machines

MDPI

Article

StairWave Transformer: For Fast Utilization of Recognition Function in Various Unmanned Vehicles

Donggyu Choi [1], Chang-eun Lee [1], Jaeuk Baek [1], Seungwon Do [1], Sungwoo Jun [1], Kwang-yong Kim [1] and Young-guk Ha [2,*]

[1] Defense ICT Convergence Research Section, Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea; dgchoi@etri.re.kr (D.C.)
[2] Department of Computer Science and Engineering, Konkuk University, Seoul 05029, Republic of Korea
* Correspondence: ygha@konkuk.ac.kr

Abstract: Newly introduced vehicles come with various added functions, each time utilizing data from different sensors. One prominent related function is autonomous driving, which is performed in cooperation with multiple sensors. These sensors mainly include image sensors, depth sensors, and infrared detection technology for nighttime use, and they mostly generate data based on image processing methods. In this paper, we propose a model that utilizes a parallel transformer design to gradually reduce the size of input data in a manner similar to a stairway, allowing for the effective use of such data and efficient learning. In contrast to the conventional DETR, this model demonstrates its capability to be trained effectively with smaller datasets and achieves rapid convergence. When it comes to classification, it notably diminishes computational demands, scaling down by approximately 6.75 times in comparison to ViT-Base, all the while maintaining an accuracy margin of within ±3%. Additionally, even in cases where sensor positions may exhibit slight misalignment due to variations in data input for object detection, it manages to yield consistent results, unfazed by the differences in the field of view taken into consideration. The proposed model is named Stairwave and is characterized by a parallel structure that retains a staircase-like form.

Keywords: autonomous driving; infrared image; transformer; object detection; classification

Citation: Choi, D.; Lee, C.-e.; Back, J.; Do, S.; Jun, S.; Kim, K.-y.; Ha, Y.-g. StairWave Transformer: For Fast Utilization of Recognition Function in Various Unmanned Vehicles. *Machines* **2023**, *11*, 1068. https://doi.org/10.3390/machines11121068

Academic Editor: Dan Zhang

Received: 31 October 2023
Revised: 30 November 2023
Accepted: 2 December 2023
Published: 4 December 2023

1. Introduction

The field of autonomous driving has evolved from being a highly challenging domain to becoming an aspect of every day, now commonly integrated as basic assistance functions in commercially available vehicles. The autonomous driving sector, built on rapidly advancing artificial intelligence and sensor technologies, continues to experience sustained growth [1–3]. The most crucial aspect among the fields utilized in autonomous driving technology is artificial intelligence, which can rapidly assess and provide solutions to issues that arise during driving. While a wide range of events can occur during driving, generally, vision-related technologies, which function much like human eyes for assessment, are the most critical [4,5]. They need to learn quickly and provide rapid responses. Training artificial intelligence to make judgments through images requires a substantial amount of data and a significant amount of time. If the performance of such functions is excellent, it often demands high-performance hardware, and the issue of resource-intensive costs has been a long-standing concern [6–8].

Just as depicted, autonomous driving, which used to rely on numerous sensors and data for performance, was challenging to implement in small-scale systems. However, over time, with improvements in hardware performance and streamlining, it is now being employed across diverse platforms [3,9,10]. One of the prominent examples is the autonomous driving algorithm used by Tesla in the United States [11]. In the design of this algorithm, it is explained that it relies solely on camera input data for perception and autonomous driving, just as a person would assess situations during regular driving with their eyes. Vehicles are

not constrained by size when it comes to equipping high-performance hardware. However, in situations with no lighting, such as at night, the accuracy significantly diminishes, and there are limitations in achieving a high level of autonomous driving [12,13]. One of the most readily available small autonomous vehicles is the robotic vacuum cleaner [14–16]. Robotic vacuum cleaners utilize LiDAR sensors to collect data in confined, small-scale environments, navigating obstacles through distance detection using infrared technology and collision recognition with bumpers. In contrast, robots that provide services, such as serving robots that can potentially harm people, need to proactively avoid critical situations and be capable of quickly adapting to various circumstances. However, conventional learning methods such as object recognition and classification aim to enhance accuracy by training on extensive datasets, enabling the recognition of diverse environmental elements within images and reducing computational losses [17–19]. Various attempts are under way to address these normal issues in the introduction of autonomous driving. There have been studies proposing Fear-Neuro-Inspired based reinforcement learning frameworks to induce defensive responses regarding threats or dangers, aiming to address crucial safety issues in driving [20]. Additionally, there have been proposals for a robust decision-making approach aimed at maintaining a single decision rather than continuously changing intentions in the flow of traffic [21].

New models are constantly being introduced in the field of artificial intelligence to optimize and enhance performance, with transformer and multi-modal being the predominant keywords recently observed in the AI domain [22,23]. The adoption of the transformer architecture has moved beyond the traditional convolution structure, introducing a new form of deep learning for both training and inference. This structure was primarily used in NLP (Natural Language Processing) previously. Since the introduction of the transformer model, efforts have been made to utilize the characteristics of this structure to integrate the meaning of multi-modal, enabling the generation of meaningful inference results by utilizing a variety of data in conjunction with images [24–26]. However, achieving high performance demands a significant amount of data, and the drawback is the lengthy training time required until it can infer the correct answer. Vehicles that require human intervention should be produced with a focus on safety and stability, necessitating strong AI-driven autonomous driving capabilities [27–30]. However, unmanned vehicles designed for various environments require a need for quick development and easy adoption.

In this paper, we propose a Stairwave Transformer model structure designed in parallel to reduce the input image size used in operations, similar to a stair-like form, enabling training with multi-sensor data collected at the same time. To efficiently apply the classification and object detection functions, while these two functions have different model structures, the mechanisms related to the implementation were designed in the same way. For classification, there is no separate backbone. Instead, it goes through three stages of reducing the image patch size and a total of eight transformer encoders. It achieves an accuracy within ±3% compared to DEtection TRansformer (DETR) while requiring roughly 6.75 times fewer computations. For object detection, ResNet50 was used as the backbone. The process involves downsizing the image patches twice and performing a total of 6 transformer encoders and decoders. This allows for faster initial learning convergence compared to DETR and effective training with smaller datasets.

2. Related Works

To design a proposed method, we examined the characteristics of representative models for each inference function, namely Vision Transformer (ViT) and DEtection TRansformer (DETR), as well as the foundational model structure, transformer. Furthermore, we confirmed the efficiency related to the utilization of additional data.

2.1. Transformer [22]

Transformer is a machine learning model that was first introduced in a paper by Google in 2017, and it revolutionized the predominant paradigm that primarily used the structure of conventional models based on CNN, which had been in use for a long time [22].

Figure 1 shows a simplified operation of the transformer. It was initially designed to perform NLP tasks, but this architecture has become a foundational model that can be extended to various fields.

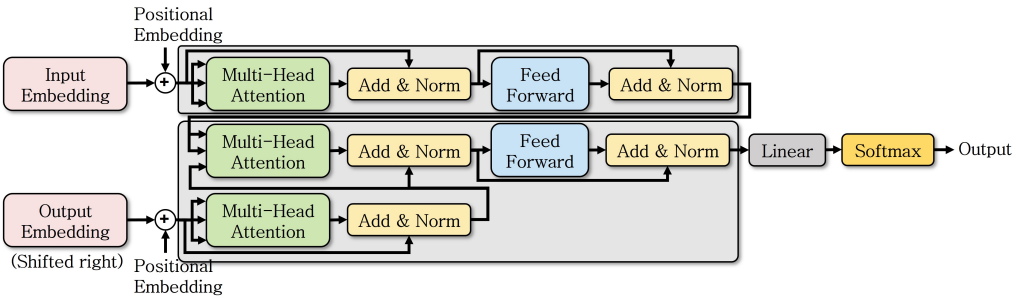


Figure 1. Transformer internal model structure.

Transformer utilizes the tokenizer approach commonly used in NLP to patchify various data for training. Instead of the conventional Recurrent Neural Network (RNN), it is based on the self-attention mechanism. By employing the transformer approach, the sequential processing method, which is a limitation of RNN, is parallelized. Furthermore, it employs multi-head attention to process input information from various perspectives and combines the extracted information. The transformer method processes elements within the input training data simultaneously, resulting in faster processing speeds compared to the RNN. It performs better with a larger amount of training data. Additionally, it can learn the correlations between similar data more clearly through the attention mechanism. However, the computational load increases depending on the length of the data required for processing.

The CNN and transformer employ different approaches for learning and inference, each with its own set of advantages and disadvantages [22,31]. In the case of the CNN, the use of convolution filters allows it to effectively capture the influence of surrounding information, which can have an impact on the results. In contrast, transformer utilizes the complete information of the data it processes, enabling it to obtain information from distant data points. From the perspective of 'inductive bias', a value that is reflected to achieve better performance when new data are introduced in continuous learning, these approaches reflect different scales of influence.

2.2. Vision Transformer (ViT) [32]

While the original transformer model was primarily utilized in the field of NLP, the model designed to extend this to the vision domain is known as the ViT [32].

Figure 2 shows the basic structure of the ViT. The ViT leverages the advantages of the transformer's inductive bias, which results in the model's versatility, to capture and process interacting elements in global image information. In the case of transformers used in NLP, a patchify process is performed to break down sentence structures into patches and use them as input. To adapt this to images, appropriately sized patches are defined and used as input to the transformer encoder. This structure features only an encoder without a decoder.

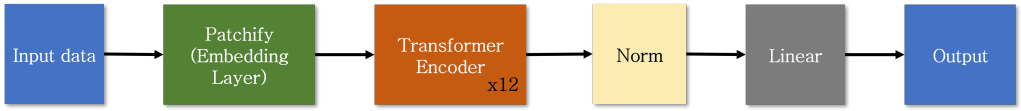


Figure 2. ViT model structure.

From an image learning perspective, the conventional CNN approach has been suitable for resource-constrained environments due to its compact model size and efficient memory utilization. Models utilizing CNN techniques still demonstrate fast processing speeds and decent accuracy on lightweight platforms. In contrast, ViTs have the drawback of larger model sizes and high memory usage. Due to the nature of transformers, they require large datasets to achieve optimal performance. ViTs, for instance, have been trained on datasets containing over 300 million images with more than 37.5 billion labeled data points. Currently, they may not seem suitable for resource-constrained, small-scale platforms. Nevertheless, to harness the advantages of transformer-based models, such as improved utilization of diverse data and overcoming the limitations of the CNN, it is necessary to design and enhance models using transformers.

2.3. DEtection TRansformer (DETR) [33]

The ViT fundamentally performs classification tasks, and a model that applies this to object detection is called DETR [33].

Figure 3 provides a brief overview of the DETR's structure. While it employs the structure of transformer, it strengthens image features by using a CNN-based resnet as its backbone. The output from resnet, along with the positional information of the divided patches, is used as input for the transformer encoder. The output obtained through resnet is similar to the result of transforming the image into 16x16-sized patches, and no separate patch processing is required. In DETR, positional embedding is added to the query after data input to the transformer encoder, without including it when patches are input. Unlike conventional object detection, DETR does not output values sequentially but produces results all at once.

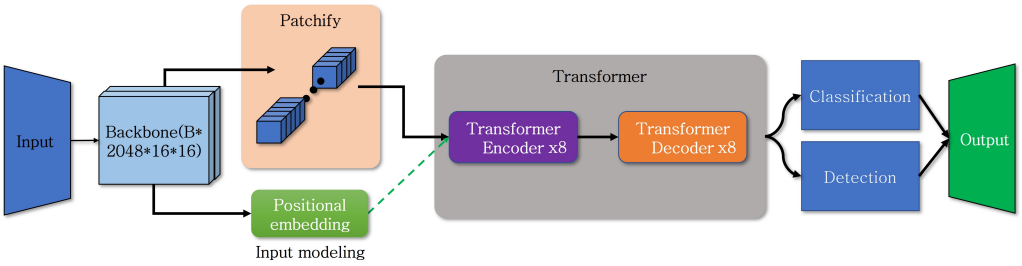


Figure 3. DETR model structure.

DETR has the advantage of recognizing large objects within an image effectively, as it utilizes the entire image's information, in contrast to models designed with a CNN. Moreover, when trained with an end-to-end model structure, it demonstrates performance similar to the Faster R-CNN. However, due to the utilization of the transformer structure, it requires longer training times, and its inference performance for small objects is suboptimal. Research aimed at addressing the shortcomings arising from the use of the transformer architecture continues to be ongoing through various methods [34–36].

2.4. Information Change Due to the Use of Additional Data

When additional data are used for training and inference with image data, it allows for obtaining a greater amount of information.

$$H = -\sum_{t=0}^{255} p_i \log_2 p_i \tag{1}$$

Equation (1) represents the value of information or the entropy calculation corresponding to a single pixel in the local context of an image [37,38]. In Equation (1), 'p_i' denotes the probability values concerning the bray scale obtained from the normalized histogram of the image.

Figure 4 is the image used to examine the value of information of Equation (1). According to the paper, when calculating the entropy of the original image, the value of information is lower for infrared images compared to RGB because of the fewer channels [37]. However, when combined, it produces a higher value. A higher value indicates that it contains more information, and typically, color data contains more features. The formula's outcome demonstrates that merely by utilizing additional data alongside the existing data, the amount of information obtained increases.

RGB(3ch 8bit) IR(1ch 16bit)

Figure 4. RGB and IR sample images for assessing image information.

3. Design

This paper presents a model designed with two structures that perform classification and object detection based on a mechanism of gradually reducing input data size. It also proposes methods for utilizing additional data in object detection. In the case of classification, the transformer's structure involves a significant computational load and lengthy training times, which led to its use in the fundamental model design for image learning. The mechanism employed in classification was later extended to object detection, and the model was designed to be applied in various environments by utilizing additional sensor image data.

Figure 5 shows the model structure for the classification function using the proposed design approach. This structure depicts the entire transformer model augmented with the process of convolution and can be characterized by two main features. It can be divided into the part that reinforces key features through convolution before executing the transformer encoder and the part that distributes them based on input data size, performing the transformer encoder in parallel. In the conventional ViT structure, the original image size is transformed only to the input size. However, in the proposed method, the grid down convolution block (GDC block) is applied to further reduce the image size while making the features within the image more distinct. The structure consists of two 3×3 convolution layers for generating general features, two convolution layers for reducing the input size for computation, and two linear convolution layers. The most computationally intensive part in the basic transformer structure is 'patchify', which divides the image into predefined patch sizes. In the ViT, after patchify, the transformer encoder is performed with the same input size. For the base model, this operation is performed a minimum of 12 times, consuming a significant amount of resources. The proposed method involves performing the GDC block a total of three times, resulting in four different input data sizes, each of which is processed twice by the transformer encoder. In this case, the total number of transformer encoder executions is reduced from 12 to 8, and the input images used for patchify are in four different sizes, significantly reducing resource usage. Furthermore, due to the smaller input size for the transformer, it enables efficient and rapid learning and inference based on various output data.

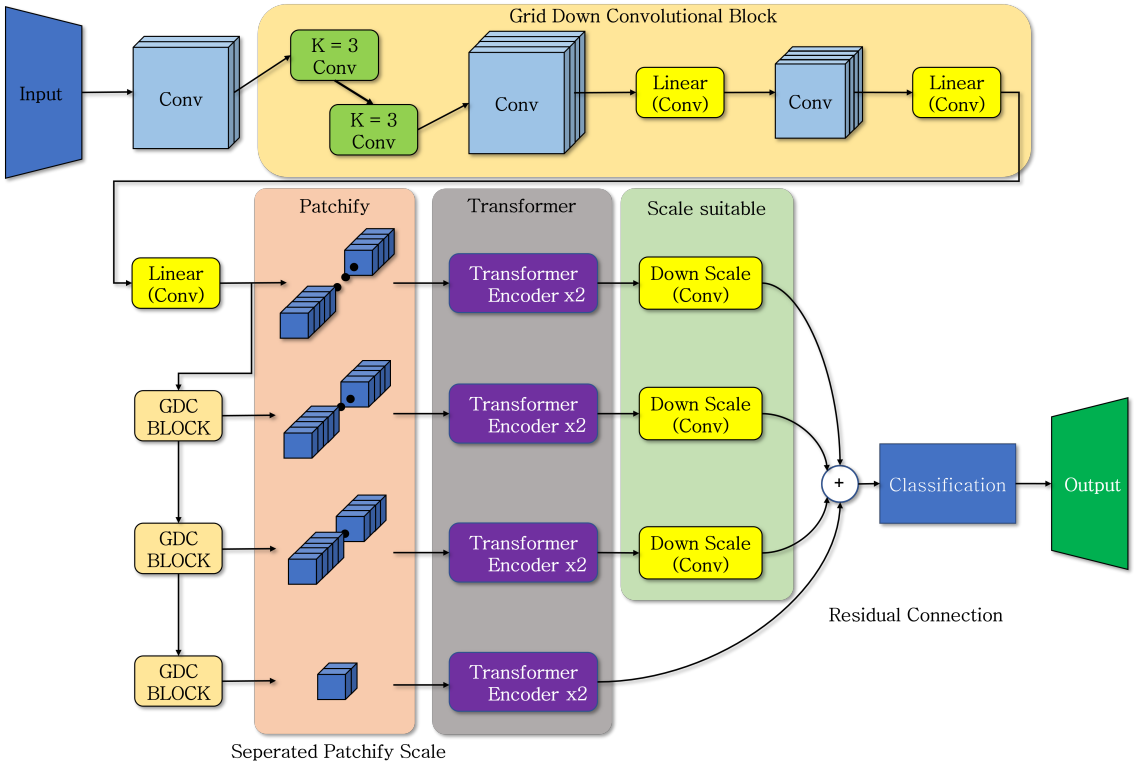


Figure 5. Proposed classification model structure.

$$N_{base} = \sum_{k=1}^{12} HW/P^2 \tag{2}$$

$$N_{proposal} = \sum_{k=1}^{4} 2(\frac{HW}{2^k}/P^2) \tag{3}$$

Equations (2) and (3) calculate the vectors, in other words, the number of patches used in the execution of the encoder layer for both the ViT and the proposed method. H and W represent the width and height of the input data, while P represents the patch size. Technically, the computation should reduce by half with each iteration. However, the input data's size is larger, specifically 256 × 256, compared to ViT-Base, which has an input size of 224 × 224. Utilizing a slightly larger resolution of the input data is aimed at preventing feature loss in the final GDC block, where the data become too small. The data used for transformer encoder execution and the results of size are based on the four blocks. The data output in different sizes is passed through the down scaling convolution layer to be resized to the same size as the smallest patch. Then, a residual connection is applied, and a multi-layer perceptron is used for classification based on the number of classes to present the results.

Figure 6 shows the structure of the object detection function model designed using the approach applied in the previously described classification. The input data utilize resnet as the backbone to highlight features within the image. The input data that have passed through the backbone results in 2048 channels with a size of 16 × 16, which is consistent with the original DETR. To use the output as input data for the transformer, an input modeling process is performed, and additionally, patchify is executed in two different sizes. When reducing the patch size, reducing it to 1 × 1 or a similarly small size completely eliminates object feature information. Therefore, the patch size is reduced to 8 × 8 and 4 × 4. The proposed method differs in terms of transformer layer execution, as input data of each size do not pass through a single transformer layer until the end. Instead, input data of different sizes pass through separate transformer layers. The transformer layer is

executed a total of six times, with two executions for each size. The results obtained after passing through the layers then pass through a residual connection layer and undergo object recognition and classification inference processes. The model designed in this way exhibits a parallel structure, resembling a staircase with steps gradually, ascending in a sequential process.

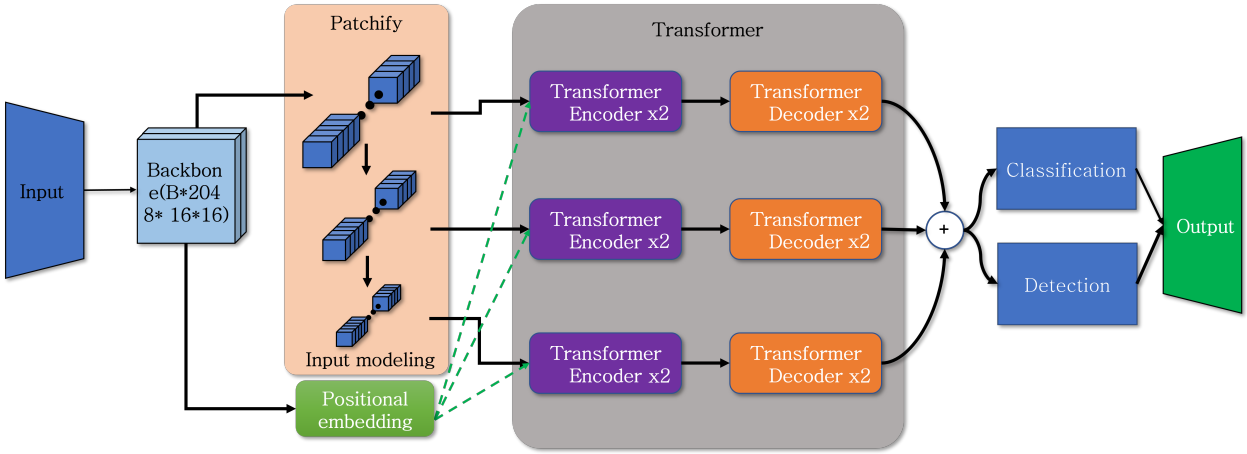


Figure 6. Proposed object detection model structure.

Figure 7 shows the location where data are modified to utilize additional data in the transformer layer for the preceding object detection process. During the execution of the transformer encoder, values corresponding to query, key, and value are utilized. Additional data, transformed to match the format of the query, is added to the key's values, along with positional embedding values that account for the position of each patch. To generate and incorporate the result into the value of the transformer encoder, the answer value from the query is utilized, along with the key values as hints to find answers, including additional data. In the decoder, the final output is produced using the value.

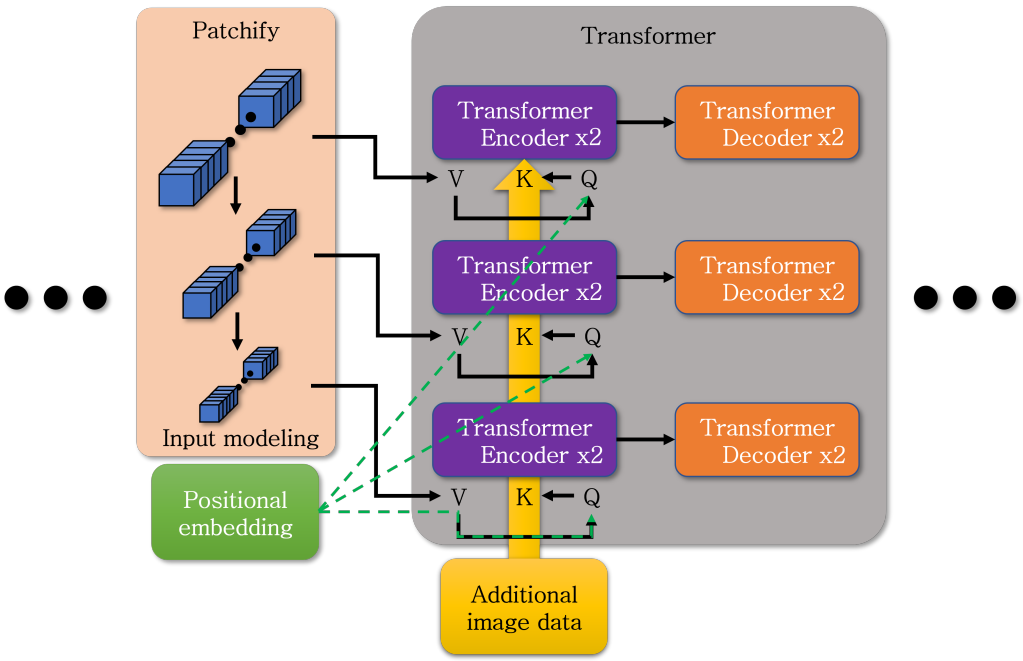


Figure 7. Approach for incorporating additional image data into object detection model.

$$AI = AdditionalImage, \quad \hat{K} = K \oplus AI, \quad Attention(Q, \hat{K}, V) = softmax(\frac{Q\hat{K}^t}{\sqrt{d^{head}}})V \quad (4)$$

Equation (4) shows the addition of extra data to the attention mechanism of the encoder. K added the value of input obtained from resnet processing and positional embeddings to the 'Q'. However, 'K *hat*' represents the value obtained by embedding additional image data within the 3 channels during data input.

4. Results

To verify the performance of the model proposed in this paper, we conducted tests and examined the results for each function.

Tables 1 and 2 show the datasets and performance specifications. The Place365 dataset was used for classification, and the BDD-100K and FLIR datasets were employed for object detection [19,39,40]. In addition, we utilized custom indoor datasets for testing. The Place365 dataset includes 1,803,460 images, each with label data for 365 different classes. The BDD-100K dataset comprises approximately 3,300,000 Bbox Label data for 79,863 images spanning 8 classes. The FLIR dataset consists of some continuous video data, capturing 3748 images with both RGB and thermal views from the same perspective. It is categorized into 10 classes and includes 84,786 bounding box Label data. Among custom indoor datasets, the one used for classification comprises 9338 images with 8 distinct classes. The dataset used for object detection encompasses 10,195 images with 33,850 bounding box Label data into 12 classes. The computational specifications used in the experiments include an Intel Xeon Silver 4210R CPU and an RTX A6000 48 GB GPU, along with 192 GB of RAM. The operating system used is Ubuntu 18.04 LTS 64-bit. The programming languages employed are python 3.8.10 and pyTorch 1.12.0.

Table 1. Specifications.

Usage	Dataset	Images	Classes	Labels (Bbox)
Classification	Place365	1,803,460	365	-
	Custom	9338	8	-
Object detection	BDD-100K	79,863	8	3,300,000
	FLIR	3748	10	84,768
	Custom	10,195	12	33,850

Table 2. System specifications.

	Specification
CPU	Intel Xeon Silver 4210R
GPU	RTX A6000 48 GB GPU
RAM	192 GB
OS	Ubuntu 18.04 LTS 64-bit
Language	Python 3.8.10

Figure 8 shows sample data from the five datasets used. The BDD-100K and FLIR datasets consist of image data acquired from the perspective of vehicle operation.

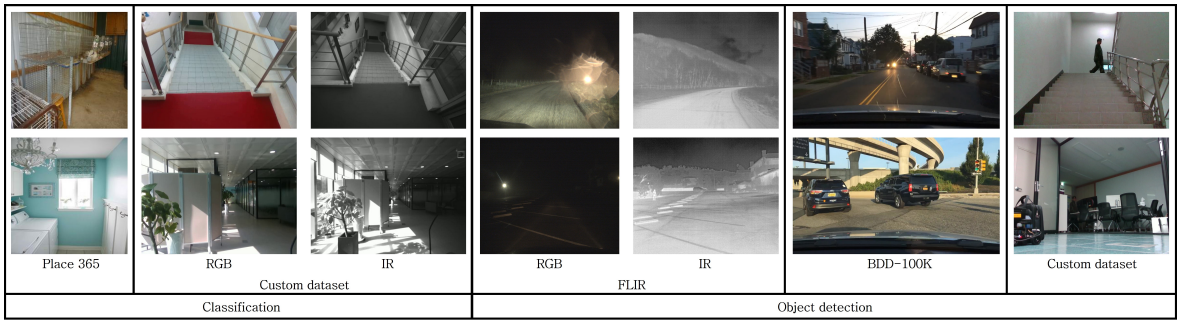


Figure 8. Sample images from the dataset used.

Table 3 shows a comparison of the structure and depth between the existing and designed models. To achieve model lightweighting for classification tasks in the proposed approach, the number of channels was reduced by half or less, and the depth of the transformer encoder was reduced by 4 compared to ViT-Base. As a result, the number of parameters could be reduced by approximately 6.75 times. After the introduction of the ViT, models such as ConViT and Swin Transformer emerged, based on ViT architecture. However, these were not designed with a focus on lightweight structures to increase accuracy. These models also exhibit parameter counts exceeding 80 million [41,42]. The object detection model, applying the proposed approach, reduces the patch size by 2 times for each operation to facilitate faster training. As the data are downsampled n times, the depth of the transformer increases by a factor of 2. This downsizing, although it slightly increases the number of parameters, is undertaken to achieve rapid training convergence.

Table 3. Comparison of features between the proposed model and the base model.

	Model	Width (Channels)	Depth	Input Size	Patch Size	Parameters
Previous	ViT (Classification)	768	Enc.12	224	10	86 M
	ConViT (Classification)	768	GPSA.10, SA.2	224	16	86 M
	Swin (Classification)	1024	SWTB. (2/2/18/2)	224	4	88 M
	DETR (Detection)	2048	Enc.6, Dec.6	max.800 × 1333	16	41.50 M
Proposal	Classification	300	Enc.8	256	10	12.58 M
	Detection	2048	n (Enc.2, Dec.2)	256	16/8/4	52.98 M

Table 4 shows the results of lightweighting the ViT using the proposed method. For the ViT, after training up to 50 epochs, the accuracy was 26.61%. In contrast, the proposed method achieved an accuracy of 33.40% as early as 19 epochs. On the custom dataset, both models achieved over 95% accuracy after the same 50 epochs, with an error of approximately ±3%. The proposed model exhibited a training speed at least three times faster.

Table 4. Performance comparison of classification functionality.

Dataset	Model	Epoch	Accuracy	Batch Size	Learning Time (Mean of Step per s)
Custom dataset	Proposal	50	96.25	20	0.037
	ViT-Base/16	50	98.39	20	0.2
Place365 dataset	Proposal	19	33.40	360	0.328
	ViT-Base/16	50	26.61	360	0.768

Figure 9 shows the loss graphs during the training of DETR and the proposed method. For DETR, there is a tendency for rapid learning from a certain epoch, but it takes a considerable amount of time to converge. The training speed for both DETR and the proposed method is approximately 7 s per step, and the convergence speed for recognition is also fast. This is reflected in the training results and is confirmed. Although the proposed method has more parameters for computation, it gains an advantage in training speed due to the use of smaller input sizes.

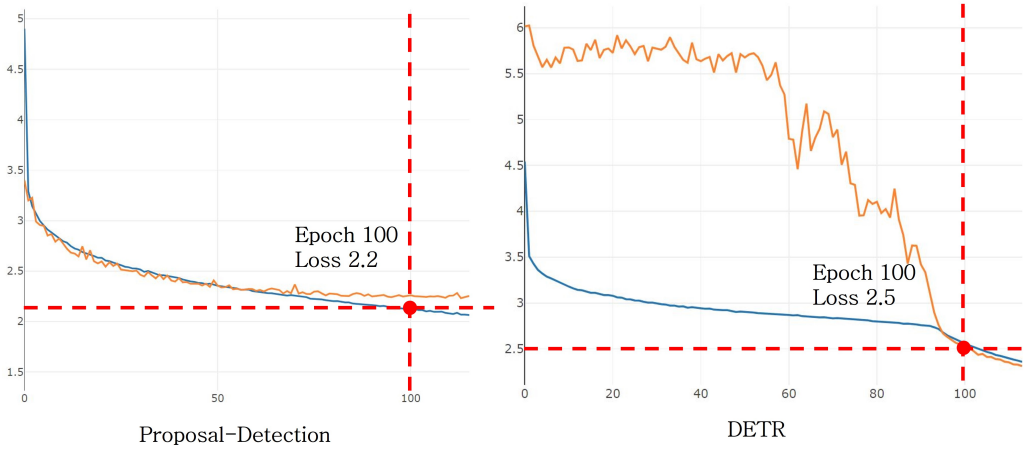


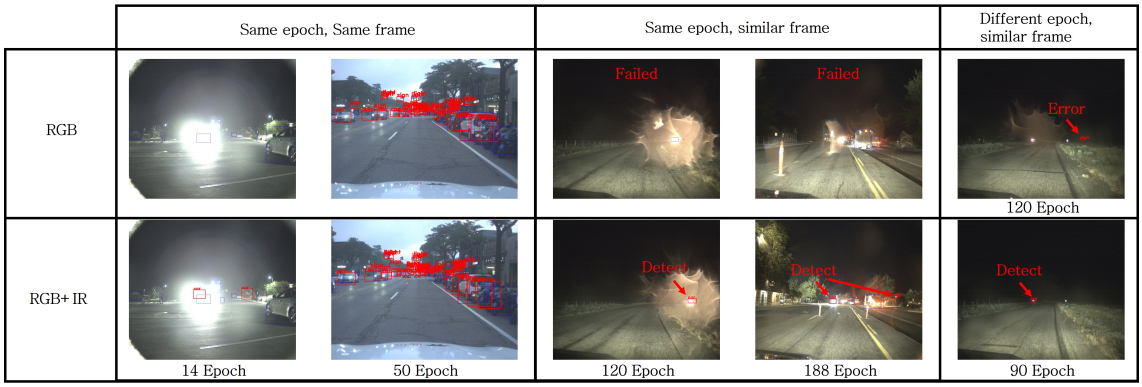
Figure 9. Loss rate for each object detection model.

Figure 10 shows the inference results of object detection designed through DETR and the proposed method at the same epoch. As evident from the results, the training convergence speed of the proposed method is fast. This is reflected in the inference results, as it begins detecting objects in the similar positions not long after the first epoch. Even up to 90 epochs, DETR did not appear to learn much about the input data. While it exhibited some level of recognition, the model utilizing our proposed method consistently demonstrated significantly higher accuracy at the same 100 epochs. In the case of training on a custom dataset, even with a small dataset of fewer than 10,000 images, we observed promising detection results starting from epoch 189. However, in the case of DETR, even after training for 500 epochs, it fails to detect objects.

Figure 10. Results across training epochs.

In Figure 11, Figure 11a represents RGB images and infrared images captured at the same time, while Figure 11b illustrates the results of training with single RGB image data using the proposed method and the results of training with both RGB image data and infrared image data. In Figure 11a, for objects that are not visible in the original RGB images due to direct sunlight, their shapes become visible when captured with an infrared camera. The proposed model, designed to utilize such data additionally, can recognize objects on the same RGB images as in Figure 11a, as seen in the results of Figure 11b. Furthermore, when additional similar images are used, it exhibits robust recognition results, even in the presence of lighting elements that may interfere with recognition, outperforming the model trained solely on RGB images. Furthermore, the training speed remains unaffected by the addition of extra information about the images, as these data are incorporated into the key in a manner that does not slow down the computation speed, except when loading the data for training.

(a)



(b)

Figure 11. Results of single use of RGB images and combined learning of RGB and additional sensor data. (**a**) original images, (**b**) processed images.

Figure 12 shows the performance metrics of the executed dataset and dataset-specific accuracy for each task. Under the dataset name, the number of images in the dataset is indicated. In the case of classification, the metrics are consistent with the previous explanation. However, examining the results of object detection, it is evident that utilizing images with additional channels performs better than using only RGB data. For the BDD-100K dataset, due to the need to detect small object sizes, it exhibits results in tracking similar positions, but the mAP metric is measured relatively low. For the custom dataset labeled for object recognition, the DETR model did not recognize objects.

Figure 12. Performance results graph of executed models for each dataset.

5. Conclusions

In this paper, we propose an effective model structure for rapidly introducing the utilization of artificial intelligence functions through multi-sensor inputs in small-scale systems. This technology is expanding into various fields of autonomous driving. We employ the transformer architecture, which has gained prominence recently. To address the drawbacks of the transformer, such as training speed and high computational load, we employ a parallel layer arrangement passing through different transformer layers for varying data sizes while gradually reducing the input image data size. We also reduce the number of transformer layers compared to the conventional approach. As a result, in the classification function, our proposed ViT exhibits a computational load that is approximately 6.75 times less than that of the basic ViT. It maintains similar or improved accuracy, and its training speed is at least three times faster, making it suitable for straightforward training and small-scale system applications. In the object detection function, our proposed model's computational load is comparable to that of DETR, but it offers rapid training and subsequent inference accuracy convergence. Notably, no separate pre-training is required to achieve these results. It does not unconditionally demand extensive data and can effectively train on small-scale datasets. If you want to further improve object recognition accuracy, you can utilize larger-scale datasets. Our modified model, taking advantage of the characteristics of the transformer architecture and using additional sensor data, demonstrates improved object detection results even in images with varying lighting conditions, interference, or nighttime scenarios when compared to the results of inference using only RGB data. This shows the model's adaptability to diverse environmental data.

The used backbone, resnet, accounts for a substantial portion, approximately half, of the overall computational load. Therefore, it is possible to improve processing speed by either designing an effective backbone for obtaining features from input data or utilizing a lightweight alternative. In the case of additional data like infrared images, constructing separate layers for feature extraction and processing to enhance results using this sensor in low-light conditions could lead to accuracy improvements.

Author Contributions: Conceptualization, writing—original draft preparation, software, visualization, D.C.; project administration, funding acquisition, C.-e.L.; methodology, J.B.; formal analysis, S.D.; data curation, S.J.; investigation, K.-y.K.; validation, supervision, writing—review and editing, Y.-g.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported by Korea Research Institute for Defense Technology planning and advancement (KRIT) grant funded by Korea government DAPA (Defense Acquisition Program Administration) (No. KRIT-CT-22-006-002, Development of the situation/environment recognition technology for micro-swarm robot).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Agrawal, A.; Gans, J.; Goldfarb, A. What to expect from artificial intelligence. *MIT Sloan Manag. Rev.* **2017**, *58*, 23.
2. Muhammad, K.; Ullah, A.; Lloret, J.; Ser, J.D.; de Albuquerque, V.H.C. Deep Learning for Safe Autonomous Driving: Current Challenges and Future Directions. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 4316–4336. [CrossRef]
3. Grigorescu, S.M.; Trasnea, B.; Cocias, T.T.; Macesanu, G. A Survey of Deep Learning Techniques for Autonomous Driving. *J. Field Robot.* **2020**, *37*, 362–386. [CrossRef]
4. Sidhwani, S.; Malkotiya, M.; Korde, N.; Unde, S.; Salunke, M. Autonomous Driving: Using a Vision based Approach. *Int. J. Comput. Appl.* **2014**, *92*, 20–24. [CrossRef]
5. Kanchana, B.; Peiris, R.; Perera, D.; Jayasinghe, D.; Kasthurirathna, D. Computer Vision for Autonomous Driving. In Proceedings of the 2021 3rd International Conference on Advancements in Computing (ICAC), Colombo, Sri Lanka, 9–11 December 2021; pp. 175–180. [CrossRef]
6. García-Martín, E.; Rodrigues, C.F.; Riley, G.; Grahn, H. Estimation of energy consumption in machine learning. *J. Parallel Distrib. Comput.* **2019**, *134*, 75–88. [CrossRef]
7. Desislavov, R.; Martínez-Plumed, F.; Hernández-Orallo, J. Trends in AI inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. *Sustain. Comput. Inform. Syst.* **2023**, *38*, 100857. [CrossRef]
8. Potok, T.E.; Schuman, C.; Young, S.; Patton, R.; Spedalieri, F.; Liu, J.; Yao, K.T.; Rose, G.; Chakma, G. A study of complex deep learning networks on high-performance, neuromorphic, and quantum computers. *ACM J. Emerg. Technol. Comput. Syst. (JETC)* **2018**, *14*, 1–21. [CrossRef]
9. Chishiro, H.; Suito, K.; Ito, T.; Maeda, S.; Azumi, T.; Funaoka, K.; Kato, S. Towards heterogeneous computing platforms for autonomous driving. In Proceedings of the 2019 IEEE International Conference on Embedded Software and Systems (ICESS), Las Vegas, NV, USA, 2–3 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–8.
10. Brown, N.E.; Rojas, J.F.; Goberville, N.A.; Alzubi, H.; AlRousan, Q.; Wang, C.; Huff, S.; Rios-Torres, J.; Ekti, A.R.; LaClair, T.J.; et al. Development of an energy efficient and cost effective autonomous vehicle research platform. *Sensors* **2022**, *22*, 5999. [CrossRef]
11. Tesla. Autopilot. Available online: https://www.tesla.com/autopilot (accessed on 10 September 2023).
12. Berecz, C.E.; Kiss, G. Dangers in autonomous vehicles. In Proceedings of the 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 21–22 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 000263–000268.
13. Coicheci, S.; Filip, I. Self-driving vehicles: Current status of development and technical challenges to overcome. In Proceedings of the 2020 IEEE 14th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 21–23 May 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 000255–000260.
14. Hendriks, B.; Meerbeek, B.; Boess, S.; Pauws, S.; Sonneveld, M. Robot vacuum cleaner personality and behavior. *Int. J. Soc. Robot.* **2011**, *3*, 187–195. [CrossRef]
15. Kang, M.C.; Kim, K.S.; Noh, D.K.; Han, J.W.; Ko, S.J. A robust obstacle detection method for robotic vacuum cleaners. *IEEE Trans. Consum. Electron.* **2014**, *60*, 587–595. [CrossRef]
16. Asafa, T.; Afonja, T.; Olaniyan, E.; Alade, H. Development of a vacuum cleaner robot. *Alex. Eng. J.* **2018**, *57*, 2911–2920. [CrossRef]
17. Feng, D.; Haase-Schütz, C.; Rosenbaum, L.; Hertlein, H.; Glaeser, C.; Timm, F.; Wiesbeck, W.; Dietmayer, K. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1341–1360. [CrossRef]
18. Rashed, H.; Mohamed, E.; Sistu, G.; Kumar, V.R.; Eising, C.; El-Sallab, A.; Yogamani, S. Generalized object detection on fisheye cameras for autonomous driving: Dataset, representations and baseline. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 2272–2280.
19. Yu, F.; Xian, W.; Chen, Y.; Liu, F.; Liao, M.; Madhavan, V.; Darrell, T. BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling. *arXiv* **2018**, arXiv:1805.04687.
20. He, X.; Wu, J.; Huang, Z.; Hu, Z.; Wang, J.; Sangiovanni-Vincentelli, A.; Lv, C. Fear-Neuro-Inspired Reinforcement Learning for Safe Autonomous Driving. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, 1–13. [CrossRef]
21. He, X.; Lou, B.; Yang, H.; Lv, C. Robust Decision Making for Autonomous Vehicles at Highway On-Ramps: A Constrained Adversarial Reinforcement Learning Approach. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 4103–4113. [CrossRef]
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
23. Ngiam, J.; Khosla, A.; Kim, M.; Nam, J.; Lee, H.; Ng, A.Y. Multimodal Deep Learning. In Proceedings of the 28th International Conference on International Conference on Machine Learning, Washington, DC, USA, 28 June–2 July 2011; Omnipress: Madison, WI, USA, 2011; pp. 689–696.
24. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models from Natural Language Supervision. In Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021.
25. Xu, P.; Zhu, X.; Clifton, D.A. Multimodal Learning with Transformers: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 12113–12132. [CrossRef] [PubMed]

26. Rahman, W.; Hasan, M.K.; Lee, S.; Zadeh, A.; Mao, C.; Morency, L.P.; Hoque, E. Integrating multimodal information in large pretrained transformers. *Proc. Conf. Assoc. Comput. Linguist. Meet.* **2020**, *2020*, 2359–2369. [PubMed]
27. Fu, Y.; Li, C.; Yu, F.R.; Luan, T.H.; Zhang, Y. A Survey of Driving Safety with Sensing, Vehicular Communications, and Artificial Intelligence-Based Collision Avoidance. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6142–6163. [CrossRef]
28. Abbasi, S.; Rahmani, A.M. Artificial intelligence and software modeling approaches in autonomous vehicles for safety management: A systematic review. *Information* **2023**, *14*, 555. [CrossRef]
29. Fernandez-Llorca, D.; Gómez, E. Trustworthy artificial intelligence requirements in the autonomous driving domain. *Computer* **2023**, *56*, 29–39. [CrossRef]
30. Parekh, D.; Poddar, N.; Rajpurkar, A.; Chahal, M.; Kumar, N.; Joshi, G.P.; Cho, W. A review on autonomous vehicles: Progress, methods and challenges. *Electronics* **2022**, *11*, 2162. [CrossRef]
31. Arkin, E.; Yadikar, N.; Xu, X.; Aysa, A.; Ubul, K. A survey: Object detection methods from CNN to transformer. *Multimed. Tools Appl.* **2023**, *82*, 21353–21383. [CrossRef]
32. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
33. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
34. Li, Y.; Mao, H.; Girshick, R.; He, K. Exploring plain vision transformer backbones for object detection. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Cham, Switzerland, 2022; pp. 280–296.
35. Wang, Y.; Zhang, X.; Yang, T.; Sun, J. Anchor detr: Query design for transformer-based detector. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 22 February–1 March 2022; Volume 36, pp. 2567–2575.
36. Zhang, Z.; Lu, X.; Cao, G.; Yang, Y.; Jiao, L.; Liu, F. ViT-YOLO: Transformer-based YOLO for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtually, 11–17 October 2021; pp. 2799–2808.
37. Choi, D.; Do, S.; Lee, C.-e. A Study on the Training Methodology of Combining Infrared Image Data for Improving Place Classification Accuracy of Military Robots. *J. Korea Robot. Soc.* **2023**, *18*, 293–298. [CrossRef]
38. Dey, S. *Hands-On Image Processing with Python*; O'Reilly Media: Sebastopol, CA, USA, 2018.
39. Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; Torralba, A. Places: A 10 million Image Database for Scene Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1452–1464. [CrossRef]
40. Teledyne FLIR. FREE Teledyne FLIR Thermal Dataset for Algorithm Training. Available online: https://www.flir.com/oem/adas/adas-dataset-form/ (accessed on 5 August 2023).
41. d'Ascoli, S.; Touvron, H.; Leavitt, M.L.; Morcos, A.S.; Biroli, G.; Sagun, L. Convit: Improving vision transformers with soft convolutional inductive biases. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 2286–2296.
42. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtually, 11–17 October 2021.