applied sciences

MDPI

Article

Fine-Grained Image Recognition by Means of Integrating Transformer Encoder Blocks in a Robust Single-Stage Object Detector

Usman Ali [1], Seungmin Oh [1], Tai-Won Um [2,*], Minsoo Hann [3] and Jinsul Kim [1,*]

[1] ICT Convergence System Engineering Department, Chonnam National University, Gwangju 61186, Republic of Korea; usman4293@gmail.com (U.A.); osm5252kr@gmail.com (S.O.)
[2] Graduate School of Data Science, Chonnam National University, Gwangju 61186, Republic of Korea
[3] Astana IT University, Astana 010000, Kazakhstan; m.hahn@astanait.edu.kz
* Correspondence: stwum@chonnam.ac.kr (T.-W.U.); jsworld@jnu.ac.kr (J.K.)

Abstract: Fine-grained image classification remains an ongoing challenge in the computer vision field, which is particularly intended to identify objects within sub-categories. It is a difficult task since there is both minimal and substantial intra-class variance. Current methods address the issue through first locating selective regions with region proposal networks (RPNs), object localization, or part localization, followed by implementing a CNN network or SVM classifier to those selective regions. This approach, however, makes the process simple via implementing a single-stage end-to-end feature encoded with a localization method, which leads to improved feature representations of individual tokens/regions through integrating the transformer encoder blocks into the Yolov5 backbone structure. These transformer encoder blocks, with their self-attention mechanism, effectively capture global dependencies and enable the model to learn relationships between distant regions. This improves the model's ability to understand context and capture long-range spatial relationships in an image. We also replaced the Yolov5 detection heads with three transformer heads at the output for object recognition using the discriminative and informative feature maps from transformer encoder blocks. We established the potential of the single-stage detector for the fine-grained image recognition task, achieving state-of-the-art 93.4% accuracy, as well as outperforming existing one-stage recognition models. The effectiveness of our approach is assessed using the Stanford car dataset, which includes 16,185 images of 196 different classes of vehicles with significantly identical visual appearances.

Keywords: fine-grained image recognition; Yolov5; transformer encoder block; attention mechanism

Citation: Ali, U.; Oh, S.; Um, T.-W.; Hann, M.; Kim, J. Fine-Grained Image Recognition by Means of Integrating Transformer Encoder Blocks in a Robust Single-Stage Object Detector. *Appl. Sci.* **2023**, *13*, 7589. https://doi.org/10.3390/app13137589

Academic Editor: Zhengjun Liu

Received: 31 May 2023
Revised: 16 June 2023
Accepted: 26 June 2023
Published: 27 June 2023

1. Introduction

Among the most significant challenges in computer vision is fine-grained image recognition, which seeks to distinguish objects from different sub-categories of a particular super-category, for instance, different consumer product categories, vehicle models, bird species, etc. In computer vision, there are numerous fine-grained image recognition applications, including fine-grained image retrieval [1], visual-based recommended systems [2,3], picture generation [4], visual search system [5], and image labelling [6]. Therefore, fine-grained image recognition is a key research topic as well as an actively emerging area of image recognition. Even though networks based on deep learning are capable of extracting important features [7], particularly CNNs [5,8], fine-grained classification is still a difficult task that needs learning to differentiate fine image features. As a result, there has always been a spotlight on learning desired features regarding both fine and discriminating information.

Present fine-grained image recognition approaches can usually be sorted as weakly supervised and strongly supervised. Weakly supervised methods gather specific local areas for part localization using just image labels, whereas strongly supervised methods train a network using extra information such as bounding boxes, image labels, and manual annotation [9–15]. In weakly supervised learning, attention-based approaches are becoming more prevalent in recent times given the ability to perform end-to-end training without additional information. Convolutional neural networks are used in attention-based approaches to construct a local sub-network that gathers important parts of the image. After that, an additional sub-network is utilized to achieve recognition at the output. These methods, however, come with certain acknowledged drawbacks. The amount of object parts must be addressed; for example, the object parts are limited and predefined, which restricts the model's efficiency and adaptability. Furthermore, constructing and training sub-networks to handle every attention element in an object is unreliable, resulting in bottlenecks within the structure. Additionally, local regions could be concatenated but cannot affect the connection between several local regions from a global perspective, which is also extremely important for fine-grained image recognition. Such restrictions need the development of a reliable model capable of extracting unrestricted main features, such as coarse-grained along with fine-grained (attention) features in a relational manner.

The vision transformer [16] recently gained incredible results in the recognition task, proving that using a simple transformer aligned to a series of image patches is capable of capturing the relevant regions because of its inherent attention mechanism. A number of expanded research projects targeting related tasks, including semantic segmentation [17,18] and object detection [19], demonstrated its capacity to extract local features as well as global features. The transformer's capabilities make it suitable for fine-grained image recognition, considering the initial distant receptive field [16], which allows it to track down minimal differences and associated spatial relationships within the initial layers. Convolutional neural networks, on the other hand, primarily leverage the image localization feature and simply locate weaker distant relationships in highly dense layers. Moreover, minor differences among fine-grained categories appear only in particular regions; it is unsuitable to construct a filter that notices minute differences across every region of an image.

The main idea of our research is to investigate the fusion of the vision transformer with the one-stage object detector and their performance in fine-grained image recognition, as there are few studies proving the viability of one-stage object detectors in the fine-grained recognition problem. The aim of our study is to exploit the vision transformer's ability to learn more discriminative and informative features, which is the most important factor considered for the fine-grained recognition problem, as well as to utilize the inference speed of the one-stage object detector simultaneously. Our method integrates transformer encoder blocks with CSP-Darknet53 [20], which results in expanding the receptive field to forecast various scale features through considering the object's local and global information. We swapped several CSP bottleneck blocks with transformer encoder blocks, and after comparing the bottleneck block with the transformer encoder block, we anticipated that the transformer encoder block accumulates both global and local contextual details. Our model learned more discriminative and informative features, leading to improved performance in downstream tasks, and also enhanced feature representations, which are beneficial for fine-grained image classification, where capturing detailed visual patterns is crucial. We utilized some recent computer vision methods, such as the transformer encoder block, multi-stage feature fusion, and various training approaches. To summarize, we have contributed some important and notable improvements to fine-grained image recognition:

- To the best of our understanding, our study is the first to demonstrate the capability of transformer encoder blocks using a one-stage object detector with respect to the fine-grained image recognition task.

- We proposed a single-stage fine-grained model to improve efficiency and minimize the level of complexity, compared to existing two-stage models for the fine-grained recognition problem.
- We introduced the transformer encoder blocks in the backbone of Yolov5 to capture detailed visual patterns and feature representations.
- We replaced the Yolov5 detection heads with three transformer heads at the output to detect discriminative fused features extracted using the transformer encoder blocks.

To validate the algorithm's performance, comparison studies were conducted, and the empirical results indicate that the model is capable of recognizing sub-classes with high precision and accuracy.

The rest of the article is structured as follows: Section 2 discusses related work in detail, providing an overview of the existing two-stage and one-stage methods to overcome the fine-grained recognition problem. Section 3 first introduces the existing Yolov5 backbone structure, subsequently modifies the backbone using transformer encoder blocks, and finally compares the proposed model to the existing model. Section 4 summarizes the experimental setup and presents the training and validation results from a fine-grained dataset using evaluation metrics, demonstrates the proposed model's results on the test set, and finally compare the results with the state-of-the-art fine-grained recognition methods. Discussion of these results is covered in Section 5, and lastly, Section 6 concludes the proposed work.

2. Related Work

Currently, there are two prevailing techniques for fine-grained image recognition. The first approach is known as localization classification subnetworks, and the other one is end-to-end feature encoding.

The two-stage method (localization classification subnetworks), in particular, depends on the object and part localization annotations, region proposal networks [21], or attention mechanisms to acquire discriminatory areas, which are subsequently fed to the classifier. Study [22] built object and part detectors using bounding box datasets to find the most effective local semantic parts, and afterward applied a classifier to retrieve final classifications. Ref. [23] generated portions using segmentation and a posture graph, followed by moving them to a classification model. The authors of [24] layered a series of branches comprising a part cascade, an object cascade, and part landmark localization to merge feature maps carrying information from each component along with the bounding box. In [25], researchers combined classification and semantic part recognition. Study [26] designed a multi-granularity algorithm for learning with two stages: a targeted search to discover ROI (regions of interest) followed by classification. Ref. [27] utilized a weakly supervised approach to locate various relevant regions coming from proposals and subsequently apply them to generate a broader representation for classification. An attention mechanism was employed by [28] to train a coarse-grained model to identify relevant regions, which were then forwarded through a fine-grained network to enhance categorization. In terms of conclusion, each of those techniques attempts to use object-level or local-level details to eliminate unnecessary information, then feed the relevant information to the classifier for the classification task.

Considering the complex two-stage pipeline and tedious and resource-intensive datasets, the present work emphasizes end-to-end feature encoding through deep learning neural networks to identify minute differences within subcategories. This strategy relies on maximizing classification outcomes through improved feature representations. A couple of research studies [29,30] proposed paired interaction learning approaches to gather semantic differences. The authors of [31] employed the self-attention mechanism to retrieve discriminative features. Study [11] suggested a hierarchical architecture that performs cross-layer bilinear pooling. A small number of studies have focused on fine-

grained image recognition with one-stage object detectors, which similarly adopt the feature encoding approach with the object localization system. Consequently, in this study, we intended to explore the performance and ability of the single-stage detector [32] to fill the void in the fine-grained image recognition problem.

3. Proposed Model

3.1. Yolov5 Backbone

The Yolov5 backbone serves the purpose of feature extraction from the given input image. The backbone includes a focused network, spatial pyramid pooling, and a cross-stage partial network, which can be seen in Figure 1. The focus structure decreases model parameters and GPU storage space for execution, which results in boosting the model speed. The spatial pyramid pooling unit has the ability to enhance the receptive field. A broad receptive field is capable of spotting object information and discriminating some of the significant relevant features. The cross-stage partial network has two different kinds of patterns; the difference between them is the reiterated ResUnit, which has more complex layers that are capable of extracting detailed information. The Yolov5 backbone, however, struggles with modeling long-range dependencies across the entire input as well as understanding detailed contextual information, which is essential for fine-grained image recognition. As a result, we proposed an improved backbone through introducing transformer encoder blocks to replace the bottleneck CSP blocks.

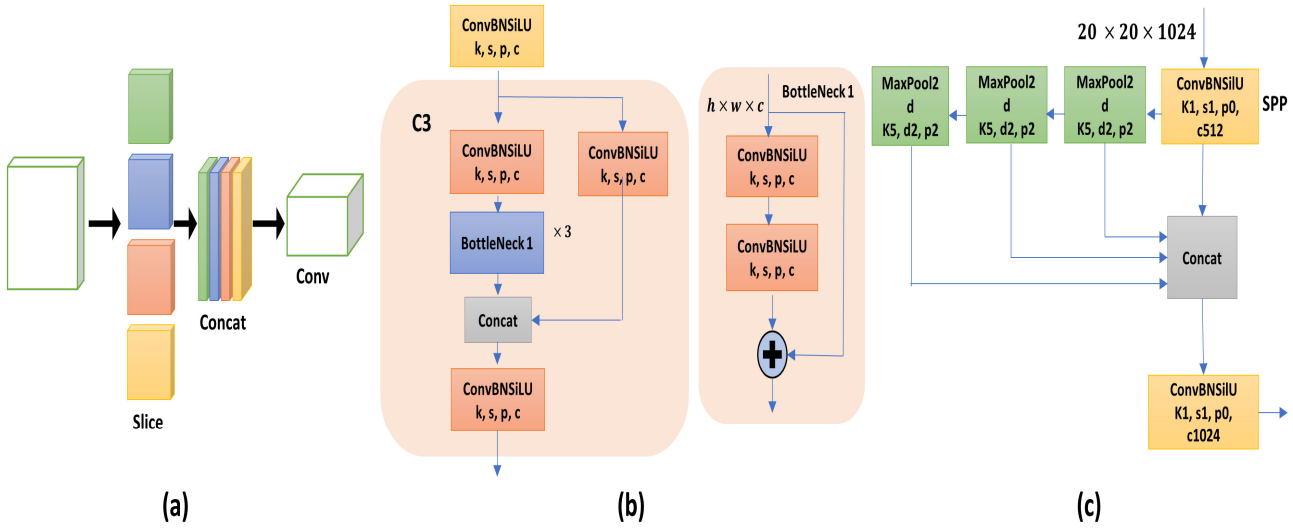


Figure 1. (**a**) The focus structure uses down-sampling and concatenation operations to capture both fine-grained details and larger spatial context. (**b**) BottleNeckCSP module is a building block that extracts features through utilizing cross-stage partial connections. (**c**) Spatial pyramid pooling utilizes multiple levels of pooling operations to capture features at different scales.

3.2. Improved Backbone with Vision Transformer

A traditional vision transformer [16] is composed of two basic components: a linear projection from an image as well as a transformer encoder block that includes numerous MLP models alongside a self-attention network.

3.2.1. Patch Embedding

The vision transformer method involves splitting the input image into different patches of identical shapes, similar to a pattern of embedded words in natural language processing. The image is broken down into image tokens using the vision transformer as

$$[X_1, X_2, X_3, \ldots\ldots X_N] \text{ by } x \in r^{n \times d} \quad (1)$$

A convolutional neural network employs pixel arrays; however, the patch size (n) must be specified. This phase involves vectorizing the received visual patches into vectors or flattening them, and then these flattened patches are projected onto a lower-dimensional space using the linear operator on each of the vectors x_n. Since w and b are two accepted parameters obtained using the training data, these individuals also append a position embedding acquired through patches $P \in 1, 2, \ldots, N$ to their respective $\vec{z}$ vectors to ensure that the $\vec{z}$ vector retains both the content as well as the position simultaneously. This result is regarded as patch embedding and is written as

$$Z_N = W_{XN} + B \quad (2)$$

Through this, nearer patches often have matching position embedding compared to other patches. In recognition tasks, including a second embedded learnable vector Z_0 into the sequential X, which represents the CLS token, enables gathering and keeping data that has been acquired through other tokens and has an identical form as the rest of the $\vec{z}$ vectors.

3.2.2. Transformer Encoder Block

The self-attention mechanism transforms a single feature into another through capturing long-range dependencies across each input via taking N instances with no contextual information and then returning N entities with contextual information. In other terms, it accepts inputs in the manner of $[X_1, X_2, X_3, \ldots\ldots X_N]$ by $x \in r^{n \times d}$, and further employs the learnable weighted matrices that are queries $w^Q \in r^D \times D_Q$, keys $w^K \in D \times D_K$, and values $w^v \in D \times D_K$. Evidently, through combining each value with weights after measuring the query across all keys, the equation below represents the self-attention output.

$$attention(q, k, v) = \vec{z} = SoftMax(\frac{q \cdot k^t}{\sqrt{D_Q}})v \quad (3)$$

upon which, $\vec{z} \in r^{N \times D}$ along with SoftMax to achieve the attention level having $v = xw^v$, $q = xw^q$, and $k = xw^k$ using the dot product calculation. Relying on adopting a self-attention layer, the vision transformer applies multi-head self-attention. Where eight headers are often used to streamline various complex connections among different components in a series and handle longer-term dependencies, this corresponds to the aggregated multiple self-attention that is independent of parameters w_i^q, w_i^k, and w_i^v and possesses similar input, where $I = 0, \ldots\ldots (H-1)$, and H is the overall length of attention blocks, respectively.

$$multihead(q, k, v) = concat(HEAD_1, \ldots\ldots, HEAD_H)\, w \quad (4)$$

while $HEAD_1 = attention(vw_i^v, qw_i^q,$ and $kw_i^q)$, and outcomes are combined to a single matrix, $[c_0, c_1, \ldots\ldots c_{H-1}] \in r^{H \cdot D \times D_K}$.

Multilayer perception layers (MLPs) in the transformer encoder block have enabled our model to narrow its attention to the relevant features while minimizing the number of parameters after integrating them into the final layer inside the feature extraction stage. The dimensions of the input image along with extracted features and the output can be seen in Figure 2, where $640 \times 640 \times 3$ represents the size of the input, and once the input image is converted into a feature map, its dimensions change to $20 \times 20 \times 512$. As a result, the transformer encoder block input size is $20 \times 20 \times 512$. The feature map's size is 400×512 (length × channel) using patch embedding, which applies a simple additive operation through a learnable vector. Therefore, transformer encoder block input vectors and output vectors are of the same size.

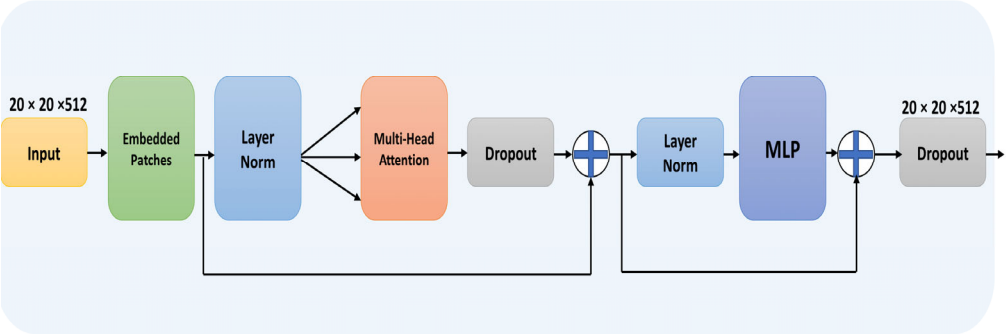


Figure 2. Transformer encoder block performs the additive operation through a learnable vector and has the same input and output size.

3.3. Improved Yolov5

3.3.1. Yolov5 Architecture Overview

In the Yolov5 model, the learning capability offered by the CSPNet (cross-stage partial network) is used to formulate the CSPDarkNet53 network to boost network performance. The results greatly minimize the model parameters, simultaneously improve residual feature information, and boost feature learning abilities compared to the ResNet model. Meanwhile, the neck primarily functions to combine information coming from multiple features to form a model with improved representation and richer features. The total number of heads is chosen by the neck, where objects of different sizes are assigned to each head for learning. The neck also maintains a multi-scale feature fusion order that improves the existing range of the features in a more effective manner than simply utilizing a single pooling method and notably distinguishes the object context.

3.3.2. Proposed Model Improvement Comparison with the Existing Model

- Multi-Head Self-Attention: Convolutional layers (CSP bottleneck module) are effective for feature extraction, but they do not capture fine-grained details and subtle differences required for accurate fine-grained recognition. The fixed receptive field size and limited context modeling of convolutional layers hinder their ability to capture fine-grained visual cues. We replaced these convolutional layers with multi-head self-attention layers which enabled the model to capture both local and global dependencies, as it allowed the model to attend to specific fine-grained details and captured the broader context.
- Feed-Forward Neural Network: The residual connection in the CSP bottleneck module helps propagate information through skip connections. However, it is not sufficient to capture the intricate relationships and dependencies present in fine-grained recognition tasks, where subtle details and local patterns play a crucial role. We replaced it with feed-forward networks, which allowed the model to refine the feature representation in a non-linear manner and enabled the model to learn complex patterns and capture subtle differences between visually similar categories.
- Layer Normalization: An additional layer normalization step helped with stabilized training and improved gradient flow. The first normalization layer enhanced the model's ability to learn discriminative features through reducing the impact of variations in scales and intensities across fine-grained images, while another layer of normalization further enhanced the stability and convergence of the model.
- Position-wise Feed-Forward Network: Additional convolutional layers in the CSP bottleneck module have limited context modeling capabilities, which limits their ability to capture the intricate details necessary for accurate fine-grained recognition. Our position-wise feed-forward network added additional non-linearity, allowing the model to capture more intricate relationships between tokens/regions.

- In our experiment, we also applied transformer heads at the last stage of the Yolov5 network because the feature maps at that stage have low resolutions, and using transformer heads on low-resolution feature maps reduces high computing costs and memory consumption. The proposed architecture can be seen in Figure 3, where transformer encoder blocks containing a multi-head self-attention module with a feed-forward neural network and layer normalization are integrated at the end of the feature extracting stage and the transformer encoder blocks included in the model's neck are used to form prediction heads.

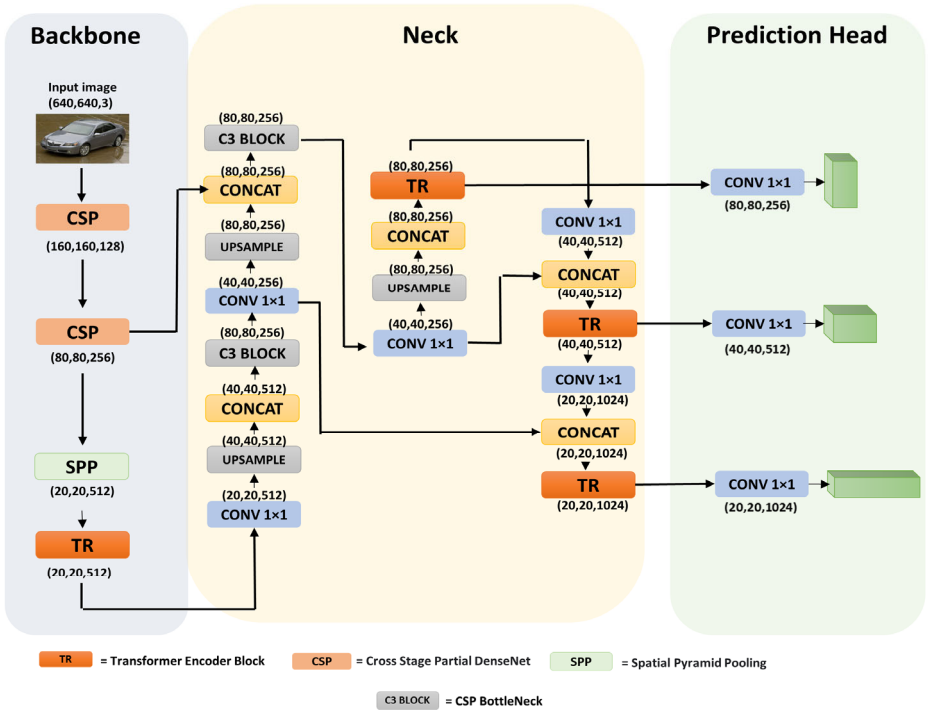


Figure 3. Proposed model based on Yolov5 for fine-grained image recognition. Three transformer encoder blocks have been added at the end of the backbone, where input features are fed through a spatial pyramid pooling layer. Replaced transformer prediction heads take the feature maps from the neck part.

4. Model Training and Results

In this section, we will explain the dataset and experimental setup, then exhibit the training outcomes, and compare the models. Finally, we will demonstrate the outcomes of the experiment.

4.1. Dataset

The Stanford car dataset, having 16,185 images of 196 classes and sometimes extended to 208 classes, is used for the experiment. This dataset contains images of vehicle brands with significantly identical visual appearances and is one of the few benchmark datasets that are specifically designed for fine-grained image recognition. The dataset images, which are in JPEG format and contain different sizes, were first converted into Yolov5 format and then split into train, validation, and test sets. Figure 4 shows the dataset visualization.

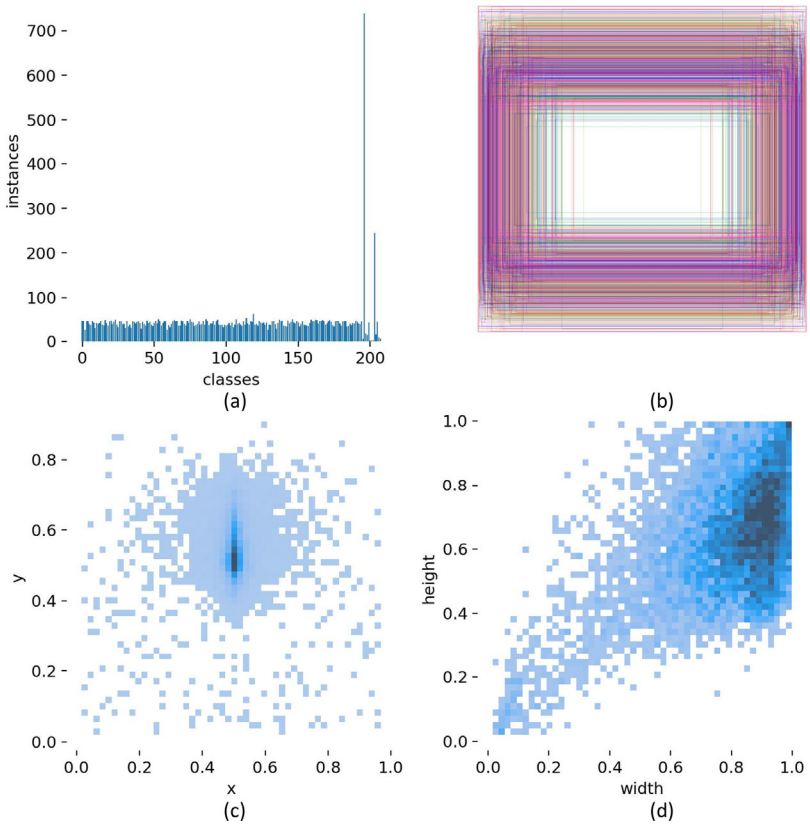


Figure 4. Visualization of the dataset. (**a**) The number of annotations for each class; (**b**) a visual representation of the location as well as the dimensions associated with each bounding box; (**c**) the statistical distribution of bounding box location; (**d**) the statistical distribution of bounding box dimensions.

4.2. Experimental Environment

Our training setup involved a Windows 10 64-bit operating system with a 13th Gen Intel(R) Core(TM) i5-13400 processor, 32 GB of RAM, NVIDIA GeForce RTX 3060 Ti GPU, and Python 3.9 with the Pytorch framework. To optimize the performance of our model and to compare and analyze it with existing Yolov5 models, we trained all the models with SGD and ADAM optimizers. Table 1 displays multiple experimental hyperparameters.

Table 1. Experimental hyperparameter details. Most of the parameters were set to the same as the default Yolov5 model; only data loaders and optimizers were tested at different stages.

Parameter	Values
Batch Size	16
Learning Rate	0.01
Learning Rate Decay	0.999
Momentum	0.937
Learning Rate Decay Step	5.e−4
Epoch	300
Workers	8

4.3. Evaluation Metrics and Model Training

Precision, recall, average precision, and F1 score are commonly used for statistical analysis to evaluate the effectiveness of a detection model. Below are the equations adopted to evaluate precision, recall, and F1 score.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{5}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{6}$$

The number of correctly detected objects refers to true positives; false positives are wrongly identified as targets, and false negatives are the number of undetected objects. If the predicted bounding box of an object differs from the ground truth, this is not evidence that the detection is incorrect; therefore, intersection over union (IoU) is a frequent approach, where intersection over union is the ratio of the detected bounding box over the ground truth (bounding box). If the value of the IoU is higher than the set threshold, the detection is accurate (true positive); else, it is incorrect (false positive).

We derived the F1 Score assuming a harmonic mean of recall and precision upon calculating the precision and recall scores for each class. The F1 Score allows us to understand how the model becomes confused while providing predictions. Equation (7) is often used to compute the F1 Score for all classes.

$$F1\ Score = \frac{2 \times Precision \cdot Recall}{Precision + Recall} \tag{7}$$

The precision–recall curve represents a curve where the x-coordinate is the recall rate, and the y-coordinate is the accuracy. The total area under precision–recall curve is referred to as the average precision (AP), and it can be calculated using Equation (8).

$$Average\ Precision = \int_0^1 Precision(Recall) dRecall \tag{8}$$

mAP indicates the average across all APs to evaluate the model's performance.

$$mAP = \sum \frac{1}{n} Average\ Precision \tag{9}$$

We trained our proposed model using both SGD and ADAM optimizers along with the Yolov5l, Yolov5x, Yolov7, and Yolov8 models on the Stanford car training data set. The training results after every epoch are displayed in Figure 5. We observed that during the first 100 epochs, all eight models' precision increased gradually, while the recall dropped and fluctuated until the 50th epoch. However, the mAP@50 and mAP@50:95 remained stable and were gradually increasing, which shows that the instability during the initial stage of the training did not affect the performance of the models. Our proposed models' accuracy peaked at 0.934 on the 220th epoch followed by the Yolov8 at 0.919, while both models maintained high recall at 0.89 and 0.88, respectively, considering stable training where the model was improving at every epoch. Consequently, we let the models train until the 300th epoch, where no such improvement in accuracy and recall was seen.

Table 2 demonstrates the overall training results of all six models, where we can better analyze the models' performance during training. Our proposed model with the transformer encoder block performed relatively well during training with higher accuracy and recall.

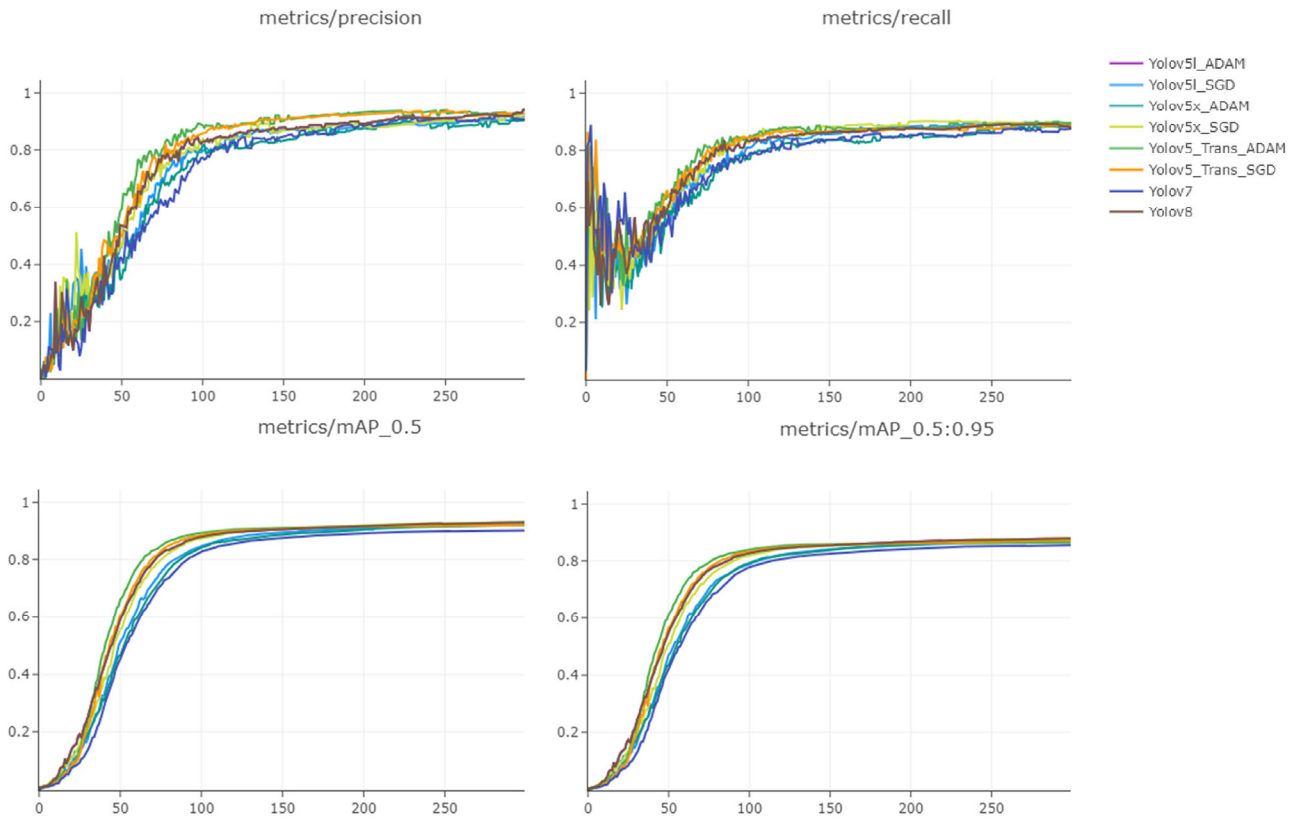


Figure 5. All models' training results after every epoch. Proposed model precision during training peaked at 0.934 along with a recall of 0.895, where average precision at 0.5 and 0.5:0.95 thresholds peaked at 0.927 and 0.878, respectively.

Table 2. Overall training results for all the models. The proposed model performed better during the training process, whereas Yolov5x slightly improved with the ADAM optimizer. Yolov7 recall dropped the most compared to other models.

Model	Precision	Recall	mAP @0.5	mAP 0.5:0.95
Yolov5l_SGD	0.890	0.885	0.912	0.863
Yolov5l_ADAM	0.911	0.880	0.912	0.864
Yolov5x_SGD	0.896	0.889	0.917	0.874
Yolov5x_ADAM	0.901	0.891	0.919	0.868
Yolov5_tr_SGD	0.931	0.892	0.921	0.873
Yolov5l_tr_ADAM	0.934	0.895	0.927	0.878
Yolov7	0.898	0.839	0.884	0.811
Yolov8	0.919	0.887	0.914	0.877

Loss Function

Three different parts form the Yolov5 loss function: object loss, bounding box loss, and class loss. These components are weighted and paired to create the final loss function and can be seen in Equation (10).

$$LOSS = a \cdot LOSS_{object} + b \cdot LOSS_{B.box} + c \cdot LOSS_{class} \quad (10)$$

The weights assigned to the loss function are represented by a, b, and c. Typically, object loss is given the highest weight, followed by bounding box loss and class loss.

Equation (11) shows the two cross-entropy losses, which are class loss and object loss.

$$LOSS_{CLASS,OBJECT} = -\frac{1}{n}\sum_{I=1}^{n} Y_I \cdot Log\big(P(Y_i)\big) + (1 - Y_i) \times Log(1 - P(Y_i)) \quad (11)$$

where the total number of categories is (n), true value is referred to as Y_i, and predicted probability is $P(Y_i)$.

The model's predicted result and the true value are compared using binary cross-entropy. The loss function value will be closer to zero if the predicted value is nearer to 1, meaning that the loss function value decreases as the gap between the expected result and the actual value reduces. On the other hand, if the predicted value moves nearer to 0, the gap between the true value and the predicted result will be greater so that the loss function value will be higher.

We calculated the losses using Equation (11) during the training, where we have seen all the models' losses (object loss, bounding box loss, and class loss) decreased gradually, apart from the bounding box loss of Yolov5l and Yolov5x with the SGD optimizer, which moderately decreased from 0.04 to 0.02 initially, but between the 40th epoch and the 110th epoch, it increased and did not converge relatively as expected. The object loss of our proposed model with the ADAM optimizer as well as the Yolov7 model instantly decreased to 0.005 after 10 epochs, which later ended up at 0.003—far better than the other models' during training—whereas during validation, different models have shown different convergence rates, where the class loss decreased gradually while box loss and object loss decreased until the 100th epoch and later settled with same rate until the end, which can be seen in Figure 6.

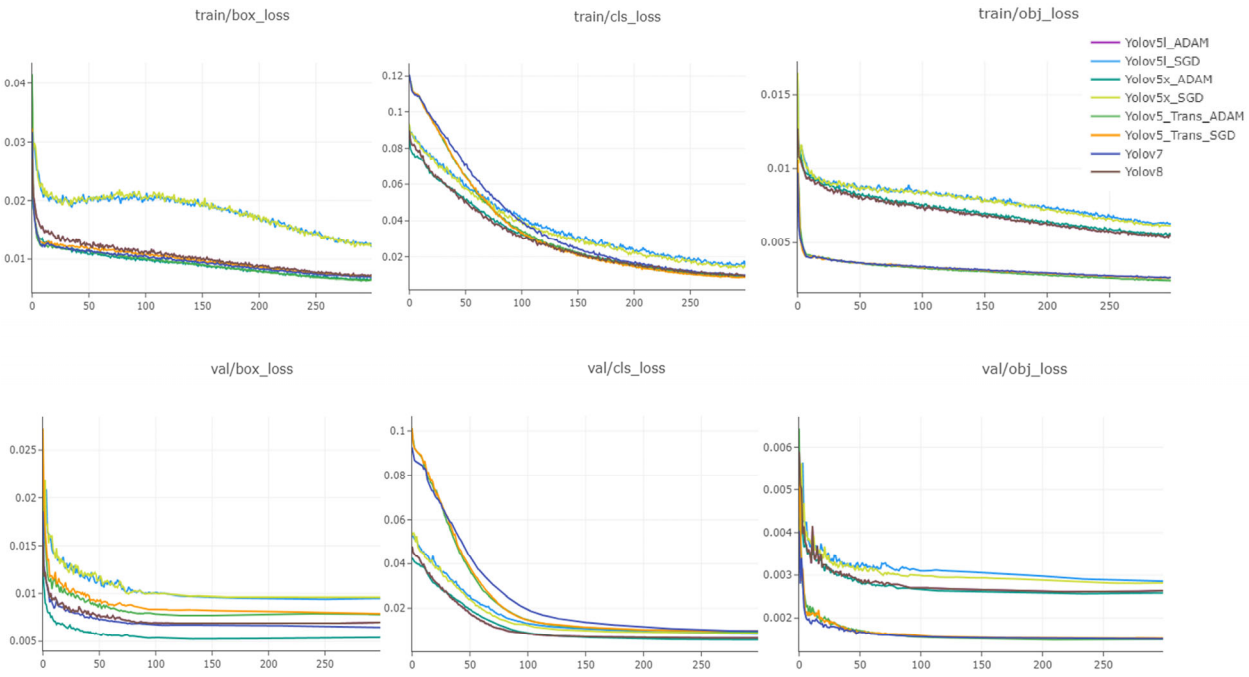


Figure 6. Representation of all models' training and validation losses after every epoch.

4.4. Model Adaptability over Test Images

To test our model's capability for fine-grained image recognition, we used the Stanford test set, having 2530 challenging identical images. Vehicle recognition results can be seen in Figure 7. Table 3 demonstrates our models' inference speed and compares the result with other trained models. Our model has a slower inference speed as well as

preprocessing time compared to other models because of the increased training parameters after including transformer encoder blocks.

Figure 7. Proposed models' recognition results with similarly shaped vehicles but different brand models and make years. The model performed better when provided with similar vehicle shapes and colors at different viewing angles.

Table 3. All models' inference speed comparison on the Stanford car test set. The latest Yolov7 and Yolov8 models with fewer training parameters have the edge of being lighter models that can detect at a minimum inference speed of 15.1 and 13.9 ms, respectively, but recognition accuracy is lower than that of our proposed model, which is moderately behind by 39.2 ms.

Model	Pre-Process (ms)	Inference Speed (ms)	NMS/ Image (ms)	Image Size
Yolov5l_SGD	0.3	15.5	0.6	640 × 640
Yolov5l_ADAM	0.3	15.5	0.6	640 × 640
Yolov5x_SGD	0.3	28.7	0.7	640 × 640
Yolov5x_ADAM	0.4	28.4	0.7	640 × 640
Yolov5_tr_SGD	0.8	39.2	0.9	640 × 640
Yolov5l_tr_ADAM	0.8	39.2	0.9	640 × 640
Yolov7	0.4	15.1	0.6	640 × 640
Yolov8	0.3	13.9	0.5	640 × 640

The normalized confusion matrix of the proposed model is presented in Figure 8, which was generated after obtaining the precision and recall scores for the test images. Predicted true positive and true negative values for all 208 sub-classes can be seen in the diagonal position, with the dark blue color indicating confidence of over 0.8 on the predicted class, whereas very few false positive and false negative predictions with light blue color indicating that confidence of below 0.4 was obtained. These false predictions occurred where the model became confused due to occlusion and low light.

The precision–recall curve and the F1 confidence score curve for all the classes are produced using Equations (7) and (8) and can be visualized in Figure 9. Intuitively, it is evident that as the recall is increasing, the rate of change in accuracy is also increasing. The PR curves are established near the upper right corner, demonstrating the proposed model's efficiency in recall and accuracy. The substantial area under the PR curves suggests that our approach works effectively. Furthermore, the smooth PR curves confirm that our model's accuracy and recall rate have a very stable relationship.

The F1 curve for all classes started to flatten at a confidence score of 0.8, as shown in Figure 9b, which means that the bounding boxes under the confidence score of 0.8 are discarded later in validation.

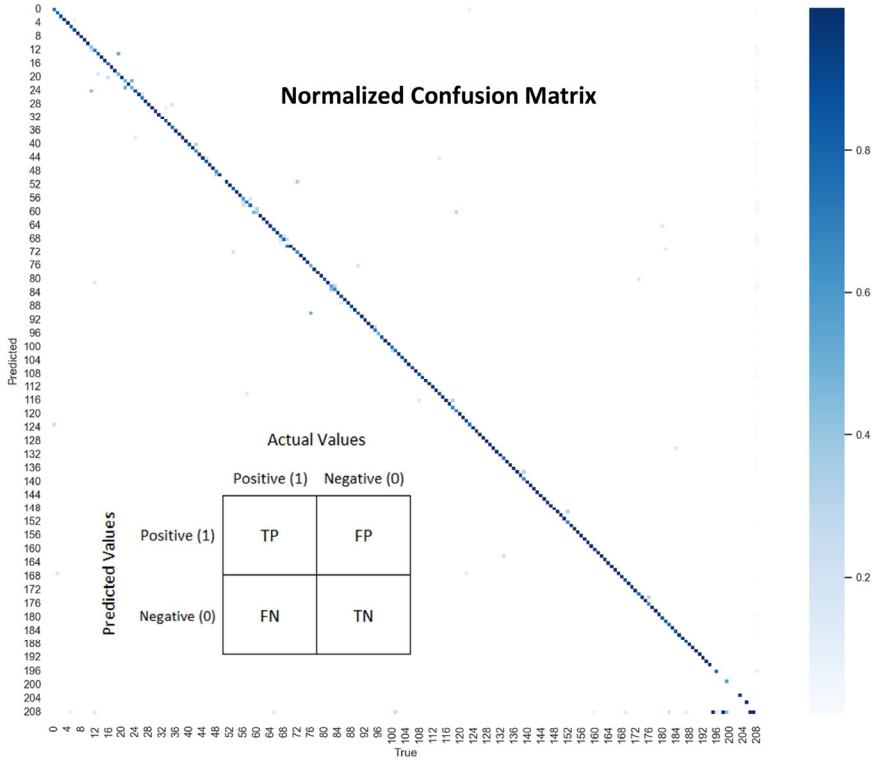
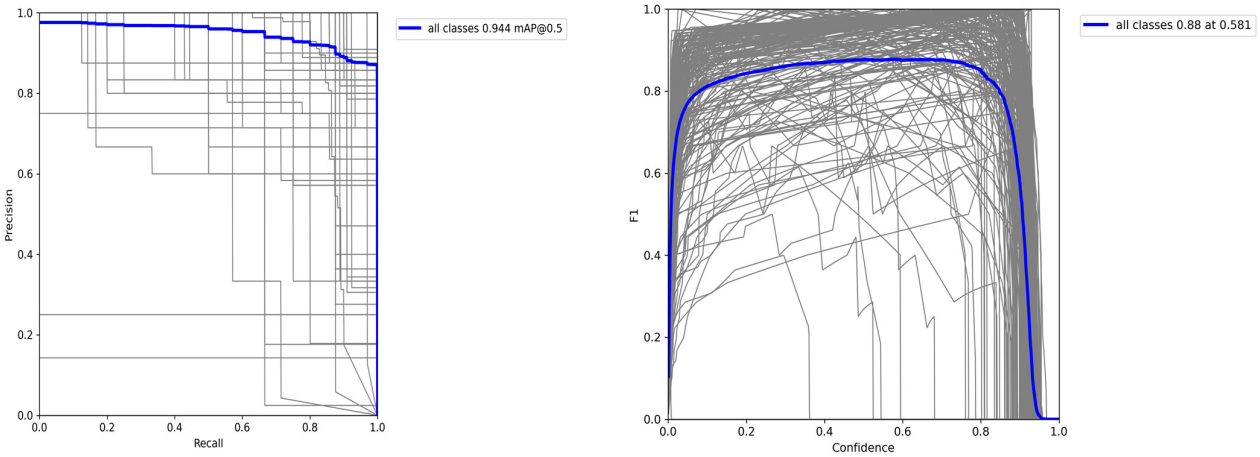


Figure 8. Proposed model confusion matrix based on the test set predictions. As can be seen in the figure, the diagonal values are the correctly predicted samples.



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

Figure 9. PR curve and F1 confidence score curve from the proposed model's test set recognition results. (**a**) represents the mAP@0.5 threshold for all classes where the average precision is 0.944. (**b**) represents the proposed model confidence score of 0.88 at the 0.581 threshold value.

We compared the trained model's performance on a Stanford car test set. We passed 2530 images to all the models with the batch size of 10 and calculated the mAP scores through averaging the precision scores for each model using Equation (9). Yolov5's existing models performed moderately well against the latest Yolov7 and Yolov8 models on unseen data. However, our model performed better than the others but at the cost of inference speed, which can be seen in Table 4.

Table 4. All models' performance comparison on the Stanford car test set, where Yolov7 and Yolov8 performed at a minimum inference speed with lower mAP compared to the proposed model, which performed better than all, scoring 0.919 at mAP@50 and 0.871 at mAP@0.5:95 thresholds.

Model	Precision	mAP @0.5	mAP 0.5:0.95	Total Inference Time (s), Test Images = 2530, Batch Size = 10
Yolov5l_SGD	0.874	0.899	0.845	40
Yolov5l_ADAM	0.871	0.896	0.847	40
Yolov5x_SGD	0.88	0.9	0.844	47
Yolov5x_ADAM	0.891	0.901	0.848	47
Yolov5_tr_SGD	0.926	0.919	0.864	56
Yolov5l_tr_ADAM	0.927	0.919	0.871	56
Yolov7	0.899	0.882	0.831	37
Yolov8	0.901	0.9	0.858	31

4.5. Comparison with State of the Art

We compared our model with some of the existing state-of-the-art fine-grained image recognition models. Most of the models have utilized the VGG-19 and Resnet-50 backbones and are weakly supervised, where no training annotations were employed. The comparison is based on the accuracy achieved using the benchmark Stanford car dataset. Our method achieved 93.4 percent accuracy with an improved CSP-Darknet53 backbone; a comparison can be seen in Table 5.

Table 5. Accuracy comparison of our model with the state of the art on Stanford car dataset.

Methods	Train Anno	Backbone	Image Resolution	Accuracy
RA-CNN		VGG-19	448 × 448	92.5%
BoT		Alex-Net	Not given	92.5%
WPA	BBox	CaffeNet	224×224	92.6%
MA-CNN		VGG-19	448 × 448	92.8%
PA-CNN		VGG-19	448 × 448	93.3%
M2DRL		VGG-16	448 × 448	93.3%
Yolov5-Trans	BBox	CSP-Darknet53	640 × 640	93.4%
DFL-CNN		VGG-16	448 × 448	93.8%
TASN		ResNet-50	224 × 224	93.8%
Hsnet	Parts	GoogleNet	224 × 224	93.9%
MGE-CNN		ResNet-50	448 × 448	93.9%
NTS-Net		ResNet-50	448 × 448	93.9%
GCL		ResNet-50+BN	448 × 448	94.0%
FDL		ResNet-50	448 × 448	94.3%
S3N		ResNet-50	448 × 448	94.7%
DF-GMM		ResNet-50	448 × 448	94.8%

5. Discussions

This research focuses on the recognition of fine-grained vehicles, which are almost visually identical. Although promising results have been achieved using a one-stage

object detector, vision transformers are known for their computational and memory requirements. The self-attention mechanism used in the transformer encoder block computes pairwise interactions between all elements in the input feature map, resulting in quadratic complexity with respect to the input size. During the training stage, our model performed considerably better but still consumed a lot of time because of the higher number of training parameters as compared to the other models, and when we tested the model on the test set, the inference speed was slightly increased as well. The implementation of a visual transformer, which costs additional speed and memory resources, is a drawback of this study. However, as we know, vision transformers divide the input image into fixed-size patches and process them individually. Through reducing the patch size, the number of patches and the subsequent memory requirements can be decreased. But taking into consideration that this reduction should be balanced with the model's ability to capture fine-grained details, smaller patches might result in a loss of information. Various methods have already been proposed to make transformer attention mechanism more efficient, such as utilizing sparse attention patterns or approximating attention mechanisms with lower complexity operations such as kernelized self-attention or linear attention. These approaches can significantly reduce memory requirements and computational overhead and can be considered to align with our future research work.

However, while testing all the models with an unseen test set, these transformer encoder blocks expressed robustness through enhancing the generalization capability of the model via ensuring consistent performance across all classes through capturing the underlying patterns and features that are characteristic of each object class, which enabled accurate recognition even on samples that differ from the training data. Although generalization capability improved with transformer encoder blocks, there are still some resilience issues faced during the testing stage, including occlusion, poor lighting conditions, and partial visibility where our model became confused and struggled with accurately detecting objects. There is another factor that came during the test phase which affected the model's performance: sensitivity to image quality and noise. In scenarios where images have low resolution, high noise levels, or significant distortions, our model, along with other models, was ineffective due to the nature of one-stage feature extraction. Despite that, strengthening the preprocessing techniques, denoising methods, or data augmentation strategies can help to improve the model's resilience to these challenges, while enhancements in multi-scale feature fusion can make the proposed model robust to occlusion handling. A BiFPN (bi-directional feature pyramid network) can be utilized in future research; it is a variant of the FPN that introduces additional connections to create a bidirectional information flow, facilitating more effective feature fusion and refinement. It aims to address both feature resolution degradation and inconsistent feature propagation issues in the FPN architecture.

6. Conclusions

In this paper, we proposed a one-stage fine-grained object recognition model based on the Yolov5 object detector. We improved the backbone of the existing Yolov5 model to effectively capture global dependencies and enabled the model to learn relationships between distant regions. This improved the model's ability to understand context and capture long-range spatial relationships in an image, which are important aspects of the fine-grained recognition task. We also replaced the Yolov5 detection heads with three transformer heads using the discriminative feature maps from transformer encoder blocks. To evaluate model improvement after adding the transformer encoder blocks, we used the famous Stanford car dataset, which is a benchmark dataset for fine-grained recognition consisting of 16,185 highly similar images of 196 different classes of vehicles, which has been updated to 208 classes. We trained the existing Yolov5l and Yolov5x models along with our proposed model for 300 iterations using both the stochastic gradient descent (SGD) and adaptive moment estimation (ADAM) optimizers. Evaluation matrices such as precision, recall, mAP, and F1 score were used to evaluate model performance and to

obtain comparisons with the existing Yolov5 and state-of-the-art models. However, further study is required to implement modern vision transformers effectively, particularly to solve the challenges of speed and extreme memory usage.

Author Contributions: Conceptualization, U.A. and S.O.; methodology, U.A.; software, U.A.; validation, J.K., T.-W.U., and M.H.; formal analysis, J.K.; investigation, U.A.; resources, S.O.; data curation, U.A.; writing—original draft preparation, U.A.; writing—review and editing, J.K.; visualization, T.-W.U.; supervision, J.K.; project administration, S.O.; funding acquisition, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Korean government (MSIT) (2022-0-00215). This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (2021-0-02068, Artificial Intelligence Innovation Hub).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Pang, K.; Yang, Y.; Hospedales, T.M.; Xiang, T.; Song, Y.Z. Solving Mixed-Modal Jigsaw Puzzle for Fine-Grained Sketch-Based Image Retrieval. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10344–10352. https://doi.org/10.1109/CVPR42600.2020.01036.
2. Zhou, W.; Mok, P.Y.; Zhou, Y.; Zhou, Y.; Shen, J.; Qu, Q.; Chau, K.P. Fashion recommendations through cross-media information retrieval. *J. Vis. Commun. Image Represent.* **2019**, *61*, 112–120. https://doi.org/10.1016/J.JVCIR.2019.03.003.
3. WMin; Jiang, S.; Jain, R. Food Recommendation: Framework, Existing Solutions, and Challenges. *IEEE Trans. Multimed.* **2020**, *22*, 2659–2671. https://doi.org/10.1109/TMM.2019.2958761.
4. Bao, J.; Chen, D.; Wen, F.; Li, H.; Hua, G. CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; Volume 2017, pp. 2764–2773. https://doi.org/10.1109/ICCV.2017.299.
5. Jing, L.; Yang, X.; Tian, Y. Video you only look once: Overall temporal convolutions for action recognition. *J. Vis. Commun. Image Represent.* **2018**, *52*, 58–65. https://doi.org/10.1016/J.JVCIR.2018.01.016.
6. Xu, N.; Liu, A.A.; Liu, J.; Nie, W.; Su, Y. Scene graph captioner: Image captioning based on structural visual representation. *J. Vis. Commun. Image Represent.* **2019**, *58*, 477–485. https://doi.org/10.1016/J.JVCIR.2018.12.027.
7. Qin, Z.; Zhang, Y.; Meng, S.; Qin, Z.; Choo, K.K.R. Imaging and fusing time series for wearable sensor-based human activity recognition. *Inf. Fusion* **2020**, *53*, 80–87. https://doi.org/10.1016/J.INFFUS.2019.06.014.
8. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015.
9. Zhao, B.; Wu, X.; Feng, J.; Peng, Q.; Yan, S. Diversified Visual Attention Networks for Fine-Grained Object Classification. *IEEE Trans Multimed.* **2017**, *19*, 1245–1256. https://doi.org/10.1109/TMM.2017.2648498.
10. Dubey, A.; Gupta, O.; Guo, P.; Raskar, R.; Farrell, R.; Naik, N. Pairwise Confusion for Fine-Grained Visual Classification. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Cham, Switzerland, 2017; Volume 11216, pp. 71–88. https://doi.org/10.1007/978-3-030-01258-8_5.
11. Yu, C.; Zhao, X.; Zheng, Q.; Zhang, P.; You, X. Hierarchical Bilinear Pooling for Fine-Grained Visual Recognition. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer, Cham, Switzerland, 2018; Volume 11220, pp. 595–610. https://doi.org/10.1007/978-3-030-01270-0_35.
12. Ji, R.; Wen, L.; Zhang, L.; Du, D.; Wu, Y.; Zhao, C.; Liu, X.; Huang, F. Attention Convolutional Binary Neural Tree for Fine-Grained Visual Categorization. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10465–10474. https://doi.org/10.1109/CVPR42600.2020.01048.
13. Fu, J.; Zheng, H.; Mei, T. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2016; Volume 2017, pp. 4476–4484. https://doi.org/10.1109/CVPR.2017.476.
14. Zhang, X.; Xiong, H.; Zhou, W.; Lin, W.; Tian, Q. Picking deep filter responses for fine-grained image recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; Volume 2016, pp. 1134–1142. https://doi.org/10.1109/CVPR.2016.128.

15. Jiao, Q.; Liu, Z.; Ye, L.; Wang, Y. Weakly labeled fine-grained classification with hierarchy relationship of fine and coarse labels. *J. Vis. Commun. Image Represent.* **2019**, *63*, 102584. https://doi.org/10.1016/J.JVCIR.2019.102584.
16. 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.
17. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H.; et al. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6877–6886. https://doi.org/10.1109/CVPR46437.2021.00681.
18. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. *arXiv* **2021**. arXiv: 2102.04306.
19. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer, Cham, Switzerland, 2020; Volume 12346, pp. 213–229. https://doi.org/10.1007/978-3-030-58452-8_13.
20. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**. arXiv: 2004.10934.
21. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.
22. Zhang, N.; Donahue, J.; Girshick, R.; Darrell, T. Part-based R-CNNs for Fine-grained Category Detection. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer, Cham, Switzerland, 2014; Volume 8689, pp. 834–849. https://doi.org/10.1007/978-3-319-10590-1_54.
23. Krause, J.; Jin, H.; Yang, J.; Li, F.F. Fine-grained recognition without part annotations. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015, pp. 5546–5555. https://doi.org/10.1109/CVPR.2015.7299194.
24. Huang, S.; Xu, Z.; Tao, D.; Zhang, Y. Part-Stacked CNN for Fine-Grained Visual Categorization. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 1–26 July 2016; pp. 1173–1182. https://doi.org/10.1109/CVPR.2016.132.
25. Zhang, H.; Xu, T.; Elhoseiny, M.; Huang, X.; Zhang, S.; Elgammal, A.; Metaxas, D. SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 1–26 July 2016; pp. 1143–1152. https://doi.org/10.1109/CVPR.2016.129.
26. Wang, D.; Shen, Z.; Shao, J.; Zhang, W.; Xue, X.; Zhang, Z. Multiple Granularity Descriptors for Fine-Grained Categorization. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 2399–2406. https://doi.org/10.1109/ICCV.2015.276.
27. Zhang, Y.; Wei, X.S.; Wu, J.; Cai, J.; Lu, J.; Nguyen, V.A.; Do, M.N. Weakly supervised fine-grained categorization with part-based image representation. *IEEE Trans. Image Process.* **2016**, *25*, 1713–1725. https://doi.org/10.1109/TIP.2016.2531289.
28. Eshratifar, A.E.; Eigen, D.; Gormish, M.; Pedram, M. Coarse2Fine: A Two-stage Training Method for Fine-grained Visual Classification. *Mach. Vis. Appl.* **2019**, *32*, 49. https://doi.org/10.1007/s00138-021-01180-y.
29. Zhuang, P.; Wang, Y.; Qiao, Y. Learning Attentive Pairwise Interaction for Fine-Grained Classification. In Proceedings of the AAAI 2020—34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 13130–13137. https://doi.org/10.1609/aaai.v34i07.7016.
30. Zheng, H.; Fu, J.; Zha, Z.J.; Luo, J. Learning Deep Bilinear Transformation for Fine-grained Image Representation. *arXiv* **2019**. arXiv: 1911.03621.
31. He, J.; Chen, J.N.; Liu, S.; Kortylewski, A.; Yang, C.; Bai, Y.; Wang, C. TransFG: A Transformer Architecture for Fine-grained Recognition. In Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI 2022, Virtual, 1 March–22 February 2022; Volume 36, pp. 1174–1182. https://doi.org/10.1609/aaai.v36i1.19967.
32. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Michael, K.; Fang, J.; Yifu, Z.; Wong, C.; Montes, D.; et al. ultralytics/yolov5: v7.0—YOLOv5 SOTA Realtime Instance Segmentation. *Zenodo* **2022**, 7347926. https://doi.org/10.5281/ZENODO.7347926.