International Journal of Internet, Broadcasting and Communication Vol.16 No.4 62-72 (2024)
http://dx.doi.org/10.7236/IJIBC.2024.16.4.62

IJIBC 24-4-7

Real time character and speech commands recognition system

Dong-jin Kwon [1] , Sang-hoon Lee [2]

[1] *Associate Professor, Department of Computer Electronics Engineering, Seoil University, Korea*
[1] *djkwon77@seoil.ac.kr*

[2] *Master, Korea Institute of Science and Technology, Korea*
[2] *lsh950223@kist.re.kr*

Abstract

With the advancement of modern AI technology, the field of computer vision has made significant progress. This study introduces a parking management system that leverages Optical Character Recognition (OCR) and speech recognition technologies. When a vehicle enters the parking lot, the system recognizes the vehicle’ s license plate using OCR, while the administrator can issue simple voice commands to control the gate. OCR is a technology that digitizes characters by recognizing handwritten or image-based text through image scanning, enabling computers to process the text. The voice commands issued by the user are recognized using a machine learning model that analyzes spectrograms of voice signals. This allows the system to manage vehicle entry and exit records via voice commands, and automatically calculate paid services such as parking fees based on license plate recognition.

The system identifies the text areas from images using a bounding box, converting them into digital characters to distinguish license plates. Additionally, the microphone collects the user's voice data, converting it into a spectrogram, which is used as input for a machine learning model to process 2D voice signal data. Based on the model’ s inference, the system controls the gate, either opening or closing it, while recording the time in real-time.

This study introduces a parking management system that integrates OCR and a speech command recognition model. By training the model with multiple users' data, we aim to enhance its accuracy and offer a practical solution for parking management.

Keywords: *Computer Vision, Character Recognition, Speech Recognition, Machine Learning system*

1. INTRODUCTION

Recently, with the advancement of machine learning technology, its utilization and importance are on the rise in modern times. As a result, it is being applied across various fields, particularly in computer vision, where tasks such as object detection, tracking, image restoration, and image compression can now yield better results through automated detection systems rather than manual algorithm-based processes.

Manuscript Received: September. 7, 2024 / Revised: September. 12, 2024 / Accepted: September. 18, 2024
Corresponding Author: lsh950223@kist.re.kr
Tel: :+82-2-490-7349, Fax: +82-2-490-7802
Master, Korea Institute of Science and Technology, Korea

This study presents a parking management system within a limited parking space by building a user interface with the Qt framework and integrating cameras and microphones to combine them with a machine learning system. Optical Character Recognition (OCR), a technology that recognizes characters in photos and videos, has made significant advancements over time. In the past, OCR operated through multiple specialized modules, such as text line detection and character segmentation, with humans manually defining the feature points that served as criteria for character recognition. However, modern AI technology, as exemplified by companies like Google, has overcome these past limitations by enabling the analysis of characters in both photos and videos. The recognition rate and accuracy have dramatically improved compared to earlier systems, and ongoing development of algorithms continues to address limitations like recognition errors.

Speech recognition technology allows computers to understand and process human speech by converting voice data collected via a microphone into digital signal data and interpreting it through an algorithm. Modern AI technology utilizes statistical machine learning and artificial neural networks to train models on large-scale speech datasets, improving the accuracy of speech recognition systems. This technology is now applied in smartphones, robots, and audio equipment systems.

This study demonstrates a solution that optimizes the real time performance of camera vision technology u sing OCR and a voice recognition system, while enhancing system adaptability by utilizing diverse datasets. The proposed solution has practical applications in management systems such as parking management and s mart homes, contributing to user convenience within limited spaces. We aim to explain the process of buildin g such a system and provide examples of potential applications.

2. BACKGROUND KNOWLEDGE

2.1 QT framework

The Qt framework is a library and development tool for building cross-platform applications and user interfaces (UIs). Qt simplifies software development at the design, development, and deployment stages, allowing developers to create applications that run on various platforms such as Windows, macOS, Linux, Android, and iOS. The GUI library provided by Qt makes it easy to build various UI elements such as buttons, text boxes, charts, and dialogs. It supports event-driven programming, which facilitates communication between objects using the signal and slot mechanism, enhancing modularity and flexibility in applications. Additionally, it can be integrated with powerful libraries such as OpenCV during development and provides a variety of tools and utilities to support code editing, debugging, and profiling.

2.2 OCR

OCR (Optical Character Recognition) is a technology that digitizes text written by hand or contained in images, allowing computers to recognize the text after scanning the image. By digitizing text, OCR enables the permanent storage of printed materials and facilitates content searching, making it a widely used technology that enhances efficiency in daily life and business operations. OCR originated in 1928 when G. Taushek patented a "pattern matching" technique in Germany. Pattern matching compares input characters to pre-stored standard pattern characters and selects the one with the highest similarity. In the past, OCR operated through specialized modules such as text line detection and character segmentation, where humans had to manually define the distinguishing features of characters, which resulted in lower recognition rates for handwritten or cursive text. With the rapid advancement of modern AI technology, machine learning has also made significant progress, leading to a dramatic improvement in OCR’ s character recognition rate and accuracy. Today, instead of manually registering character features, computers learn from large datasets and

build their own rules to recognize text in photos and videos. While OCR still experiences recognition errors, it demonstrates a very high level of accuracy, and ongoing algorithm development continues to enhance its performance. As a result, OCR is being applied in various fields with growing success. [1]

2.3 Speech recognition

Speech recognition is a technology that enables computers to understand and process human speech. It is applied in various fields such as voice commands and control, voice search, automatic translation, and speech-based systems. A speech recognition system converts voice signals into digital signals, followed by a preprocessing stage to extract features. These extracted features are used as data representing the linguistic content of the voice signal and serve as input data for the model. Recently, deep learning and artificial neural network technologies have been utilized to improve the accuracy of speech recognition. Neural network architectures such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), and Convolutional Neural Networks (CNNs) are commonly employed, and these techniques maximize performance by training models on large-scale speech datasets. Additionally, modern natural language processing (NLP) models like BERT and GPT, which are based on transformer architectures, have contributed to improving the performance of speech recognition. These models can better understand the context and meaning of voice signals, enabling more sophisticated speech recognition.

Speech recognition technology is applied in a wide range of devices and applications, including smartphones, IoT devices, in-vehicle navigation systems, and smart speakers, and its usage continues to expand across various industries. [2]

3. SYSTEM SUGGESTION

3.1 Setup Environment

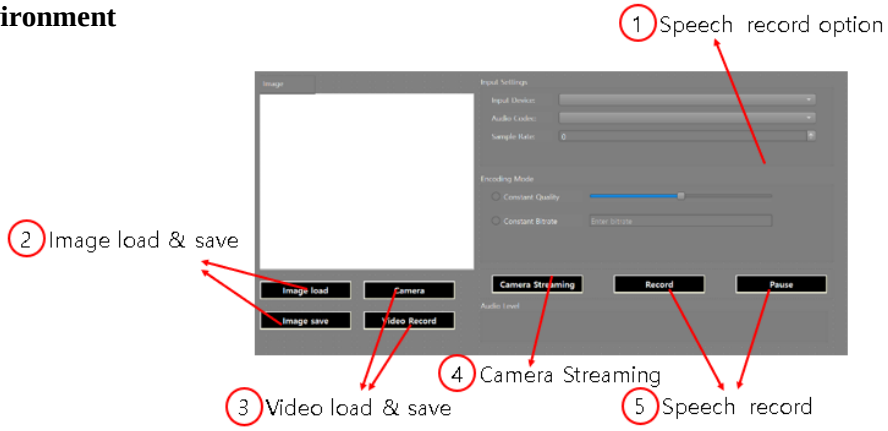


Figure 1. System UI

As following Figure 1, in the System UI, users can utilize functionalities through a connected microphone and camera. The "Speech Record Option" detects a connected microphone, enabling users to control audio quality by adjusting the audio codec, sampling rate, and encoding mode. The "Image load & save" allows users to upload standard images without using a camera. This feature is designed to output recognized text from the images using an OCR algorithm, and users can save the resulting text-embedded image. The "Video laod & save" enables users to upload videos with OCR algorithms applied to individual frames. Users can then save the video with embedded recognized text. The "Camera Streaming" connects to a live camera feed, enabling real-time OCR processing of video streams, allowing users to observe recognized text in real-time. Lastly, The "Speech record" provides a function for users to record spoken audio.

This program developed in this study is based on Windows OS and provides a user-friendly and intuitive interface using the Qt framework. Windows OS allows easy access for a wide range of users, and Qt contributes to cross-platform development and enhanced usability.

This program is integrated with essential libraries for image processing and deep learning model execution. OpenCV offers powerful image processing capabilities, enabling the recognition of characters from vehicle license plates captured in video or photo data through an OCR algorithm, converting them into text. It also provides the ability to detect and track vehicles within the space. These results are dynamically displayed on a screen within the Qt UI, offering a viewer function that shows the current position and information of the detected characters. CUDA utilizes the Single Instruction Multiple Threads (SIMT) functionality to execute multiple threads simultaneously for parallel processing, efficiently leveraging NVIDIA' s GPU architecture. This significantly reduces the inference time of deep learning models while providing highly improved results, enabling the acceleration of deep learning models within the program. [3 - 4]

3.2 OCR Process

Today, OCR is a technology that detects characters from images or videos and converts them int o text. We performed OCR inference using Google's Cloud Vision API. As shown in Figure 2, the parts of the captured image corresponding to text are marked with bounding boxes, and the characte rs are converted to text and returned to the user. Unlike humans, computers cannot intuitively distin guish characters and instead recognize clusters of pixels with similar brightness through color analys is. Therefore, the greater the contrast in color, the higher the recognition accuracy. To enhance this, the image undergoes a preprocessing stage where the color image is converted to grayscale, and pixel values are analyzed to increase brightness and contrast. The pixel values are th en binarized by dividing them into two ranges, classifying them as either 0 or 1. In addition to thi s, various techniques such as noise removal, line removal, and layout analysis are used in combinati on during the image preprocessing stage.

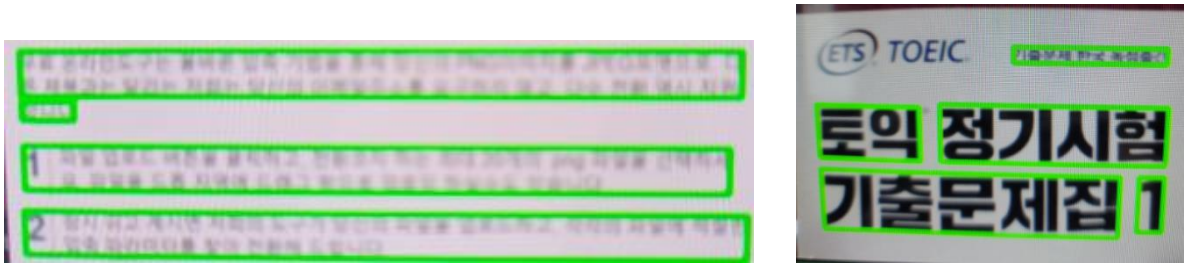


Figure 2. Character recognition using OCR

The preprocessed image, as shown in Figure 3, detects text areas and calculates the rotation angle of the respective areas in order to align the text horizontally. A bounding box is then drawn around the portions corresponding to the text, allowing the system to recognize the characters.

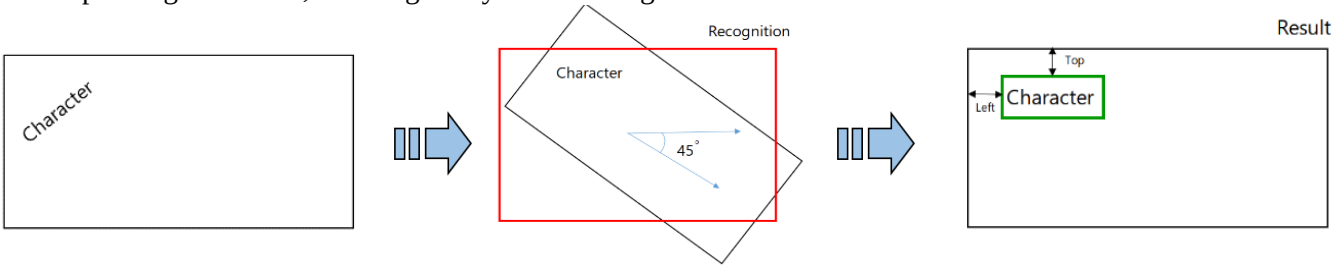


Figure 3. Instruction classification process

In the character detection process, bounding box regions corresponding to the extracted characters are isolated, resulting in a division of the characters as shown in Figure 4. Each separated character is recognized as an individual character, and the model, trained on a large dataset, learns various features to identify the specific character. This information is then returned to the user as a result.

Finally, in the post-processing stage, the content of the output text is examined, and if any unnatural words or characters are present, they are corrected to enhance the overall accuracy. [5]

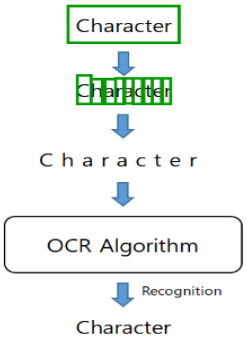


Figure 4. Character recognition

3.3 Speech command

To enable speech recognition, we utilize a process that converts audio signals into digital signals through spectrogram transformation. A spectrogram is a graphical representation that visually illustrates the frequency changes over time, as shown in Figure 5. Frequency indicates the pitch of sound waves, while time refers to the elapsed duration since the sound wave was produced. This technique is widely used in audio signal processing and speech recognition fields, aiding in the visualization of frequency and time variations to understand and analyze the characteristics of speech signals. It finds applications in various areas, such as word classification and music genre classification.

The spectrogram intuitively displays the frequency components, energy distribution, and patterns of frequency variations in speech signals, enabling users to understand and analyze the audio data. This allows it to be used as input data for machine learning systems, where models can process and infer speech data, conducting classifications of commands spoken by individuals. [6 - 7]

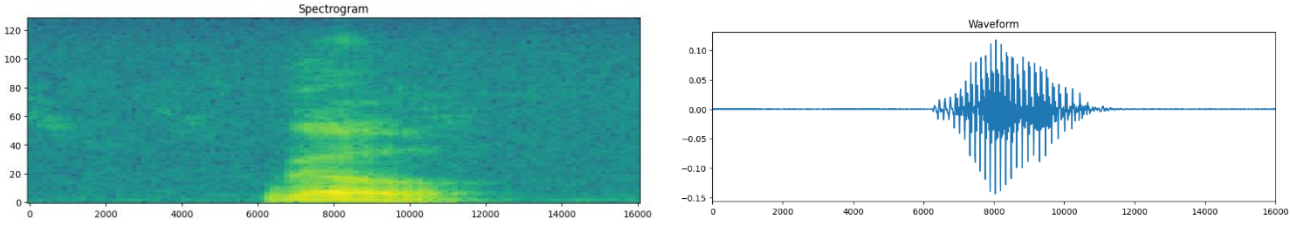


Figure 5. Speech command spectrogram (Command : Go)

To convert to a spectrogram, the first step is to segment the signal through windowing, as described in Equation (1). This process involves dividing the original signal into small segments of fixed length, and then applying a window function to each segment for transformation. Here, x[n] represents the orig inal signal, while w[n] denotes the window function.

$$x[n] \cdot w[n] \tag{1}$$

The second step is to perform a Fast Fourier Transform (FFT) on each segment, as described in Equation (2), to extract the frequency components. Here, X[k,m] represents the k-th frequency component of the m-th window, x[n] is original signal, w[n -m] is window function n-m represents the shift of the window across different frames, e is base of the natural logarithm approximately equal to 2.718, j is the imaginary unit, N is the length of the window and k allowing computation of the spectral content for each frequency.

$$X[k,m] = \sum_{n=0}^{N-1} x[n] \cdot w[n-m] \cdot e^{-j2\pi \frac{kn}{N}} \tag{2}$$

Equation (3) illustrates the process of generating the spectrogram by squaring the magnitude of ea ch frequency component. Each pixel in the spectrogram represents the intensity of the signal at a sp ecific time and frequency. Ultimately, the transformed spectrogram is used for classification tasks rel ated to commands within a DenseNet architecture model. Here, S[m,k] is the spectrogram value at time frame m and frequency k. $|X[k,m]|^2$ is denotes the squared magnitude of the STFT (Short-Time Fourier Transform) k is frequency and m is time frame.

$$S[m,k] = |X[k,m]|^2 \tag{3}$$

We conducted experiments using the dataset provided by Kaggle's Speech Commands, working wi th a total of eight classes. The training, validation, and test sets were divided in a ratio of 8:1:1. A s shown in Table 1, the simplest model, the CNN, achieved a result of 89.64%. Through experimen ts with various network architectures, we obtained results of 92.92% for ResNet50 [8] and 94.22% f or DenseNet. Based on these experimental results, we chose to utilize the DenseNet model for the vehicle passage system in a parking management system, allowing users to interact with the system via voice commands through audio devices.

Table 1. Instruction classification model result

	CNN	Resnet50	**Densenet**
Learning rate	0.001	0.001	0.001
Epsilon	1e-07	1e-07	1e-07
Loss function	Sparse Categorical Cross entropy		
Max Epochs	200	200	200
Batch size	32	32	32
Early stopping	Patience = 5, min_delta = 0.0001		

Optimizer function	Adam	Adam	Adam
Accuracy	89.64%	92.92%	**94.22%**
Recall	0.89	0.93	**0.94**
Precision	0.89	0.91	**0.93**
F1 Score	0.89	0.92	**0.94**

As shown in Figure 6, DenseNet features a unique architecture where each layer is connected to all preceding layers. This design allows outputs from earlier layers to be continuously fed into subsequent layers, effectively preventing the vanishing gradient problem.

In Figure 7, the structure of the Dense Block and Transition Layer can be observed. To perform pooling operations between Dense Blocks, it is necessary to design a Transition Layer.

The bottleneck architecture of DenseNet utilizes 1x1 convolution operations instead of traditional

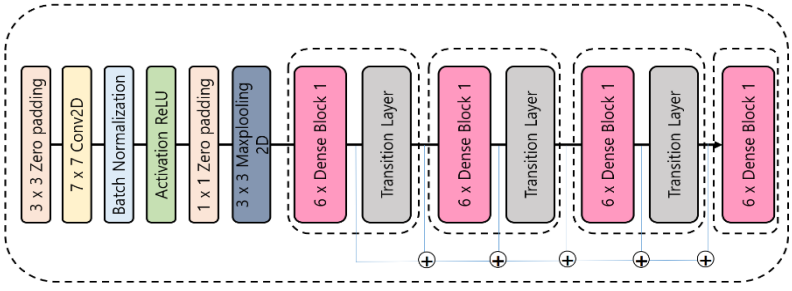


Figure 6. Speech command Densenet structure

convolution operations to compute feature maps. This approach allows for greater flexibility in combining input feature maps in various ways. Additionally, the bottleneck structure can be employed to construct networks of varying sizes and depths, achieving high accuracy with a minimal number of parameters, thereby ensuring an efficient network architecture. Furthermore, the input feature maps leverage the dense connectivity of layers, facilitating more effective sharing of feature maps by connecting them directly to the output feature maps. In this bottleneck structure, the variable 'K' represents the growth rate, which determines the number of feature maps added to each layer's input within the dense block, thereby indicating how much each layer contributes to the overall output. [9]

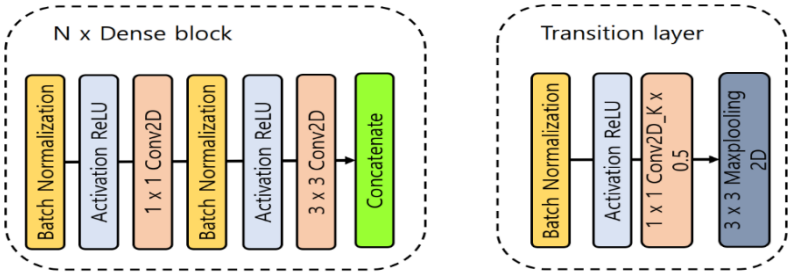


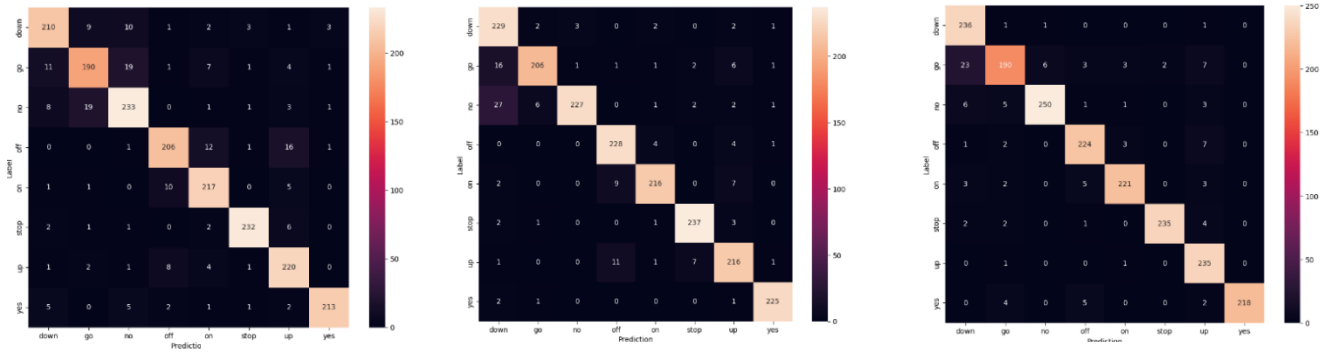
Figure 7. (a) is Dense block structure and (b) is Transition layer structure

4. RESULT

We attempted to recognize vehicle license plates using Optical Character Recognition (OCR). Figu re 8 shows experiment the detection of text segments within the license plate, for example classified i nto bounding boxes as NEW, YORK, 234ABC,NEW, YORK and CITY. This process allows us to c onvert characters visible on a computer display into digital character-type data, which can then be p resented to the user.

Figure 8. Vehicle license plate recognition through OCR

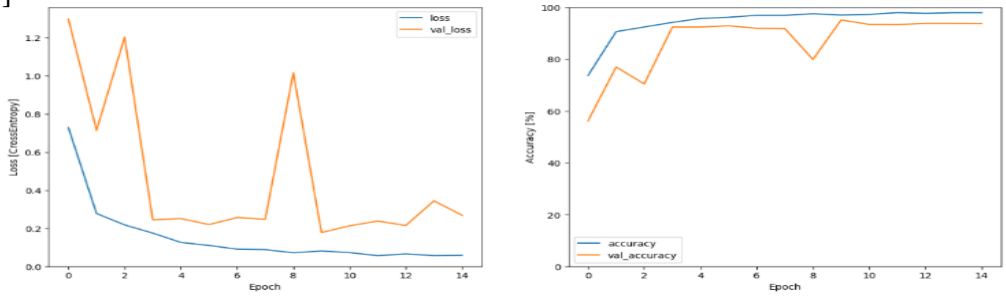
We visualized the results by extracting the confusion matrix, as shown in Figure 9. The obtained values were an F1 Score of 0.94, Precision of 0.93, and Recall of 0.94. When comparing the results of CNN and ResNet, DenseNet demonstrated the best performance, achieving an accuracy of 94.22%.



(a) **(b)** **(c)**

Figure 9. (a) is CNN, (b) is Resnet and (c) is Densenet confusion matrix

Figure 10 displays the Loss and Accuracy of DenseNet. To prevent overfitting during data training, we utilized a Patience option. The Patience was set to stop the training immediately if the Loss did not improve over 5 epochs. Additionally, we configured the maximum number of epochs to 200 for the training process. [10]



(a) **(b)**

Figure 10. (a) is Densenet Loss and (b) is Densenet Accuracy

5. CONCLUSION

In this study, we developed a program utilizing deep learning models for speech commands and Optical Character Recognition (OCR) within a Windows OS environment using the Qt framework. This program integrates deep learning elements to provide a user-friendly interface that recognizes commands spoken by users in real-time through audio devices and visualizes them on a display. Additionally, it employs OCR to capture input data from a camera, enabling recognition of characters on vehicle license plates.

To ensure high performance from our model, we combined OpenCV and CUDA libraries. Furthermore, DenseNet demonstrated superior performance compared to conventional CNNs across all metrics, including Loss, Accuracy, F1 Score, Recall, and Precision, while exhibiting low memory consumption. This makes it suitable for use in small devices like embedded boards.

By introducing a system that integrates computer vision and speech recognition, we aimed to build a practical application for real-time fields leveraging widely used technologies in object and speech recognition. Additionally, this system can efficiently handle tasks in parking management by allowing users to issue simple commands through audio devices. Moreover, with potential applications, OCR can read recognized text aloud through speakers, assisting visually impaired individuals by enabling them to provide commands.Thus, our system is expected to be applied in various fields within real-time recognition systems.

Acknowledgement

The present research has been conducted by the Research Grant of Seoil University

References

[1] Thi Tuyet Hai Naguyen, Adam jatowt, Mickael Coustaty and Antoine Doucet, " Survey of Post-OCR Processing Approaches" , ACM Computing Surveys (CSUR) , vol. 54, pp. 1 - 37, 2021.
DOI: https://doi.org/10.1145/3453476

[2] Juyoung Kim, Dai Yeol Yun, Oh Seko Kwon, Seok Jae Moon and CHio gon Hwang " Comparative Analysis of Speech Recognition Open API Error Rate" International Journal of Advanced Smart Convergence (IJASC), vol. 10, pp. 79-85, 2021
DOI: https://doi.org/10.7236/IJASC.2021.10.2.79

[3] Felipe R. Monteiro, Mario A. P. Garcia, Lucas C. Cordeiro and Eddie B. de Lima Filho, " Bounded model checking of C++ programs based on the Qt cross-platform framework" , ASE 18 Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering , pp. 954, 2015.
DOI: https://doi.org/10.1109/GCCE.2015.7398699

[4] Hancheng Wu, John Ravi and Michela Becchi " Compiling SIMT Programs on Multi- and Many-Core Processors with Wide Vector Units: A Case Study with CUDA" International Conference on High Performance Computing(HiPC), pp. 123-132, 2018
DOI: https://doi.org/10.1109/HiPC.2018.00022

[5] Chirag Patel, Atul Patel and Dharmendra Patel" Optical Character Recognition By Open Source OCR Tool Tesseract: A Case Study" International Journal of Computer Applications(IJCA), vol. 55, 2012
DOI: https://doi.org/10.5120/8794-2784

[6] Lonce Wyse" Audio Spectrogram Representations for Processing with Convolutional Neural Network" Joint with the International Joint Conference on Neural Networks (IJCNN), pp. 1 - 65, 2017.
DOI: https://doi.org/10.48550/arXiv.1706.09559

[7] Y. Zhang, B. Li, H. Fang and Q. Meng, " Spectrogram Transformers for Audio Classification" , International Sustainability Transitions(IST) , 2022
DOI:https://doi.org/10.1109/IST55454.2022.9827729

[8] J. Liang," Image classification based on RESNET" , International Conference on Computer Information Science and Application Technology (CISAT) , pp. 1-6 , 2020.
DOI: https://doi.org/10.1088/1742-6596/1634/1/012110

[9] Z. Zhong, M. Zheng, H. Mai, J. Zhao and X. Liu," Cancer image classification based on DenseNet model" , International Conference on Artificial Intelligence Technologies and Application (ICAITA) , pp. 1-6 , 2020.
DOI: https://doi.org/10.1088/1742-6596/1651/1/012143

[10] Xue Ying," An Overview of Overfitting and its Solutions" , Journal of Physics: Conference Series , vol. 1168 , 2019.
DOI: https://doi.org/10.1088/1742-6596/1168/2/022022