scientific reports

Check for updates

OPEN

Multitasking vision language models for vehicle plate recognition with VehiclePaliGemma

Nouar AlDahoul[1], Myles Joshua Toledo Tan[2], Raghava Reddy Tera[3], Hezerul Abdul Karim[4], Chee How Lim[5], Manish Kumar Mishra[5] & Yasir Zaki[1]✉

License Plate Recognition (LPR) automates vehicle identification using cameras and computer vision. It compares captured plates against databases to detect stolen vehicles, uninsured drivers, and crime suspects. Traditionally reliant on Optical Character Recognition (OCR), LPR faces challenges like noise, blurring, weather effects, and closely spaced characters, complicating accurate recognition. Existing LPR methods still require significant improvement, especially for distorted images. To fill this gap, we propose utilizing visual language models (VLMs) such as OpenAI GPT-4o (Generative Pre-trained Transformer 4 Omni), Google Gemini 1.5, Google PaliGemma (Pathways Language and Image model + Gemma model), Meta Llama (Large Language Model Meta AI) 3.2, Anthropic Claude 3.5 Sonnet, LLaVA (Large Language and Vision Assistant), NVIDIA VILA (Visual Language), and moondream2 to recognize such unclear plates with close characters. This paper evaluates the VLM's capability to address the aforementioned problems. Additionally, we introduce "VehiclePaliGemma", a fine-tuned Open-sourced PaliGemma VLM designed to recognize plates under challenging conditions. We compared our proposed VehiclePaliGemma with state-of-the-art methods and other VLMs using a dataset of Malaysian license plates collected under complex conditions. The results indicate that VehiclePaliGemma achieved superior performance with an accuracy of 87.6%. Moreover, it is able to predict the car's plate at a speed of 7 frames per second using A100-80GB GPU. Finally, we explored the multitasking capability of VehiclePaliGemma model to accurately identify plates containing multiple cars of various models and colors, with plates positioned and oriented in different directions.

License plate recognition (LPR) systems, also known as automatic number plate recognition (ANPR), utilize optical character recognition on images to read vehicle registration plates. This widely recognized technique is instrumental in traffic management systems and has heaped significant focus on itself due to its real-time applications [1]. An advanced LPR system not only effectively recognizes car plates but also contributes significantly to improving traffic efficiency by distinguishing different classes of vehicles [2]. The adoption of LPR systems in various areas has been growing over the years due to their wide-ranging benefits [3]. In law enforcement, for instance, LPR systems are employed to monitor traffic compliance, find stolen vehicles, and manage access control [4]. In the area of toll systems, car plate recognition enables automatic toll collection, reducing congestion at toll booths. In parking management, ANPR reduces the need for manual ticketing and enables the efficient tracking of vehicles [5].

Despite the importance of this LPR system, there are a few limitations that still pose challenges. The advanced LPR system should be able to handle real-world conditions such as low illumination and weather changes (e.g., rain and snow). Additionally, the recognition system should be able to adapt to various other real-life limitations, such as the usage of low-quality cameras, unclear car plates, and complex backgrounds [6].

The historical evolution of car plate recognition systems showcases a fascinating trajectory of technological advancements aimed at enhancing accuracy, speed, and adaptability. The inception of these systems can be traced back to the use of optical character recognition (OCR)-based approaches, which marked the early efforts to automate the extraction of textual information from vehicle registration plates [7]. These early methods relied heavily on image processing techniques to detect, segment, and recognize characters on the plates, offering a foundational step towards automation. As technology progressed, the field witnessed significant enhancements

[1]Computer Science, New York University Abu Dhabi, Abu Dhabi, UAE. [2]Department of Electrical and Computer Engineering, Herbert Wertheim College of Engineering, University of Florida, Florida, USA. [3]Yo-Vivo Corporation, Bacolod City, Negros Occidental, Philippines. [4]Centre for Image and Vision Computing, Centre of Excellence for Artificial Intelligence, Faculty of Artificial Intelligence and Engineering, Multimedia University, Cyberjaya, Selangor, Malaysia. [5]Tapway Sdn Bhd, Petaling Jaya, Selangor, Malaysia. ✉email: yasir.zaki@nyu.edu

with the integration of traditional machine-learning techniques [4]. These algorithms, including support vector machines (SVMs) and neural networks, offered more robust feature extraction and classification methods, considerably improving the recognition rates under varied and challenging conditions. This era of car plate recognition was characterized by the deliberate shift from rule-based processing to data-driven approaches, enabling systems to learn from examples rather than follow explicitly programmed instructions [8].

Language models are fundamental elements of natural language processing (NLP). They predict the likelihood of a sentence by computing the probability distribution of the next word in the sentence given the words already seen [9]. With developments in deep learning, language models have begun to handle complex tasks in various sectors. In healthcare, for instance, language models help to improve healthcare delivery by analyzing electronic health records [10]. Similarly, in the education sector, language models are used to develop intelligent tutoring systems [11].

Parallel to the advancements of car plate recognition systems, the domain of NLP saw the introduction of large language models (LLMs) [12,13]. These models, powered by deep learning architectures, have revolutionized the way machines understand human language. LLMs, such as the generative pre-trained transformer (GPT) by OpenAI [14] and bidirectional encoder representations from transformers (BERT) by Google [13], exhibit an unprecedented capacity to generate coherent text, comprehend context, and perform language understanding tasks with remarkable accuracy. The general capabilities of LLMs extend beyond text generation to include language translation, question answering, and text summarization, showcasing their versatility across various fields.

Pushing the boundaries of AI capabilities, visual language models (VLMs) are built upon the foundational work done in LLMs. VLMs are designed to process and understand both visual and textual data simultaneously. For instance, VLMs can generate descriptive texts from images, which could then be parsed for relevant information, including car plate data, effectively bridging the gap between visual data and language [15].

Exploring the potential of VLMs in car plate recognition systems presents an innovative research direction. The integration of VLMs could address some of the limitations of traditional methods, such as the handling of obscured or distorted plates and the adaptation to new plate formats without extensive retraining. The rationale behind leveraging VLMs lies in their ability to understand and interpret context, which could be beneficial in deciphering partially visible or damaged plates. Furthermore, their adaptability and generative capabilities suggest potential benefits in terms of accuracy and robustness, making them a promising tool in the continual evolution of car plate recognition technologies.

In this study, our proposed license plate recognition system utilizes state-of-the-art visual language models such as GPT-4o [14], Google's Gemini 1.5 [16], Google PaliGemma [17] , Meta Llama 3.2 [18], Anthropic Claude 3.5 Sonnet [19], LLaVA-NeXT [20,21], VILA [22], and moondream2 [23] to recognize plate's characters that are too close to each other and were captured under various challenging conditions. Our contributions can be summarized as follows:

1. We explored the OCR capability of visual language models and employed them in the task of license plate recognition.
2. We evaluated state-of-the-art visual language models such as GPT-4o, Google Gemini 1.5, Google PaliGemma, Meta Llama 3.2, Anthropic Claude 3.5 Sonnet, LLaVA-NeXT, VILA, and moondream2 in terms of plate-level recognition accuracy and character-level accuracy.
3. We utilized an image dataset of plates that were collected in real-life under various challenging conditions, including low illumination, low-quality cameras, unclear car plates, and close characters.
4. We proposed two multitasking VLMs, namely "VehicleGPT" and "VehiclePaliGemma" for localizing and recognizing plates' characters from images of multiple cars using a prompt engineered for a car with a specific color and modal.

The rest of the paper is organized as follows: In Section, we review previous works on OCR and LPR. Section presents our research motivation. In Section, we describe the plate images collected to run the experiments and the methodology used by our LPR system. Section discusses the experimental results and compares the proposed solution with other baseline methods. We discussed our findings and concluded with a summary of key takeaways in Section. Finally, limitations and future work are indicated in Section.

Related work

Traditional methods of car plate recognition

Before the widespread application of deep learning techniques, car plate recognition systems largely hinged on optical character recognition (OCR) and traditional machine learning methods such as SVMs [1] and k-nearest neighbor (KNN) models [24]. These technologies are aimed at identifying and classifying the characters of the license plates from the images. OCR methods were pivotal in converting different styles of vehicle number plate fonts into machine-encoded text. Machine learning methods like SVMs excelled at classifying segmented characters into recognizable letters and digits based on feature extraction from the input images [4].

Edge detection methods, such as the Canny edge detector [25], have been widely used for identifying car parts in images by highlighting significant transitions in intensity. Similarly, color analysis techniques, such as histogram-based methods, are employed to distinguish cars from the background based on their color distribution [26].

Template matching, which is another traditional method, involves comparing portions of the image with pre-defined templates of car shapes. Although this is useful in specific scenarios, template matching is computationally intensive and less adaptable to diverse real-world conditions [4].

Despite their successes, traditional methods faced notable limitations. The accuracy of these systems significantly declined in suboptimal conditions such as poor lighting, varied angles, motion blur, and diverse

plate formats. These methods also struggled with the generalization needed to cope with the worldwide variety of license plate designs, requiring considerable manual tuning to adapt to each new format [3].

Deep learning approaches

The advent of deep learning has significantly transformed car plate recognition systems, offering enhanced accuracy and robustness. The emergence of Convolutional Neural Networks (CNNs) has substantially advanced the field of image recognition [27]. CNNs have been instrumental due to their hierarchical feature extraction capabilities, which accurately identified salient features in images without the need for manual feature design [28]. In the realm of car plate recognition, CNNs have demonstrated superior performance in detecting and recognizing number plates under various challenging conditions, outperforming traditional machine learning methods [29].

Several notable studies have emphasized the efficacy of CNNs in this domain. For instance, researchers developed a system employing CNNs that achieved remarkable accuracy in recognizing Brazilian car plates using two (You Only Look Once) YOLO-CNNs [29]. This success underscores the CNNs potential to drastically mitigate the previous limitations through their adeptness at learning complex, variable patterns in data.

AlexNet [27], a pioneering CNN architecture, demonstrated the potential of deep learning in large-scale image classification tasks, setting the stage for its application in car plate recognition [30]. Subsequent architectures like VGGNet [31] and ResNet [32] further improved the recognition performance by introducing deeper and more complex network structures [30].

Region-based CNNs (R-CNNs) [33] and their variants, such as Fast R-CNN [34] and Faster R-CNN [35], have been specifically tailored for object detection tasks, making them highly effective in identifying and localizing cars in images [36]. These models use region proposal networks to suggest potential bounding boxes, which are then refined by the CNN.

The YOLO family of models [37,38], known for their real-time detection capabilities, have also been applied to car plate recognition with impressive results [39]. YOLO's unified architecture, which performs detection and classification in a single forward pass, offers a balance between speed and accuracy.

More recently, transformers, originally designed for natural language processing, have been adapted for image recognition tasks. The Vision Transformer (ViT) [40] leverages self-attention mechanisms to capture the global context in images, showing promise in car plate recognition applications [41].

Emerging use of LLMs in image processing

The application of Large Language Models (LLMs) like GPT [14] and BERT [13] transcends the barriers of text processing, venturing into non-text-based tasks including image recognition and processing. This expansion has been facilitated by the models' ability to understand and generate human-like text, providing a novel approach to interpreting and analyzing images [15].

Recent interdisciplinary studies have begun to explore the feasibility of LLMs for image-related tasks. For example, researchers have demonstrated the capabilities of GPT in generating textual descriptions from images, opening new pathways for image understanding and processing through natural language descriptions [15].

Large language models (LLMs), like GPT and its successors, have primarily been recognized for their prowess in natural language understanding and generation. However, recent research has begun exploring their potential in image recognition tasks, often through multimodal learning approaches [42]. The integration of LLMs with car plate recognition systems is a nascent area of exploration that holds the potential to redefine the efficiencies of these systems.

Multimodal models, such as CLIP (Contrastive Language-Image Pretraining) [15], combine the strengths of LLMs and CNNs by training on pairs of images and their textual descriptions. CLIP has demonstrated state-of-the-art performance on a variety of image recognition benchmarks, including car plate recognition [15]. By leveraging large-scale datasets of images and text, CLIP learns a joint representation space, enabling robust recognition even in zero-shot scenarios.

DALL-E [43], another multimodal model, generates images from textual descriptions, showcasing the potential of LLMs in understanding and creating visual content [43]. While primarily a generative model, the principles underlying DALL-E's training could inform the development of more sophisticated car plate recognition systems.

The integration of LLMs with traditional vision models has also been explored through techniques like visual question answering (VQA) [44], where models are trained to answer questions about images. These systems require a deep understanding of visual and textual information, highlighting the synergy between LLMs and image recognition [44].

Recent work utilized three pre-trained OCR models, namely Tesseract [45], EasyOCR [46], and KerasOCR [47] and evaluated their performance in recognizing characters in complex car plates [6]. These models failed to recognize the characters in plate images under challenging conditions and produced low recognition accuracy [6].

Our solution of utilizing VLMs for car plate recognition is proposed to address recognition problems under challenging conditions such as close characters and unclear plates and to improve the recognition accuracy largely using textual and visual understanding, as well as the OCR capability of VLMs for this purpose.

Research motivation

Although direct applications of VLMs in car plate recognition have yet to be extensively documented, the principles of the case studies—mentioned earlier in the related work section—offer intriguing prospects. The adaptability and contextual understanding of VLMs could potentially address complex challenges in car plate recognition, such as deciphering obscured or damaged plates and recognizing plates from diverse global formats without extensive reprogramming for each new case.

The insights from these studies suggest that VLMs, with their deep understanding and generation capabilities, could offer complementary, if not substitutive, solutions to traditional and CNN-based approaches in car plate

recognition systems. By leveraging the advanced language comprehension and contextual analytics of VLMs, researchers could pave the way for breakthroughs in accuracy, efficiency, and adaptability in car plate recognition technologies.

Materials and methods

Dataset overview

Complex plate dateset

The license plate dataset used in this work consists of 258 labeled images of Malaysian license plates that are blurry, not clear, and have close characters. The dataset was collected by a Malaysian company called Tapway Sdn Bhd [48]. These images were considered complex and difficult to recognize by state-of-the-art OCR methods. Figure 1 shows examples of these plates.

This set of 258 images was collected for evaluation purposes only to test if the proposed solution is able to address the previous limitations and recognize the plates correctly (i.e., the gold set). Our researchers manually labeled the images to identify the characters in each one. This process was repeated three times, involving three different individuals, to ensure data consistency and accuracy. The final labels were determined using a voting technique to confirm the correct characters. The plate images have a width range of 64–181 pixels and a height range of 24–72 pixels.

Fine-tuning dateset

We developed a synthetic image dataset to fine-tune PaliGemma. This dataset comprises 600 images of Malaysian license plates, created with a black background and white alphanumeric characters (letters and numbers). Each image has a resolution of 50x120 pixels. Two plate formats were generated: a single line containing three letters followed by four numbers, and a two-line format where the first line includes three letters, and the second line contains four numbers. The letters and numbers were selected randomly. The images were rotated by 5 degrees in both directions, blurred, and subjected to Gaussian and salt-and-pepper noise.

Diverse car dateset

We scraped a dataset consisting of 140 images of single or multiple cars from the web with the key word "Malaysian car plates". We labeled these images by three evaluators with a majority voting technique as follows: if at least two evaluators, out of the three, gave the same label to the character, then this label is deemed to be correct. Otherwise, the character is checked again to have an agreement from at least two evaluators. This dataset was utilized to evaluate the multitasking capability of VehicleGPT and VehiclePaliGemma.

Methods

The proposed solution for license plate recognition is an artificial intelligence system that combines both language and visual processing to provide an enhanced understanding and generation capabilities and to extract characters from car plate images given a proper prompt. We employed VLMs to utilize their natural language processing capabilities to interpret and analyze the context within the images. The solution utilizes the OCR

Fig. 1. Sample complex license plates from the used dataset.

capability of VLMs to understand the text, including the characters in the license plate, directly from the plate images without any preprocessing. Figure 2 shows the block diagram of the proposed solution.

As shown in Fig. 2, the license plate image and text (i.e., the prompt) are applied to the inputs of each VLM, namely GPT-4o [14,49], Google's Gemini 1.5 [16,50], Google PaliGemma [17], Meta Llama 3.2 [18], Anthropic Claude 3.5 Sonnet [19], LLaVA-NeXT [20,21,51], VILA [22], and moondream2 [23,52].

We evaluated each of these VLMs separately and compared their outcomes against the ground truth. These VLMs represent the well-known VLMs available in the literature in both small- and large-size models.

Each of these VLMs has the OCR capability to understand the contents of the image, such as their characters, and the language processing capability to understand the prompt given to the VLM asking them to perform a specific action on the given image. The VLM processes the plate image to recognize its characters and also uses its contextual understanding to ensure that the extracted text makes sense and aligns with the prompt's requirements. In this work, various VLMs such as GPT-4o, Google's Gemini 1.5 Pro, Meta Llama 3.2 11b, Anthropic Claude 3.5 Sonnet, LLaVA-NeXT-34b have been evaluated and compared to find the best model that can produce the highest recognition accuracy. Additionally, we also evaluated the performance of small vision language models (such as GPT-4o-mini, Gemini 1.5 Flash, Google PaliGemma 3b, LLaVA-NeXT-7b, VILA, and moondream2), which are designed to run efficiently on laptops or edge devices. In this section, a summary of each used VLM is presented. The prompt that was used for the comparison is "`Extract three letters and four numbers from this car's plate; print the result in one word as: letters followed by numbers`".

OpenAI generative pre-trained transformer 4 omni

The Generative Pre-trained Transformer 4 Omni (GPT-4o) [14,49] is the first VLM used in this study. It has vision capabilities and is a big step forward in AI because it combines powerful language processing with complex image analysis. This multimodal model integrates visual understanding with textual analysis, expanding the functionality of AI applications. GPT-4o excels in visual question answering (VQA), allowing users to input images alongside questions to receive contextually relevant answers. Additionally, GPT-4o demonstrates strong optical character recognition (OCR) capabilities, effectively extracting and interpreting text from images, which benefits document digitization and reading signs in images [14,49]. The model's ability to combine image and text processing enables comprehensive and nuanced responses. For example, GPT-4o can describe image contents, generate captions, or analyze charts and graphs for insights. Its improved contextual understanding enhances its utility in continuous engagement applications [14,49]. Additionally, we used GPT-4o mini, which is the most advanced model in the small models category [14]. It is the cheapest, most affordable, and most intelligent small model for fast and lightweight multimodal tasks (accepting text or image inputs and outputting text).

Google Gemini-1.5

The second VLM utilized in this work is Google Gemini-1.5 [16]. This paper explored two versions of Gemini-1.5: the large Gemini 1.5 Pro and the small Gemini 1.5 Flash. The Gemini 1.5 Pro is a mid-size multimodal model optimized for a wide range of tasks [16]. It features a context window of up to one million tokens, enabling it to seamlessly analyze, classify, and summarize large amounts of content within a given prompt. When compared to the largest 1.0 Ultra model [16,50] on the same benchmarks, it performs at a broadly similar level. Additionally,

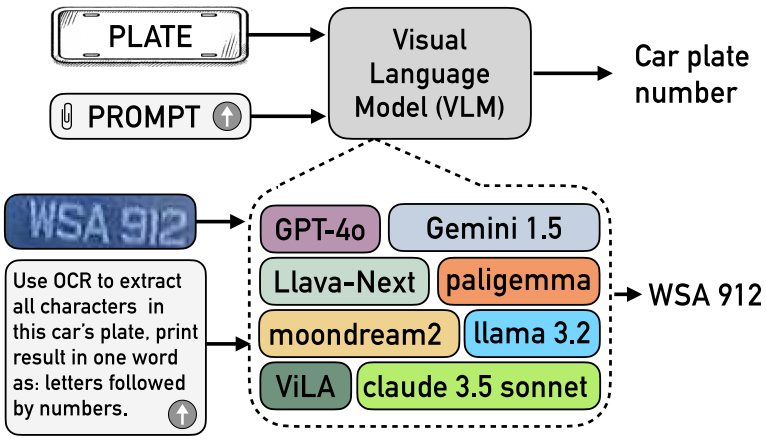

Fig. 2. The proposed solution block diagram. The diagram illustrates the process of recognizing car plate numbers using a VLM. The system takes as input an image of a license plate and a textual prompt instructing the VLM to extract and format the plate number. OCR is first applied to detect the characters in the plate image, and the extracted text is formatted as a single-word output, with letters preceding numbers. Multiple VLMs, including GPT-4o, Gemini 1.5, Llava-Next, PaliGemma, Moondream2, Llama 3.2, ViLA, and Claude 3.5 Sonnet, are used to process the OCR result and generate the final plate number, such as WSA912. The extracted text is then returned as the recognized car plate number.

Gemini 1.5 Pro demonstrates impressive in-context learning abilities, allowing it to acquire new skills from information provided in a long prompt without requiring additional fine-tuning.

On the other hand, Gemini 1.5 Flash [16,50] represents a significant leap in AI technology by integrating multimodal capabilities with an emphasis on speed and efficiency. This model is designed to handle high-frequency tasks at scale, making it ideal for applications requiring rapid, real-time processing of both text and visual data. One of the standout features of Gemini-1.5 Flash is its long context window, which can process up to one million tokens [16,50]. In terms of strengths, Gemini-1.5 Flash excels in multimodal reasoning, effectively integrating text and visual information to deliver accurate and insightful outputs. Its efficiency is bolstered by a streamlined architecture using a "distillation" process, where essential knowledge from larger models is transferred to this smaller, more efficient model. This makes it highly cost-effective and accessible for a wide range of users, from developers to enterprise customers.

Google PaliGemma

Google's PaliGemma is an open vision-language model (VLM) that extends the PaLI series by integrating it with the Gemma family of language models. Built upon the SigLIP-So400m vision encoder and the Gemma2b language model, PaliGemma serves as a versatile and broadly applicable base model, excelling in transfer learning [17]. It showcases strong performance across diverse open-world tasks, leveraging multi-task learning through task prefixes. The prefix-LM approach, which uses task prefixes and supervises only suffix tokens, proves to be an effective pre-training objective for VLMs.

While fine-tuning is useful for solving specific tasks, a generalist model with a conversational interface is often preferred. Instruction tuning, achieved by fine-tuning on a diverse dataset, typically facilitates this versatility. PaliGemma has been shown to be well-suited for such transfer learning [17].

GPT-4o, GPT-4o Mini, Gemini 1.5 Pro, Gemini 1.5 Flash, and Claude 3.5 Sonnet are closed-source models that are costly to fine-tune and require payment for inference. Additionally, at the time of our experiments, fine-tuning these models was not an available option. In contrast, among open-source LLMs such as LLaVA, VILA, and Moondream2, the pre-trained PaliGemma demonstrated significantly superior recognition performance. Consequently, we selected PaliGemma for fine-tuning, as it not only outperformed other open-source models but is also freely available.

In this work, we employed two versions of PaliGemma: the pre-trained PaliGemma and a fine-tuned version named VehiclePaliGemma, specifically optimized for the car's license plate recognition task. The VehiclePaliGemma was fine-tuned using 600 synthetic plate images (see the Dataset section above). All parameters in PaliGemma, including Vision tower and multimodal projector, were updated during fine-tuning. The hyperparameters were set as follows: the number of training epochs was 5, the training batch size was 16, the learning rate was 0.00002, and the optimizer used was Adam. No parameter-efficient fine-tuning methods, such as LoRA, or quantization-based tuning, such as QLoRA, were applied. The fine-tuning was run on A100-80GB GPU.

The outcome of the fine-tuning was fine-tuned PaliGemma, "VehiclePaliGemma", that we open-sourced on the Hugging Face platform https://huggingface.co/NYUAD-ComNets/VehiclePaliGemma

Llama instruct

Llama 3.1, developed by Meta, is an auto-regressive language model built on an optimized transformer architecture [53]. It includes multilingual LLMs that offer both pre-trained and instruction-tuned generative models, designed to handle text inputs and outputs effectively.

Llama 3.2 Instruct with vision capability [18] extends the Llama 3.1 text-only model into a multi-modal generative framework capable of processing both text and image inputs to generate text outputs. Optimized for tasks like visual recognition, image reasoning, captioning, and answering questions about images, Llama 3.2 Instruct employs instruction tuning. It integrates a separately trained vision adapter to handle image recognition, which works in conjunction with the pre-trained Llama 3.1 language model. In this study, we evaluated Llama 3.2 11b model to support our efforts in recognizing complex car's plate by combining object recognition in images with semantic analysis of text.

Claude 3.5 sonnet

Claude 3.5 Sonnet establishes new industry standards [19]. It demonstrates significant advancements in understanding nuance, humor, and intricate instructions, excelling at producing high-quality content with a natural and relatable tone. Operating at twice the speed of Claude 3 Opus, Claude 3.5 Sonnet delivers a substantial performance boost. Its enhanced efficiency, paired with cost-effective pricing, makes it an excellent choice for complex tasks.

Claude 3.5 Sonnet is the most advanced Anthropic vision model to date, outperforming Claude 3 Opus on standard vision benchmarks. Its significant enhancements are particularly evident in tasks requiring visual reasoning, such as analyzing charts and graphs. Additionally, Claude 3.5 Sonnet excels at accurately transcribing text from imperfect images–a critical capability for industries like retail, logistics, and financial services. In this work, we explored and evaluated the capability of Claude 3.5 Sonnet model to recognize complex car's plates.

LLaVA-NeXT

The third VLM demonstrated in this work is Large Language and Vision Assistant (LLaVA) [51]. LLaVA-NeXT [20] represents a significant advancement in multimodal AI models, designed to integrate and enhance both language and vision capabilities. This model is built upon the success of its predecessor, LLaVA, incorporating improvements in reasoning, optical character recognition (OCR), and overall world knowledge. LLaVA-NeXT excels in visual question answering (VQA) and image captioning, leveraging a combination of a pre-trained

large language model (LLM) and a vision encoder. The model's architecture enables it to handle high-resolution images dynamically, preserving intricate details that improve visual understanding [20,21,51]. The model's efficiency is another key strength. LLaVA-NeXT achieves state-of-the-art performance with relatively low training costs, utilizing a cost-effective training method that leverages open resources [20]. Despite its strengths, LLaVA-NeXT faces challenges in handling extremely complex visual tasks that may require specialized models for optimal performance. Additionally, while it has shown strong results in zero-shot scenarios, further refinement is needed to consistently match or exceed the performance of commercial models in all contexts [20,21,51]. Several versions of LLaVA are available based on the number of parameters (i.e., the model's size). We utilized two versions in our experiments: large 34 billion LLaVA and small 7 billion LLaVA.

Visual language model (VILA)
It is notably worth considering the computational requirements of VLMs, which are usually important for the practical implementation of such systems in real-world scenarios [54]. Therefore, in this work, small versions of VLMs such as VILA [22] have also been explored for plate recognition. VILA is a very recent VLM pre-trained with interleaved image-text data at scale, enabling multi-image VLM [22]. It unveils appealing capabilities, including multi-image reasoning, visual chain-of-thought, and video understanding. VILA was found to outperform state-of-the-art models like LLaVA-1.5 across various benchmarks. Furthermore, VILA is deployable on the edge via AWQ 4bit quantization. In this work, we utilized the Llama-3-VILA1.5-8B [22] version to recognize characters in plate images.

Moondream2
Another VLM that is used in this work is moondream2 [23,52]. It is an open-source tiny and compact visual language model incorporating weights from the Sigmoid Loss for Language Image Pre-Training (SigLIP) and Phi-1.5 small language models. moondream2 is specifically engineered for efficient operation on devices with limited computational capabilities, such as edge devices with very little memory [23,52].

Experimental setup
Several performance metrics were calculated for evaluation and comparison, including plate-level accuracy, which measures the proportion of correctly predicted license plates, and character-level accuracy, which measures the proportion of correctly predicted characters. All open-source models, such as PaliGemma, LLaVA, VILA, and Llama, were run on an A100-80GB GPU. For closed-source LLMs, inference was conducted via their respective APIs using a CPU.

Results and discussion
This section presents the results of evaluating and comparing our proposed solution, which leverages the OCR capabilities of VLMs to address the challenging problem of car plate recognition. Several VLMs were evaluated and compared in terms of plate-level accuracy and character-level accuracy. Additionally, we compared the proposed solution with three pre-trained deep learning OCR models, namely Tesseract [45], EasyOCR [46], and KerasOCR [47]. The comparison was done using a complex plate dataset that contains complex Malaysian license plates (see the Dataset section above).

We conducted several experiments to evaluate the vision capabilities of the VLMs for: 1) the OCR task in general, and 2) license plate recognition in particular. In the first experiment, we examined GPT-4's vision capabilities and employed OCR to extract characters from the plate images. Integrating OCR with GPT-4 allows the extracted text to be combined with the language model, enhancing the model's understanding and processing of both the image and any associated text. Table 1 shows the character-level accuracy of GPT-4o (97.1%) by recognizing 1700 correct characters out of 1751 characters. Similarly, the GPT-4o mini version gave a close accuracy of 96.7%. Additionally, we investigated the Google Gemini 1.5 Pro model to study the OCR capability of Gemini for our plate recognition task. The results indicate degradation in character-level accuracy in both Gemini 1.5 Pro (93.8%) and Gemini 1.5 Flash (93.8%). Similarly, Llama 3.2 Instruct and Claude 3.5 Sonnet produced less recognition accuracy (93.38% and 92.8%, respectively) compared to Gemini 1.5. Likewise, LLaVA-NeXT has less recognition accuracy compared to the previously mentioned VLMs, producing a character-level accuracy of 85.9% in the 34b version and 80.94% in the 7b version. In contrast, small VLM versions such as VILA show better recognition performance than the LLaVA-NeXT 7b with accuracy of 83.21%. Furthermore, the tiny moondream2 has less recognition capability than VILA with a character-level accuracy of 76.58%. The results indicate that the two small versions of VLMs, namely GPT-4o mini and Gemini 1.5 Flash, outperformed other small VLMs such as VILA and moondream2 in our plate recognition task. The number of correctly predicted characters for each VLM is shown in Table 1.

Using the pre-trained PaliGemma model, a character-level accuracy of 90.92% was achieved, correctly recognizing 1,592 characters out of 1,751. In contrast, the fine-tuned version, VehiclePaliGemma, demonstrated a significant improvement, increasing character-level accuracy by 7% to reach 97.66%, with 1,710 characters correctly identified. This performance surpasses other VLMs in general, including GPT-4o, as detailed in Table 1.

In the second experiment, we analyzed the performance of three widely-used pre-trained deep learning OCR models namely, Tesseract, EasyOCR, and KerasOCR against our proposed VLM-based approach. The evaluation was conducted using the complex plate dataset. This comparison considered challenging conditions in the plates such as lighting, blurring, varying degrees of distortion, and closely spaced characters.

Comparing traditional approaches with VLM-based methodologies reveals substantial differences in potential outcomes, as seen in Table 2. Three pre-trained deep learning models, namely KerasOCR, EasyOCR, and Tesseract, are considered baseline methods in this work and were used for comparison. These models that showed promising performance in various OCR tasks [55–57] failed to recognize the characters in plate images in

Method	Number of correctly Predicted characters	Character-level Accuracy %
Moondream2	1341	76.58 %
VILA	1457	83.21 %
LLaVA-NeXT-7b	1417	80.93 %
Gemini 1.5 flash	1643	93.8 %
GPT-4o-mini	1693	96.7 %
LLaVA-NeXT-34b	1504	85.9 %
Gemini 1.5 Pro	1643	93.8 %
GPT-4o	1700	97.1 %
Llama 3.2 Instruct	1635	93.38 %
Claude 3.5 Sonnet	1625	92.80 %
Pre-trained PaliGemma	1592	90.92 %
VehiclePaliGemma	1710	97.66 %

Table 1. Character-level accuracy results of several VLMs.

Method	Number of correctly Predicted plates	Plate-level Accuracy %
EasyOCR (baseline)	79	32.95%
Tesseract (baseline)	97	36.74%
KerasOCR (baseline)	107	40.53%
Moondream2	102	39.5 %
LLaVA-NeXT-7b	144	55.8 %
VILA	147	57 %
Gemini 1.5 flash	200	77.5 %
GPT-4o-mini	220	85.7 %
LLaVA-NeXT-34b	152	58.9 %
Gemini 1.5 Pro	185	71.7 %
GPT-4o	222	86 %
Llama 3.2	175	67.83 %
Claude 3.5 Sonnet	186	72.1 %
Pre-trained PaliGemma	178	69 %
VehiclePaliGemma	226	87.6 %

Table 2. Plate-level accuracy, comparing the performance of several VLMs against multiple baseline methods.

our dataset [6]. Tesseract 4.0 is an OCR engine based on Long Term Short Memory (LSTM) neural networks [6]. EasyOCR detects Text using the Character-Region Awareness for Text detection (CRAFT) algorithm [6]. After that, EasyOCR utilizes Convolutional Recurrent Neural Network for recognition. Its recognition model contains several components: feature extraction (Resnet and VGG), sequence labelling (LSTM) and decoding (Connectionist Temporal Classification). KerasOCR utilizes CRAFT to detect text areas by analyzing each character region and the affinity between characters [6]. To locate text-bounding boxes, minimum-bounding rectangles are identified on the binary map after thresholding the scores of the character regions and their affinities. For text recognition, it employs either the original CRNN model or a spatial transformer network layer to rectify the text.

The results in Table 2 show the plate-level accuracy and the number of correctly predicted plates. EasyOCR predicted correctly only 87 images [6] and tends to confuse visually similar characters, such as 'I' and '1' or 'B' and '8'. Moreover, it lacks an integrated text detection feature, making it unable to directly recognize text on license plates that contain two lines of characters [46].On the other hand, Tesseract predicted correctly 97 images [6], demonstrating weak resilience against noise, complex visual distortions and inconsistent illumination. This Tesseract OCR usually requires pre-processing techniques such as binarization, noise reduction, and deskewing (aligning the text properly) [58]. In contrast, KerasOCR was able to recognize better with 107 images out of 258 images [6], but it still struggles with rotated text and can better recognize straight, and horizontal text lines. However, all of these three methods have low recognition accuracy and limitations that have been addressed in this work by leveraging the OCR capability of VLMs, as shown in Table 2.

The proposed VLM-based solution addresses these OCR limitations by integrating both visual perception and language-based contextual understanding. As a result, the recognition accuracy is significantly improved even in visually challenging scenarios. Unlike traditional OCR methods that strictly rely on visual character

recognition pipelines, VLMs inherently leverage semantic reasoning, enabling more accurate predictions of partially visible or distorted text, as demonstrated by superior performance in Table 2.

Among large pre-trained VLMs, GPT-4o achieved the highest plate accuracy at 86%, correctly recognizing 222 out of 258 plates in the dataset. Claude 3.5 Sonnet ranked second with a plate accuracy of 72.1%, followed by Gemini 1.5 Pro in third place at 71.7%. VILA-NEXT 34b ranked last among them, achieving a plate accuracy of 58.9%. On the other hand, among the small VLMs, GPT-4o mini achieved the highest plate accuracy at 85.7%, followed by Gemini 1.5 flash with an accuracy of 77.5%, outperforming its larger counterpart, Gemini 1.5 Pro. Pre-trained PaliGemma 3b secured third place with a plate accuracy of 69%, while Llama 3.2 11b ranked fourth at 67.83%. Furthermore, other small VLMs such as VILA, LLaVA-NeXT, and moondream2 have accuracies of 57%, 55.8%, and 39.5%, respectively. All small VLMs except moondream2 were able to outperform the three baseline methods.

The pre-trained PaliGemma model achieved a plate-level accuracy of 69%, correctly recognizing 178 plates out of 258. In comparison, the fine-tuned version, VehiclePaliGemma, exhibited a substantial improvement, increasing plate-level accuracy by 18% to 87.6%, with 226 plates accurately identified. This performance notably exceeds that of other VLMs, including GPT-4o, as shown in Table 2.

The number of correctly predicted plates for each VLM utilized is shown in Table 2. The heatmap of each character's accuracy for each VLM is shown in Fig. 3. The heatmap helps in quickly identifying which models perform consistently across all characters and which ones have variability in their recognition. The lighter colors indicate any particular characters where the models have struggled to identify them.

The results show that traditional systems relying on optical character recognition and machine learning face challenges in adaptability and require extensive manual tuning to maintain high accuracy under varied conditions. On the other hand, VLMs, with their sophisticated understanding of context and nuance, hypothetically promise greater adaptability and accuracy, especially in interpreting obscured or complex plate images. In the end, while VLMs offer a promising avenue for enhancing car plate recognition systems, their integration demands careful attention to computational feasibility and ethical standards.

Integrating VLMs into such plate recognition systems requires careful consideration of ethical standards, as follows:

1. Ensuring that the deployment of these systems respects individuals' privacy, especially in public spaces where data might be collected without consent.
2. addressing any potential biases in the model that could lead to unfair treatment of certain groups, particularly in law enforcement contexts.
3. maintaining transparency in how these models make decisions and ensuring there is accountability for any errors or misuse.
4. safeguarding the data collected and used by these systems to prevent unauthorized access or misuse.
5. adhering to local and international laws regarding data collection, storage, and usage, particularly in relation to surveillance and data protection.

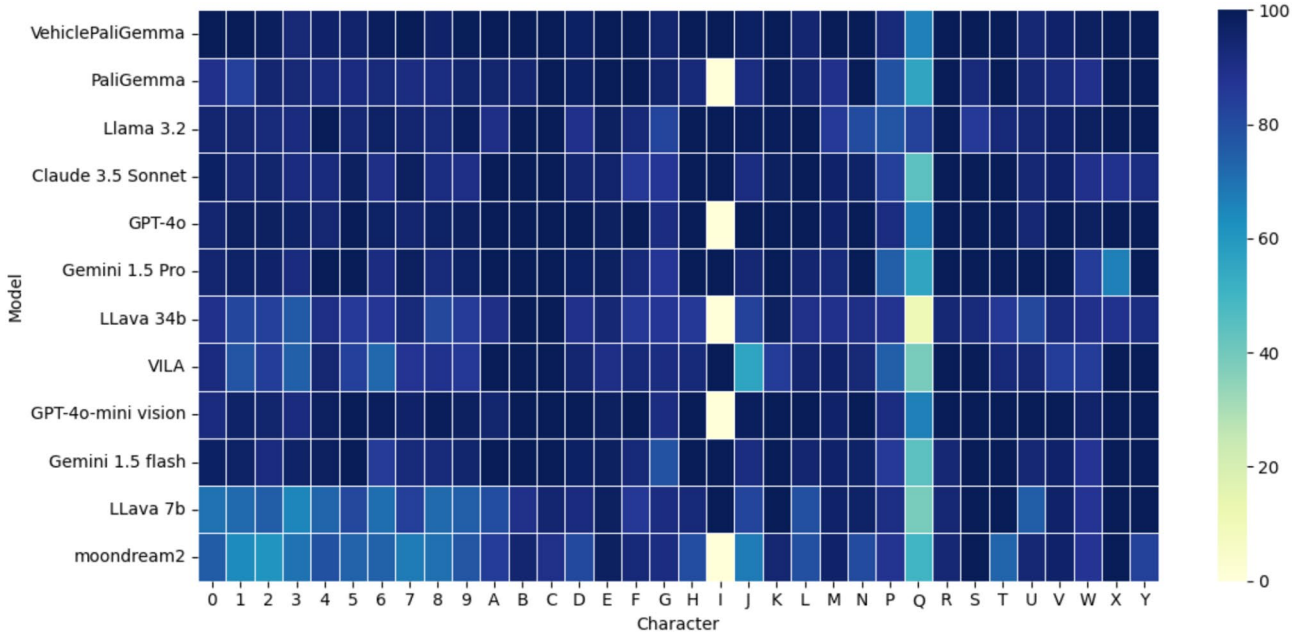

Fig. 3. Character-level accuracy heatmaps for different vision models.

Method		Number of correctly Predicted plates	Plate accuracy (%)
GPT-4o	Prompt1	216	83.7 %
	Prompt2	222	86 %
	Prompt3	**227**	**88 %**
Gemini 1.5 Pro	Prompt1	177	68.6 %
	Prompt2	**186**	**72.1 %**
	Prompt3	176	68.2 %
Pre-trained PaliGemma	Prompt2	119	46.12 %
	Prompt4	**178**	**69** %

Table 3. Prompt sensitivity in GPT-4o and Gemini 1.5 Pro. Significant values are in bold.

Fig. 4. Examples of license plates with Rs predicted as Ps.

Prompt sensitivity

In this section, we studied the impact of prompts in VLMs on our plate recognition task. We chose three VLMs: Pre-trained PaliGemma, GPT-4o and Gemini 1.5 Pro due to their demonstrated superior performance in license plate recognition, as evidenced in prior results. We evaluated four prompts as follows:

- Prompt1: "`extract characters in this car's plate, print result in one word as: letters followed by numbers`"
- Prompt2: "`extract three letters and four numbers from this car's plate; print the result in one word as: letters followed by numbers`"
- Prompt3: "`use OCR to extract all characters in this car's plate, print result in one word as: letters followed by numbers`"
- Prompt4: "`extract the text from the image`"

In the first prompt, we asked both GPT-4o and Gemini 1.5 Pro to extract characters in general without determining the number of letters and numbers in the license. On the other hand, prompt2 explicitly determined the exact number of letters and characters, i.e., four letters and three numbers, which can help in identifying all characters in the plates without missing any, thus increasing the number of correctly recognized plates as shown in Table 3. The previous advantages can be achieved only if all plates under evaluation have the same format (four letters followed by three numbers). Otherwise, the second prompt fails if we have plates with various formats. In the third prompt, we asked both GPT-4o and Gemini 1.5 Pro to use OCR to extract all characters, and the results in Table 3 show the capability of GPT-4o to recognize 227 plates correctly out of 258 plates with an accuracy of 88% using prompt3 which has more recognition capability when used in comparison to prompt2. In contrast, Gemini 1.5 Pro performed better with prompt2 compared to prompt3. Moreover, we evaluated Pre-trained PaliGemma with two prompts: prompt2 (that both GPT-4o and Gemini 1.5 Pro show good performance utilizing it) and prompt4. The plate accuracy with prompt4 was better than one with prompt2 by 23%. The results show that VLMs are sensitive to prompts used to recognize characters in the plate images, and that careful attention should be given to the prompt to achieve the highest performance.

To study the limitations of VLMs, we chose Pre-trained PaliGemma, which was the top recognition model in our experiments. First, we show the limitations using prompt4 as follows:

1. Actual P is predicted as R, such as these pairs of examples (actual, predicted): (PJG90, RJG90), (PJW6633, RJW6633), (PJV8666, RJV8666), (PJC5688, RJC5688). It is clear in most cases that when J comes after P, the model predicts P as R, as shown in Fig. 4.
2. In few cases, When plates have only six characters (three letters and 3 numbers), Pre-trained PaliGemma added one letter, such as these pairs of examples (actual and predicted): (PJN214, PJN2114), (KCJ112, KCJ1112), and (PLA113, PLA1113), as shown in Fig. 5.
3. If a letter comes at the end, Pre-trained PaliGemma will reorder them according data fine-tuned on and put letters before numbers (actual: W1209G, predicted: WI2096).

The use of Visual Language Models (VLMs) for OCR in general, and specifically for license plate recognition, demonstrates significant potential for future applications that remain challenging for traditional machine learning models. Future advancements aimed at enhancing the visual analysis capabilities of VLMs could significantly

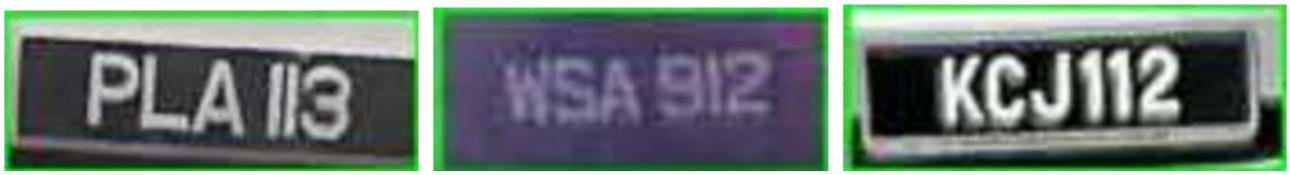

Fig. 5. Examples of license plates to which pre-trained PaliGemma added one extra letter.

Method	Deployment	Speed (second)
Moondream2	Local	0.09
LLaVA-NeXT-7b	Local	0.84
VILA	Local	0.35
Gemini 1.5 flash	API	1.65
GPT-4o-mini	API	1.7
LLaVA-NeXT-34b	Local	10
Gemini 1.5 Pro	API	1.85
GPT-4o	API	1.6
Llama 3.2	Local	0.42
Claude 3.5 Sonnet	API	1.8
Fine-tuned PaliGemma	Local	0.135

Table 4. Inference speed comparison of various VLMs across different deployment environments. The table presents both locally deployed models and API-based models, with inference times measured in seconds. Lower values indicate faster inference.

increase their applicability for image analysis and understanding tasks, such as license plate recognition or any other complex use cases. However, to enhance their capabilities, more diverse and high-quality data are required to further improve the model's generalization capabilities.

Table 4 indicates Inference Speed for several VLMs , highlighting notable differences in latency across models. The latency of API-based LLMs such as GPT-4o and Google Gemini 1.5 is influenced by multiple external factors, including network latency, server processing time, and the current load on remote infrastructures. Consequently, their real-world deployment can be challenging, particularly in latency-sensitive or resource-constrained scenarios. In contrast, locally deployed models like Moondream2 and Fine-tuned PaliGemma exhibit significantly lower latency. This efficiency is primarily due to the removal of network-related delays, allowing these models to achieve faster inference times. Such locally deployed models are particularly advantageous in real-time applications, edge computing scenarios, or situations where network reliability cannot be guaranteed. The results underscore an essential trade-off in model deployment: Fine-tuned PaliGemma achieves a balance between high-speed inference and high recognition accuracy, making it particularly suitable for real-world applications requiring rapid decision-making and accurate performance. The feasibility of deploying such models depends not only on computational efficiency but also on hardware availability, scalability, and maintenance complexity. Thus, selecting an appropriate VLM requires careful consideration of latency requirements, accuracy targets, and deployment environment constraints.

VehicleGPT and VehiclePaliGemma

In this section, we propose "VehicleGPT" (a multitasking GPT-4o) and "VehiclePaliGemma" (a multitasking PaliGemma) with a car's plate recognition capability. It was able to detect (localize and recognize) cars' plates in images with single or multiple cars. We chose both LLMs due to their demonstrated superior performance in license plate recognition, as evidenced in prior results.

In Table 3, we analyzed the prompt sensitivity across three pre-trained LLMs–GPT-4o, Gemini 1.5 Pro, and Pre-trained PaliGemma–to examine the impact of prompt wording on model performance. Our findings indicate that Prompt 4 outperforms Prompt 2 in Pre-trained PaliGemma. Based on this result, we selected Prompt 4 ("extract the text from the image/") to fine-tune PaliGemma, resulting in a specialized model named VehiclePaliGemma.

Figure 6 illustrates the block diagram of the proposed solution of VehiclePaliGemma. In this analysis, the input is an image of a single or multiple car(s), and the prompt used was "`Extract all characters from the plate of the white Nissan car(s)`". The output is the extracted characters from the specific car(s) referred to in the prompt. To detect cars and plates, and then recognize characters in the plates, our proposed solution VehiclePaliGemma followed several steps:

1. using '`detect car`" prompt to utilize the detection capability of pre-trained PaliGemma to localize all cars available in the images.

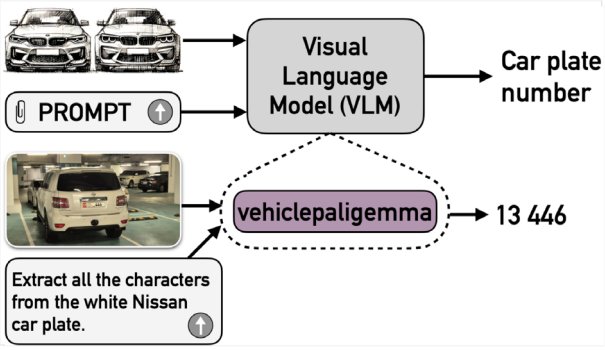

Fig. 6. VehiclePaliGemma's block diagram.

2. using '`detect license plate`" prompt to leverage the pre-trained PaliGemma model's detection capabilities for localizing the plate of an already detected car.
3. using '`extract the text from the image`" prompt with VehiclePaliGemma to recognize characters in the detected plate.
4. if the main prompt has a specific color or model of the car, pre-trained PaliGemma was asked to check the color and model before steps 2 and 3. For example, '`Is this car red/Toyota?`".

In contrast, VehicleGPT integrates the previous steps internally, operating as a black box since we only interact with it through its API.

First, both VehicleGPT and VehiclePaliGemma were evaluated with Diverse car dataset (see the Dataset section above) using "`Extract all characters from the car plates`" prompt targeting all plates for all cars displayed in the image. The accuracy is calculated as follows: if the model recognizes all plates in the image correctly, the counter that counts the number of correctly recognized images is incremented by one. Otherwise, even if one plate in the image is not properly recognized, the counter is not incremented. The percentage of correctly identified images over the total number of images in the dataset determines the final accuracy. VehicleGPT identified successfully 171 plates among the 176 cars or plates present in 140 images, resulting in a plate-level accuracy of 97.16%. Similarly, VehiclePaliGemma correctly recognized 166 plates, resulting in a plate-level accuracy of 94.32%. The performance gap likely stems from the need to fine-tune VehiclePaliGemma for car and plate detection task, ensuring better localization of cars and plates to minimize missed detections.

Secondly, we evaluated both VehicleGPT and VehiclePaliGemma in several additional scenarios using other prompts, as follows:

- Prompt1: "`Extract all characters from plates of red cars`".
- Prompt2: "`Extract all characters from plates of BMW blue cars`".
- Prompt3: "`Extract all characters from plates of PERODUA cars`".

Both VehicleGPT and PaliGemmaGPT show superior performance and produces accurate outcomes in these scenarios. This experiment underscores their ability to link the description provided in the prompts with the objects' attributes in the image to identify the specific cars' model and/or color, localize the cars and then the plates, and extract the characters from the plates.

The strength of both VehicleGPT and and VehiclePaliGemma lies in its multitasking ability, allowing it to perform several functions simultaneously, including car localization, license plate localization, the car's model recognition, color recognition, and plate recognition. All of these functions can be driven by a prompt provided to the model along with an image. By combining multiple tasks into a single processing pipeline, organizations can save on computational costs and reduce the need for separate models for each task.

The challenging problems that VehicleGPT and VehiclePaliGemma were able to address are:

1. Recognizing all cars' plates in the images, which had several cars and/or plates.
2. Identifying multiple license plates that appeared at various angles and orientations due to the different positions and movements of the cars in real-life image captures.
3. Being robust against the presence of various objects and textures in the background.

Discussion and conclusion

This paper demonstrated the challenging problem of recognizing unclear, distorted license plates with close characters. Various VLMs have been explored to evaluate their OCR capability. We compared these VLMs with other baseline methods. The experimental results showed that the OCR capabilities of VLMs outperformed other OCR baseline methods in terms of plate-level recognition accuracy. It was found that 226 plate images out of 258 images were recognized correctly with a plate accuracy of 87.6% using VehiclePaliGemma, which showed superior performance compared to others. Additionally, the VehiclePaliGemma was able to correctly recognize 1710 characters out of 1751 characters with a character-level accuracy of 97.66%. In summary, While both

VehiclePaliGemma and VehicleGPT offer excellent recognition performance, VehiclePaliGemma distinguishes itself with superior speed, affordability, and efficiency, which opens door to integrate it on edge devices for real-life scenarios. Moreover, we explored the multitasking capability of both "VehicleGPT" and "VehiclePaliGemma" to recognize plates in challenging conditions given an image that has multiple cars with various models and colors, as well as plates in several positions and orientations in cluttered backgrounds.

The images used in this study were collected by Tapway, a Malaysian company, and they encompass various real-world challenges such as noise, low illumination, blurring, weather effects, and closely spaced characters. Acquiring real-world license plate images under these conditions is difficult, which constrained the dataset size to 258 images. While this number may be small, it still provides valuable insights into the model's performance across challenging scenarios.

To the best of our knowledge, no prior research has explored the use of Large Language Models (LLMs) for vehicle plate recognition (for both the small and large models), particularly in challenging conditions. We compared several state-of-the-art LLMs and explored their capability in plate recognition task. We introduce PaliGemmaVehicle, a fine-tuned LLM specifically designed for recognizing Malaysian license plates under complex conditions. Our model was trained on a synthetic dataset of 600 plates and demonstrated superior performance compared to state-of-the-art LLMs, including GPT-4o, Gemini 1.5, Claude 3.5 Sonnet, and Llama 3.2.

Overall, the VLM-based approach surpasses conventional OCR systems by effectively handling complex scenarios typical of real-world license plate datasets, providing robust recognition accuracy and demonstrating enhanced adaptability to diverse visual conditions.

Limitations and future work

This work focused on recognizing close characters in unclear Malaysian license plates. In future work, we plan to extend the proposed solution to recognize more complex plates in other countries. Furthermore, we plan to modify the prompt to address specific instances of plates that require individual handling.

To enhance the proposed solution and ensure no car or plate is missed, future work could involve fine-tuning PaliGemma for car and plate detection tasks. Additionally, the current solution involves multiple steps, including detecting cars and plates, recognizing the color and model of cars, and then identifying the cars. Even though all these steps are completed in under one second, further improvement could be achieved by fine-tuning PaliGemma to directly recognize plates from images containing multiple cars. However, this would require annotating a large dataset to achieve the desired performance. Such tuning should ensure that VLMs are fine tuned on diverse and representative datasets and should consider ethical implications to prevent bias and maintain privacy and security in processing such potentially sensitive information.

Data availability

Data will be available upon request. You can request it from yasir.zaki@nyu.edu.

Received: 14 December 2024; Accepted: 7 July 2025
Published online: 18 July 2025

References

1. Anagnostopoulos, C.-N.E., Anagnostopoulos, I. E., Psoroulas, I. D., Loumos, V. & Kayafas, E. License plate recognition from still images and video sequences: A survey. *IEEE Trans. Intel. Transp. Syst.* **9**, 377–391 (2008).
2. Lubna, Mufti N. & Shah, S. A. A. Automatic number plate recognition: A detailed survey of relevant algorithms. *Sensors* **21**, 3028 (2021).
3. WIJERS, P. J. Implementing automated enforcement in emerging economies. 17th International Road Federation World Meeting (2013).
4. Du, S., Ibrahim, M., Shehata, M. & Badawy, W. Automatic license plate recognition (ALPR): A state-of-the-art review. *IEEE Trans. Circuits Syst. Video Technol.* **23**, 311–325 (2012).
5. Kamaruzaman, M. & Nasir, N. R. M. Parkey: Ticket-less parking system using license plate recognition approach. *J. Phys. Conf. Ser.* **1860**, 012006 (2021).
6. Idrose, H., AlDahoul, N., Karim, H. A., Shahid, R. & Mishra, M. K. An evaluation of various pre-trained optical character recognition models for complex license plates. In Multimedia University Engineering Conference (MECON 2022), 21–27 (Atlantis Press, 2022).
7. Sugiyono, A. Y., Adrio, K., Tanuwijaya, K. & Suryaningrum, K. M. Extracting information from vehicle registration plate using OCR tesseract. *Procedia Comput. Sci.* **227**, 932–938 (2023).
8. Bishop, C. M. & Nasrabadi, N. M. *Pattern Recognition and Machine Learning* Vol. 4 (Springer, Ney York, 2006).
9. Bengio, Y., Ducharme, R. & Vincent, P. A neural probabilistic language model. *Adv. Neural Inf. Process. Syst.* **13**, 1137 (2000).
10. Yang, X. et al. A large language model for electronic health records. *NPJ Digit. Med.* **5**, 194 (2022).
11. Modran, H., Bogdan, I. C., Ursu?iu, D., Samoila, C. & Modran, P. L. LLM intelligent agent tutoring in higher education courses using a rag approach. Preprints (2024).
12. Vaswani, A. et al. Attention is all you need. Advances in neural information processing systems **30** (2017).
13. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).
14. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/ / (2024).
15. Radford, A. et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 8748–8763 (PMLR, 2021).
16. Introducing Gemini 1.5, Google's next-generation AI model. https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024/#architecture (2024).
17. Beyer, L. et al. Paligemma: A versatile 3b VLM for transfer. arXiv preprint arXiv:2407.07726 (2024).
18. Face, H. meta-llama/llama-3.2-11b-vision-instruct. https://huggingface.co/meta-llama/Llama-3.2-11B-Vision-Instruct.
19. Anthropic. Claude 3.5 sonnet. https://www.anthropic.com/news/claude-3-5-sonnet.
20. Liu, H. et al. LLAVA-next: Improved reasoning, OCR, and world knowledge (2024).

21. Dang, P. Multimodal (visual and language) understanding with LLAVA-next. https://rocm.blogs.amd.com/artificial-intelligence/llava-next/README.html (2023).
22. Lin, J. et al. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26689–26699 (2024).
23. Moondream2. https://huggingface.co/vikhyatk/moondream2 (2024).
24. Gunawan, D., Rohimah, W. & Rahmat, R. Automatic number plate recognition for Indonesian license plate by using k-nearest neighbor algorithm. *IOP Conf. Ser. Mater. Sci. Eng.* **648**, 012011 (2019).
25. Mousa, A. Canny edge-detection based vehicle plate recognition. *Int. J. Signal Process. Image Process. Patt. Recogn.* **5**, 1–8 (2012).
26. Aruna, V., Ravi, S. & Suruthi, M. Detection and recognition of license plates using color image processing. In *International Conference on Communications and Cyber Physical Engineering 2018*, 133–140 (Springer, 2024).
27. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems **25** (2012).
28. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
29. Montazzolli, S. & Jung, C. Real-time Brazilian license plate detection and recognition using deep convolutional neural networks. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, 55–62 (IEEE, 2017).
30. He, H., He, S. & Huang, T. License plate recognition based on three different neural networks. In *2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 215–220 (IEEE, 2022).
31. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).
32. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778 (2016).
33. Girshick, R., Donahue, J., Darrell, T. & Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580–587 (2014).
34. Girshick, R. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, 1440–1448 (2015).
35. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems **28** (2015).
36. Saidani, T. & Touati, Y. E. A vehicle plate recognition system based on deep learning algorithms. *Multimedia Tools Appl.* **80**, 36237–36248 (2021).
37. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788 (2016).
38. Redmon, J. & Farhadi, A. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018).
39. Hendryli, J., Herwindiati, D. E. et al. Automatic license plate recognition for parking system using convolutional neural networks. In 2020 *International Conference on Information Management and Technology (ICIMTech)*, 71–74 (IEEE, 2020).
40. Dosovitskiy, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020).
41. Zhang, T. & Jia, W. Automatic license plate recognition using transformer. In *Fourteenth International Conference on Graphics and Image Processing (ICGIP 2022)*, vol. 12705, 129–138 (SPIE, 2023).
42. Abdelhamed, A., Afifi, M. & Go, A. What do you see? Enhancing zero-shot image classification with multimodal large language models. arXiv preprint arXiv:2405.15668 (2024).
43. Ramesh, A. et al. Zero-shot text-to-image generation. In Meila, M. & Zhang, T. (eds.) *Proceedings of the 38th International Conference on Machine Learning, vol. 139 of Proceedings of Machine Learning Research*, 8821–8831 (PMLR, 2021).
44. Antol, S. et al. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, 2425–2433 (2015).
45. Tesseract documentation. https://tesseract-ocr.github.io/ (2022).
46. Easyocr. https://github.com/JaidedAI/EasyOCR (2022).
47. Keras-ocr. https://github.com/faustomorales/keras-ocr (2022).
48. Vehicletrack. https://gotapway.com/solutions/vehicletrack (2024).
49. Gpt-4o: The comprehensive guide and explanation. https://blog.roboflow.com/gpt-4o-vision-use-cases/ (2024).
50. Gemini Team, G. Gemini 1.5 technical report. https://storage.googleapis.com/deepmind-media/gemini/gemini_v1_5_report.pdf (2024).
51. Llava: Large language and vision assistant explained. https://encord.com/blog/llava-large-language-vision-assistant/ (2024).
52. Vision AI for Devs. https://moondream.ai/ (2024).
53. Face, H. meta-llama/llama-3.1-8b-instruct. https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct.
54. Brown, T. et al. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. & Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, 1877–1901 (Curran Associates, Inc., 2020).
55. Ocr using pytesseract and opencv. https://nanonets.com/blog/ocr-with-tesseract/ (2024).
56. Smelyakov, K., Chupryna, A., Darahan, D. & Midina, S. Effectiveness of modern text recognition solutions and tools for common data sources. In COLINS, 154–165 (2021).
57. Vedhaviyassh, D., Sudhan, R., Saranya, G., Safa, M. & Arun, D. Comparative analysis of easyocr and tesseractocr for automatic license plate recognition using deep learning algorithm. In *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, 966–971 (IEEE, 2022).
58. Barozai, D. K. Tesseract OCR: Understanding its features, applications, and limitations. https://www.folio3.ai/blog/tesseract-ocr/.

Author contributions

Conceptualization by N.A., Y.Z.; data curation by N.A., C.H.L., M.K.M.; formal analysis by N.A., Y.Z.; funding acquisition by Y.Z.; investigation by N.A.; methodology by N.A.; project administration by Y.Z.; software by N.A.; validation by N.A.; visualization by N.A., Y.Z.; writing–original draft preparation by N.A., M.J.T.T., R.R.T.; writing, review and editing by N.A., Y.Z., H.A.K.

Competing interests

The authors declare that there are no conflicts of interest relevant to this article.

Additional information

Correspondence and requests for materials should be addressed to Y.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.