

A Lightweight Visual Understanding System for Enhanced Assistance to the Visually Impaired Using an Embedded Platform

Adel Jalal Yousif ^{1,*}, Mohammed H. Al-Jammas ²

¹ Department of Computer Engineering, University of Mosul, Mosul, Iraq

² College of Electronics Engineering, Ninevah University, Mosul, Iraq

ARTICLE INFO

Article history:

Received May 26, 2024

Revised August 7, 2024

Accepted August 19, 2024

Available online September 1, 2024

Keywords:

Visually Impaired

Assistive Technologies

Deep learning

Transformers

CNN

Video Description

Jetson Nano

ABSTRACT

Visually impaired individuals often face significant challenges in navigating their environments due to limited access to visual information. To address this issue, a portable, cost-effective assistive tool is proposed to operate on a low-power embedded system such as the Jetson Nano. The novelty of this research lies in developing an efficient, lightweight video captioning model within constrained resources to ensure its compatibility with embedded platforms. This research aims to enhance the autonomy and accessibility of visually impaired people by providing audio descriptions of their surroundings through the processing of live-streaming videos. The proposed system utilizes two distinct lightweight deep learning modules: an object detection module based on the state-of-the-art YOLOv7 model, and a video captioning module that utilizes both the Video Swin Transformer and 2D-CNN for feature extraction, along with the Transformer network for caption generation. The goal of the object detection module is for providing real-time multiple object identification in the surrounding environment of the blind while the video captioning module is to provide detailed descriptions of the entire visual scenes and activities including objects, actions, and relationships between them. The user interacts via a headphone with the proposed system using a specific audio command to trigger the corresponding module even object detection or video captioning and receiving an audio description output for the visual contents. The system demonstrates satisfactory results, achieving inference speeds between 0.11 to 1.1 seconds for object detection and 0.91 to 1.85 seconds for video captioning, evaluated through both quantitative metrics and subjective assessments.

1. Introduction

The major influence of visual information on human thinking and decision-making has dramatically increased the importance of visual data in our daily lives. Vision impairment, also known as visual impairment, refers to a reduction in the ability to see, which leads to difficulties that cannot be corrected by conventional means like glasses or contact lenses [1]. Vision impairment encompasses various degrees of visual loss. While some

individuals may suffer from a total loss of sight, others may retain the ability to perceive light, discern shapes, or have no visual perception at all. Visually impaired individuals encounter various challenges in their everyday activities depending on their degree of visual impairment. According to the World Health Organization (WHO) in 2017, approximately 253 million individuals were living with visual impairment, with 36 million among them classified as completely blind [2]. This emphasizes the critical necessity to enhance the quality of life

* Corresponding author.

E-mail address: ajalal289@gmail.com

DOI: [10.24237/djes.2024.17310](https://doi.org/10.24237/djes.2024.17310)

This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).



for people with visual impairments. Assistive technology plays a vital role in achieving this goal.

Various assistive technologies have been previously introduced to aid visually impaired individuals in navigation. Traditional assistive tools like white canes allow users to feel the way around and specially trained dogs help them reach some places. Although these are great inventions for the visually impaired, they have limitations. Canes require constant physical contact with things all the time, and dogs require extensive training and care. Undoubtedly, technological advances such as GPS and 3D audio systems have improved the lives of visually impaired individuals around the world. However, these technologies have limited functionality, focusing on basic tasks such as measuring distance and providing limited awareness of the surrounding environment [3, 4]. These limitations highlight the need for more comprehensive assistive technologies that can provide users with reasonable assistance and a deeper understanding of the surroundings.

With significant advancements in deep learning algorithms, particularly in computer vision, techniques such as object detection [5], image captioning [6], and video captioning [7] have become essential tools for enhancing accessibility for people with visual impairments after converting visual information to speech. Object detection can help visually impaired people in navigating their surroundings safely and confidently by providing real-time identification of objects and potential hazards. By describing everyday objects such as chairs, tables, food items, and other personal belongings, object detection allows visually impaired people to move through their environment with more independence. The most popular object detection algorithms are mainly based on the CNN architecture [8, 9] including R-CNN (Region-Based Convolutional Neural Networks) [10], SSD (Single Shot MultiBox Detector) [11], and YOLO (You Only Look Once) [12]. The R-CNN and its extensions (Fast R-CNN and Faster R-CNN) are a two-stage detector, that works by starting with region proposal generation followed by the classification of these regions to identify

objects. Faster R-CNN improved the process by using a Region Proposal Network (RPN) to generate proposals with less computational time. This leads to increased speed and accuracy in detecting objects. While SSD and YOLO are a single-stage detector, they integrate region proposal and object classification into a single process. This integration allows them to achieve high detection speeds with lower computational overhead. This single-shot approach allows YOLO to achieve real-time object detection speeds while maintaining reasonable accuracy. SSD is known for its balance between accuracy and speed. However, the real-time performance of YOLO has motivated researchers to refine and release several versions of the model. For example, YOLOv5 is known for its speed and accuracy, and it has been successfully used in various computer vision applications such as autonomous vehicles [13], video surveillance [14], and drone navigation [15]. Building on this progress, YOLOv7 [16] was released, offering significant improvements in both speed and accuracy. It also excels at detecting multiple objects within a single image or video frame, making it a valuable choice for complex object detection tasks [17, 18]. Furthermore, the study in [19] has shown that YOLOv7 outperforms Faster R-CNN in terms of accuracy, in addition to offering real-time performance.

On the other hand, image/video captioning provides another way for helping visually impaired people by creating detailed descriptions of entire scenes and activities including objects, actions, and relationships between them. Video captioning, in particular, offers distinct advantages over image captioning due to its ability to provide a more dynamic and comprehensive description of visual content. Unlike static images, videos capture temporal information, allowing for richer contextual understanding and more detailed descriptions of dynamic scenes and actions [7]. The early works of video captioning are based on a predefined template in sentence generation. Based on visual concepts, template-based methods identify triplets of subject, verb, and object in videos. Then, a predefined template is filled with the recognized triplets to generate the output caption. However, the details of generated

captions in these methods are restricted by the structures of predefined language templates, which limits the variety and creativity of the generated captions [20]. To overcome these limitations, the application of deep learning techniques has emerged as a promising approach in video captioning and other related tasks. Numerous existing works [21-25] are based on the well-known encoder-decoder framework. Typically, a video encoder translates video frames or clips into feature vectors using 2D/3D CNN, while a decoder converts these vectors into relevant sentences based on Recurrent Neural Network (RNN) [26]. Though 3D-CNNs are efficient at capturing motion features in videos, they tend to be time-consuming and often incompatible with real-time processing. Additionally, RNNs, although traditionally used for sequence modeling tasks such as video captioning, are prone to issues like vanishing gradients and have difficulty managing long-range dependencies [27].

However, most existing work in this area focuses on software solutions without considering the challenges of implementing such algorithms on hardware with limited computational resources. Current research does not adequately address the application of video captioning techniques on hardware platforms such as the Jetson Nano, which offers a cost-effective, portable, and low-power solution. To overcome these limitations, we have developed a lightweight model specifically designed for constrained environments using the Jetson Nano. In this work, we incorporate more advanced deep neural architectures, including the Video Swin Transformer [28] with the convolutional EfficientNet [29] for video feature extraction and the Transformer network [30] for description generation. The Video Swin Transformer offers a unique approach to processing video data by using a shifted window mechanism, enabling efficient computation and improved scalability. This model has achieved state-of-the-art performance in video recognition tasks across multiple datasets [27, 28, 31]. The EfficientNet architecture has several advantages over other CNNs including high accuracy, compound scaling, and reduced

model parameters [30, 32]. These advantages make it a valuable choice for various computer vision tasks, especially with resource limitations. In Natural Language Processing (NLP), the Transformer architecture has outperformed other models with its self-attention mechanism, providing superior parallelization and robustness in managing long sequences. Both the Video Swin Transformer and the Transformer network consistently achieve state-of-the-art results in terms of speed and accuracy for video recognition and NLP tasks [27]. Hence, these features motivated us to utilize these advanced architectures instead of 3D-CNNs and RNNs.

Visually impaired individuals face significant challenges in independently navigating their environments due to limited access to visual information. Existing assistive technologies do not effectively utilize video captioning on resource-constrained hardware like the Jetson Nano, limiting their portability and accessibility. Current solutions often lack the capability to provide fast, detailed audio feedback about visual scenes, including objects, actions, and their relationships. There is a need for a cost-effective and portable solution that can process live video streams to deliver descriptive audio feedback, enhancing autonomy for visually impaired users. In this paper, we employ YOLOv7 object detection and a new video captioning method to create an assistive tool designed to support visually impaired people. By combining these advanced technologies, our system can provide audio descriptions of objects, scenes, and activities, enabling a more comprehensive understanding of their surroundings. The main contributions of this paper are listed as follows:

- Development of a scalable, efficient, and cost-effective framework for video processing aimed at assisting visually impaired individuals in enhancing their daily lives.
- Combining of video captioning and object detection techniques utilizing state-of-the-art deep learning models, including YOLOv7, EfficientNet, Video

Swin Transformer, and Transformer architecture.

- Extensive experimentation and comparison of different methods and models to evaluate speed, accuracy, and computational requirements.
- Creation of a simple prototype of the proposed system with a detailed analysis of the system architecture and hardware components.

The rest of this paper is organized as follows: Section 2 reviews related work in video captioning and object detection, highlighting existing solutions and their limitations. In section 3, a detailed description of the proposed framework is provided. Section 4 presents the experimental setup and results, including performance metrics and comparisons with different methods and models. The potential limitation of this work is summarized in section 5. Finally, section 6 concludes the paper with suggestions for future research.

2. Related work and limitations

This section examines recent literature on computer vision-based solutions for assisting the visually impaired. V. Kumar et al. [1] presented a method for assisting blinds based on an image captioning technique. The integration of ResNet50-LSTM networks is used in this method as an encoder-decoder framework. Similarly, the study in [3] used another image captioning technique for describing visual content based on the VGG16-LSTM pipeline. However, both methods lack an attention layer in their deep architectures, which is important when processing sequential data such as video and text. To address this issue, M. Sarkar et al. [33] adopted a deep learning-based image captioning method that incorporates an attention mechanism. The deep model is based on the pre-trained Inception-ResNet network for feature extraction, followed by a Gated Recurrent Unit (GRU) network for caption generation. In the research [34], the authors presented a method for assisting blinds based on describing a single video frame through an image captioning technique. The model processes one frame for

every 50 frames to reduce complexity. Additionally, the authors incorporated a method to measure the distance of detected objects from the camera using YOLOv5 and a triangular similarity approach. However, the captioning technique relied on a conventional architecture, using VGG16 [35] for feature extraction and LSTM [36] for word generation. Furthermore, a better performance of this model could have been achieved by incorporating an attention mechanism in the deep neural architecture. A. Bodi et al. [37] developed a video description system to aid blind and low-vision individuals. The system combines multiple pre-trained models, including YOLOv3 for object detection and a pre-trained image captioning model. However, the system is primarily designed to process video frames from a pre-recorded media file instead of directly from a camera, which could limit its real-time applications. The study in [38], proposed a deep learning model to assist the blind in recognizing environmental information using video captioning techniques. The system is built with an encoder-decoder framework using the VGG16 network for encoding and RNN-LSTM combination for caption decoding. The MSVD [39] dataset is customized into five categories to train the proposed model. However, an attention mechanism is missing in the adopted system in addition to the use of traditional models.

Notably, the studies referenced above share a common limitation: none of them employ a specific hardware implementation for the proposed system, limiting their practical use and evaluation in real-world scenarios. On the other hand, P. Shameem et al. [40] implemented a guidance system for blind individuals via a smartphone application that uses an image captioning technique based on the Inceptionv3-LSTM framework. The user can capture images with their mobile phone, which are then sent to a Python server connected to the user's phone to run the deep learning model. The server processes the images and sends the corresponding audio descriptions back to the user's mobile phone. The work in [4] adopted a similar smartphone-server architecture to perform object detection on streaming video captured from the user's phone. However, this

approach requires a stable internet connection to send images to the server for processing, which can cause delays and interruptions in areas with poor connectivity. It also raises privacy and security concerns due to the transmission of potentially sensitive user data to an external server. A. Papanai and H. Kaushik [41] developed wearable IoT devices that transform information extracted from a camera mounted on the user into audio signals to aid the mobility of the blind based on object detection and lane detection techniques. The system uses the pre-trained Yolov5 model for object detection, while lane detection is primarily based on the Canny edge detection method. The implementation and evaluation of the system were carried out on a Jetson Nano board. The system introduced by K. Safiyaet et al. [42] aids the user in navigating independently through the use of image captioning techniques based on the VGG16-LSTM pipeline. The system deployed on a Raspberry Pi board using Keras and TensorFlow frameworks. However, the hardware limitations of the Raspberry Pi make it challenging to deploy larger deep learning models, which affects both real-time usability and system accuracy. B. Arystanbekov et al. [2] developed an assistive tool for visually impaired individuals using a pre-trained image captioning model in the Kazakh language. According to the evaluation results, the system achieved real-time response for captioning the captured image via a USB camera mounted on the user's head, which was connected to a Jetson Xavier NX 8GB board.

Upon reviewing the relevant works proposed by various studies, it's clear that while several investigations offer different solutions to specific challenges, some limitations still exist, leaving substantial gaps yet to be explored as follows:

- 1- Lack of temporal feature processing: previous works often focus on image captioning techniques applied to individual frames, neglecting the analysis of temporal or motion features across a sequence of frames. This limits the ability to capture dynamic changes and context within video sequences, which is crucial for accurate video

description in hardware implementations.

- 2- Conventional description generation methods: many studies utilize traditional architectures such as VGG16-LSTM for description generation. These methods do not use more advanced and effective architectures like Transformers, which can provide improved performance in capturing complex patterns and dependencies in video data.
- 3- Insufficient experimentation and evaluation: existing approaches frequently lack comprehensive experimentation and detailed evaluation of their performance, especially for the computational cost. This inadequacy affects the reliability and generalization of their results, leaving a need for more thorough analysis and validation.

Thus, this study aims to address these limitations by adopting a more robust approach with hardware implementation that combines advanced video description and object detection techniques.

3. Proposed methodology

The proposed system is an assistive tool designed to process audio commands and video frames, providing users with audio-based feedback. Figure 1 illustrates the system architecture and the interconnection between the hardware components. The system consists of two deep learning modules which are video captioning and object detection. These modules are accessed by the audio commands of the user. The hardware component of the proposed system consists of a Jetson Nano board (equipped with 128 NVIDIA CUDA cores and 4 GB of memory), a USB camera, USB Headphones (earphones/microphone), a USB Wi-Fi dongle, and a power source.

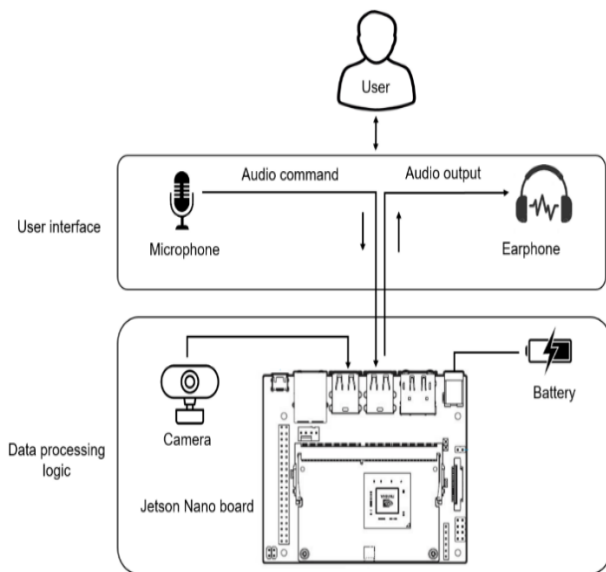


Figure 1. System architecture overview

The system begins by booting up the Jetson Nano, initializing the camera, and loading all necessary models into memory. The system prompts the user with an audio message saying, "Please choose between 'describe' or 'detect'," to select either the video captioning or object detection mode. The audio command of the user is translated into text data through speech

recognition software. In case of an invalid command, the system prompts the user to re-enter the command. Upon receiving a valid command (e.g. describe or detect) the corresponding module is activated (see Figure 2). The camera captures video frames, which are then served as input data for the model's processing. Subsequently, the resulting output is converted from text to audio format using the gTTS (google text to speech) package, and transmitted to the user through the earphones as illustrated in Fig. 2. Notably, gTTS supports a wide range of languages which allows users to receive descriptions in their native languages. The Jetson Nano was selected as the embedded platform for this project due to its balance of performance, power efficiency, and affordability. The key advantages involve its capability to run deep learning models with 128 NVIDIA CUDA cores and 4 GB of memory, making it suitable for real-time processing in a low-power setting. Its compact size and low cost also make it an ideal choice for deploying assistive technologies in resource-constrained environments. Notably, the code and trained models are available at <https://github.com/vision-research/vid-desc-jet>.

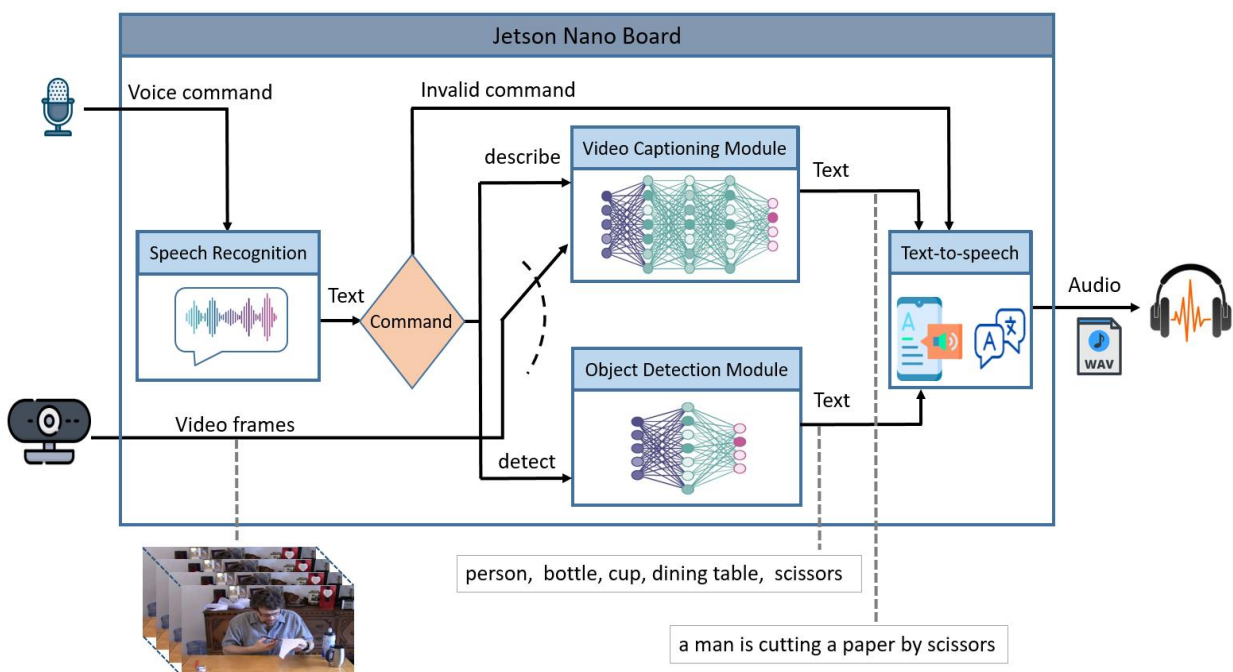


Figure 2. Framework for the hardware and software architecture of the assistive system

3.1 Video captioning module

The proposed video captioning module is mainly based on the model architecture from our previous work [27], which utilizes the encoder-decoder framework, with model compression and acceleration. Figure 3 shows the structural design of the proposed video captioning method. Our captioning module involves two distinct components including a video encoder and a caption decoder. The architecture of the proposed video captioning model integrates the Video Swin Transformer, 2D-CNN, and Transformer network to generate accurate and detailed captions. Initially, the 2D-CNN (EfficientNet) extracts spatial features from individual video frames, representing each frame as a high-level feature vector that captures detailed visual information. Subsequently, the Video Swin Transformer processes sequences of frames to capture temporal dynamics and motion features, analyzing the context and relationships between frames to provide a comprehensive understanding of the video. Finally, the Transformer network combines the spatial and temporal features from the previous components to generate the final captions, creating coherent and contextually relevant descriptions of the visual content.

3.1.1 Video encoder

To generate accurate and informative video descriptions, it is necessary to extract visual features that capture a high level of video understanding, including spatial features (i.e. appearance and static scene) along with temporal features (i.e. motion and dynamic events). Hence, the encoder in our model comprises three components: a 2D CNN for spatial feature extraction, a Video Swin Transformer for temporal feature extraction, and a neural feature fusion network to efficiently unify the extracted features.

For the 2D CNN, the EfficientNet architecture is used, producing a feature vector for each processed frame. While the Video Swin Transformer produces a single feature vector for the entire sequence of video frames. To unify the extracted feature vectors, we construct a trainable neural network that consists of two

fully connected networks with batch normalization, an activation function, and a dropout function in between. The main objective of this fusion network is to significantly reduce the computational time in addition to generating a more compact and informative video representation.

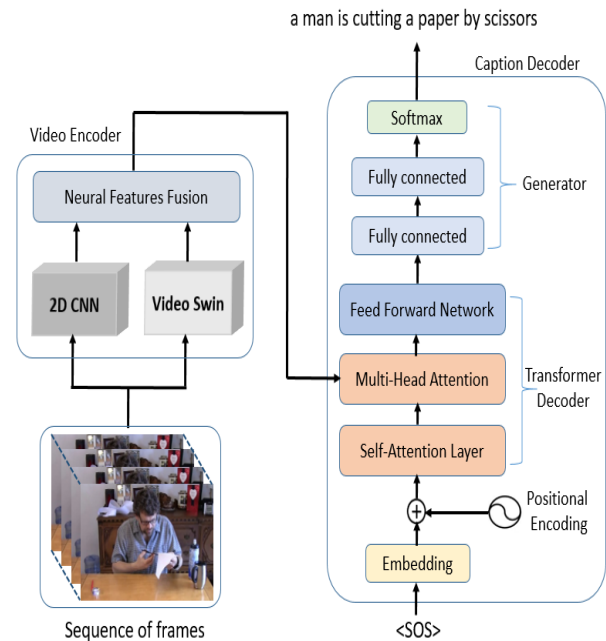


Figure 3. Structure of the proposed video captioning model

3.1.2 Caption decoder

The caption decoder comprises two main components: the decoder and the generator. The decoder, employing the Transformer decoder architecture, takes the video representation features and word embedding vectors as input, generating caption features. As in Transformer architectures, positional encoding is used to add position information to the embedding vectors. Meanwhile, the generator predicts the subsequent word in the caption by computing the probability of each word across the constructed vocabulary using softmax activation, based on the decoder output. The word with the highest probability is chosen as the predicted word. The caption decoder works progressively, starting with an initial token <SOS>. At each iteration, it takes in the video representation features (from the video encoder) and the partially completed caption that has already been predicted. It then predicts the next

word with the highest probability by passing through the decoder component and then through the generator component. This process continues until a distinctive token <EOS> (i.e. ending token) is encountered.

3.2 Object detection module

For the object detection task in the proposed system, we employ the pre-trained YOLOv7 as the primary framework based on transfer learning. YOLOv7 is well-known for its efficiency and accuracy in real-time object detection tasks, making it well-suited for real-time applications on embedded systems. This model identifies objects within the frame and categorizes them with labels such as "person," "car," "chair," etc. The text output of YOLO is modified by cleaning and removing punctuation, which ensures that the detected object names are formatted in a manner suitable for accurate and clear audio conversion. Finally, the processed labels are translated into audio descriptions using a text-to-speech conversion.

4 Experimental analyses

4.1 Video captioning

In this subsection, we evaluate the performance of the video captioning module adopted in the proposed system using the Microsoft Research Video Description Corpus (MSVD) [39] dataset with the most frequently used metrics. Furthermore, we provide a comparative analysis between our proposed method and other existing video captioning approaches.

4.1.1 Dataset and evaluation metrics

The MSVD dataset consists of a 1970 video clip with a duration mostly ranging from 10 to 25 seconds with a single activity. Each video in MSVD is associated with about 40 captioning sentences that describe the salient objects, actions, and relations in the video. For benchmarking, the dataset is split into 1200 videos for training, 100 videos for validation, and 670 videos for testing. For evaluating the model's performance, we investigate the most commonly used assessment metrics in the video captioning approach which estimates the

similarity between the predicted and the human-annotated captions. These metrics are Bilingual Evaluation Understudy (BLUE) [43], Metric for Evaluation of Translation with Explicit Ordering (METEOR) [44], Recall Oriented-Understudy for Gisting Evaluation (ROUGE) [45], and Consensus-based Image Description Evaluation (CIDEr) [46].

4.1.2 Implementation details

After performing hyperparameter optimization on the MSVD validation set, we adopted the following configuration. For keyframes extraction, we set $N = 8$ keyframes extracted from each video in the dataset. For the video encoder, the pre-trained EfficientNet-B3 is used as the 2D CNN, producing a feature vector with dimensions of $N \times 1280$. While, the pre-trained Swin-S (Swin-Small) model serves as the Video Swin Transformer, producing a feature vector with a dimension of $768 \times 8 \times 7 \times 7$. These feature vectors are obtained by removing the final (i.e. classification) layer from both EfficientNet-B3 and Swin-S networks. Regarding the concatenation purposes, average pooling is applied to the feature vectors from EfficientNet-B3 across the N frames. For the Video Swin model, 3D average pooling is applied across the last three dimensions, which represent the 3D-shifted window size. This process yields a single concatenated feature vector with a final dimension of 2048. This vector is then passed to the feature fusion network, which reduces it to a more compact vector with a dimension of 1024. The hidden dimension of the fusion network is set to 1024, with a dropout ratio of 0.2, and ReLU is used as the activation function.

On the caption decoder side, a trainable embedding layer with a dimension of 1024 is for word embedding. The Transformer decoder contains a single decoder layer with one attention head in the self-attention layer and one attention head in the multi-head attention layer. The feed-forward network comprises two fully connected layers with hidden dimensions of 1024, where ReLU is applied after the first layer. For the generator, the hidden dimensions of the fully connected layers correspond to the size of the constructed vocabulary from the

MSVD dataset, which is 2,831. These fully connected layers are separated by a ReLU activation followed by a dropout with a probability of 0.2. The final output layer is a softmax, which translates the feature vector into a probability distribution across the vocabulary. To obtain the most likely words for the caption, a beam search algorithm with a beam size of 3 is applied.

For sentence preprocessing, all punctuation marks are eliminated from each sentence in the MSVD captions. Subsequently, each sentence is

segmented by blank spaces, and all words are converted to lowercase. All the sentences that exceed 15 words are truncated. Regarding training details, we used the Adam optimizer with a learning rate of $2e-5$, a batch size of 64, and 50 training epochs. Figure 4 shows the training loss across training epochs for the video captioning model using the MSVD dataset. This model is trained on a single NVIDIA RTX 3060 mobile GPU for later deployment on a Jetson Nano board.

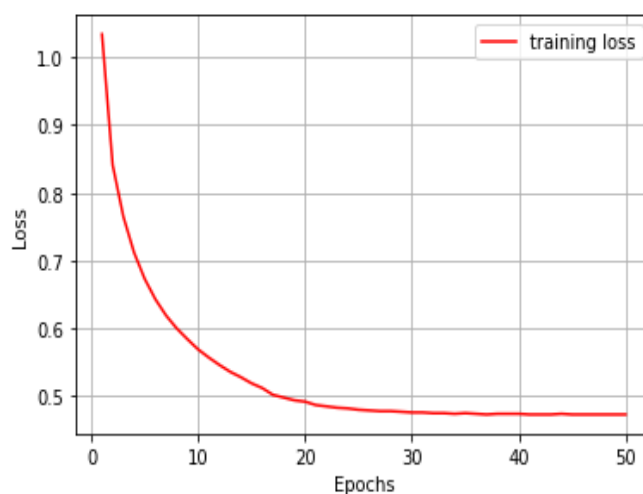


Figure 4. Evolution of training loss of video captioning model on the MSVD dataset

4.1.3 Results and comparisons

To assess the effectiveness of our video captioning module, a detailed comparison is conducted with a set of existing videos captioning techniques, including RecNet [20], GRU-EVE [21], SibNet [22], and SAAT [23], GMNet [24], VADD [25], ADL [47], MM-AT [48], and the approach in [49], as presented in Table 1.

To identify the most effective architecture for our video captioning system, we conducted an extensive experiment evaluating various models, including Swin-T, Swin-S, MobileNetV2, VGG16, InceptionV3,

ResNet50, and EfficientNetB3. This experiment considered both computational efficiency and accuracy in caption generation as presented in Table 2. The inference speed in Table 2 involves the encoding-decoding process for 8 frames. This experiment provides a comprehensive analysis of the proposed method to select the best configurations that offer a balance between efficiency and accuracy. For qualitative evaluation, Figure 5 visualizes examples of descriptions generated by our model and other investigated video encoding models. This figure includes both accurate and imperfect predictions for the MSVD dataset.

Table 1: Comparative analysis of video captioning methods on MSVD dataset

Methods	Year	Metrics			
		BLUE4	ROUGE-L	METEOR	CIDEr
RecNet [20]	2018	52.3	69.8	34.1	80.3
GRU-EVE [21]	2019	47.9	71.5	35.0	78.1
SibNet (Conv+S) [22]	2020	54.2	71.7	34.8	88.2
SAAT [23]	2020	46.5	69.4	33.5	81.0
GMNet [24]	2021	52.1	70.7	33.5	83.1
VADD [25]	2022	51.5	72.1	34.8	91.5
ADL (Inception-V4) [47]	2022	54.1	70.4	35.7	81.6
MM-AT (R+O+S) [48]	2023	53.6	73.5	35.0	87.4
Research in [49]	2024	47.1	62.0	30.4	59.9
Proposed (EffecientNetB3 + Swin-S)	2024	55.9	74.0	36.5	94.9

Table 2: Efficiency comparison of various video encoder architectures in the proposed method

Video Encoder	Metrics					
	Accuracy				Inference Speed	
	BLUE4	ROUGE_L	METEOR	CIDEr	On RTX 3060 GPU	On Jetson Nano
Only Swin-T	51.1	70.1	33.8	81.9	43 ms	0.91 s
Only Swin-S	54.2	72.6	35.0	90.5	54 ms	1.37 s
MobileNetV2 + Swin-T	51.7	71.4	34.2	82.8	52 ms	1.06 s
MobileNetV2 + Swin-S	54.5	72.7	35.5	90.8	65 ms	1.52 s
VGG16 + Swin-S	54.8	72.9	35.3	89.1	93 ms	2.80 s
InceptionV3 + Swin-S	55.2	73.8	35.6	90.9	72 ms	1.83 s
ResNet50 + Swin-S	53.8	73.1	35.2	91.4	82 ms	1.92 s
EffecientNetB3 + Swin-T	54.4	72.3	35.1	88.0	63 ms	1.40 s
EffecientNetB3 + Swin-S	55.9	74.0	36.5	94.9	78 ms	1.85 s









Examples of accurate descriptions	Examples of imperfect descriptions
 <p>GT: a man is mixing something with a blender MobileNet+Swin-T: a man is cooking Only Swin-S: a man is mixing something VGG16+Swin-S: a man cooking his kichen Proposed: a man is mixing something</p>	 <p>GT: a car explodes behind some men MobileNet+Swin-T: a man is singing Only Swin-S: a man is dancing VGG16+Swin-S: a man is <UNK> a car Proposed: a group of people are playing a fire</p>
 <p>GT: a cat is playing in a box MobileNet+Swin-T: a cat is playing Only Swin-S: a cat is playing VGG16+Swin-S: a cat is playing with a ball Proposed: a cat is playing with a box</p>	 <p>GT: a man is hitting a block of ice with a sword MobileNet+Swin-T: a man is cutting a sword Only Swin-S: a man is cutting a piece of wood VGG16+Swin-S: a man is cutting a sword Proposed: a man is cutting wood with a sword</p>
 <p>GT: a man is riding a motorcycle MobileNet+Swin-T: a man is riding a bike Only Swin-S: a man is riding a motorcycle VGG16+Swin-S: a man is riding a motorcycle Proposed: a man is riding a motorcycle</p>	 <p>GT: a cheetah is running very fast MobileNet+Swin-T: a dog is running Only Swin-S: a dog is running VGG16+Swin-S: a dog is running Proposed: a dog is running</p>
 <p>GT: a panda is climbing MobileNet+Swin-T: a panda bear is walking Only Swin-S: a panda is climbing VGG16+Swin-S: a panda is eating Proposed: a panda is climbing</p>	 <p>GT: two men are moving tires MobileNet+Swin-T: a man is riding a skateboard Only Swin-S: a man is running VGG16+Swin-S: a man is skating Proposed: a man is doing some tricks</p>

Figure 5. Qualitative comparison of the proposed video description on the MSVD benchmark across different model

Based on the results in Table 1, our approach demonstrates superior performance across all four metrics on the MSVD benchmark. Referring to Table 2, our analysis indicates that utilizing only the Swin-T model for video encoding yields the highest inference speed while maintaining acceptable accuracy compared to methods outlined in Table 1. Conversely, the EfficientNetB3-Swin-S backbone demonstrates superior performance across all four accuracy metrics while still achieving acceptable inference speed. Hence, we adopt the EfficientNetB3-Swin-S combination as our video encoder. Notably, the time calculated in Table 2 represents the time consumed for the encoding-decoding processes, excluding the preprocessing time.

Figure 6 illustrates the performance of our captioning approach across 50 epochs on the

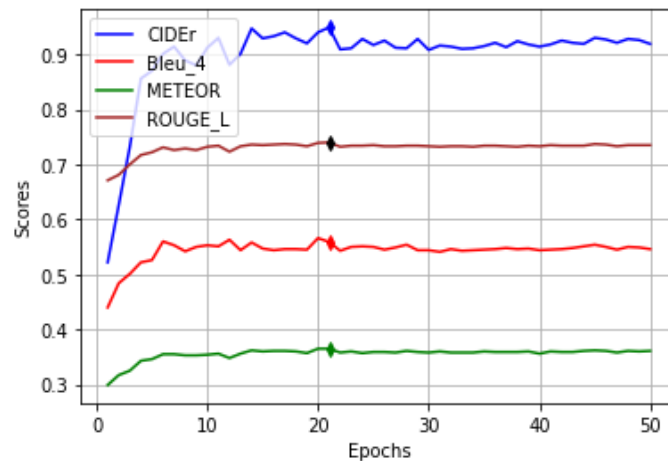


Figure 6. Model performance on MSVD testing set over training epochs

According to Figure 6, our model reaches its peak performance at epoch 21 on average, particularly for the CIDEr metric. Notably, the CIDEr metric is specifically developed for evaluating image captioning tasks. Moreover, Figure 6 illustrates the fast convergence of the proposed method, requiring fewer training epochs compared to LSTM-based models. One of the key reasons for this behavior is the utilization of the Transformer architecture in the caption decoding part instead of the LSTM network, which typically requires longer training times to achieve similar performance.

testing set of the MSVD dataset. Referring to Figure 5, the proposed model utilizing the EfficientNetB3-Swin-S combination achieves the best performance in video description accuracy compared to other models. Nevertheless, Figure 5 also shows that the proposed model encounters limitations with some scenes. This can be attributed to the limited training data for such scenes within the MSVD benchmark. For instance, the model struggles with accurately describing cheetahs. Since the MSVD training set comprises only two videos featuring cheetahs, our model misclassifies them as dogs, which are frequently present in the dataset. This example highlights the importance of addressing data scarcity for specific scenarios when developing video description models.

4.2 Object detection

In this section, the performance of the object detection module in the proposed system is analyzed and evaluated. Our object detection module is mainly based on the pre-trained Yolov7 model, which is trained on the well-known MS COCO dataset. To ensure efficiency, the system processes video frames at a rate of 1fps and applies the YOLOv7 model for object detection on the sampled frames. YOLOv7 offers different versions, ranging from the smallest YOLOv7-tiny to the largest YOLOv7-E6E (see Table 3), with a trade-off between speed and accuracy [50]. Table 3 provides a comparison of these models across different

metrics, including computational efficiency on both the Jetson Nano device and a mobile NVIDIA RTX 3060 GPU, Average Precision (AP) test, model size, and total number of parameters. Furthermore, Figure 7 depicts the qualitative results from real-world scenarios using YOLOv7-tiny and YOLOv7-W6, which are specifically designed for edge GPUs. The figure demonstrates the capability of both models for real-time detection of multiple

objects with high accuracy. However, as shown in Figure 7, YOLOv7-W6 offers superior accuracy compared to YOLOv7-tiny. For instance, the dining table is missing in the YOLOv7-tiny detection, and the bottle is misclassified as two separate objects (bottle and cup). Therefore, we adopted YOLOv7-W6 in our system due to its favorable trade-off between speed and accuracy.

Table 3: Comparative analysis of YOLOv7 versions in terms of accuracy, model size, and inference speed

Model	AP test (on MS COCO)	Size	Number of parameters	Inference speed on RTX 3060 GPU		Inference speed on Jetson Nano	
YOLO-tiny	38.7%	12.3 MB	6.2M	4 ms	250 fps	117 ms	8.19 fps
YOLOv7	51.4%	73.8 MB	36.9M	12 ms	83 fps	547 ms	1.81 fps
YOLO-X	53.1%	139.7 MB	71.3M	19 ms	52 fps	912 ms	1.08 fps
YOLO-W6	54.9%	137.9 MB	70.4M	11 ms	90 fps	514 ms	1.93 fps
YOLO-E6	56.0%	190.4 MB	97.2M	17 ms	58 fps	748 ms	1.32 fps
YOLO-D6	56.6%	261.9 MB	133.7M	21 ms	47 fps	977 ms	1.01 fps
YOLO-E6E	56.8%	297.2 MB	151.7M	27 ms	37 fps	1184 ms	0.84 fps



Figure 7. Object detection comparison: (a) YOLOv7-tiny, (b) YOLOv7-W

Table 4: Performance evaluation of YOLO-tiny and YOLO-W6 on objects in Figure 7.

Detected object	Accuracy using YOLO-tiny	Accuracy using YOLO-W6	Improvement (W6-tiny)
Green apple	0.75	0.87	12%
Orange	0.60	0.82	22%
Banana	0.81	0.93	12%
Yellow apple	0.57	0.88	31%
Cell phone	0.72	0.92	20%

Mouse	0.93	0.95	2%
Laptop	0.93	0.96	3%
Cup	0.92	0.96	4%
Dining table	Missing	0.70	-
Bottle	0.81(bottle) + 0.30 (cup)	0.94	13%
Left chair	0.60	0.92	32%
Right chair	0.74	0.84	10%

5. Limitations

Even if our research offers a new direction for scene description specifically developed for aiding blind and visually impaired individuals, there are limitations to address. These shortcomings present valuable opportunities for further exploration in this field. The key shortcomings of our proposed solution are presented below:

- 1- **Applicability Scope:** This approach is limited by the diversity of scenes found in the MSVD dataset and the recognized objects in the MS COCO dataset (i.e., only 80 objects). This narrows the range of scenes where users can receive assistance from this system.
- 2- **Limited Hardware Resources:** The Jetson Nano used in this research has limited resources, such as relatively small memory shared between CPU and GPU in addition to fewer CUDA cores. This restricts the proposed video captioning method to using smaller or medium-sized models. Employing larger and more complex models, like Video Swin-B and EfficientNet-B7, would likely result in out-of-memory errors.

6. Conclusion and future work

This research introduces a hardware implementation system for visually impaired assistance, utilizing state-of-the-art deep learning techniques. We further analyze relevant studies from the literature, highlighting their limitations and paving the way for advancements in this field. Our approach is mainly based on video captioning and object detection to provide meaningful audio

descriptions of the environment for enhancing the mobility and independence of users. By utilizing advanced architectures involving the Transformers network, EfficientNet, and YOLOv7, our system demonstrates a significant improvement in accuracy and efficiency. The combination of EfficientNetB3 with Swin-T and YOLOv7-W6 creates a favorable balance between computational time and accuracy. The implementation on a Jetson Nano board proves the feasibility of deploying a powerful deep learning-based system on resource-constrained hardware. Moreover, the proposed system is cost-effective and easy to set up in addition to requiring no special skills to operate. Our extensive experiments demonstrate the adaptability and effectiveness of the proposed system across various scenarios.

For future work, we plan to explore the use of the TensorRT engine to accelerate system performance. Additionally, employing more robust hardware like the Jetson Xavier NX 8GB (with 384 CUDA cores) would facilitate the use of larger deep models to improve accuracy and response times. Furthermore, adding other assistive technologies to the system, such as text recognition and facial recognition, could enhance its utility for visually impaired individuals.

Ethical considerations will be an integral part of these future developments. We aim to address privacy concerns by ensuring that all data processing occurs locally, thereby protecting user data. To reduce bias, we will incorporate more diverse datasets during model training, ensuring equitable performance across different user demographics. Increasing user trust in the system's accuracy will be a priority, achieved through extensive testing and incorporating user feedback to refine and improve the models continuously. Continued

research in this area could lead to more comprehensive solutions for providing greater accessibility and independence.

References

- [1] V. V. N. V. P. Kumar, V. P. Teja, A. R. Kumar, V. Harshavardhan and U. Sahith, "Image Summarizer for the Visually Impaired Using Deep Learning," 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), Puducherry, India, pp. 1-4, 2021.
- [2] B. Arystanbekov, A. Kuzdeuov, S. Nurgaliyev and H. A. Varol, "Image Captioning for the Visually Impaired and Blind: A Recipe for Low-Resource Languages," 2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Sydney, Australia, pp. 1-4, 2023
- [3] A. Chharia and R. Upadhyay, "Deep Recurrent Architecture based Scene Description Generator for Visually Impaired," 2020 12th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT), Brno, Czech Republic, pp. 136-141, 2020.
- [4] C. Chaitra, Chennamma, R. Vethanayagi, K. M. V. Manoj, B. S. Prashanth, T. Likewin, and D. S. L. Shiva, "Image/Video Summarization in Text/Speech," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, pp. 1-6, 2022.
- [5] D. N. Jyothi, G. H. Reddy, B. Prashanth and N. V. Vardhan, "Collaborative Training of Object Detection and Re-Identification in Multi-Object Tracking Using YOLOv8," 2024 International Conference on Computing and Data Science (ICCDs), Chennai, India, pp. 1-6, 2024.
- [6] J. Sudhakar, V. V. Iyer and S. T. Sharmila, "Image Caption Generation using Deep Neural Networks," 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 2022, pp. 1-3, 2022.
- [7] X. Hao, F. Zhou and X. Li, "Scene-Edge GRU for Video Caption," 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, pp. 1290-1295, 2020.
- [8] T. A. Tuib, B. H. Saoudi, Y. M. Hussein, T. H, Mandeel, F, T, Al-Dhief, "Convolutional neural network with binary moth flame optimization for emotion detection in electroencephalogram." *Int J Artif Intell* ISSN 2252.8938: 1173.
- [9] A. K. S. Alsajri, and A. V. Hacimahmud, "Review of deep learning: Convolutional Neural Network Algorithm." *Babylonian Journal of Machine Learning*, 19-25, 2023.
- [10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, pp. 580-587, 2014.
- [11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," In *European conference on computer vision*, pages 21–37. Springer, 2016.
- [12] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 779-788, 2016.
- [13] B. Xiao, J. Guo, and Z. He, "Real-Time Object Detection Algorithm of Autonomous Vehicles Based on Improved YOLOv5s," 2021 5th CAA International Conference on Vehicular Control and Intelligence (CVCI), Tianjin, China, pp. 1-6, 2021.
- [14] P. Zhang, W. Hou, D. Wu, B. Ge, L. Zhang, and H. Li, "Real-Time Detection of Small Targets for Video Surveillance Based on MS-YOLOv5," 2023 6th International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, pp. 690-694, 2023.
- [15] Y. Yang, "Drone-View Object Detection Based on the Improved YOLOv5," 2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, pp. 612-617, 2022.
- [16] C. Wang, A. Bochkovskiy, H. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7464-7475. 2023.
- [17] S. Chourasia, R. Bhojane and L. Heda, "Safety Helmet Detection: A Comparative Analysis Using YOLOv4, YOLOv5, and YOLOv7," 2023 International Conference for Advancement in Technology (ICONAT), Goa, India, pp. 1-8, 2023.

- [18] T. Reddy Konala, A. Nammi and D. Sree Tella, "Analysis of Live Video Object Detection using YOLOv5 and YOLOv7," 2023 4th International Conference for Emerging Technology (INCET), Belgaum, India, pp. 1-6, 2023.
- [19] I. Hilali, A. Alfazi, N. Arfaoui and R. Ejbali, "Tourist Mobility Patterns: Faster R-CNN Versus YOLOv7 for Places of Interest Detection," in *IEEE Access*, vol. 11, pp. 130144-130154, 2023.
- [20] B. Wang, L. Ma, W. Zhang, and W. Liu, "Reconstruction Network for Video Captioning," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 7622-7631, 2018.
- [21] N. Aafaq, N. Akhtar, W. Liu, S. Z. Gilani and A. Mian, "Spatio-Temporal Dynamics and Semantic Attribute Enriched Visual Encoding for Video Captioning," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 12479-12488, 2019.
- [22] S. Liu, Z. Ren and J. Yuan, "SibNet: Sibling Convolutional Encoder for Video Captioning," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, pp. 3259-3272, 1 Sept. 2021.
- [23] Q. Zheng, C. Wang, and D. Tao, "Syntax-Aware Action Targeting for Video Captioning," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, pp. 13093-13102, 2020.
- [24] X. Zhang, C. Liu and F. Chang, "Guidance Module Network for Video Captioning," 2021 40th Chinese Control Conference (CCC), Shanghai, China, pp. 7955-7959, 2021.
- [25] Z. Sun, S. Chen and L. Zhong, "Visual-Aware Attention Dual-Stream Decoder for Video Captioning," 2022 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, pp. 1-6, 2022.
- [26] N. Xu, A. Liu, Y. Wong, Y. Zhang, W. Nie, Y. Su, and M. Kankanahall, "Dual-Stream Recurrent Neural Network for Video Captioning," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2482-2493, Aug. 2019.
- [27] A. Yousif and M. Al-Jammas, "Real-time Arabic Video Captioning Using CNN and Transformer Networks Based on Parallel Implementation," *Diyala Journal of Engineering Sciences* vol. 17, No 1, March 2024.
- [28] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.
- [29] M. Tan and Q. Le, EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114, 2019.
- [30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advance Neural Inf. Process. Syst.* 30 (2017).
- [31] X. Chen, M. Zhao, F. Shi, M. Zhang, Y. He, and S. Chen, "Enhancing Ocean Scene Video Captioning with Multimodal Pre-Training and Video-Swin-Transformer," *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, Singapore, pp. 1-6, 2023.
- [32] S. Chaudhary, S. Sadbhawna, V. Jakhetiya, B. N. Subudhi, U. Baid and S. C. Guntuku, "Detecting Covid-19 and Community Acquired Pneumonia Using Chest CT Scan Images with Deep Learning," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, pp. 8583-8587, 2021.
- [33] M. Sarkar, S. Biswas and B. Ganguly, "A Hybrid Transfer Learning Architecture Based Image Captioning Model for Assisting Visually Impaired," 2023 IEEE 3rd Applied Signal Processing Conference (ASPCON), India, pp. 211-215, 2023.
- [34] A. S. Alva, R. Nayana, N. Raza, G. S. Sampatrao and K. B. S. Reddy, "Object Detection and Video Analyser for the Visually Impaired," 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, pp. 1405-1412, 2023.
- [35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [36] S. Hochreiter, J. Schmidhuber, long short-term memory, *Neural Compute.* 9 (1997) 1735–1780.

- [37] A. Bodi, P. Fazli, S. Ihorn, Y. Siu, A. Scott, L. Narins, Y. Kant, A. Das, and I. Yoon, "Automated Video Description for Blind and Low Vision Users," In Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems. 1–7.
- [38] Y. -H. Huang and Y. -Z. Hsieh, "The Assisted Environment Information for Blind based on Video Captioning Method," 2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan), Taoyuan, Taiwan, pp. 1-2, 2020.
- [39] D. Chen and W. Dolan, "Collecting highly parallel data for paraphrase evaluation". In ACL: Human Language Technologies- Volume 1. ACL, 190-200, 2011.
- [40] P. Muhammad Shameem, M. F. Imthiyaz, P. Abshar, K. Ijassubair, and A. K. Najeeb, "Real time visual interpretation for the blind," 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, pp. 1655-1660, 2021.
- [41] A. Papanai and H. Kaushik, "Hybrid Image Processing Device as Wearable Aide for Visually Impaired," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2022, pp. 733-738, 2022.
- [42] K. M. Safiya and R. Pandian, "Computer Vision and Voice Assisted Image Captioning Framework for Visually Impaired Individuals using Deep Learning Approach," 2023 4th IEEE Global Conference for Advancement in Technology (GCAT), Bangalore, India, pp. 1-7, 2023.
- [43] K. Papineni, S. Roukos, T. Ward and W. Zhu, "Bleu: a method for automatic evaluation of machine translation," In: Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics, 2002;311–318.
- [44] S. Banerjee, and A. Lavie, "Meteor: An automatic metric for MT evaluation with improved correlation with human judgments," In: Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, 2005;65–72.
- [45] C. Lin, "Rouge: A package for automatic evaluation of summaries," In: Proceedings of Workshop on Text Summarization Branches Out, Post2Conference Workshop of ACL 2004.
- [46] R. Vedantam, C. L. Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015;4566–4575.
- [47] W. Ji, R. Wang, Y. Tian, and X. Wang, "An attention based dual learning approach for video captioning," Applied Soft Computing, vol. 117, p. 108332, 2022.
- [48] H. Munusamy and C. Sekhar, "Multimodal attention-based transformer for video captioning," Applied Intelligence (2023): 23349-23368, 2023.
- [49] N. Alrebdi and A. Al-Shargabi, "Bilingual video captioning model for enhanced video retrieval," Journal of Big Data 11.17, 2024.
- [50] M. A. A. Albadr, M. Ayob, S. Tiun, F. T. AL-Dhief, A. Arram, and S. Khalaf, "Breast cancer diagnosis using the fast-learning network algorithm," Frontiers in Oncology, vol. 13, p. 1150840, 2023.