTS technology enables the conversion of digital text into spoken words, serving as an assistive technology commonly referred to as "read-aloud" technology. Individuals with visual impairments often rely on others due to their inability to read handwritten or printed material. Although Braille paper documents are accessible, Braille street signs remain limited. To alleviate these challenges, technologies like TTS and OCR offer significant benefits to the blind and visually impaired community by enabling the audible reading of content from various sources such as books, newspapers, or documents.

The system comprises two main components: image captioning and image-based text detection. Users select their preferred language for translation and receive speech-to-text conversions [32]. Thanks to this system, blind or partially-sighted people can capture photographs accompanied by captions. Figure 15 illustrates the general architecture of the speech system. The module effectively addresses the needs of visually impaired individuals by providing access to digital text through spoken words, thereby enhancing their independence and accessibility to information.

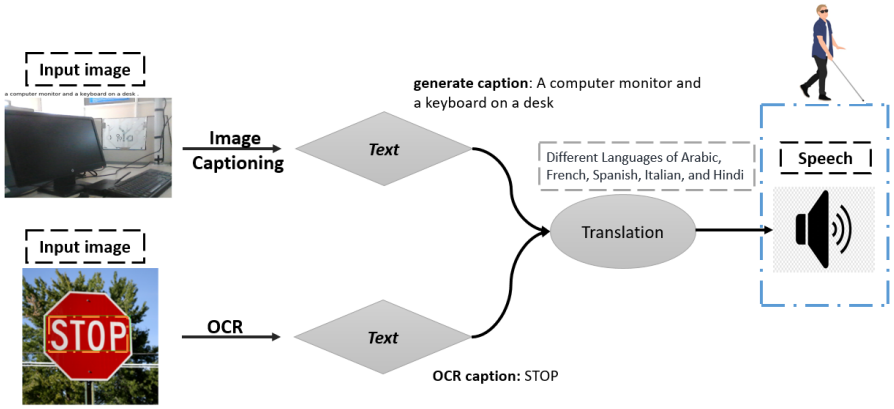



Figure 15. TTS conversion for accessibility: integrating OCR and captioning

## 4. REQUIREMENTS

### 4.1. Hardware requirements

The "SeeAround" smartphone app requires a functioning camera and a minimum of 1.86 GB of RAM. For optimal performance, the app is compatible with Windows 10 (64-bit) PCs featuring a Core i7 processor and 16 GB of RAM.

### 4.2. Software requirements

For the development of our smartphone app, several Java and Python programs and libraries are essential. Google has implemented a set of deep learning techniques called TensorFlow using an open-source library. To simplify the training and deployment process, Google created the TensorFlow API, which supports various machine learning models. This API makes it easy to train and deploy machine-learning models. Users can perform a wide range of mathematical and numerical calculations with this API.

For its operations, Google makes use of TensorFlow lite, which consists of two main components: an interpreter and a converter. The interpreter is responsible for executing optimized models on low-resource devices, while the converter transforms TensorFlow models into a format usable by the interpreter. The ultimate goal is to enable optimized execution of these models on low-resource devices. This optimization is of particular importance in the context of integration with OpenCV, facilitating efficient use of the TensorFlow Lite and OpenCV libraries on mobile devices. We pay particular attention to visually impaired users to ensure a clearer experience when developing captions for images.

## 5. RESULT AND DISCUSSION

The first step involves training the model using a suitable dataset to predict the captioning state of the images. Subsection 5.1 details the datasets and preprocessing performed. The evaluation measures are described in subsection 5.2, which comprises the three modules required in our "SeeAround" system. Subsection 5.3 presents the results of the various pre-trained image captioning models, as well as the OCR and translation modules, in an overview table. Finally, subsection 5.4 offers a general discussion to explain the performance, speed, and tolerance of our system.

### 5.1. Dataset and performance

To evaluate the proposed picture captioning system, a large dataset with reference captions is required. Picture captioning systems often use datasets like MSCOCO [33] because of their extensive content. With 118,000 images accessible for training and 5 images used for testing per test case, we have 41,000 tests with 5,000 reference annotations total in this example. MSCOCO is the largest visual recognition dataset in particular, and previous studies on picture captioning have demonstrated high semantic and grammatical performance when grouping images from MSCOCO.

### 5.2. The evaluation metrics

To assess the training quality of the pre-trained model, we use four evaluation metrics:

– BLEU is a popular metric that analyzes the co-occurrences of n-grams between the candidate and reference sentences. Computing the BLEU score involves analyzing the co-occurrences of n-grams between the candidate and reference sentences.

$$\text{BLEU} - N(ci, Si) = b(ci, Si) \exp\left(\sum_{n=1}^{N} \omega_n \log P_n(ci, Si)\right) \tag{5}$$

where,

$$b(ci, Si) = \begin{cases} 1 & \text{if } lc > ls \\ e^{1-\frac{ls}{lc}} & \text{if } lc \leq ls \end{cases} \text{ is a brief penalty; lc is}$$

The total length of candidate sentences ci; lS is the length of the corpus-level effective reference length; when multiple references for a candidate sentence are available, the closest reference length is selected. $\omega$n is typically held constant for [34]:

$$\text{for all } n; N = 1, 2, 3, 4: \quad P_n(ci, Si) = \frac{\sum_k \min(h_k(ci), \max(h_k(S_{ij})))_{j \in \mathcal{M}}}{\sum_k h_k(ci)}$$

precision score, and it favors short sentences. Hence, B-N (abbreviation of BLEU-N) measures the fraction of n-grams (up to 4-grams) that are in common between a candidate sentence and a reference sentence or a set of reference sentences.

– METEOR computes unigram precision and recall, extending exact word matches to include similar words based on WordNet synonyms and stemmed tokens. Compute the METEOR score as (6) [35]:

$$\text{METEOR} = (1 - \text{Pen}) \times \text{Fmean} \tag{6}$$

where Pen = $\gamma(\frac{ch}{m}^m)$ is a penalty item, m is a set of alignments, ch is the number of chunks of contiguous and identically ordered tokens in the sentence pair.

$$F_{\text{mean}} = \frac{P_m \cdot R_m}{\alpha P_m + (1 - \alpha) R_m}, \quad P_m = \frac{|m|}{\sum_k h_k(ci)}, \quad R_m = \frac{|m|}{\sum_k h_k(S_{ij})}$$

The METEOR metric is the harmonic mean of precision and recall between the best scoring reference and candidate.

– ROUGE-L: the metric is based on the longest common sequence (LCS). Given the length l(ci, sij) of the LCS between a pair of sentences, the definition of this metric is (7):

$$ROUGEL(ci, Si) = \frac{(1 + \beta^2) RlPl}{Rl + \beta^2 Pl} \tag{7}$$

where

$$Rl = \max_j \left( \frac{l(ci; S_{ij})}{|S_{ij}|} \right) \text{ stands for the recall of LCS} Pl = \max_j \left( \frac{l(ci; S_{ij})}{|Ci|} \right)$$

represents the precision of LCS, and E is a constant that is usually set to favor recall. Similar to ROUGE-L, there are some other metrics like ROUGE-N and ROUGE-S [35].

– CIDEr-D: the metric for n-grams of length n is computed using the average cosine similarity between the candidate sentence and the reference sentences, which accounts for both precision and recall. This metric is defined as (8) [36]:

$$CIDEr - D_n(ci, Si) = \sum_{n=1}^{N} \omega_n \text{CIDEr-D}(ci, Si) \tag{8}$$

$$CIDEr - D_n(ci, Si) = \frac{10}{m} e^{-\frac{(l(ci) - l(S_{ij}))^2}{2\omega^2}} \sum_j \left( \frac{\min(g^n(ci), g^n(S_{ij})) g^n(S_{ij})}{|g^n(ci)||g^n(S_{ij})|} \right)$$

is a vector formed by $g^k(ci)$ corresponding to all n-grams of length n; $| g^n(ci)|$ is the magnitude of the vector $g^n(ci)$; Similarly, $|g^n(sij)|$ is the magnitude of the vector $g^n(sij)$; l(ci) and l(sij) denote the lengths of candidate and reference sentences, respectively; $\sigma$ is a constant defined in advance.

The measure of the performance of the module's OCR and translation processes is its accuracy, defined respectively as (9) and (10):

$$\text{Accuracy for OCR } (\%) = \frac{\text{No. of correctly identified characters}}{\text{Total no. of characters}} \tag{9}$$

$$\text{Accuracy for translation } (\%) = \frac{\text{No. of correctly translated words}}{\text{Total no. of words}} \tag{10}$$

OCR accuracy measures the ability to correctly identify characters in the digitized text, reflecting the system's effectiveness in minimizing recognition errors. Translation accuracy, on the other hand, assesses the accuracy of converting words from one language to another, underscoring the importance of faithfully translating the meaning and context of the source text. These measures are vital to ensuring the quality and reliability of OCR and translation technologies, which are essential criteria for content accessibility and comprehension.

### 5.3. Results

We tested our suggested method using the BLEU, ROUGE-L, METEOR, and CIDEr metrics to assess its effectiveness with three pre-trained multilayer GRU models: Inception-V3, Inception-V4, and ResNet152-V2. The results of these performance measures are presented in Table 1, where the highest scores are highlighted in bold. We compared three different multilayer RNNs, including a three-layer model, and found that adding layers significantly improved the performance of our method. Compared with previous methods, our approach showed better performance on the BLEU-1, BLEU-2, BLEU-3, BLEU-4, METEOR, CIDEr, and ROUGE-L metrics. ResNet152-V2 turned out to be the best CNN architecture model on ImageNet, with the highest scores for the various evaluation metrics.

Table 1. Performance metric results of Inception-V3, Inception-V4, and Resnet152-V2

| Model | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | ROUGE-L | CIDEr | METEOR |
|---|---|---|---|---|---|---|---|
| Inception-V3 3-layer | 0.6379 | 0.4476 | 0.3038 | 0.2045 | 0.4640 | 0.6524 | - |
| Inception-V4 3-layer | 0.650 | 0.4690 | **0.3620** | 0.2270 | 0.480 | **0.8540** | - |
| Resnet152-V2 3-layer | **0.675** | **0.492** | 0.350 | **0.248** | **0.491** | - | **0.782** |

The performance of our system was analyzed in terms of response time and model quality for each module. The results are summarized in Table 2. The system takes around 3 to 4 seconds for the OCR and image captioning models, and 2 to 3 seconds for initializing and loading the translation models into memory. The average response time for speech synthesis is 1-2 seconds. These results show that the system delivers fast, accurate results under ideal conditions. To assess the quality of the image captioning model, we calculated the sum of the performances of the different pre-trained models.

Table 2. System "SeeAround" performance observations

| Module name | Response time (seconde) | Performance model quality (%) |
|---|---|---|
| Image captioning | 3.11 | 78.92 |
| OCR | 3.11 | 93.2 |
| Translation | 2.22 | 95.7 |

By calculating an overall score by combining the performances of the three image captioning modules, we can assign weights to each evaluation metric (BLEU, CIDEr, and METEOR) according to their relative importance for our specific application. In our evaluation of image captioning models, we chose the BLEU, CIDEr, and METEOR metrics due to their relevance in our context of voice assistance for the visually impaired. We considered the availability of annotated data, ease of interpretation, and sensitivity to the specific characteristics of our task. While the ROUGE-L metric may be relevant in other contexts, it was not selected for our evaluation due to its lesser suitability for our specific application and its relative insensitivity to the nuances of image captions. The overall score is obtained by performing a linear combination of the modules' performances, where equal weights are assigned to each evaluation metric.

$$\text{Score global for image captioning} = \frac{\text{BLEU} + \text{CIDEr} + \text{METEOR}}{3} \tag{11}$$

The evaluation results demonstrated exceptional accuracy across image captioning, OCR, and translation modules, based on nearly 220 images tested.

To calculate the overall accuracy of our multi-stage "SeeAround" system as a function of the accuracy of each stage, we use the method known as the "probability product." We therefore have three modules with respective accuracies of 0.7856, 0.928, and 0.957. We calculate the overall accuracy as follows: overall accuracy=0.7856 * 0.928 * 0.957=0.704. This means that the overall accuracy of the entire process is approximately 70.4%.

### 5.4. Discussion

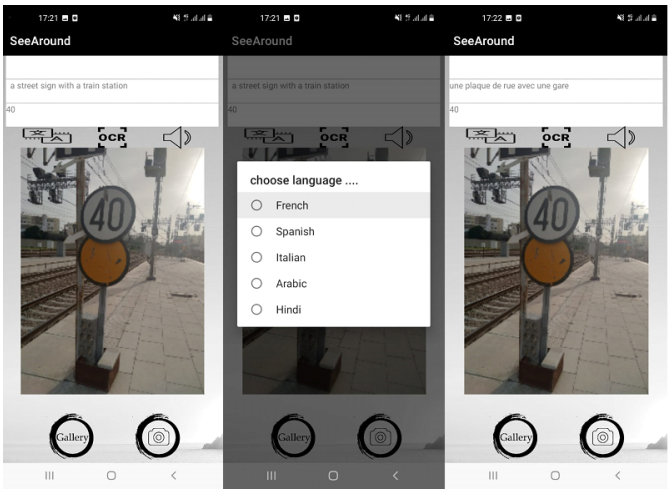
Our "SeeAround" system aims to enhance accessibility for visually impaired or blind people by providing real-time image captioning and OCR. This technology enables them to interact more easily with their

environment by converting visual information into text and speech. In addition, "SeeAround" enhances accessibility by offering translations in the five most widely spoken languages worldwide: French, Arabic, Spanish, Hindi, and Italian, enabling users to receive information in their native language. This feature is essential to facilitate communication and understanding for the visually impaired. Significant enhancements have been introduced in the "SeeAround" mobile application, focusing on the user interface, OCR, image captioning, text-to-speech conversion, and translation. These updates aim to optimize usability and portability, enabling instant access to subtitles and key features right from the home screen. The application uses the SIFT algorithm to improve system performance by reducing noise and visual redundancies, ensuring the production of accurate and relevant captions.

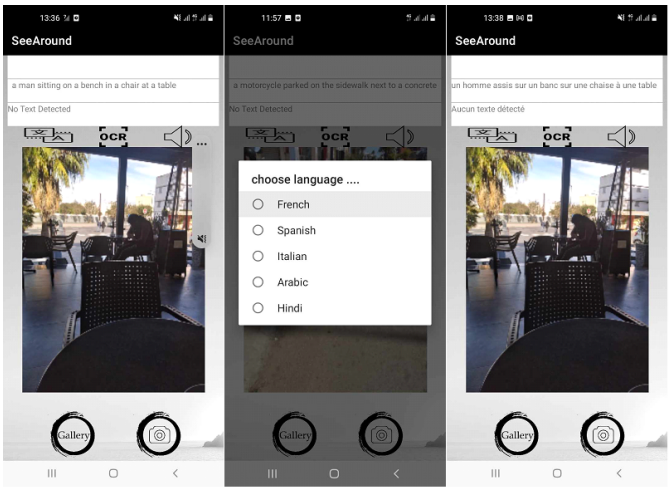
By leveraging the OpenCV library and integrating pre-trained image caption templates into a comprehensive system including OCR and real-time translation, we have significantly improved the quality and efficiency of our service. The use of the SIFT algorithm, in particular, enables efficient detection and extraction of unique image features, improving text recognition in variable lighting or quality conditions. The integration of the SIFT algorithm is a central pillar of our "SeeAround" system, playing a crucial role in improving the accuracy and relevance of the information provided to users. Thanks to its ability to identify and extract distinctive features from images, SIFT enables detailed and precise analysis, essential for the generation of captions and high-quality OCR. SIFT's role in "SeeAround" not only enhances image capture but also strengthens the basis of our commitment to providing an unprecedented user experience by facilitating access to critical visual information, converted into text and speech, for visually impaired or blind people. We tested our image captioning system using automated multi-GRU models and by integrating Keras and the TensorFlow library into the "SeeAround" mobile application, which we had already developed. The use of TensorFlow Lite via the Android inference library API facilitates efficient implementation on mobile devices, guaranteeing a smooth and responsive user experience.

Our comparison with other studies [6], [7], [9], [10] revealed promising results. When the SIFT algorithm was added, the accuracy of the image descriptions got a lot better, going beyond what systems that don't use noise reduction techniques can do. In addition, user studies indicated that the real-time functionality and offline translation capabilities significantly improved the user experience and autonomy compared to network-dependent solutions such as [8]. Importantly, our mobile system is the first to integrate multiple modules and use the SIFT algorithm, a feature absent in previous mobile applications [6]-[10]. We observed better results when comparing our system to other studies, especially with the addition of the SIFT algorithm, which reduces noise and image repetition. Unlike other applications that require a network connection to translate texts, our system operates offline in real time.

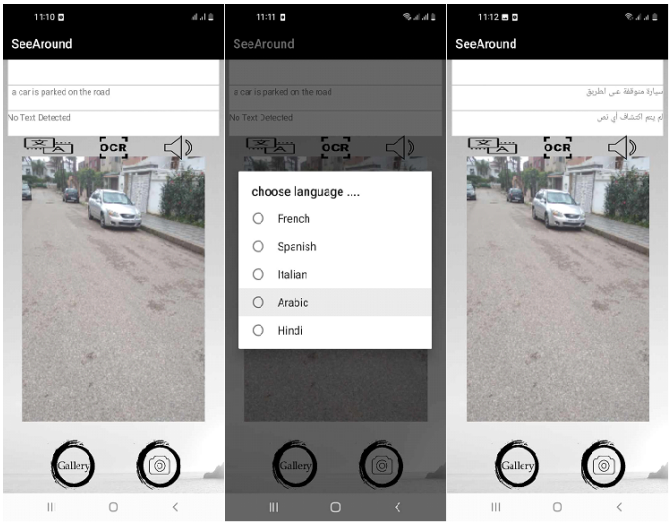
Enhancing accessibility for blind or visually impaired individuals, the "SeeAround" solution integrates OCR and real-time picture captioning, as illustrated in Figure 16. It takes a significant step forward by including real-time translation, enabling users to access information in their native languages. This advancement fosters improved communication and comprehension across various settings. Users can hear the received caption that shows beneath the image on the home page by hitting the mobile camera button to take a picture of the input image. By swiping down from the home screen, the user can also access an integrated camera. By default, "SeeAround" generates the caption in English, and it also provides text-to-speech conversion. Users can translate the English caption into the mobile device's display language by adjusting the language in the visual impairments settings. Furthermore, "SeeAround" introduces OCR. For Figures 16(a), the process begins with image captioning and text detection OCR from image captures in real time. This means that an image is captured, and then the text present in that image is extracted using OCR technology and image captioning. In Figures 16(b), the extracted text is translated. It seems that this step is intended to help visually impaired people understand the text in their native language. Finally, in Figures 16(c), the translated results are presented to the user using speech. This may mean that the translated text is converted into speech.




(a)




(b)




(c)

Figure 16. The SeeAround process; (a) OCR and generate caption, (b) select language for translation, and (c) OCR and generate caption with audio speech after translation

## 6. CONCLUSION

Our research shows how technology can significantly improve the quality of life of people who are blind or partially blind. In developing our mobile app, we aimed to ensure that users had quick and easy access to essential information through an intuitive design. Our program aims to increase accessibility and use of technology for a wider audience by integrating features such as voice recognition, image subtitle conversion, and several language options. In addition, our research highlights the crucial importance of optimization for mobile platforms, using tools such as TensorFlow lite to ensure optimal performance. By adopting this method, we can increase the accessibility of our image captioning, OCR, translation, and text-to-speech systems for visually impaired users. By rigorously evaluating our method using real data and examining the impact of the number of layers on the quality of the image descriptions generated, we were able to identify potential improvements for future iterations of our application. We are convinced that these technological advances will further enhance the experience of visually impaired users and promote their digital inclusion in our society. Ultimately, our work illustrates the importance of accessibility-focused technological innovation, and we look forward to continuing our efforts to create even more effective and inclusive solutions in the future. Notably, our 'SeeAround' system has demonstrated an efficiency of 70.4%, marking an important step towards our goal of making the world more accessible to visually impaired people. We plan to improve our subtitling models by adjusting the training parameters to form a more advanced model that performs better than the models already presented and by adding more advanced algorithms for OCR and translation, such as Firebase ML Kit and Google Cloud. In addition, we will improve image capture using key point extraction techniques such as BRISK, SURF, and ORB on mobile devices. These enhancements aim to better meet the needs of visually impaired users by optimizing accessibility and autonomy through innovative technological solutions.

## ACKNOWLEDGEMENT

We would like to thank the members of the research team for their helpful discussions and contributions, which made it possible to carry out this study.

## REFERENCES

[1] C. Rane, A. Lashkare, A. Karande, and Y. S. Rao, "Image captioning based smart navigation system for visually impaired," in *Proceedings - International Conference on Communication, Information and Computing Technology, ICCICT 2021*, Jun. 2021, pp. 1–5, doi: 10.1109/ICCICT50803.2021.9510102.

[2] B. Makav and V. Kilic, "A new image captioning approach for visually impaired people," in *ELECO 2019 - 11th International Conference on Electrical and Electronics Engineering*, *IEEE*, Nov. 2019, pp. 945–949, doi: 10.23919/ELECO47770.2019.8990630.

[3] M. F. Zamir, K. B. Khan, S. A. Khan, and E. Rehman, "Smart reader for visually impaired people based on optical character recognition," *Communications in Computer and Information Science*, vol. 1198, pp. 79–89, 2020, doi: 10.1007/978-981-15-5232-8_8.

[4] A. S. Agbemenu, J. Yankey, and E. O. Addo, "An automatic number plate recognition system using OpenCV and Tesseract OCR engine," *International Journal of Computer Applications*, vol. 180, no. 43, pp. 1–5, May 2018, doi: 10.5120/ijca2018917150.

[5] C. Chaitra *et al.*, "Image/video summarization in text/speech for visually impaired people," in *MysuruCon 2022 - 2022 IEEE 2nd Mysore Sub Section International Conference*, Oct. 2022, pp. 1–6, doi: 10.1109/MysuruCon55714.2022.9972653.

[6] V. Kiliç, "Deep gated recurrent unit for smartphone-based image captioning," *Sakarya University Journal of Computer and Information Sciences*, vol. 4, no. 2, pp. 181–191, Aug. 2021, doi: 10.35377/saucis.04.02.866409.

[7] J. Bagrecha, T. Shah, K. Shah, T. Gandhi, and S. Palwe, "VirtualEye: android application for the visually impaired," in *Recent Trends in Intensive Computing*, 2021, doi: 10.3233/apc210204.

[8] R. Keskin, O. T. Moral, V. Kilic, and A. Onan, "Multi-GRU based automated image captioning for smartphones," in *SIU 2021 - 29th IEEE Conference on Signal Processing and Communications Applications*, Jun. 2021, pp. 1–4, doi: 10.1109/SIU53274.2021.9477901.

[9] B. Makav and V. Kilic, "Smartphone-based image captioning for visually and hearing impaired," in *ELECO 2019 - 11th International Conference on Electrical and Electronics Engineering*, Nov. 2019, pp. 950–953, doi: 10.23919/ELECO47770.2019.8990395.

[10] S. Ubarhande, A. Magdum, H. Pawar, S. Phutane, and S. Sengupta, "NAYAN-narrator for the visually impaired," *SSRN Electronic Journal*, 2020, doi: 10.2139/ssrn.3735885.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[12] T. T. D. S. Barbieri and R. Goularte, "KS-SIFT: a keyframe extraction method based on local features," in *2014 IEEE International Symposium on Multimedia, ISM 2014*, Dec. 2015, pp. 13–17, doi: 10.1109/ISM.2014.52.

[13] H. M. Blanken, *Multimedia Retrieval*, Berlin: Springer-Verlag, 2007.

[14] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video-retrieval systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 580–588, Jun. 1999, doi: 10.1109/76.767124.

[15] P. Mathur, A. Gill, A. Yadav, A. Mishra, and N. K. Bansode, "Camera2Caption: A real-time image caption generator," in *ICCIDS 2017 - International Conference on Computational Intelligence in Data Science*, Jun. 2017, pp. 1–6, doi: 10.1109/IC-

CIDS.2017.8272660.
[16] M. Karpagam and A. Maheshwari, "Novel Computer Vision approach for identifying diseased Tomato leaves by classifyingleaf images for pest identication using ResNet152V2 architecture," *Research Square*, pp. 1–16, 2022, doi: 10.21203/rs.3.rs-1927876/v1.
[17] B. Uslu, Ö. Çayli, V. Kilic, and A. Onan, "Resnetbased deep gated recurrent unit for image captioning on smartphone," *European Journal of Science and Technology*, Apr. 2022, doi: 10.31590/ejosat.1107035.
[18] J. Joseph, S. Vineetha, and N. V. Sobhana, "A survey on deep learning based sentiment analysis," *Materials Today: Proceedings*, vol. 58, pp. 456–460, 2022, doi: 10.1016/j.matpr.2022.02.483.
[19] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing-Proceedings*, Apr. 2015, pp. 4520–4524, doi: 10.1109/ICASSP.2015.7178826.
[20] A. Shewalkar, D. nyavanandi, and S. A. Ludwig, "Performanceevaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 4, pp. 235–245, Oct. 2019, doi: 10.2478/jaiscr-2019-0006.
[21] R. Keskin, Ö. Çayli, Ö. T. Moral, V. Kiliç, and A. Onan, "A benchmark for feature-injection architectures in image captioning," *European Journal of Science and Technology*, Dec. 2021, doi: 10.31590/ejosat.1013329.
[22] G. Hoxha, F. Melgani, and J. Slaghenauffi, "A New CNN-RNN framework for remote sensing image captioning," in *2020 Mediterranean and Middle-East Geoscience and Remote Sensing Symposium, M2GARSS 2020-Proceedings*, Mar. 2020, pp. 1–4, doi: 10.1109/M2GARSS47143.2020.9105191.
[23] M. Tanti, A. Gatt, and K. P. Camilleri, "Where to put the image in an image caption generator," *Natural Language Engineering*, vol. 24, no. 3, pp. 467–489, May 2018, doi: 10.1017/S1351324918000098.
[24] R. Reeve Ingle, Y. Fujii, T. Deselaers, J. Baccash, and A. C. Popat, "A scalable handwritten text recognition system," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, Sep. 2019, pp. 17–24, doi: 10.1109/ICDAR.2019.00013.
[25] R. Smith, "An overview of the Tesseract OCR engine," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, Sep. 2007, pp. 629–633, doi: 10.1109/ICDAR.2007.4376991.
[26] S. Deshpande and R. Shriram, "Real time text detection and recognition on hand held objects to assist blind people," in *International Conference on Automatic Control and Dynamic Optimization Techniques, ICACDOT 2016*, Sep. 2017, pp. 1020–1024, doi: 10.1109/ICACDOT.2016.7877741.
[27] C. Yi, Y. Tian, and A. Arditi, "Portable camera-based assistive text and product label reading from hand-held objects for blind persons," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 808–817, Jun. 2014, doi: 10.1109/TMECH.2013.2261083.
[28] L. Fei, H. Chen, K. Wang, S. Lin, K. Yang, and R. Cheng, "Scene text detection and recognition system for visually impaired people in real world," in *Target and Background Signatures IV*, K. U. Stein and R. Schleijpen, Eds., SPIE, Oct. 2018, p. 29, doi: 10.1117/12.2325523.
[29] Y. K. Lee, J. Song, and Y. Won, "Improving personal information detection using OCR feature recognition rate," *Journal of Supercomputing*, vol. 75, no. 4, pp. 1941–1952, Apr. 2019, doi: 10.1007/s11227-018-2444-0.
[30] U. R. Khamdamov, M. N. Mukhiddinov, A. O. Mukhamedaminov, and O. N. Djuraev, "A novel method for extracting text from natural scene images and Tts," *European Science Review*, pp. 30–33, Feb. 2019, doi: 10.29013/esr-19-11.12.1-30-33.
[31] R. O. Ogundokun, J. B. Awotunde, S. Misra, T. Segun-Owolabi, E. A. Adeniyi, and V. Jaglan, "An android based language translator application," *Journal of Physics: Conference Series*, vol. 1767, no. 1, p. 012032, Feb. 2021, doi: 10.1088/1742-6596/1767/1/012032.
[32] K. Ragavi, P. Radja, and S. Chithra, "Portable text to speech converter for the visually impaired," *Advances in Intelligent Systems and Computing*, vol. 397, pp. 751–758, 2016, doi: 10.1007/978-81-322-2671-0_71.
[33] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: lessons learned from the 2015 MSCOCO image captioning challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 652–663, Apr. 2017, doi: 10.1109/TPAMI.2016.2587640.
[34] O. González-Chávez, G. Ruiz, D. Moctezuma, and T. Ramirez-delReal, "Are metrics measuring what they should? An evaluation of Image Captioning task metrics," *Signal Processing: Image Communication*, vol. 120, p. 117071, Jan. 2024, doi: 10.1016/j.image.2023.117071.
[35] M. S. Wajid, H. Terashima-Marin, P. Najafirad, and M. A. Wajid, "Deep learning and knowledge graph for image/video captioning: a review of datasets, evaluation metrics, and methods," *Engineering Reports*, vol. 6, no. 1, 2024, doi: 10.1002/eng2.12785.
[36] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: consensus-based image description evaluation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2015, pp. 4566–4575, doi: 10.1109/CVPR.2015.7299087.

## BIOGRAPHIES OF AUTHORS

**Othmane Sebban** is a software engineer who graduated with honors from Sidi Mohamed Ben Abdellah University in 2020. He specializes in intelligent decision-making systems. He is a Ph.D. student at the same university. He is simultaneously specializing in computer science and deep learning while working as a software engineer at the Morrocan Ministry of the Interior (MI). His main areas of interest in this study are the applications of contemporary AI algorithms to mechanical component design. He can be contacted at email: othmane.sebban@usmba.ac.ma.

**Ahmed Azough** is an Associate Professor at DeVinci Engineering School in Paris, where he also holds the position of Program Director for the M.Sc. in Computer Science-Data Science and is co-founder of the De Vinci VR-Lab. His professional career includes 3 years as a research engineer at France Telecom R&D (Paris), 1 year as an Assistant Professor at the University of Lyon, 5 years as a researcher at Genomic Vision (Paris), and 5 years as an associate professor at the University of Fez. He can be contacted at email: ahmed.azough@devinci.fr.

**Mohamed Lamrini** is a full Professor of computer science and researcher in the Department of Computer Science at the Dhra ElMahraz Faculty of Science at Sidi Mohamed Ben Abdellah University. He can be contacted at email: mohamed.lamrini@usmba.ac.ma.