# An Improved Vehicle Fine-Grain Identification Algorithm Based on Stochastic Weight Average

**Yankaiqi Li**

*St. Cloud State University, Saint Cloud, Minnesota, 56301, USA*

***Abstract:*** *In the field of motor vehicle recognition, the use of neural network models has become the standard, and the tuning of hyperparameters and loss functions has been shown to be an effective way to improve the performance of these models. However, when using classical convolutional network architectures (e.g., ImageNet) and training them on motor vehicle images with random labels, the overparameterization problem can lead to suboptimal results and an increased risk of recognition failure. P. Ismailova et al. proposed a solution to this problem with the use of weight averaging, which resulted in the development of the simple and effective Stochastic Weight Averaging (SWA) optimizer. In this paper, we apply the SWA method to optimize the original recognition model and demonstrate significant improvements in accuracy through the use of different learning rate schemes with various traditional optimizers. We also identify suitable hyperparameter values to enhance the model's generalization abilities through several experiments, reducing the waste of resources in the motor vehicle recognition task and improving the recognition accuracy of fine-grained images in general, thus increasing the efficiency of related fields.*

***Keywords:*** *Random Average Weighting, Fine-Grained Recognition, Generalization capability, Computer Vision*

## 1. Introduction

With the rapid development of urbanization, the traffic system has generated intelligent hardware needs, and the probability of accidents is gradually increasing in the face of the increasing urban traffic. And more is hanging you bring a huge workload in motor vehicle type identification. Although the occurrence of the accident contains human irresistible factors, but the inability to effectively monitor the road traffic is the main drawback at present. In the current existing system, the main steps focus on the identification and judgment of the captured vehicle, and the judgment of the model is often more accurate and effective than the traditional identification based on license plate number. Based on this motivation, an accurate and highly automated recognition system is in demand, and fine-grained recognition can be more accurate than traditional manual feature extraction methods in such scenarios where the focus is on the features of a given object subclass. In the field of vehicle subclass classification, it is relatively easy to develop algorithms that achieve high levels of accuracy on the training set, but it is more challenging to improve the model's generalization abilities and maintain high levels of performance on unseen data. In addition to increasing the sample size, the use of state-of-the-art (SOTA) modules or losses may not always improve the model's performance. To address this problem, we need to explore new approaches that can improve the model's generalization capabilities. One potential solution is to average multiple models, as this has been shown to provide better robustness and generalization performance than using a single model. However, this approach comes with a higher computational cost.

## 2. Loss Function and Generalization

In machine learning, a loss function, also known as a cost function, is a special type of function that maps the predicted values of a model to non-negative real numbers to indicate the "risk" or "loss" associated with those predictions. The loss function is used to evaluate how well the model's predictions match the true values, with a better loss function typically leading to a better-performing model. Different models may use different loss functions, depending on the specific task they are designed to solve. In general, the loss function is an important component of a machine learning model, as it provides a way to measure the model's performance and guide its optimization.

The curvature of a model's loss function can be mathematically quantified using the Hessian matrix, which is a square matrix of second-order partial derivatives that measures the curvature of the function in each direction. The figure below shows the loss function for a model, which illustrates how sensitive the loss function is to changes in the model's parameters. A low value of Loss indicates that the curvature of the loss function is low, and the model's parameters are not very sensitive to changes in the training data, leading to better generalization and a lower risk of overfitting. In contrast, a high value of Loss indicates that the curvature of the loss function is high, and the model's parameters are very sensitive to changes in the training data, which may cause the model to overfit the training data and perform poorly on new, unseen data. In general, models whose parameters converge to a flat minimum region, such as around the saddle $P_1$, Keskar and Nitish Shirish et al. [2] noted that those points tend to have better generalization abilities because the low curvature of the flat minimum region makes the model's parameters less sensitive to changes in the training data, resulting in a model that does not overfit the training data and is more likely to perform well on new, unseen data. (As shown in  Figure 1)

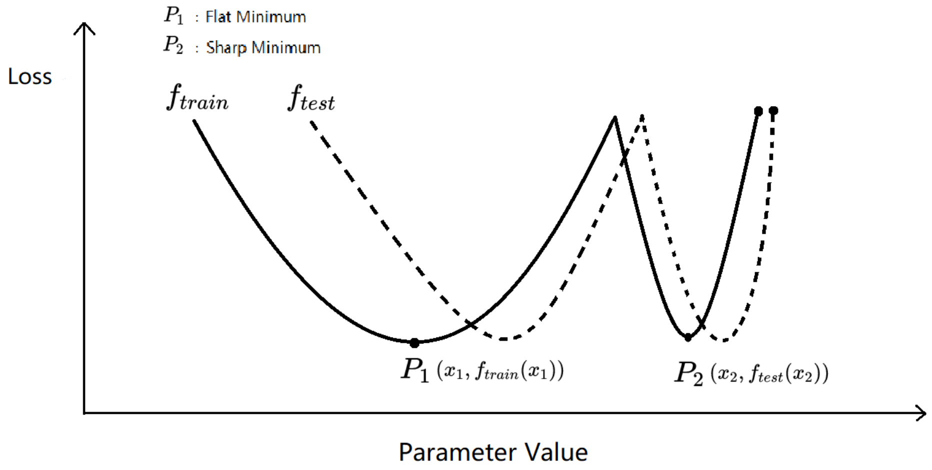


*Figure 1: The solution of the loss function located in different regions of the function image has a different impact on the generalization ability of models*

## 3. Stochastic Weight Average

### *3.1 SWA Program introduction*

Stochastic Weight Averaging (SWA) is a method for improving the generalization ability of deep learning models using stochastic gradient descent, and it does not add additional computational overhead during training. The results of the SWA paper [1] show that taking a simple average of multiple points along the SGD trajectory at a periodic or constant learning rate leads to better generalization than traditional training methods. SWA has been shown to significantly improve the generalization ability of common computer vision tasks, including VGG and Dense Nets on the ImageNet and CIFAR benchmarks, and it can be a valuable alternative to other optimization methods. This method offers a simple and effective way to improve the generalization of deep learning models without requiring additional computational resources.

### *3.2 Principle of SWA Program*

The idea behind Stochastic Weight Averaging (SWA) is based on the observation that, in many experiments, the loss values at the end of each learning cycle tend to accumulate at the edges of the loss plane and are generally far from the center. The loss values on these edge regions can be approximated as points W1, W2, and W3 on the red regions in the figure below, which are associated with low loss. By averaging the set of these values, we can obtain lower loss values with a higher probability and produce more uniform generalization results for the experiment, which in turn improves the model's generalization ability. This approach offers a simple and effective way to improve the generalization of deep learning models, and it has been shown to be effective in a variety of computer vision tasks. (As shown in  Figure 2)

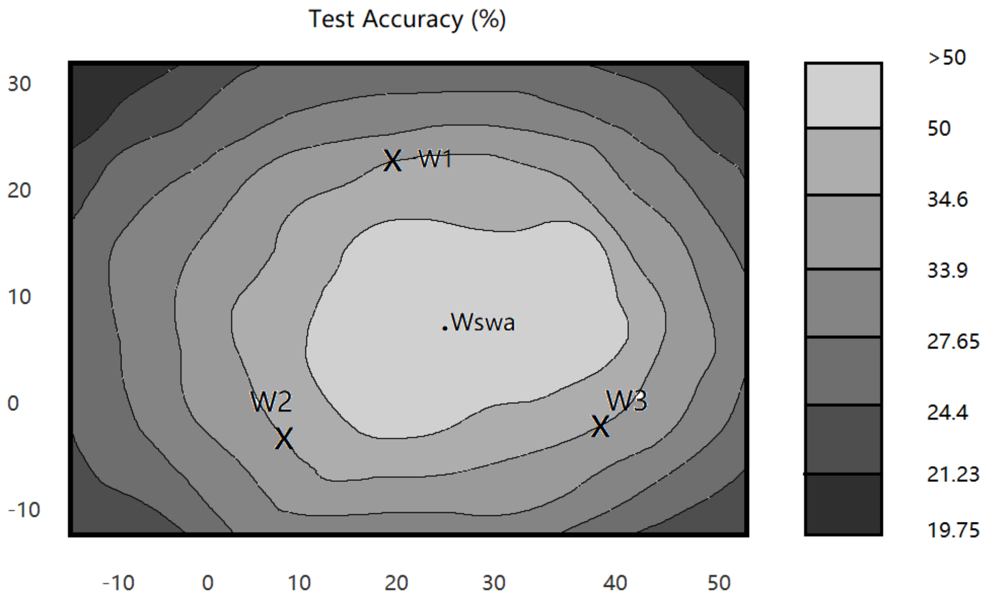


*Figure 2: Illustration of Stochastic Weight Average*

Training deep neural networks (DNNs) typically involves using stochastic gradient descent (SGD) to optimize the model weights θ.

$$\Delta\theta_t = -\eta_t \left(\frac{1}{B}\sum_{i=1}^{B} \nabla_\theta \log p\left(y_i \mid f_\theta(x_i)\right) - \frac{\nabla_\theta \log p(\theta)}{N}\right) \quad (1)$$

Where the learning rate is $\eta$, The $i$-th input (e.g., a specific image) and the tag can be $\{x_i, y_i\}$

The size of the entire training set is denoted as $N$， The size of each Batch is expressed as $B$。

For a deep neural network (DNN) $f$, with a weight parameter θ.[2]

The loss function for this can be expressed as a negative log-likelihood combining the regularizer $\log p\,(\theta)$, expressed as the following function.

$$\sum_i \log p\left(y_i \mid f_\theta(x_i)\right) \quad (2)$$

This type of maximum likelihood training does not represent prediction uncertainty or parameters $\theta_i$.

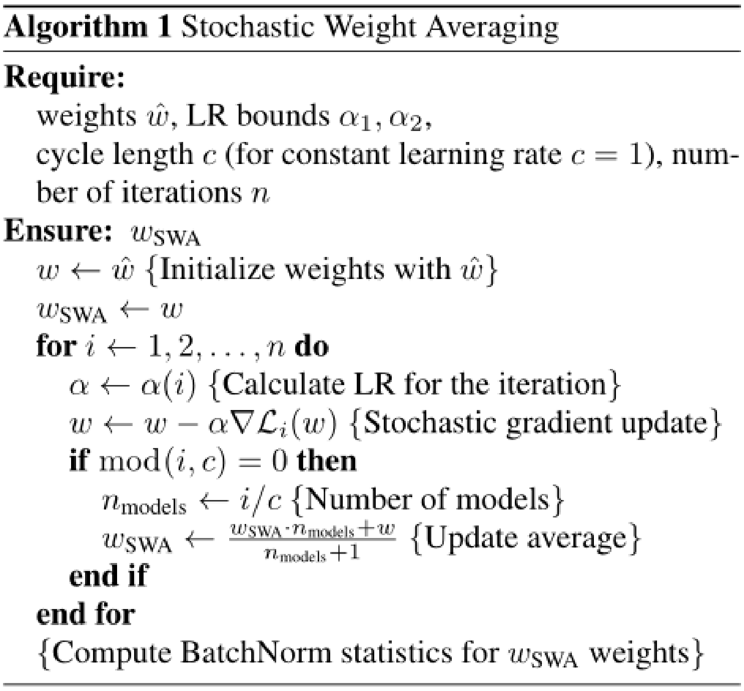
The main idea of SWA is to first start with a pre-trained solution, perform SGD with a constant learning rate schedule, and average the weights of the models it traverses. The weights of the network obtained after the $i$ epoch of SWA training is denoted as $\theta_i$, while the SWA solution after $T$ Epochs is given by the following equation.

So far, we have obtained the weight expressions after SWA processing.

$$\theta_{\mathrm{SWA}} = \frac{1}{T}\sum_{i=1}^{T} \theta_i \quad (3)$$

### *3.3 Implementation of SWA method*

In the paper, P. Izmailov provides a pseudo-code [1] for the SAM algorithm, and we have created flow chart to represent it more intuitively. (As shown in Figure 3)



```
Algorithm 1 Stochastic Weight Averaging
Require:
    weights $\hat{w}$, LR bounds $\alpha_1, \alpha_2$,
    cycle length $c$ (for constant learning rate $c = 1$), number of iterations $n$
Ensure: $w_{\text{SWA}}$
    $w \leftarrow \hat{w}$ {Initialize weights with $\hat{w}$}
    $w_{\text{SWA}} \leftarrow w$
    for $i \leftarrow 1, 2, \ldots, n$ do
        $\alpha \leftarrow \alpha(i)$ {Calculate LR for the iteration}
        $w \leftarrow w - \alpha \nabla \mathcal{L}_i(w)$ {Stochastic gradient update}
        if $\mathrm{mod}(i, c) = 0$ then
            $n_{\text{models}} \leftarrow i/c$ {Number of models}
            $w_{\text{SWA}} \leftarrow \frac{w_{\text{SWA}} \cdot n_{\text{models}} + w}{n_{\text{models}} + 1}$ {Update average}
        end if
    end for
    {Compute BatchNorm statistics for $w_{\text{SWA}}$ weights}
```

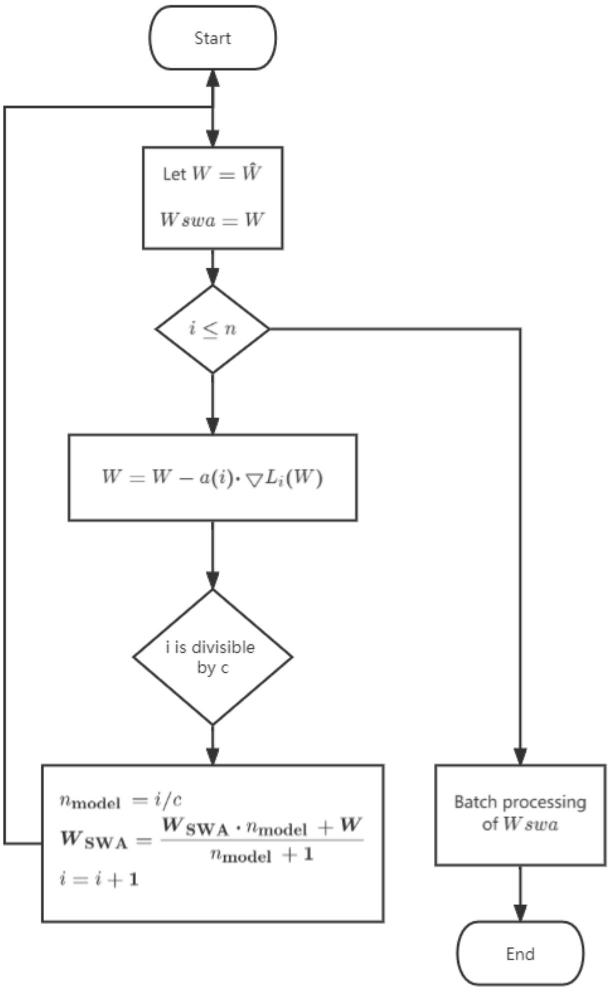

Start
Let $W = \hat{W}$
$Wswa = W$
$i \leq n$
$W = W - a(i) \cdot \nabla L_i(W)$
i is divisible by c
$n_{\text{model}} = i/c$
$W_{\text{SWA}} = \frac{W_{\text{SWA}} \cdot n_{\text{model}} + W}{n_{\text{model}} + 1}$
$i = i + 1$
Batch processing of $Wswa$
End

*Figure 3: The flow chart*

## 4. Experimental

### *4.1 Experiment purpose*

The SWA method has been shown to be effective in fine-grained classification tasks by allowing us to adjust the random weight learning rate (SWALR) to achieve optimal accuracy on the current model. Meanwhile, we obtained the source code from SWA's GitHub details page and how to use it to experiment with the unchanged version [5]. By using different value strategies, we can average the data from multiple states during the training process to find a solution that is centered on a flat region close to the expected loss, as well as a specific value. The use of a modified learning rate scheme in conjunction with other optimizers has been shown to improve the model's generalization capabilities and maintain consistent performance on enhanced test sets that differ significantly from the source images.

### *4.2 Experimental equipment*

To verify the effectiveness of the SWA method under different learning rate schedules and SWA_LR value settings, we conducted experiments using the Stanford Cars, The Cars dataset contains 16,185 images of 196 classes of cars. The data is split into 8,144 training images and 8,041 testing images, where each class has been split roughly in a 50-50 split [3]. And CIFAR-10, and CIFAR-100 datasets have been used with the Resnet-50, Resnet-101, and Resnet-152 models on a test platform consisting of a CPU: i5-10500F, GPU: RTX-1060 Laptop, RAM: 24 GiB, and CUDA: Version 11.8. We selected a portion of the car images from the network that matched the training set labels for training and provided a processed test target to increase the test difficulty, allowing us to evaluate the model's performance under non-ideal recognition conditions.

### *4.3 Result of experiment*

The results of our experiments, including the hyperparameters used, are shown in Table 1. These results demonstrate the effectiveness of the SWA method in improving the accuracy of the models on the test set. The same networks were trained by the conventional SGD method and SWA method, respectively. According to Zhang Xiang's research, random cropping and horizontal flipping of the images in the training set before training can achieve data augmentation [7], so a similar method is used for data pre-processing in this paper. The recognition accuracy of each individual sub-network using different methods on the Stanford Car test set is then recorded. All the results reflect a significant improvement compared to SGD and run faster for different SWALR values.

*Table 1: SWA enhancement compared to conventional SGD at different SWA LR*

| Model | Epoch | Optimizer | SWA LR | SGD LR | Accuracy |
|---|---|---|---|---|---|
| Resnet-50 | 20 | SWA+SGD | 1.00e-6 | 0.005 | 83.62 |
| | | SWA+SGD | 1.00e-5 | | 83.76 |
| | | SWA+SGD | 1.00e-4 | | 83.94 |
| | | SWA+SGD | 1.00e-3 | | 84.04 |
| | | SWA+SGD | 1.00e-2 | | 84.13 |
| | | SWA+SGD | 1.00e-1 | | 84.21 |
| | | SGD | N/A | | 83.16 |

### *4.4 Comparison of test results*

We tested more complex image sets using a migration learning approach. The images in the Craigslist dataset [4] and the Stanford Car samples share many common features, making it possible to use existing models trained on these datasets to address the problem of specific object classification. We consider two strategies for adjusting the learning rate during training. Haoyang et al. summarized the relevant learning rate strategies in their study related to SWA [6]. The first is a fixed learning rate schedule, where we use learning rates of 0.02, 0.002, and 0.0002, corresponding to the learning rates used in the different training phases of the pre-trained model. Table 2 shows the accuracy rates for different models and different learning rate strategies. These results demonstrate the effectiveness of the proposed approach in improving the performance of the model on the Craigslist dataset. The second strategy is a cyclic learning rate schedule, where the learning rate starts at a high value, decreases to a minimum value, and then increases again to a maximum value. It is important to note that the learning rate decreases at each iteration, rather than at each epoch. Table 3 reflects the results of different Resnet models after adopting the cosine annealing algorithm cosine annealing learning) selected two groups (lrmax, lrmin), i.e. (0.01, 0.0001), (0.02, 0.0002), (0.03, 0.0003) and selecting 1 epoch as the loop length.

*Table 2: Test results on Resnet network with different fixed learning rates*

| Model | Epoch | Optimizer | Strategy | AP |
|---|---|---|---|---|
| Resnet-50 | 24 | SWA+SGD | FixedLr=0.02 | 81.88 |
| | | | FixedLr=0.002 | 82.16 |
| | | | FixedLr=0.0002 | 82.98 |
| Resnet-102 | 24 | | FixedLr=0.02 | 78.71 |
| | | | FixedLr=0.002 | 79.35 |
| | | | FixedLr=0.0002 | 79.82 |
| Resnet-101 | 24 | | FixedLr=0.02 | 80.11 |
| | | | FixedLr=0.002 | 80.96 |
| | | | FixedLr=0.0002 | 81.45 |

*Table 3: Accuracy results of applying the cyclic learning rate scheme on Resnet*

| Model | Epoch | Optimizer | Strategy | AP |
|---|---|---|---|---|
| Resnet-50 | 48 | SWA+SGD | cyclr=0.01, 0.0001<br>cyclen=1 | 81.93 |
| | | | cyclr=0.02, 0.0002<br>cyclen=1 | 82.14 |
| | | | cyclr=0.03, 0.0003<br>cyclen=1 | 81.07 |
| Resnet-102 | 48 | | cyclr=0.01, 0.0001<br>cyclen=1 | 80.43 |
| | | | cyclr=0.02, 0.0002<br>cyclen=1 | 80.84 |
| | | | cyclr=0.03, 0.0003<br>cyclen=1 | 80.29 |
| Resnet-101 | 48 | | cyclr=0.01, 0.0001<br>cyclen=1 | 81.18 |
| | | | cyclr=0.02, 0.0002<br>cyclen=1 | 81.22 |
| | | | cyclr=0.03, 0.0003<br>cyclen=1 | 81.68 |

## 5. Conclusion

In this paper, we systematically study the effectiveness of SWA in vehicle target detection and model recognition. We find that using both SWA and SGD, the accuracy of model recognition can be improved by changing only the SWA learning rate while keeping the SGD learning rate constant, and the computation time can be significantly reduced. Additionally, training a vehicle classification model with a cyclic variable learning rate during the experiment can improve the accuracy of this learning model by about 1 percentage point on the Stanford Car test set and the Craigslist test set after averaging the computation as the final weights of the model. We achieved an accuracy of 84.2 on the Stanford Car dataset and 82.98 on Craigslist and found the most suitable cyclic learning rate pair (0.02, 0.0002), respectively. Our experimental results show that this technique is applicable to a variety of network models used in image recognition, including Resnet-50, Resnet-101, and Resnet-102. We hope that our work will make more computer vision practitioners and car recognition experts aware of this simple yet effective method and help them train better neural network models for production.

## References

*[1] Izmailov P., Ashukha A., Gulin A., & Vetrov D. (2018). Averaging weights leads to wider optima and better generalization. In Conference on uncertainty in artificial intelligence (pp. 412-421).*

*[2] Keskar N. S., Dauphin Y. N., & Socher R. (2017). On large-batch training for deep learning: Generalization gap and sharp minima. arXiv preprint arXiv:1609.04836.*

*[3] Krause J., Stark M., Deng J., & Fei-Fei L. (2013, December). 3D object representations for fine-grained categorization. In 2013 IEEE international conference on computer vision workshops (pp. 554-561). IEEE.*

*[4] Information on: https://github.com/AustinReese/UsedVehicleSearch.*

*[5] Timgaripov. (n.d.). Timgaripov/SWA: Stochastic weight averaging in Pytorch. Retrieved from https://github.com/timgaripov/swa.*

*[6] Zhang, H., Zhu, Y., Hu, S., He, T., & Sun, J. (2020). SWA object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5104-5112).*

*[7] Zhang X., Shi Z., & Chen L. (2020). Expression recognition method based on cascade network optimized by SWA. Electronic Science and Technology, 33(9), 16-20.*