Spectral Proximal Method and Saliency Matrix for Robust Deep Learning Optimization

Cherng-Liin Yong, Ban-Hoe Kwan, Danny-Wee-Kiat Ng, and Hong-Seng Sim

Abstract—**This paper presents a new training optimizer for deep learning models, called the Spectral Proximal (SP) method with saliency matrix, that aims to improve their ability to generalize to new data. Generalization is the measure of how well a model can perform on data that it has not seen during training. The SP method addresses a pair of hurdles affecting generalization: the problem of gradient confusion within complex model architectures and the limited availability of training data. The key innovation of the SP method is the use of a proximal operator with a saliency matrix, which adjusts the descent direction based on the importance of each parameter and avoids overfit issues. This leads to improved performance on image classification (MNIST and CIFAR-10) and object detection (YOLOv7) tasks and better ability to generalize to new data. We conducted a comprehensive inquiry by performing experiments on various configurations while controlling for potential confounding factors. The SP method consistently outperformed the baseline method based on the results.**

Index Terms—**spectral proximal method, saliency matrix, deep learning, machine vision, optimization algorithm.**

I. Introduction

IMAGE classification and object detection tasks can benefit from the use of deep network models, which can learn from extensive datasets and perform elaborate computations to generate reliable results. However, these models have some drawbacks, such as requiring a lot of computing power and being susceptible to poor optimization result. To overcome these challenges, different optimization methods have been proposed, including Stochastic Gradient Descent (SGD) method and Adam method.

The SGD method is light and robust, therefore it does not need much hyperparameter tuning, but it can be affected by noisy datasets because it scales the gradients uniformly in all directions. In contrast, Adam method is a more advanced method that incorporates moment estimation and bias corrective measures. Adam has been proven to work well on large-scale optimization problems and to converge faster than SGD [1]. However, Adam also has some disadvantages, such as over-optimizing to train data, which results in poor generalization compared to SGD [2]–[4].

Manuscript received October 13, 2023; revised January 19, 2024. Date of publication February 26, 2024. Date of current version February 26, 2024. The associate editor prof. Renata Lopes Rosa has been coordinating the review of this manuscript and approved it for publication.

Authors are with the Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Jalan Sungai Long, Bandar Sungai Long, Kajang, Selangor, Malaysia (e-mails: y_c_l_@live.com, kwanbh@utar.edu.my, ngwk@utar.edu.my, simhs@utar.edu.my).

Digital Object Identifier (DOI): 10.24138/jcomss-2023-0124

Some researchers have tried to use second-order methods that take the Hessian matrix of the parameters into account to improve the performance of first-order methods. However, these methods can be very costly in terms of computation [5].

Hence, within the scope of this paper, we introduce an innovative training method tailored for deep learning models, denoted as the Spectral Proximal method with a saliency matrix. The principal novelty underlying the SP method lies in its integration of a proximal operator in tandem with a saliency matrix. This dynamic strategy orchestrates adjustments to the descent direction during the optimization process, factoring in the significance of each model parameter, thereby effectively mitigating the potential for overfitting. Consequently, the SP method yields notable enhancements in model performance across an array of tasks, encompassing image classification challenges as well as object detection task.

Our method has a high convergence rate and excellent generalization properties, outperforming established and well-tested methods across various small and large-scale optimization challenges, as demonstrated through comprehensive experimental evaluations. Furthermore, we have analyzed our method's strengths and weaknesses and provided comprehensive information to support future research and improvement.

The paper is organized as follows: Section II reviews the related literature. Section III presents the derivation steps for the Spectral Proximal (SP) method and the Saliency matrix. Then, the convergence analysis of the novel method is provided. Section V shows our experimental results and comparisons with other methods. Section VI summarizes our contributions and highlights the advantages of the SP method. We also include supplementary data in the appendices, such as the ResNet-18 training results in Appendix A and Yolov7 training results in Appendix B, to give a complete overview of our work.

II. Related Work

This section provides an overview of the most relevant optimization methods used in machine learning and computer vision. We first present second-order methods, which use the curvature of the objective function to achieve faster convergence and better accuracy. We also discuss quasi-Newton methods, which approximate the second-order information using gradient updates. Additionally, we present spectral gradient methods, which minimize the gradient spectral norm of the cost function. Finally, we introduce proximal algorithms, which are useful for dealing with non-smooth and constrained optimization problems. We also explain how saliency maps can be used to visualize and highlight the important parameters.

A. Second Order Methods

Newton's approach represents a second-order optimization method that employs the Hessian matrix to enhance the optimization procedure. The updating mechanism for Newton's method is characterized by the following rule:

$$\theta_t = \theta_{t-1} - H^{-1} \cdot g_t, \tag{1}$$

where the previous weight vector θ_{t-1} subtract the multiplicative product of inverse Hessian matrix and loss gradient vector g_t yields the next weight vector θ_t

As highlighted by Wang et al. [6], a significant limitation of this approach lies in its substantial computational demand for Hessian matrix inversion, which can hinder its practical applicability. In an effort to mitigate computational expenses, Wang et al. [6] introduced a subsampling technique, resulting in a method that attains results on par with the SGD method during the training of convolutional neural networks (CNNs).

One of the main drawbacks of the second order method is that it requires the computation of the Hessian matrix, which is the matrix of second derivatives of the objective function. The Hessian matrix can be very large and expensive to compute, especially for high-dimensional problems. Moreover, the Hessian matrix may not be positive definite, which means that the second order method may not converge to a local minimum. In contrast, the first order method only requires the computation of the gradient vector, which is much cheaper and easier to obtain.

B. Quasi-Newton Methods

The Quasi-Newton (QN) method is a second-order optimization algorithm that uses the secant equation with the finite difference approach to approximate the Hessian matrix. This approach provides the benefit of second-order data utilization in optimization while simultaneously avoiding explicit Hessian calculation and reducing computational needs.

The secant equation is:

$$\delta g_t = B_{t+1} \delta\theta_t, \tag{2}$$

where $\delta g_t = g_{t+1} - g_t$, $\delta\theta_t = \theta_{t+1} - \theta_t$, and the Hessian approximation is B_{t+1}.

C. Spectral Gradient Methods

The Barzilai-Borwein (BB) method is a spectral gradient method that utilizes the secant equation for an estimation of the inverse Hessian matrix while adjusting the learning rate. The BB method is using $\lambda_{t+1} \cdot I$ to approximate the inverse Hessian matrix, where the learning rate is λ_{t+1}. Here are the two learning rate updating formula:

$$\lambda_1 = \frac{\delta\theta_{t-1}^T \delta\theta_{t-1}}{\delta\theta_{t-1}^T \delta g_{t-1}}, \tag{3}$$

$$\lambda_2 = \frac{\delta\theta_{t-1}^T \delta g_{t-1}}{\delta\theta_{t-1}^T \delta g_{t-1}}. \tag{4}$$

Sim et al. [7] suggested a spectral gradient approach that incorporates a damping matrix for the adaptation of the descent direction during the training process. The damping matrix is computed by using the spectral decomposition of the Hessian matrix and applying a damping function to the eigenvalues. The damping function is designed to reduce the influence of small and negative eigenvalues, which can cause instability and slow convergence. In a numerical experiment conducted, the method exhibited exceptional performance when compared to other widely used approaches.

Dai et al. [8] introduced another spectral gradient method which incorporates a blend of both learning rate formulas for updating the learning rate within the BB method framework. The method uses the following equation and multiple policies for choosing γ :

$$\lambda_{t+1} = \gamma\lambda_1 + (1-\gamma)\lambda_2. \tag{5}$$

The method balances the influence of both formulas by using a parameter λ, which can be selected by different policies, such as the Armijo rule, constant, adaptive, or random. The method has been shown to have better convergence properties and robustness than the BB method.

Wang et al. [9] unveil a pioneering spectral conjugate gradient approach, meticulously tailored to address the complexities of large-scale unconstrained optimization challenges. The method draws inspiration from the approximate optimal stepsize strategy commonly employed in gradient-based methods leading to the formulation of a novel truncating framework for determining spectral and conjugate parameters. The method outperforms several established methods across an extensive battery of 130 test problems.

Laylani et al. [10] introduces a fresh perspective on spectral descent methodologies, specifically addressing spectral conjugate gradient estimation through equation (6). Our approach is characterized by its simplicity and its capacity to enhance the efficiency of gradient-based algorithms while keeping storage demands to a minimum. The techniques yield substantial reduction in number of iterations (NI) and the number of function evaluations (NF).

$$0 < \beta_t = \frac{g_{t+1}^T d_{t+1}}{g_t^T d_t}, \tag{6}$$

where, β_t is the conjugate parameter and d_t is the descent direction.

D. Proximal Algorithm

Proximal algorithms constitute a category of optimization techniques that can solve large and non-smooth optimization problems. These problems are often difficult to solve by using gradient-based methods, such as SGD, because the gradients may not exist or may be noisy. Proximal algorithms use proximal operators to find the optimal solution by iteratively minimizing a sequence of simpler subproblems.

The proximal operator is a mathematical function that takes a point and a parameter as inputs and returns another point that is closer to the optimal solution. The proximal operator is defined as:

$$\mathbf{prox}_f(v) = \underset{\theta}{\operatorname{argmin}}(f(\theta) + (1/2)|\theta - v|_2^2), \quad (7)$$

The proximal operator is a function that Generates a convex approximation with smooth characteristics of the objective function and calculate the optimal solution with it. The proximal operator can be seen as a regularization technique that imparts a "smoother" quality to the function prior to optimization [11].

Santos & Souza [12] delves into a novel proximal point method for solving quasi-equilibrium problems in Hilbert spaces, expanding upon prior methodologies. By iteratively addressing equilibrium problems, we achieve weak convergence to QEP solutions under mild assumptions. Our numerical experiments provide empirical evidence of the method's efficacy.

In their research, Yang et al. [13] utilized the proximal operator to enhance the performance of SGD with momentum. Their findings revealed that this modified approach produced outcomes on par with established training methods commonly employed in the field.

In addition to incorporating the proximal operator, Tang and Scheinberg [14] introduced a Quasi-Newton (QN) technique featuring a strategic selection approach that prioritizes greedy active elements. This approach entails the selection of indices that meet the following criteria:

$$I^{(1)} = i \in P|(\delta f(\theta))i \neq 0, \quad I^{(2)} = i \in P|(\theta)i \neq 0, \quad (8)$$

This approach entails the periodic selection of indices that fulfill two criteria: they must be non-zero elements ($I^{(2)}$) and demonstrate substantial descent in the objective function ($I^{(1)}$). This selection strategy aims to achieve a sparse solution.

Yun, Lozano & Yang [15] introduced a pioneering approach to train structured neural networks, venturing into the domain of stochastic proximal gradient methods aimed at addressing challenges posed by non-smooth and non-convex problems. This is facilitated by the integration of a proximal operator that accommodates both non-convex loss functions and regularizers. Notably, the empirical findings emphasize the heightened effectiveness of proximal methodologies compared to their subgradient counterparts, particularly in scenarios entailing non-convex regularization, thus shedding light on their prospective value in complex neural network designs.

E. Saliency Map

A saliency map visually represents the significance or pertinence of distinct regions within an image or other visual data, frequently employed in computer vision and machine learning to gain insights into the decision-making processes of neural networks and other models. The saliency map is crafted by emphasizing specific regions within the image or dataset that exert an impact on the model's output or prediction. In our methodology, we derive inspiration from both network parameter saliency and visual saliency to calculate the saliency matrix.

In their research, LeCun et al. [16] introduced the pioneering concept of neural network pruning as a method for complexity reduction while maintaining performance. They introduced an algorithm explicitly crafted to detect and remove the least impactful connections within a pre-trained network. Their findings demonstrated a substantial reduction in the number of connections without compromising performance. This research has been instrumental in advancing the field of neural network pruning and has left a lasting impact on the evolution of efficient neural network architectures.

In their research, Jia et al. [17] presented an innovative adaptive framework designed to detect prominent objects in images. This framework dynamically chooses an optimal strategy for every pixel, taking into account its contextual connections. The process commences by generating an initial saliency map using the global contrast technique and then analyzing the neighboring context of each region. This information guides the application of various methods to process individual regions, ultimately generating the conclusive saliency map. Extensive experimentation conducted on three benchmark datasets demonstrates the superior performance of this approach compared to other contemporary methods.

Amorim et al. [18] addresses the interpretability of deep learning models in medical image analysis, particularly in the context of histopathological image classification. While saliency maps are commonly used for interpretation, the lack of systematic evaluation tools has been a challenge. To tackle this issue, the authors propose an innovative approach that evaluates the reliability of saliency maps through the incorporation of natural image perturbations via a counter-class substitution technique. This results in the generation of meaningful evaluation metrics. Their validation on a breast cancer metastases detection dataset demonstrates the sensitivity and relevance of this approach, making it a promising solution for saliency map validation in medical imaging and beyond.

Kremer et al. [19] tackles the essential challenge of simulating visual attention in virtual humans and introduces an innovative contribution through the development of a parametric model and methodology for the real-time generation of saliency maps as perceived by virtual agents. In contrast to traditional approaches, this model aggregates saliency assessments using user-defined parameters related to objects and characters visible to the agent, resulting in the creation of a 2D saliency map. Its distinctive feature lies in its capability to incorporate three-dimensional data and the character's level of attentiveness via an attention field. The adaptable and parameterized framework of this technique provides users with the ability to replicate a wide spectrum of agents while retaining the flexibility to extend the model by incorporating supplementary layers and parameters. Furthermore, its integration with standard and abnormal models of the human visual field and gaze control mechanisms can significantly enhance its applicability across a wide range of use cases.

III. Method Derivation

We address the problem of stochastic optimization. To find the optimal value of the parameter, θ, that minimizes the loss function, $L(\theta)$

$$O(\theta) = \underset{\theta}{\operatorname{argmin}} \mathbb{E}[L(\theta)], \quad (9)$$

Optimizing θ, that minimizes the smooth, real-valued loss function, $L(\theta)$, where $L(\theta) : \mathbb{R} \to \mathbb{R}$ and the expected loss value is $\mathbb{E}[L(\theta)]$. The gradient of the loss function with respect to θ is $g(\theta) = \nabla_\theta L(\theta)$, and the second-order loss function derivative is $H(\theta) = \nabla^2_\theta L(\theta)$, i.e. the Hessian matrix.

A. Notation

We use the following notations to simplify the expressions: let $x \in \mathbb{R}$, $y \in \mathbb{R}$ be output, $\theta \in \mathbb{R}$ be the model variables, $L(\theta)$ be the cost function, $g(\theta) = L^{'}(\theta)$ be the derivative of the loss function, i.e. gradient, $H(\theta) = L^{''}(\theta)$ be the Hessian matrix, $S(\theta)$ be the saliency matrix, and $Q_p = p(n+1)$ term of the saliency matrix in ascending order be the p-quantile of the saliency matrix.

B. Multiple Damping Gradient

One drawback of the SGD method is that it can converge slowly due to the bad conditioning of the Hessian matrix and the variance of the stochastic gradient [5]. To address this issue, we embrace the multiple damping gradient method introduced by Sim et al. [7]. The formula for updating in this method is as follows:

$$\theta_{t+1} = \theta_t - B^{-1} \cdot g_t, \tag{10}$$

where B^{-1} is the inverse damping matrix as approximation of Hessian matrix inverse in Equation (1).

By projecting the difference between matrices B and H along the search direction and constraining it within the bounds of matrix H, the damping matrix B undergoes an update. The objective of this technique is to reduce the gradient while adhering to the limitation imposed by matrix H, thereby averting the oscillatory behavior often associated with SGD descent. By adding a constraint on the elements of B_t, based on minimizing the log-determinant norm and following the secant equation, we can get an updated formula for B_t, denoted as $B_(t+1)$.This updated solution applies to any positive matrix B.

$$\min \ \mathrm{tr}(B_{t+1}) - \ln(\det(B_{t+1})) \tag{11}$$

$$\text{s.t.} \quad \delta\theta_t^T B_{t+1} \delta\theta_t = \delta\theta_t^T \delta g_t. \tag{12}$$

Suppose $B_{t+1} = \mathrm{diag}(B^{(1)}_{t+1}, ..., B^{(n)}_{t+1})$ and $\delta\theta_t = \delta\theta_t^{(1)}, ..., \delta\theta_t^{(n)}$, The minimization problem (11) and (12) is express as:

$$\min(\sum_{i=1}^{n} B^{(i)}_{t+1}) - \ln(\prod_{i=1}^{n} B^{(i)}_{t+1}) \tag{13}$$

$$\text{s.t.} \quad (\sum_{i=1}^{n} (\delta\theta_t^{(i)})^2 B^{(i)}_{t+1}) - \delta\theta_t^T \delta g_t = 0. \tag{14}$$

Therefore, the definition of Lagrangian corresponding to equations (11) and (12) is as follows:

$$\begin{aligned} L(\alpha, \omega) = (\sum_{i=1}^{n} D^{(i)}_{t+1}) - \ln(\prod_{i=1}^{n} D^{(i)}_{t+1}) \\ + \omega[(\sum_{i=1}^{n} (s_t^{(i)})^2 D^{(i)}_{t+1}) - s_t^T y_t]. \end{aligned} \tag{15}$$

ω is the Lagrange multiplier. After that, equating the partially differentiate (15) w.r.t. $B^{(i)}_{t+1}$ to zero:

$$\frac{\delta L}{\delta B^{(i)}_{t+1}} = 1 - \frac{1}{B^{(i)}_{t+1}} + \omega(\delta\theta_t^{(i)})^2 = 0, \quad i = 1, 2, ..., n \tag{16}$$

$$B^{(i)}_{t+1} = \frac{1}{1 + \omega(\delta\theta_t^{(i)})^2}, \quad i = 1, 2, ..., n \tag{17}$$

Equation (17) is substitute into the constraint (14) yields:

$$F(\omega) = \sum_{i=1}^{n} \frac{(\delta\theta_t^{(i)})^2}{1 + \omega(\delta\theta_t^{(i)})^2} - \delta\theta_t^T \delta g_t. \tag{18}$$

The value of the Lagrange multiplier ω can be found by solving the equation $F(\omega) = 0.$, which is the nonlinear equation. The equation is approximated using single iteration of Newton-Raphson, with initial condition $\omega = 0$. If the condition $\delta\theta_t^T \delta\theta_t > \delta\theta_t^T \delta g_t$ holds, equation (18) has a unique positive solution, and hence, the Lagrange multiplier ω_k approximation is as follow:

$$\omega_t = \omega - \frac{F(\omega)}{F'(\omega)} \tag{19}$$

$$\omega_t = \frac{\delta\theta_t^T \delta\theta_t - \delta\theta_t^T \delta g_t}{\sum_{i=0}^{n} (\delta\theta_t^i)^4}. \tag{20}$$

We observe that $\delta\theta_t^T \delta\theta_t > \delta\theta_t^T \delta g_t$, $\omega > 0$ and $B^i_{t+1} > 0$ for all $i = 1, 2, ..., n$ from Equation (20). The Oren-Luenberger scaling [20], which is the fraction $\delta\theta_t^T \delta g_t / \delta\theta_t^T \delta\theta_t$, is also utilized as updating formula, in case $B^i_{t+1} < 0$. If the condition $\delta\theta_t^T \delta g_t > \delta\theta_t^T \delta\theta_t$ is met, then equation (18) is positive definite. Therefore, B_{t+1} can be expressed as follows:

$$B_{t+1} = \begin{cases} \mathrm{diag}(B^{(1)}_{t+1}, ..., B^{(n)}_{t+1}), & \text{if } \delta\theta_t^T \delta\theta_t > \delta\theta_t^T \delta g_t \\ \dfrac{\delta\theta_t^T \delta g_t}{\delta\theta_t^T \delta\theta_t} I, & \text{otherwise.} \end{cases} \tag{21}$$

C. Saliency Matrix

The saliency matrix shows how much the parameters matter for the learning process. There will be minimal changes to the loss value when delete parameters with low saliency [16]. The aim is to find and remove these parameters at every step to make the model more general and avoid overfitting problems [21].

$$\hat{L}(\theta) = \sum_{i=1}^{n} \frac{1}{n!} L^i(c)(\theta - c)^i. \tag{22}$$

consider up to the 2nd-order:

$$\hat{L}(\theta) = L(\theta) + L'(c)(\theta - c) + \frac{H(c)}{2!}(\theta - c)^2, \tag{23}$$

if c is a local minimum point, then the saliency matrix $S(\theta)$ will be as follows:

$$S(\theta) = H(\theta)\frac{(\delta\theta)^2}{2}. \tag{24}$$

At the minimum point c, the Hessian matrix $H(\theta)$ is positive definite. This means that the loss value will stay the same or go up if we change the parameters. Also, after a step, the parameters that change more will have higher saliency values according to Equation (24). So, we can use this to tell how important the parameters are for the loss function.

D. Proximal Method

Our method applies the proximal algorithm to streamline the issue and mitigate overfitting concerns. Provided a non-convex, 0-Norm loss function $L(\theta) = \beta|\theta|_0$, the following is the proximal operator at point c.

$$\mathbf{prox}(c) = \underset{\theta}{\text{argmin}}(\beta||\theta||_0 + \frac{1}{2}||\theta - c||_2^2), \tag{25}$$

where the proximal operator is thresholding each element,

$$\mathbf{prox}(c, \beta) = \begin{cases} c, & \text{if} \frac{1}{2}||c||^2 > \beta \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

From equation (26), β corresponds to importance of parameter, therefore adding it to Equation (24) and the saliency matrix become

$$S(\theta) = B(\theta)\frac{(\delta\theta)^2}{2}\frac{(\theta)^2}{2}, \tag{27}$$

To reduce the effecto of ill-conditioned Hessian, $H(\theta)$ is replaced with $B(\theta)$. Then we can adjust procimal operator with the p-quantile Q_p of $S(\theta)$ as follows:

$$\mathbf{prox}(\theta, Q_p) = \begin{cases} \theta^i, & \text{if} S(\theta^i) > Q_p \\ 0, & \text{otherwise,} \end{cases} \quad i = 1, 2, ..., n \tag{28}$$

E. Spectral Proximal with Saliency Matrix

We introduce the Spectral Proximal optimizer enhanced by a saliency matrix. This method leverages the MDG approach on Hessian matrix approximation to reduce the gradient while adhering to the Hessian limitation, thereby averting the oscillatory behavior often associated with SGD descent, as detailed in Subsection III-B. To prevent overfiting and ensure generalization, we employ the saliency matrix to identify insignificant learning as elucidated in Subsection III-C. After that, a proximal operator generates a sparsified solution by removing the insignificant elements as presented in Subsection III-D. These features in optimizer aims to improve convergence rate and increase optimizer robustness.

The main innovation of this novel method is the integration of saliency matrix in the sparcification component to reduce redundent learning and increase learning efficiency. Algorithm 1 shows the pseudocode for the SP method with the saliency matrix. The source code for Pytorch implementation is presented in the following github repository: https://github.com/LeoYong95/Spectral-Proximal-Method.

Algorithm 1 Spectral Proximal method using saliency matrix

```
Require: Initial parameters $\theta \in \mathbb{R}$,
Require: Learning rate $\lambda > 0$
Require: Quantile parameter $0 < p < 1$
 1: $t \leftarrow 0$
 2: while stopping criterion not met do
 3:     Sample mini-batch: $x = \{x^1, ..., x^n\}, y^i$
 4:     Compute gradient: $g_t \leftarrow \frac{1}{n}\nabla\theta_t \sum L(f(x, \theta_t), y)$
 5:     Compute damping matrix $B_t$ with Equation (21)
 6:     $d_t \leftarrow B_t^{-1} g_t$
 7:     $\theta_{t+1} \leftarrow \theta_t - \lambda d_t$
 8:     Compute saliency matrix: $S_t \leftarrow B_t \frac{(\delta\theta_t)^2}{2}\frac{(\theta_t)^2}{2}$
 9:     Compute sparse parameters with Equation (28): $\theta_{t+1} \leftarrow \mathbf{prox}(\theta_{t+1}, Q_p)$
10:     $t \leftarrow t + 1$
11: end while
```

F. Convergence Analysis

In this section, we investigate the convergence of the SP with a saliency matrix approach, which is the proposed algorithm. The examination involves a theorem and several lemmas that establish the criteria for algorithm convergence. The theorem and lemmas are as follows:

Lemma 3.1: If $B_t \succeq L \cdot I$ for $\forall z$, the update equation $\theta_{t+1} = \mathbf{prox}_{Q_p, B_t}(\theta_t - B_t^{-1}\nabla L(\theta_t))$ is

$$\begin{aligned} L(\theta_{t+1}) &\le L(\theta_t) - \frac{1}{2}||G_{B_t}(\theta)||_{B_t}^2 \\ &= L(\theta_t) - \frac{1}{2}||\theta_{t+1} - \theta_t||_{B_t}^2 \end{aligned}$$

The proof of this lemma can be found in [22]. The next step is to show that B_k has a lower bound. The analysis relies on the following assumptions.

Assumption 3.2:

i The loss function L has second-order continuous derivatives.

ii The set $\Theta = \{\theta \in \mathbb{R} : f(\theta) < f(\theta_0)\}$ is convex.

iii There are possitive constants C_1 and C_2 such that for $\forall z \in \mathbb{R}$ and $\forall z \in \Theta$ where

$$C_1||z||^2 \le z^T G(\theta) z \le C_2||z||^2. \tag{29}$$

This implies that L has a unique solution θ^* in Θ.

This lemma establishes the minimum value for B_k.

Lemma 3.3: Positive constants ξ_1 and ξ_2 exist that bound B_t defined in (21). The loss function L meets Assumption 3.2, and initial Damping matrix $B_0 = I$.

Proof: First, we know that B_0 is bounded. Then we define G as

$$G = \int_1^0 \nabla^2 L(\theta_t + \tau\delta\theta_t) d\tau. \tag{30}$$

Applying the mean value theorem, we obtain

$$\delta g_t = G\delta\theta_t. \tag{31}$$

We can express Assumption 3.2(iii) differently as

$$C_1||\delta\theta_t||^2 \le \delta\theta_t^T \delta g_t \le C_2||\delta\theta||^2. \tag{32}$$

Case 1: if $\delta\theta_t^T\delta\theta_t > \delta\theta_t^T\delta g_t$, B_{t+1} is express as in (21) and (20)

$$B_{t+1}^i = \frac{1}{1+\dfrac{\delta\theta_t^T\delta\theta_t-\delta\theta_t^T\delta g_t}{\sum\limits_{i=0}^{n}(\delta\theta_t^i)^4}(\delta\theta_t^i)^2}. \quad i=1,2,...,n, \quad (33)$$

Denote $(\delta\theta_t^M)^2 = \max\{(\delta\theta_t^1)^2,...,(\delta\theta_t^n)^2\}$. We know that $\delta\theta_t^T\delta\theta_t = \sum_{i=1}^{n}(\delta\theta_t^i)^2$, therefore $||\delta\theta_t||^2 \le n(\delta\theta_t^M)^2$. Besides that, the term $\dfrac{\delta\theta_t^T\delta\theta_t-\delta\theta_t^T\delta g_t}{\sum\limits_{i=0}^{n}(\delta\theta_t^i)^4}(\delta\theta_t^i)^2$ is positive, we can express B_{t+1} as

$$\frac{1}{1+n(1-C_1)} \le B_{t+1}^i \le 1. \quad (34)$$

Case 2: Assuming that $\delta\theta_t^T\delta\theta_t \le \delta\theta_t^T\delta g_t$, we can directly obtain B_{t+1} from Assumption 3.2(iii)

$$C_1 \le B_{t+1}^i \le C_2. \quad (35)$$

Hence, (34) and (35) provide

$$\xi_1 = \min\{\frac{1}{1+n(1-C_1)}, C_1\} \le B_t^i \le \max\{1, C_2\} = \xi_2, \quad (36)$$

which gives the boundness of B_t. ∎

The convergence analysis of Algorithm 1 is based on the theorem from [22]. The theorem is as follows:

Theorem 3.4: For SP in Algorithm 1, $L(\theta_t)$ converges to the optimal value L^*, i.e., $\lim\limits_{t\to\infty} L(\theta_t) := L^*$.

Proof: Since the B_t is bounded, and we pair SP algorithm with proven learning rate scheduler (Section III-E). Following lemma 3.1 and lemma 3.3 we have,

$$\begin{aligned} L(\theta_{t+1}) &\le L - \frac{1}{2}||\theta_{t+1}-\theta_t||_{B_t}^2 \\ &= L - \frac{1}{2}||G_{B_t}(\theta)||_{(B_t)^{-1}}^2, \end{aligned}$$

for this function at limit we have,

$$\begin{aligned} 0 &= \lim_{t\to\infty} ||\theta_{t+1}-\theta_t||_{B_t}^2 \\ &= \lim_{t\to\infty} ||G_{B_t}(\theta_t)||_{(B_t)^{-1}}^2. \end{aligned} \quad (37)$$

∎

The limit point L^* satisfies $G_{B^*}(\theta^*) = 0$, which means θ^* is a stationary point. Moreover, the convexity of L from Assumption 3.2(ii) implies that θ^* is the global minimum.

IV. Experiment

In this section, we present the experiment conducted to evaluate the novel optimizer. The Dataset subsection introduces the datasets used in training process. Deep Learning Models subsection highlights the diverse architectures used in experiment. The Experimental Setup subsection details hardware, training parameters, and configurations, laying the groundwork for a comprehensive assessment of the novel optimizer's performance.

A. Dataset

The MNIST dataset is a collection of 70,000 handwritten digits from 0 to 9, each represented by a 28x28 pixel grayscale image. The dataset is widely used as a benchmark for testing various machine learning algorithms, especially those related to image recognition and computer vision. The dataset is divided into 60,000 training examples and 10,000 test examples, and each digit has roughly the same number of samples. Figure 1 shows example images from MNIST dataset.

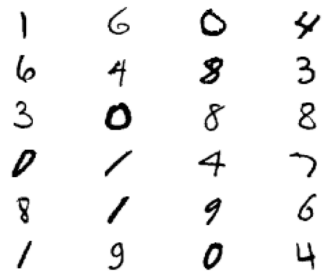


Fig. 1. Example images from MNIST dataset.

The CIFAR-10 dataset [23] is a collection of 60,000 color images of size 32x32 pixels, divided into 10 classes of objects. The classes are: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The dataset is commonly used for image classification and recognition tasks, such as deep learning models. In the training process using this dataset, we used random horizontal flipping, random cropping to 32 x 32 pixels, and normalization with mean (0.5, 0.5, 0.5).Figure 2 shows example images from CIFAR-10 dataset.

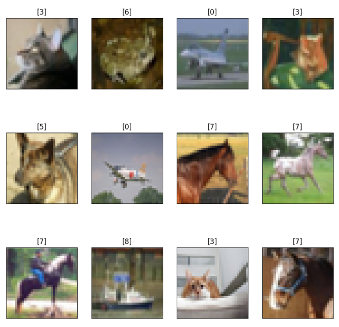


Fig. 2. Example images from CIFAR-10 dataset

The Yale-CMU-Berkeley (YCB) dataset is a self-generated dataset based on the YCB objects and model set [24] in a simulated household environment. The dataset consist of 7,000 training images and 35 test images of size 640x480 pixels divided into 63 different classes. During training process, we adopted the established mosaic data augmentation method to enhance training for YOLOv7. A visual representation of YCB training images following mosaic augmentation is portrayed in Figure 3.

B. Deep Learning Models

In this experiment, we aim to assess the effectiveness of the novel optimizer by employing a diverse set of models. These

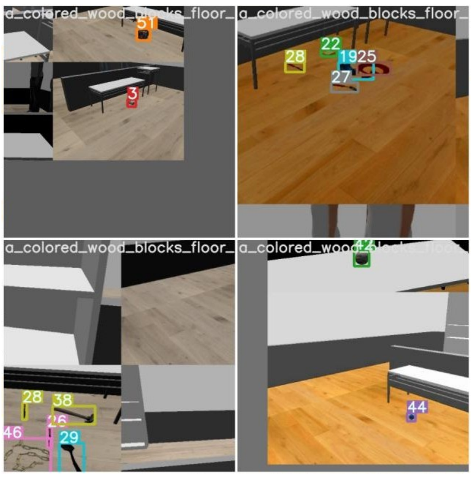


Fig. 3. Mosaic augmentation applied to training images.

models, each tailored for specific tasks and exhibiting varying degrees of complexity.

A small custom CNN we design ourselves to fit our specific task and data. It is made of two convolutional layers and two fully connected layer at the end. This model is designed to handle simple image classification tasks, such as recognizing handwritten digits in the MNIST dataset.

MobilenetV2 [25] is a model that was developed by Google researchers to perform efficient image classification and object detection on mobile devices. It uses an inverted residual structure with linear bottlenecks to reduce the number of parameters and computations, while preserving the feature richness of the network. This model is suitable for more complex image classification tasks, such as identifying objects in the CIFAR-10 dataset.

ResNet [26] is another model that was proposed by Microsoft researchers to solve the problem of vanishing gradients and degradation in very deep neural networks. It introduces residual blocks that allow the network to learn identity mappings between layers, and skip connections that enable the network to bypass some layers if they are not useful. In this experiment, we selected the ResNet-18 and ResNet-50 model variant to evaluate the effect of model depth on performance.

YOLOv7 [27] is a recent version of the popular YOLO (You Only Look Once) model that performs real-time object detection using a single neural network. It is based on the PyTorch framework and incorporates several improvements over previous versions. YOLOv7 can detect multiple objects of different sizes and shapes in an image with high accuracy and speed. Therefore, we select this model to train with the YCB dataset.

We choose to use these models because they represent different levels of complexity, pre-training, and architecture design. By testing the optimizer on these models, we can evaluate how well it can optimize different types of networks, how fast it can converge to a good solution, and how robust it is to different tasks and datasets.

C. Experimental Setup

The experimental setup involved training various neural network models on different datasets, each with specific configurations outlined in Tables I, II, and III. For the MNIST dataset, the training utilized an Intel(R) Xeon(R) W-1350 CPU, an Nvidia RTX A4000 GPU, 16GB of RAM, and PyTorch 1.10.0. The CIFAR-10 dataset employing the same hardware specifications with different training parameters, such as 300 epochs and batch size of 32. For the YCB dataset, YOLOv7 was trained with an Intel(R) Xeon(R) CPU @ 2.00GHz x 2, an Nvidia Tesla P100 GPU, 12GB of RAM, and PyTorch 1.11.0. The training parameters are, image size of 480, 300 epochs, and batch size of 32. For each dataset, we applied three distinct learning rate scheduling techniques, aligning with established practices outlined in the literature [28]–[30]. These setups aimed to comprehensively evaluate the novel optimizer's performance across diverse models, datasets, and training configurations.

TABLE I
THE CONFIGURATION OF THE TRAINING SETTINGS FOR THE MODELS ON MNIST DATASET.

Core processors (CPU)	Intel(R) Xeon(R) W-1350
Graphics Card (GPU)	Nvidia RTX A4000 x 1
RAM	16GB
Framework	pytorch 1.10.0(GPU support)
Python version	3.7.12
Architecture	Custom CNN (Convolutional layer x 2 ,Fully-connected layer x 2)
Epoch	15
Batch size	128
Scehduler	StepLR (gamma = 0.7)

TABLE II
THE CONFIGURATION OF THE TRAINING SETTINGS FOR THE MODELS ON CIFAR-10 DATASET.

Core processors (CPU)	Intel(R) Xeon(R) W-1350
Graphics Card (GPU)	Nvidia RTX A4000 x 1
RAM	16GB
Framework	pytorch 1.10.0(GPU support)
Python version	3.7.12
Architecture	MobileNetV2, ResNet18 . and ResNet50
Epoch	300
Batch size	32
Scehduler	ReduceLROnPlateau tenthing

TABLE III
THE CONFIGURATION OF THE TRAINING SETTINGS FOR THE MODELS ON YCB DATASET.

Core processors (CPU)	Intel(R) Xeon(R) CPU @ 2.00GHz x 2
Graphics Card (GPU)	Nvidia Tesla P100 x 1
RAM	12GB
Framework	pytorch 1.11.0(GPU support)
Python version	3.7.12
Architecture	YOLOv7
Epoch	300
Batch size	32
Image size	480
Scehduler	Simplified cosine annealing

V. Results and Discussion

This section demonstrates the resilience of the SP method we propose across two image classification datasets (MNIST, as discussed in Section V-A, and CIFAR-10, as outlined in Section V-B), as well as one object detection dataset (the Household Objects dataset, as detailed in Section V-C) in a real-world application context. We employ diverse model architectures (including custom CNN, MobileNetV2, ResNet-18, ResNet-50, and YOLOv7) throughout our experiments to assess the efficacy of our approach across a range of models.

We conduct a performance evaluation of our approach in comparison to well-established methods (SGD, Adam, and SGD+Nesterov) across three key metrics: test accuracy, test loss, and training loss. These metrics serve as indicators for assessing the training effectiveness, model generalization capabilities, and potential overfitting. Our experimental setup adheres to standard step size reduction and hyperparameter configurations as documented in the literature for each dataset and model. You can find further information on the experimental setup in Subsection IV-C.

The objective of these experiments is not to attain optimal outcomes but rather to showcase the capabilities of the proposed SP method across various training scenarios involving diverse models and datasets. In the majority of instances within these experiments, the SP method outperforms the alternative approaches.

A. MNIST Classification

The MNIST dataset serves as a widely utilized image classification benchmark within the machine learning community. In this specific experiment, we devised a concise CNN model, characterized by a mere 1.5 million parameters in total.

We evaluated two main types of methods: optimizers without momentum and optimizers with momentum. The SP method consistently performs better than the other methods in both types.

The Adam method tends to cause overfitting, as shown by the graphs of the training loss and test accuracy. This is attributed to Adam's tendency to converge towards suboptimal local solutions. In contrast, the SP+Nesterov variant demonstrates robust generalization capabilities within this experimental setup and even exhibits slight performance enhancements compared to the SGD+Nesterov method, as visually depicted in Figure 4.

B. CIFAR-10 Classification

In this Subsection, we present the findings in training with CIFAR-10 dataset. The selection of MobileNetV2 [25], ResNet18, and ResNet50 [26] architectures for this investigation was deliberate due to their established efficacy in handling CIFAR-10 image classification tasks. Our primary aim centered on evaluating the performance of our method across models representing different levels of parameterization: underparameterized (MobileNetV2), adequately parameterized (ResNet18), and overparameterized (ResNet50). We only report the findings related to MobileNetV2 and ResNet50 in this section. The results for ResNet18 are in Appendix A.

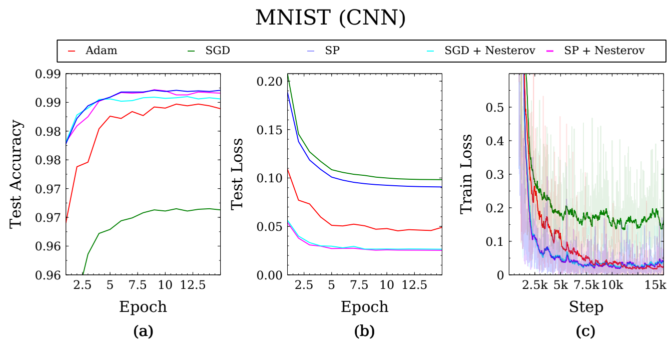


Fig. 4. Comparison of different methods on the MNIST dataset using a CNN model. The graphs show the test accuracy **(a)**, test loss **(b)** and train loss **(c)** over 15 epochs. The SP method and SP + Nesterov method achieve the best performance, with a test accuracy of 99.1% and 99.2%, respectively, and a test loss of 0.03 and 0.02, respectively. The other methods have lower accuracy and higher loss on the test set.

Anticipations were that the momentum-based optimizers would exhibit subpar performance during MobileNetV2 training, given their inclination to converge towards suboptimal local minima. Likewise, there were expectations of overfitting in the case of ResNet50 due to its overparameterization, resulting in diminished test accuracy and elevated training loss values for this particular model.

As depicted in Table IV, the outcomes of MobileNetV2 training reveal the superior performance of our approach, denoted as SP, surpassing other methods and achieving a substantial enhancement compared to SGD. As previously conjectured, the SP+Nesterov variant exhibited diminished test accuracy and elevated training loss values, as illustrated in Figure 5, signifying its convergence towards a suboptimal solution. Notably, a significant limitation of SP+Nesterov lies in its incompatibility with training narrow models. This limitation stems from the adverse influence of the 0-Norm proximal operator and momentum component on the descent direction within compact networks.

Furthermore, the slim and deep structure inherent to MobileNetV2 can introduce negative gradient correlations across training batches, posing a hindrance to momentum-based optimizers due to their historical gradient memory, as discussed in [31]. In this particular scenario, SGD also grapples with discrepancies in the descent direction due to its lack of corrective mechanisms. Our approach uses damping correction, which avoids gradient confusion during the training process by maintaining a consistent rate of gradient descent across the Hessian spectrum.

In the context of training the overparameterized model, our approach exhibited remarkable advancements when compared to the array of other methods under scrutiny. Specifically, the momentum-based optimizers excelled in achieving the highest test accuracy. The broader architectural framework of ResNet50 mitigates issues related to gradient confusion, as elaborated upon in [31], thereby affording gradient-dependent momentum-based optimizers greater accuracy in determining the descent direction.

As Figure 6 shows, optimizers based on Nesterov momentum performed better than Adam, achieving faster convergence in both the graphs of the training loss and test accuracy. This accelerated convergence can be attributed to the fact that at each iteration, optimizers that use Nesterov momentum produce better descent directions than Adam's Heavy-Ball method [32]. This is a consistent advantage of Nesterov momentum-based optimizers.

The SP+Nesterov method achieved the highest test accuracy in this experimental setting. This may be due to the use of the saliency matrix proximal operator. This operator serves to prevent overfitting while enhancing generalization capabilities.

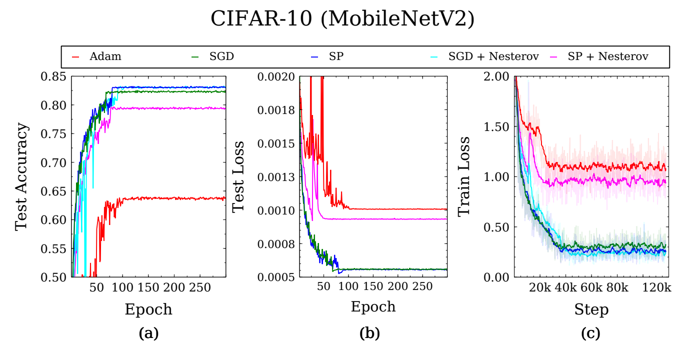


Fig. 5. Comparison of different methods on the CIFAR-10 dataset using a MobileNetV2 model. The graphs show the test accuracy **(a)**, test loss **(b)** and train loss **(c)** over 300 epochs.

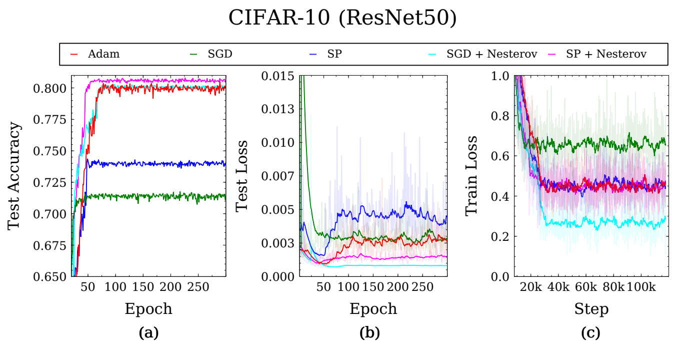


Fig. 6. Comparison of different methods on the CIFAR-10 dataset using a ResNet50 model. The graphs show the test accuracy **(a)**, test loss **(b)** and train loss **(c)** over 300 epochs.

TABLE IV
A COMPARISON OF MOBILENETV2 AND RESNET50 MODELS USING DIFFERENT TRAINING METHODS FOR CIFAR-10 IMAGE CLASSIFICATION TASK.

	MobileNetV2	Resnet 50
SGD	82.56	71.76
SP	**83.26 (+0.7)**	**74.37 (+2.61)**
Adam	64.05	80.53
SGD + Nesterov	83.32	80.30
SP + Nesterov	**67.25 (-16.07)**	**80.81 (+0.51)**

C. *Detecting Household Objects with YOLOv7 Model*

In this particular experimentation, we put our method to the test in a real-world scenario, focusing on object detection—a critical aspect of numerous contemporary vision-based applications, including self-driving vehicles, autonomous robotics, and surveillance technology. We used a subset of objects from the YCB object dataset [24], which is designed for robot manipulation. We applied the YOLOv7 [27] model, which is the latest in object detection. Our main goal was to test how well our method can handle a common industry problem: training a large network with a small dataset.

The results presented in Figure 7 shed light on the performance of our method. Although the SP method did not exhibit as swift initial convergence as Adam, it consistently maintained lower training loss values compared to SGD. This observation underscores the speedy convergence capabilities of the SP method and its proficiency in achieving more cost-effective solutions when juxtaposed with other approaches. This efficacy can be attributed to the unique proximal operator inherent to the algorithm, which effectively eliminates superfluous elements while minimizing the learning burden.

Furthermore, the SP method demonstrated formidable generalization attributes, boasting the highest test mean Average Precision (mAP) values, as depicted in Figure 8 and quantified in Table V. Models with robust generalization characteristics tend to produce fewer false positives and false negatives. A comparative analysis of the confusion matrices for both the SP-trained and SGD-trained models in Figure 10 corroborates this, indicating a higher level of accuracy in the SP-trained model.

In the training process, computational expenses hold significant importance. Although our approach incurs a greater computational cost, the model trained using Stochastic Proximal (SP) demonstrates superior performance compared to the model trained through Stochastic Gradient Descent (SGD), as detailed in Table V. We also conducted tests on the SP-trained model after the completion of SGD training, which required 29.25 hours of wall time. At this juncture, the SP method exhibited a higher mAP0.5 than the SGD method and matched the SGD method in terms of mAP0.5:0.95. This implies that the SP-trained model produces fewer false-negative predictions (refer to the confusion matrix in Figure 10) and generates bounding boxes of equivalent accuracy to those generated by the SGD-trained model. Figure 9 provides visual representations of the predictions made by the SP-trained model on the YCB dataset testing images.

TABLE V
THE MAP PERFORMANCE OF YOLOV7 MODEL TRAINED WITH SP AND SGD METHODS ON YCB OBJECT DETECTION TASK.

	mAP0.5(%)	mAP0.5:0.95(%)	mAP0.5 @29.25 hours (%)	mAP0.5:0.95 @29.25 hours (%)
SGD	88.1	68.5	88.0	68.1
SP	**90.1 (+2.0)**	**71.0 (+2.5)**	**90.6 (+2.6)**	**68.1**

VI. CONCLUSION

We have put forward a fresh approach to training deep learning models, christened the Spectral Proximal (SP) method with

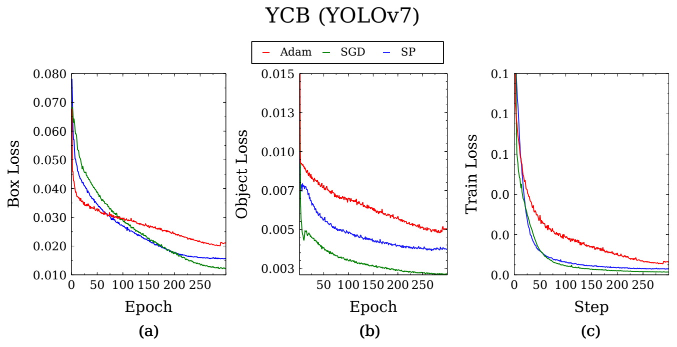


Fig. 7. The training losses of YOLOv7 model on YCB dataset using different experimental settings.

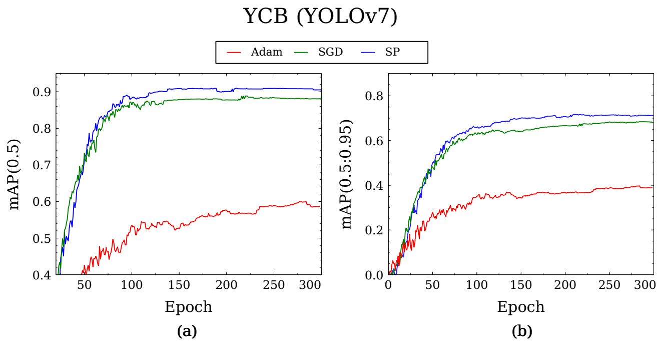


Fig. 8. The mAP performance of YOLOv7 model on YCB dataset for different IoU thresholds. The **(a)** graph shows the mAP0.5 and the **(b)** graph shows the mAP0.5:0.95.

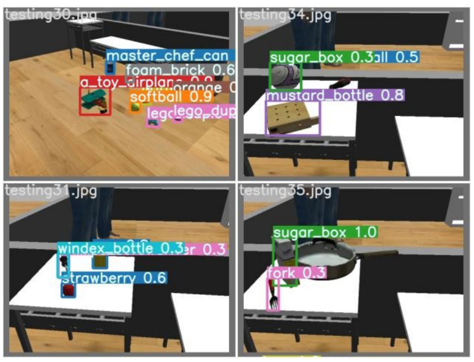


Fig. 9. YCB dataset test images annotated with the predictions of the model trained with SP method.

the saliency matrix. This SP method stands out as a superior optimizer when compared to SGD and Adam, boasting a noteworthy convergence rate and delivering exceptional generalization outcomes. Many of today's machine vision applications grapple with the complexities of overparameterization and the constraints of limited datasets. Here, the SP method steps in as a potent solution, effectively surmounting these hurdles by crafting high-accuracy models with robust generalization capabilities. What's more, our approach exhibits a keen awareness of resource constraints during training sessions, outperforming other methodologies by achieving top-tier model performance while staying within predefined resource limits.

Nevertheless, our experiments have unveiled a caveat when employing the SP method with momentum: the proximal

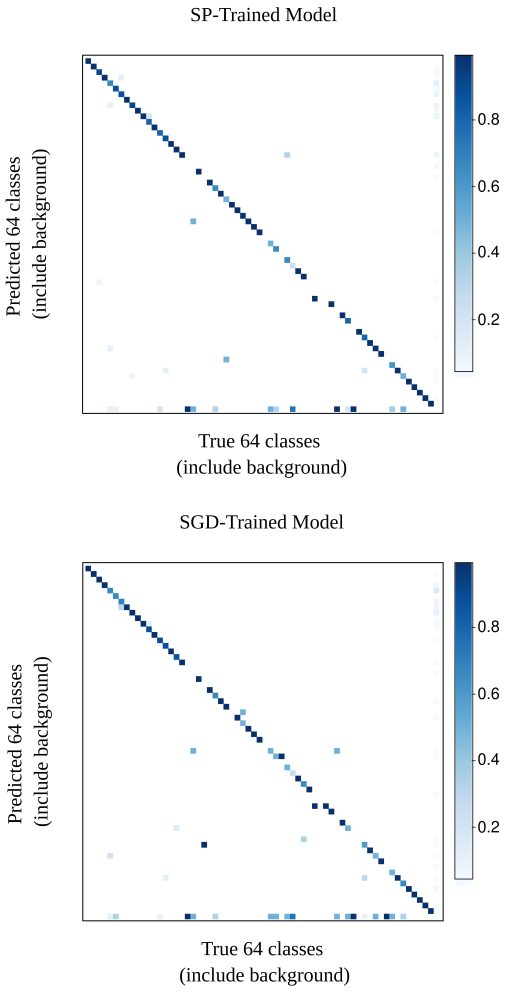


Fig. 10. The comparison of the confusion matrices for the models trained with sp and SGD methods. The sp-trained model's confusion matrix is shown above and the SGD-trained model's confusion matrix is shown below.

operator inherent to SP has a tendency to perturb the descent direction in narrow models due to heightened gradient confusion. As a precautionary measure, we advise against utilizing the SP method with momentum when training slender models.

To summarize, we have introduced the novel Spectral Proximal (SP) method supplemented by the saliency matrix as an optimizer for deep learning model training. The SP method excels in swift convergence and yields superior generalization capabilities during testing. Our comprehensive array of test results serves as a testament to the optimizer's performance across a spectrum of scenarios, where it frequently outperforms baseline methods, particularly in the realm of training overparameterized models. However, we are committed to addressing the SP method's performance limitations in narrow model training with momentum in our future research endeavors.

Appendix A
Experiment CIFAR-10 (ResNet18)

In this experimental phase, we gauge the SP method's effectiveness when applied to training a sufficiently-sized model. The outcomes, as presented in Table VI and Figure 11, mirror the findings from our MobileNetV2 training endeavors. Notably, the SP method outperforms SGD and closely rivals SGD+Nesterov in performance. Additionally, scrutiny of the test accuracy graph reveals a narrower gap between vanilla optimizers and momentum-based optimizers when compared to the MobileNetV2 setup. This observation is attributed to reduced gradient confusion, a consequence of the broader architecture of ResNet18 as opposed to MobileNetV2.

TABLE VI
A COMPARISON OF DIFFERENT METHODS FOR TRAINING RESTNET18 ON CIFAR-10 DATASET USING TEST ACCURACY METRIC.

	Resnet 18
SGD	84.28
SP	**84.63 (+0.35)**
Adam	83.55
SGD + Nesterov	84.65
SP + Nesterov	**84.10 (-0.55)**

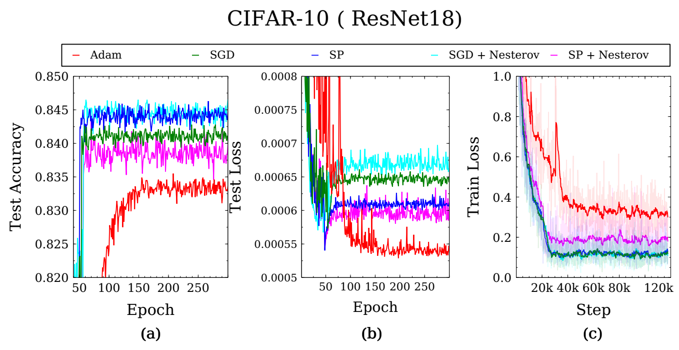


Fig. 11. The performance of ResNet18 on CIFAR-10 dataset classification performance. The test accuracy **(a)**, test loss **(b)** and train loss **(c)** graphs.

APPENDIX B
EXPERIMENT YCB (YOLOV7) USING ADAM METHOD

Displayed in Table VII are the test mAP values obtained from training the model using the Adam method. Notably, the Adam method is notorious for its subpar generalization capabilities, rendering it ill-suited for the YCB dataset in the YOLOv7 framework, which necessitates a model with the capacity to effectively generalize to novel data. This model's shortcomings become evident as it struggles to consistently distinguish objects from the background. As depicted in Figure 12, the model trained with the Adam method records a higher frequency of false negative predictions within the background class.

TABLE VII
THE MAP PERFORMANCE OF YOLOV7 MODEL TRAINED WITH ADAM METHOD ON YCB OBJECT DETECTION TASK.

	mAP0.5 (%)	mAP0.5:0.95 (%)	mAP0.5 @29.25 hours(%)	mAP0.5:0.95 @29.25 hours(%)
Adam	59.8	39.5	84.4	58.6

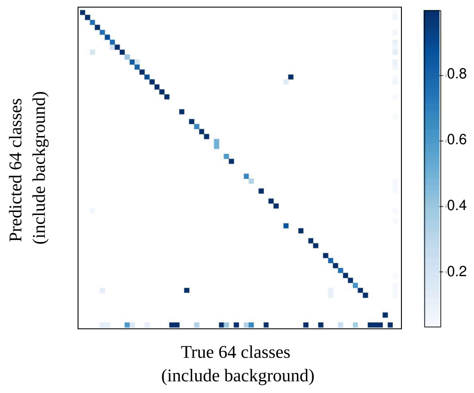


Fig. 12. The performance of the model trained with Adam method on the test images. The confusion matrix shows how the model classifies the images into different categories.

ACKNOWLEDGMENT

The authors express their gratitude for the backing provided by the Malaysian Ministry of Higher Education (MoHE) via the Fundamental Research Grant Scheme (FRGS/1/2022/STG06/UTAR/02/2), along with supplementary assistance from the Research Fund of Universiti Tunku Abdul Rahman (IPSR/RMC/UTARRF/2021-C1/K03).

REFERENCES

[1] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.
[2] J. Chen, D. Zhou, Y. Tang, Z. Yang, Y. Cao, and Q. Gu, "Closing the generalization gap of adaptive gradient methods in training deep neural networks," in *International Joint Conference on Artificial Intelligence*, 07 2020, pp. 3239–3247.
[3] S. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *International Conference on Learning Representations*, 2018.
[4] P. Zhou, J. Feng, C. Ma, C. Xiong, S. Hoi, and E. Weinan, "Towards theoretically understanding why sgd generalizes better than adam in deep learning," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS'20. Red Hook, NY, USA: Curran Associates Inc., 2020.
[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.
[6] C.-C. Wang, K. L. Tan, and C.-J. Lin, "Newton methods for convolutional neural networks," *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 2, p. 1–30, 2020.
[7] H. S. Sim, W. J. Leong, and C. Y. Chen, "Gradient method with multiple damping for large-scale unconstrained optimization," *Optimization Letters*, vol. 13, no. 3, p. 617–632, 2018.
[8] Y.-H. Dai, Y. Huang, and X.-W. Liu, "A family of spectral gradient methods for optimization," 2018.
[9] L. Wang, M. Cao, F. Xing, and Y. Yang, "The new spectral conjugate gradient method for large-scale unconstrained optimisation," *Journal of Inequalities and Applications*, vol. 2020, no. 1, apr 2020. [Online]. Available: https://doi.org/10.1186/s13660-020-02375-z
[10] Y. Laylani, B. A. Hassan, and H. Khudhur, "Enhanced spectral conjugate gradient methods for unconstrained optimization," *International Journal of Mathematics and Computer Science*, pp. 163–172, 01 2023.
[11] N. Parikh, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, p. 127–239, 2014.
[12] P. J. S. Santos and J. C. de O. Souza, "A proximal point method for quasi-equilibrium problems in hilbert spaces," *Optimization*, vol. 71, no. 1, pp. 55–70, aug 2020. [Online]. Available: https://doi.org/10.1080/02331934.2020.1810686
[13] Y. Yang, Y. Yuan, A. Chatzimichailidis, R. J. van Sloun, L. Lei, and S. Chatzinotas, "Proxsgd: Training structured neural networks under regularization and constraints," in *International Conference on Learning Representations ICLR 2020*, 2020.

[14] X. Tang and K. Scheinberg, “Efficient quasi-newton proximal method for large scale sparse optimization,” in *Conference on Neural Information Processing Systems*. NIPS, 2013.

[15] J. Yun, A. Lozano, and E. Yang, “Adaptive proximal gradient methods for structured neural networks,” in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=Qijzj3WqUl3

[16] Y. Lecun, J. Denker, and S. Solla, “Optimal brain damage,” in *Advances in Neural Information Processing Systems*, vol. 2, 01 1989, pp. 598–605.

[17] N. Jia, X. Liu, W. Zhao, H. Zhang, and K. Zhuo, “An adaptive framework for saliency detection,” *International Journal of Imaging Systems and Technology*, vol. 29, no. 3, pp. 382–393, Jun. 2019. [Online]. Available: https://doi.org/10.1002/ima.22351

[18] J. P. Amorim, P. H. Abreu, J. Santos, M. Cortes, and V. Vila, “Evaluating the faithfulness of saliency maps in explaining deep learning models using realistic perturbations,” *Information Processing and Managment*, vol. 60, no. 2, p. 103225, mar 2023. [Online]. Available: https://doi.org/10.1016/j.ipm.2022.103225

[19] M. Kremer, P. Caruana, B. Haworth, M. Kapadia, and P. Faloutsos, “Automatic estimation of parametric saliency maps (psms) for autonomous pedestrians,” *Computers & Graphics*, vol. 104, pp. 86–94, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0097849322000486

[20] D. G. Luenberger, *Linear and nonlinear programming*, 2nd ed. New York, NY: Springer, Sep. 2003.

[21] S. Gonzalez and R. Miikkulainen, “Optimizing loss functions through multi-variate taylor polynomial parameterization,” in *Genetic and Evolutionary Computation Conference*, 06 2021, pp. 305–313.

[22] Y. Park, S. Dhar, S. Boyd, and M. Shah, “Variable metric proximal gradient method with diagonal barzilai-borwein stepsize,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 05 2020, pp. 3597–3601.

[23] M. S. Hanif and M. Bilal, “Competitive residual neural network for image classification,” *ICT Express*, vol. 6, no. 1, pp. 28–37, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959519300694

[24] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. Dollar, “Yale-cmu-berkeley dataset for robotic manipulation research,” *The International Journal of Robotics Research*, vol. 36, p. 027836491770071, 04 2017.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 06 2018, pp. 4510–4520.

[26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2016, pp. 770–778.

[27] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,” *arXiv preprint arXiv:2207.02696*, 2022.

[28] Y. Bengio, *Practical Recommendations for Gradient-Based Training of Deep Architectures*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 437–478. [Online]. Available: https://doi.org/10.1007/978-3-642-35289-8_26

[29] M. Zaheer, S. Reddi, D. Sachan, S. Kale, and S. Kumar, “Adaptive methods for nonconvex optimization,” in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/90365351ccc7437a1309dc64e4db32a3-Paper.pdf

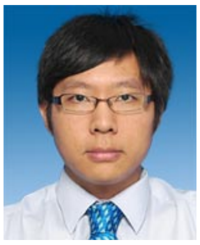
[30] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, “Bag of tricks for image classification with convolutional neural networks,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2019, pp. 558–567. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00065

[31] K. A. Sankararaman, S. De, Z. Xu, W. R. Huang, and T. Goldstein, “The impact of neural network overparameterization on gradient confusion and stochastic gradient descent,” in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML’20. JMLR.org, 2020.

[32] X. Liu, W. Tao, J. Wang, and Z. Pan, “A high-resolution dynamical view on momentum methods for over-parameterized neural networks,” *arXiv preprint arXiv:2208.03941*, 2022.

Yong Cherng Liin received his Bachelor of Engineering (Mechanical) degree from the Universiti Tunku Abdul Rahman (UTAR). He is presently pursuing his Ph.D. studies at UTAR. His research interests include robotics, deep learning, machine vision, artificial intelligence and optimization algorithm.

Kwan Ban Hoe received his Bachelor of Engineering (Electrical) degree, the Master of Engineering Science degree, and the Ph.D. degree in engineering from the University of Malaya (UM). He is currently working at Universiti Tunku Abdul Rahman (UTAR) as an Assistant Professor. His research interests include image processing, artificial intelligence, medical signal processing, Internet of Things, and robotics.



Danny Ng Wee Kiat (member, IEEE) is an assistant professor in the Department of Mechatronics and Biomedical Engineering at UTAR, where he currently serves. He completed his Ph.D. in Engineering in 2021 and specializes in the field of robotics and its practical applications. Since 2019, he has been a registered professional engineer with the Board of Engineer Malaysia. He actively engages in industry-focused research collaborations with notable companies in Malaysia.

Sim Hong Seng obtained his Ph.D. degree in mathematics from Universiti Putra Malaysia, Malaysia, in 2017. He is currently an Assistant Professor in the Department of Mathematical & Actuarial Sciences, Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Malaysia. His research interests involve optimization problems and machine learning. Dr. Sim is also the chairperson of the Centre for Mathematical Sciences, Universiti Tunku Abdul Rahman.