PG-Prnet: A Lightweight Parallel Gated Feature Extractor Based on An Adaptive Progressive Regularization Algorithm

Zhe Zhang[1], Ming Ye[1*], Yongsheng Xie[1] and Yan Liu[2]

[1]*College of Artificial Intelligence, Southwest University, Chongqing, 400700, China*
[2]*Chongqing Market Supervision Administration Archives Information Center, Chongqing, 400700, China*

Abstract

The residual block in deeper DNNs has a positive effect on feature extraction, but it is limited by practical computational resources. Deeper structures have limited performance gains in later stages, while residuals in lightweight DNNs reduce the abstract feature representation capability. We propose a lightweight parallel gating framework (PG-PRNet) based on the adaptive progressive regularization algorithm (APR), which changes the constant mapping of residual, increases the representation of structural information, and compresses the structure by Hard-Sigmoid, layer pruning, etc. The APR algorithm avoids the irrationality of using the same regularization rules in different cases. This better preserves the shallow spatial location information and deep abstract semantic information, improving the performance of the lightweight model for different specification. PG-PRNet is embedded in two vision tasks. It outperforms the listed models on the GTSRB and BDD100K datasets while maintaining low storage and computational overhead.

Keywords

parallel gating; progressive regularization; feature extraction; residual block

1. Introduction

DNNs can learn the intrinsic properties and underlying semantic features of data from a large number of samples. To a certain extent, the more complex the network is, the more high-dimensional abstract semantic features are obtained. Researchers have proposed many methods to design deeper models. EfficientNetv2 finds a balance between depth, width and resolution to build complex structures [1]. Performs well after pre-training on large datasets. However, practical hardware conditions limit this. In this paper, we propose a GhostModule-based parallel gated feature extractor (PG-PRNet) to selectively control feature embedding into branches, change the traditional constant mapping of residual branches to lighten the network, introduce stochastic depth to prevent network overfitting [2]. We also use Hard Sigmoid and Layer Pruning to further reduce the model parameters. Due to the variety in the dimensionality of the inputs and the depth of the network, it is not reasonable to train the model using the same regularization rules. Therefore, an Adaptive Progressive Regularization (APR) algorithm is also proposed to solve this problem. The effectiveness of PG-PRNet in two vision tasks was experimentally demonstrated. The main contributions of this paper can be summarized as follows.

- We propose a parallel gating unit (PG and Fused-PG) consisting of GhostModule, SE and DepthwiseConv as an intermediate module of the network, improve the constant mapping of residual branches, and configure three specifications of PG-PRNetB0 to PG-PRNetB2.
- We use an adaptive progressive regularization algorithm to solve the unreasonable problem of using the same regularization rules for features of different sizes, resolutions, and network

ICBASE2022@3rd International Conference on Big Data & Artificial Intelligence & Software Engineering, October 21-23, 2022, Guangzhou, China
zhangandzhe@foxmail.com (Zhe Zhang), 2323247608@qq.com (Yongsheng Xie), 12167292@qq.com (Yan Liu)
*Corresponding author: zmxym@swu.edu.cn (Ming Ye)

specifications. The shallow spatial location information and deep abstract semantic information are better preserved.

- We embed the proposed feature extraction framework into image recognition and object detection, and validate the performance of PG-PRNet on GTSRB and BDD100K datasets.

2. Related Work

In its early years focused on improving accuracy by building more complex neural networks. AlexNet designed a DNNs with 60 million parameters and 60,000 neurons, which earned first place in the ImageNetLSVRC-2012 competition [3]. Parallel computing on dozens of devices by Google confirms that distributing the model across multiple devices is another solution [4]. However, in recent years, scholars have found that simply increasing the depth of the model can lead to performance degradation. ResNet shows that as the depth of the network increases, the accuracy gain obtained later decreases due to overfitting, gradient disappearance, etc [5]. The residual structure adopted by ResNet preserves the shallow spatial location information as much as possible. This avoids the above problems to a large extent.

SOTA models usually use neural network structure search (NAS) to find the best structural parameters for building the network [6]. This places higher demands on the hardware. Some researchers are working on network compression. DepthwiseConv most assigning only one set of convolutional kernels to each channel can achieve great speedups with little loss of accuracy [7]. GhostModule presents a plug-and-play module that reduces intermediate feature maps and allows models to be easily deployed on mobile devices. In this paper, GhostModule and DepthwiseConv are used to build PG-PRNet lightweight networks, which combines layer pruning and Hard Sigmoid.

3. Methodology

The overall network structure is shown in figure 1. In the feature extraction part, in order to avoid the computational overload caused by the large feature embedding in the later stage, the input image is first passed through a CBR block, which increases the channel dimension and reduces the width and height scales. Then there are multiple Fused-PG and PG units proposed in this paper. The detailed description of the improvement points is as follows:

3.1. Model compression.

In this paper, layer pruning is used to reduce the overall scale. In order to minimize the sacrifice of accuracy, parallel gating is used, and the representation of residual branches is added. Multiple parallel gating units form a cascade feature representation. According to GhostNet, each trained DNN contains many similar intermediate feature maps. We start by generating only half of the intermediate feature maps, generating the same number of features by linear mapping, called Ghost. Finally, connect the

two parts in series. Extensive use of GhostModule and DepthwiseConv in the PG unit reduces the amount of computation.

The squeeze and excitation module (SE) is inserted into the PG unit to impose an attention mechanism with low computational cost [8]. The Squeeze part of the main branch compresses the features in the channel dimension, and the Excitation part learns the feature weights of the channel. The core idea is that the model learns the attention weight of the channel by loss, so that the weight of the effective feature map is relatively large, and the weight of the invalid feature map is relatively small. We use two 1x1 convolutions instead of fully connected layers, and use Hard Sigmoid activation instead of ReLU, which reduces the amount of computation.

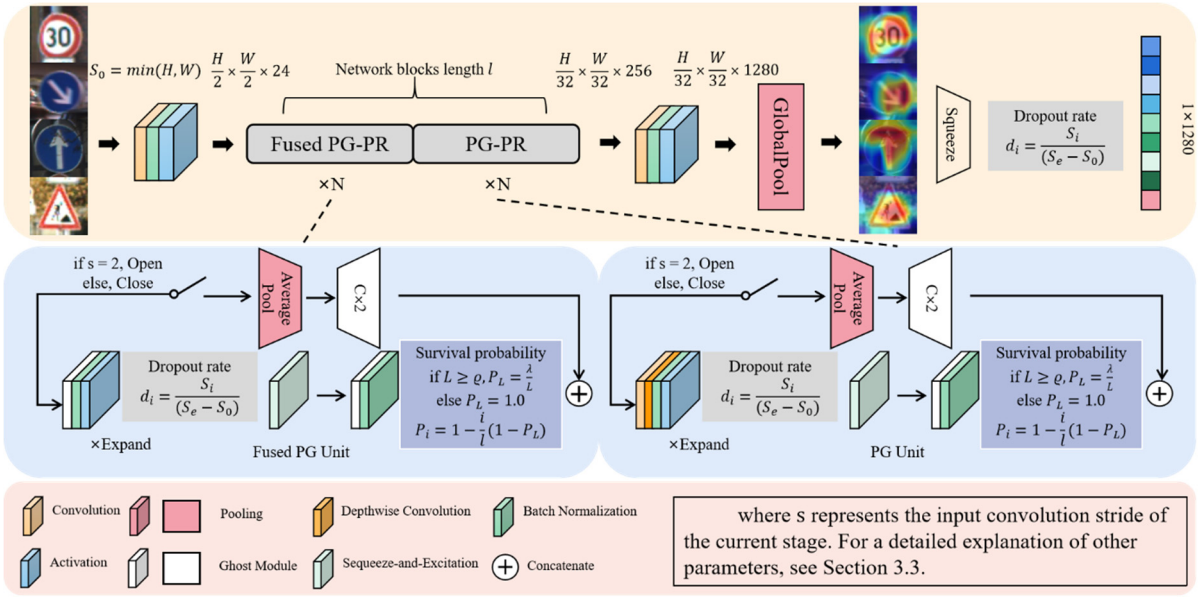

Figure 1. The pipeline of PG-PRNet. Four images from the GTSRB dataset are listed here. Grad-CAM activates and weights the output of the last convolutional layer and visualizes the result in different colors, which shows which parts of the image the model focuses more on [9]. Due to the effectiveness of the proposed method, most of the categories can be highly focused.

3.2. PG and Fused-PG Units

MBConv differs from the traditional process of dimensionality reduction of residuals [10]. The features input to the inverse residual block are first expanded to higher dimensions and then deeply mapped to the lower dimensional space. The PG unit inherits this process. As shown in figure 1, the parameters of the network structure are reduced in the PG unit by using a modified GhostModule instead of CNN. Constant mapping is redundant in the lightweight model (which means that the input features go to the next layer without modification) because it deprives the branching part of the ability to obtain abstract features. We add Pooling and GhostModule to the branch part to selectively output the generated branch-specific folded embeddings by thresholding, which can freely choose the relationship to the backbone part, which is called parallel gating.

Table 1. Compare Top-1 classification accuracy with Fused or not (224 pixels on GTSRB).

Resolution:224	B0(%)	B1(%)	B2(%)
All-Fused	97.3	98.0	97.9
No-Fused	96.8	96.9	95.4
Partial-Fused	98.3	98.4	99.0

3.2.1. PG unit.

In the backbone part, a DepthwiseConv of size 3x3 extends the previous feature. The attention score is computed in the SE module, which makes the model focus on features that are more important to the channel. Then reduce the dimension with 1x1GhostModule. The average pooling layer in the branch section selectively compresses features, acting as gating and local area feature aggregation. Downsampling and fusion are performed using the Ghost module. Finally connect the trunk and branch parts. Stochastic depth is used to prevent network model degradation. The simplified mathematical expression of the whole process is equation (1) [11].

$$F_{out}^{l} = F_{m}^{l} + P^{l}\left(F_{b}^{l}\right) \tag{1}$$

Where F_m^l, F_b^l represent the generated feature by the L-th module backbone and branc. P[l] represents the survival probability of F_b^l, which fits the Bernoulli distribution, $P^l \in [0,1]$. F_m^l, F_b^l are expressed as.

$$F_m^l = GM\{SE[Depth(F_{in}^l)]\} \tag{2}$$

$$F_b^l = \begin{cases} GM\left[Pool\left(F_{in}^l\right)\right], s = 2 \\ GM\left(F_{in}^l\right) \quad\quad , else \end{cases} \tag{3}$$

3.2.2. Fused-PG.

PG uses DepthwiseConv to reduce computation, but it is limited in the early stages. As can be seen from table 1, if all modules use Depthwise, the performance will drop. Therefore, we only use it in the first few stages of the model. In the Fused-PG module, the 1x1 CNN and Depthwise are replaced by 3x3 for convolution to reduce computation, and DepthwiseConv is removed. The simplified mathematical expression is equation (4).

$$F_m^l = GM\{SE[GM\left(F_{in}^l\right)]\} \tag{4}$$

Algorithm 1. Adaptive progressive regularization (APR)

Input: Network blocks length L, initial image size S_0, final image size S_e, initial regularization dropout rate d_0, adjustment factor $\lambda, \beta, \mu, \varrho$
Output: Trained model.

if $L \geq \varrho$ **then**
  Last blockc survival probability: $P_L \leftarrow \frac{\lambda}{L}$
else
  Last blockc survival probability: $P_L \leftarrow 1.0$
end if
for $i = 1$ to L **do**
  Image size or feature map size: $S_i \leftarrow S_0 - (S_e - S_0)\frac{i}{L}$
  Dropout rate: $d_i \leftarrow \frac{S_i}{(S_e - S_0)}$
  Survival probability: $P_i \leftarrow 1 - \frac{i}{L}(1 - P_L)$
  Train model with d_i and P_i
end for

3.3. Adaptive Progressive Regularization

Similar to EfficientNetv2, we consider the regularization problem for the training of a multi-granularity variable model. First, we add the regularization to the network depth. Second, the survival probability problem at stochastic depth is considered. Third, the adaptive probability calculation expression of dropout is improved. In PG-PRNet, the head has more redundant information, and a larger regularization factor is required to improve the generalization ability. In the tail, the features are mapped to a high-dimensional abstract space with smaller features, so a smaller regularization factor is used. For lightweight models, residuals are very important. When the model is very shallow, try to keep the residuals. When the model is complex, the residuals are discarded appropriately. Therefore it is not reasonable to use the same regularization rules all the time. Therefore, the survival probability and dropout rate need to be flexibly adjusted to fit the feature size and network depth. There are identifiers defined as.

- The length of the network module is l, and if l is larger, a higher regularization rate is required, and the ratio of the two is controlled by λ.
- The whole model has M stages. And the features of the middle hidden layer gradually decrease from the first stage to the last stage, and the dropout rate is positively related to the feature map size.

The scale coefficients of feature map size and survival probability are β, μ, respectively, and the overall steps can be described as algorithm 1. The ablation experiments in Section 4.3 further elaborate and demonstrate the effectiveness of APR.

4. Experiments

All our experiments were done on a Nvidia RTX 2080Ti server using Pytorch. In the parameters of adaptive regularization, we set the threshold $\varrho = 11, u = 0.25, \beta = 1, \lambda = 7$. We validate the PG-PRNet feature extraction performance on two tasks on two datasets.

4.1. PG-PRNet for Image Recongnition

The recognition of traffic signs is a challenging real-world problem related to intelligent transportation

systems. The German Traffic Sign Recognition Benchmark (GTSRB) contains more than 50,000 images of daytime and nighttime scenes from 43 categories [12]. Images that are too similar are removed using the Structural Similarity Index (SSMI) algorithm. The mean and variance of the local and global luminance of each image were calculated for adaptive luminance and contrast enhancement, and the distribution of each category was approximated after processing. Using the cross-entropy loss function, Adam optimizer and Cosine Annealing scheduler, we set the weight decay factor = 0.0005, initial learning rate = 0.001, batch size = 64, epoch = 100. the resolution of the design varies from 48 to 224. The training set was preprocessed using random cropping, Gaussian noise. To validate the performance of the feature extractor, a PG-PRNet feature extractor with classification head was added to evaluate its image recognition performance. It mainly includes a global average pooling layer, aggregated features and features compressed by a fully connected layer, and softmax output of category probabilities.

4.2. Result analyse

We use the inference time of a single image with 224 resolution and the amount of parameters as an indicator of network complexity, perform five calculations, and finally take the average. The results are shown in table 2. The PG-PRNet model uses the SE module and GhostModule, so the amount of parameters has been improved, but due to the calculation amount of the two, as well as the use of Hard Sigmoid, layer pruning and DepthwiseConv, therefore, the picture The inference speed is the best (56 ms < 70 ms < 74 ms), where the number of parameters of B0 is second only to EfficientNetV1, but the accuracy of the latter is much lower than our method.

Thanks to the parallel gating unit, our model can obtain good shallow spatial position information

while keeping light weight. Because of the parallel gating, it also has the function of selecting input features in the branch part, and mapping the features to high dimensions.

Table 2. Performance comparison of image recognition tasks on the GTSRB dataset (TOP-1 accuracy (%)).

Methods	48	96	160	224	Params(M)	Infer-time(ms)
PG-PRNet_B0(Ours)	**92.7**	**93.4**	96.8	98.3	3.2	**56**
PG-PRNet_B1(Ours)	92.5	93.1	96.9	98.4	5.4	81
PG-PRNet_B2(Ours)	91.8	93.3	**98.4**	**99.0**	7.2	100
Vision Transformer(P=16)	86.3	89.7	87.5	87.0	10.2	110
EfficientNet V1	89.5	92.3	94.6	96.5	**0.7**	74
EfficientNet V2	92.0	93.4	97.8	98.3	22.4	245
GhostNet	80.6	89.7	96.5	97.7	4.0	70

4.3. Ablation experiments

Two adaptive regularization methods are considered: Dropout and Stochasitc depth. Larger p is used for larger features and smaller p is used for smaller features. The lower dimension contains more spatial location information, but the higher dimension contains more abstract semantic information.

Both kinds of information are very important for inference. It is not reasonable to use the same p and d for the whole structure. In algorithm 1, the survival probability p and dropout rate d are adaptively adjusted according to the size of the feature map. This problem is mitigated to some extent.

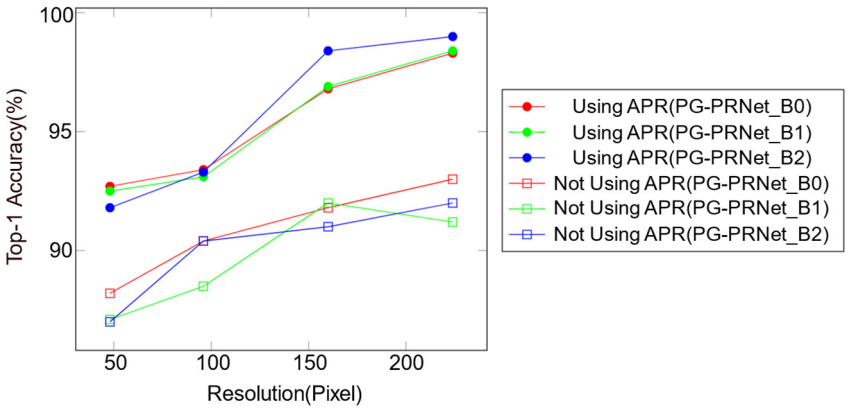

Figure 2. Comparison of Top-1 accuracy with and without adaptive progressive regularization algorithm. Solid dots: With APR. Hollow rectangle: Not APR.

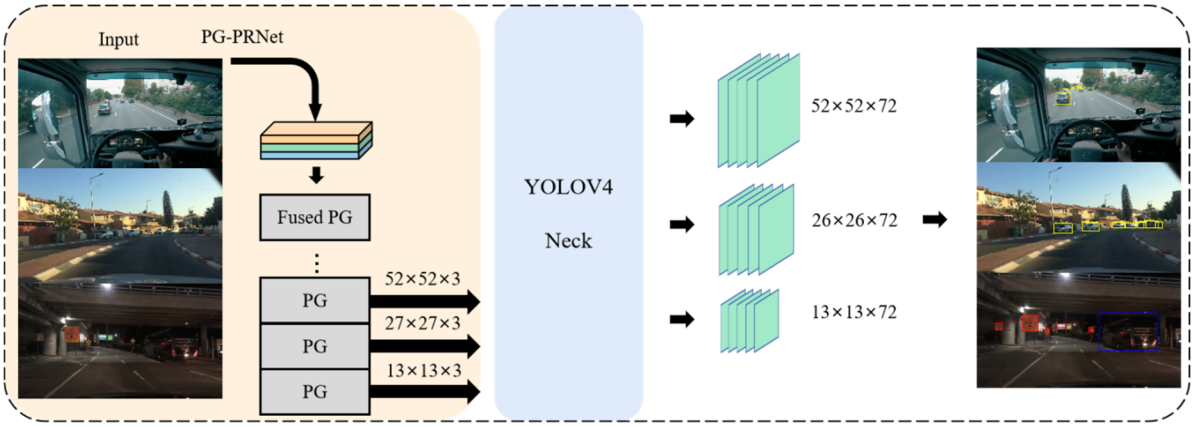

Figure 3. Pipeline to embed PG-PRNet into YOLOv4 model. Using PG-PRNet to generate the feature vectors of the last three layers, through the Neck of YOLOv4, the three output feature matrices are obtained, and the final detection results are generated after some complex post processing.

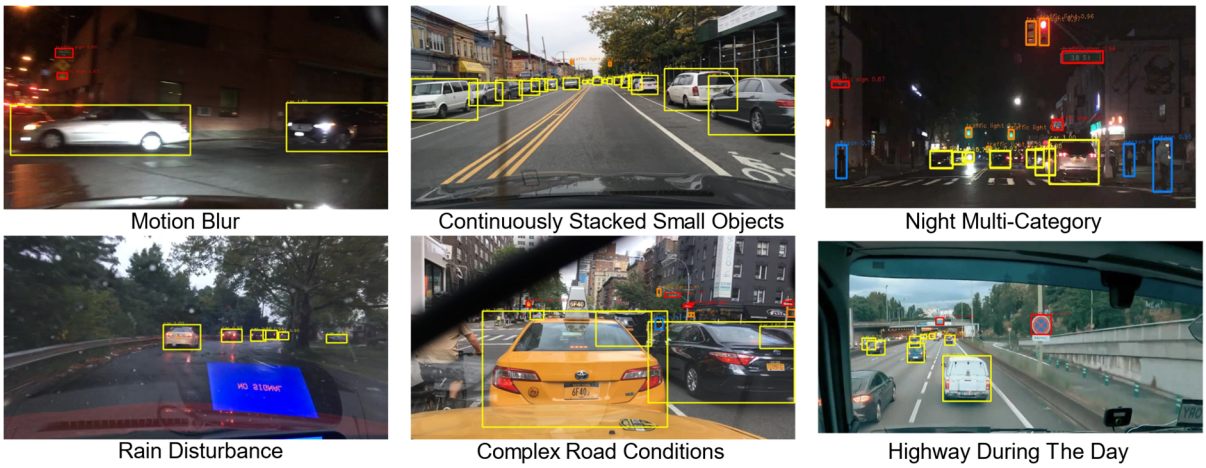

Figure 4. List the detection results under 6 typical target detection difficulties (Video Motion Blur, Continuously Stacked Small Objects, Multi-Category At Night, Rain Disturbance, Complex Road Conditions, Highway During At Day).

4.4. PG-PRNet for Object Detection

BDD100K is a traffic driving video dataset that can be used for a variety of autonomous driving task scenarios, containing up to 100,000 images for 10 task scenarios. In this paper, for the use of lightweight models, we use a subset of 10,000 of these images of autonomous driving scenarios to test performance in terms of target detection. As shown in figure 3, the output features of the last three layers of PG-PRNet are extracted using the Neck of YOLOv4 [13]. Using Mosaic data enhancement, we introduced copy-and-paste data enhancement to improve the detection accuracy of small targets [14]. Finally, the three scales of features are output and the corresponding target detection results are obtained after post-processing (NMS). We list three images as a reference for the results. The parameters are set to batch size = 16, loop scheduler and Adam optimizer.

Table 3. We use the listed model as the Backbone, connected to the YOLOv4 Neck. Compare our proposed method (Bold) with other methods. We complete mAP50, mAP75, single image parameter and inference time calculation on a 608-pixel image.

Method	mAP50(%)	mAP75(%)	Params(M)	Infer-time(ms)
PG-PRNet_B0(Ours)	**55.6**	**27.7**	**11.5**	**311**
PG-PRNet_B1(Ours)	**56.2**	**28.9**	13.0	332
PG-PRNet_B2(Ours)	**56.8**	**30.2**	15.8	356
GhostNet	43.8	19.7	11.9	344
MobileNetv1	49.3	22.5	12.5	320
MobileNetv2	41.9	16.2	10.2	372
MobileNetv3	42.2	18.3	11.4	363
DenseNet121	48.8	20.4	16.5	645
DenseNet169	49.1	20.8	22.6	873
DenseNet201	54.7	22.1	27.8	946

4.5. Results Analyze.

It can be seen that in the case of training only 300 epoches, our model is advanced. The optimum is achieved with 11.5 millions number of parameters and 311 ms inference time, and, the deepened model has a significant performance improvement. This demonstrates that the parallel gating unit effectively improves the feature representation of the branch, making the feature extraction capability of PG-PRNet still highly applicable even after many model compression methods. In figure 4, six

typical difficulties of target detection in real-time traffic scenarios are listed. Our approach maintains high detection accuracy and robustness. A parallel gating unit is used in combination with an adaptive progressive regularization algorithm. The Copy-Paste and Mosaic based approach reduces overfitting, improves model generalization, and enhances performance in scenes with occlusion, too many small targets, rain, multiple categories, and video motion blur.

5. Conclusion

In this work, we propose a lightweight parallel gated feature extraction framework to represent the residual branching information of a given feature in a new cascade, which changes the constant mapping of the residual structure in lightweight networks. In addition, an adaptive progressive regularization algorithm is used to adapt the regularization rules for different size features and different scale networks, called PG-PRNet. The framework is embedded into image recognition and object detection to verify its feature extraction capability, and our model achieves optimality in model volume and accuracy. Its efficiency at variable resolution is demonstrated.

6. References

[1] Tan, M., Le, Q. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. Long Beach, California. 6105-6114.

[2] Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C. and Xu, C. 2020. Ghostnet: More features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. Seattle, WA, USA. (pp. 1580-1589).

[3] Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

[4] Huang, Y., Cheng, Y., Bapna, A., Firat, O., Chen, D., Chen, M. and Wu, Y., (2019). Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems*, 32.

[5] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Las Vegas, NV, USA. (pp. 770-778).

[6] Zoph, B. and Le, Q. V., 2016. Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.

[7] Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Honolulu, HI, USA. (pp. 1251-1258).

[8] Hu, J., Shen, L. and Sun, G., 2018. Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Salt Lake City, UT, USA. (pp. 7132-7141).

[9] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. Venice, Italy. (pp. 618-626).

[10] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. ArXiv, abs/1704.04861.

[11] Huang, G., Sun, Y., Liu, Z., Sedra, D. and Weinberger, K. Q., 2016. Deep networks with stochastic depth. In: European conference on computer vision. Amsterdam, Netherlands. (pp. 646-661).

[12] Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M. and Igel, C., 2013. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In: The 2013 international joint conference on neural networks (IJCNN). Dallas, TX, USA. (pp. 1-8).

[13] Bochkovskiy, A., Wang, C. Y. and Liao, H. Y. M., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[14] Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T. Y., Cubuk, E. D. and Zoph, B., 2021. Simple copy-paste is a strong data augmentation method for instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Online Meeting. (pp. 2918-2928).