AIMS Mathematics, 11(7): 20502–20534.
DOI: 10.3934/math.2026834
Received: 22 April 2026
Revised: 30 June 2026
Accepted: 01 July 2026
Published: 13 July 2026

https://www.aimspress.com/journal/Math

Research article

Double-inertial bilevel optimization for hybrid deep learning: Accelerating chest X-ray classification

Suthep Suantai[1] **and Kobkoon Janngam**[2,3,*]

[1] Research Center in Optimization and Computational Intelligence for Big Data Prediction, Department of Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand

[2] Department of Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand

[3] Office of Research Administration, Chiang Mai University, Chiang Mai 50200, Thailand

* **Correspondence:** Email: kobkoon.j@cmu.ac.th.

Abstract: Convex bilevel optimization, in which an upper-level problem is optimized over the solution set of a lower-level convex program, naturally arises in machine learning tasks such as hyperparameter tuning, meta-learning, and regularized classification. Existing first-order bilevel solvers typically employ single-inertial acceleration, which can exhibit oscillatory behaviors and premature stagnation. In this paper, we propose the Double-Inertial proximal Gradient Method with Viscosity Approximation (DIPGM-VA), a novel algorithm that combines double-inertial extrapolation with viscosity approximation within a forward–backward splitting framework. We establish the strong convergence of the iterates to the unique solution of the bilevel problem in real Hilbert spaces under standard assumptions. As an application, we develop a hybrid deep learning architecture that integrates EfficientNetV2-B0 with an Extreme Learning Machine (ELM) classifier whose output weights are determined by solving a bilevel optimization problem via DIPGM-VA, and validate it on a pediatric chest X-ray pneumonia benchmark, addressing a leading cause of death among children under five in developing countries. Numerical experiments show that DIPGM-VA outperforms other state-of-the-art bilevel solvers by reaching a lower inner-level objective value within the same iteration budget. Concurrently, the hybrid model delivers a competitive F1-score of 0.9001 and accelerates the training process by an order of magnitude relative to standard end-to-end approaches.

Keywords: chest X-ray classification; convex bilevel optimization; double-inertial algorithm; extreme learning machine; fixed-point iteration; forward–backward splitting; proximal gradient method; strong convergence; variational inequality; viscosity approximation
Mathematics Subject Classification: 47H09, 47H10, 47J25, 65K10, 90C25, 68T07

1. Introduction

In data-limited clinical settings, two persistent difficulties arise in deep medical image classification: the computational cost of end-to-end training and the lack of a principled mechanism for regularization [1, 2]. Despite the remarkable progress brought by deep networks in this domain, these issues call for a hierarchical formulation in which regularized feature selection and classifier training are treated jointly; bilevel optimization offers precisely such a framework.

Over the past decade, bilevel optimization has emerged as a versatile tool in machine learning, with prominent applications to hyperparameter tuning [3, 4], meta-learning [3], and data classification [5]. In its general form, such a problem couples an outer objective with an inner optimization problem that constrains the feasible set. Here, the convex bilevel instance considered is

$$\min_{x\in S_*} \varphi(x), \tag{1.1}$$

with S_* denoting the argmin set of the inner (lower-level) problem as follows:

$$S_* = \operatorname{argmin}_{x\in H}\{f(x) + g(x)\}, \tag{1.2}$$

where H is a real Hilbert space, $\varphi\colon H \to \mathbb{R}$ is strongly convex and differentiable, $f\colon H \to \mathbb{R}$ is convex with an L_f-Lipschitz continuous gradient, and $g\colon H \to \mathbb{R} \cup \{\infty\}$ is proper, convex, and lower semicontinuous.

Denote the solution set of (1.1) by Λ. Any $x^* \in \Lambda$ is characterized by the following variational inequality [6]:

$$\langle \nabla\varphi(x^*), x - x^*\rangle \geq 0, \text{ for all } x \in S_*.$$

Problem (1.2) can be effectively tackled via proximal splitting, which reformulates the minimization as the fixed-point equation $x^* = Tx^*$ for the forward–backward operator $T = \operatorname{prox}_{cg}(I - c\nabla f)$, where f and g are as in (1.2) and $c > 0$ is a step size. Indeed, any fixed point of T, namely any x with

$$x = \operatorname{prox}_{cg}(x - c\nabla f(x)),$$

minimizes $f + g$ [7, 8], so the search for minimizers reduces to the fixed-point iteration

$$x_{n+1} = \operatorname{prox}_{cg}(x_n - c\nabla f(x_n)), \tag{1.3}$$

which is weakly convergent whenever $0 < c < 2/L_f$. In a fixed-point form, Iteration (1.3) reads $x_{n+1} = T(x_n)$, and the nonexpansive operator T satisfies $\operatorname{Fix}(T) = S_*$.

Strong convergence of the above proximal iteration was obtained by Moudafi [9] via a viscosity approximation scheme of the following form:

$$x_{n+1} = \eta_n S(x_n) + (1 - \eta_n)T(x_n),$$

in which T is nonexpansive while S is a strict contraction that satisfies standard assumptions. A viscosity step in the same spirit will be adopted in the algorithm proposed in Section 3.

To further accelerate convergence, our scheme also employs inertial extrapolation, a technique with a long history in optimization that reuses past iterates at every step. Classical milestones along this line

include Polyak's heavy-ball method [10], Nesterov's accelerated gradient method [11], and the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) of Beck and Teboulle [12], the latter extending inertial acceleration to proximal operators.

Single-inertial methods use one previous iterate in the extrapolation step. They have been extensively analyzed in recent years. In this direction, Bussaban et al. [13] established strong convergence of an inertial forward–backward algorithm based on nonexpansive mappings in the context of image restoration, Yatakoat et al. [14] proposed accelerated fixed-point schemes for convex minimization, and Janngam et al. [5] developed an inertial framework tailored to convex bilevel optimization. More recent works have investigated two-step inertial variants: Sae-jia and Suantai [15] for data classification, Wattanataweekul et al. [16] for image recovery, and Janngam et al. [17] for deep learning data classification. In the related setting of bilevel variational inequalities, Peng et al. [18] proposed an accelerated subgradient extragradient algorithm with a non-monotonic step size for problems that involve non-Lipschitz operators.

Several algorithms have been specifically developed for convex bilevel optimization problems. Sabach and Shtern [6] introduced the Bilevel Gradient Sequential Averaging Method (BiG-SAM), later improved by Shehu et al. [19] with inertial extrapolation, termed inertial BiG-SAM (iBiG-SAM):

$$\begin{cases} y_n = x_n + \theta_n(x_n - x_{n-1}), \\ p_n = \mathrm{prox}_{cg}(y_n - c\nabla f(y_n)), \\ q_n = y_n - \lambda\nabla\varphi(y_n), \\ x_{n+1} = \gamma_n q_n + (1-\gamma_n)p_n. \end{cases}$$

Duan and Zhang [20] advanced the iBiG-SAM algorithm by introducing an alternated inertial step, thus leading to the alternated inertial Bilevel Gradient Sequential Averaging Method (aiBiG-SAM):

$$y_n = \begin{cases} x_n, & \text{if } n \text{ is even}, \\ x_n + \theta_n(x_n - x_{n-1}), & \text{if } n \text{ is odd}, \end{cases}$$

and

$$\begin{cases} p_n = \mathrm{prox}_{cg}(I - c\nabla f)(y_n), \\ q_n = y_n - \lambda\nabla\varphi(y_n), \\ x_{n+1} = \gamma_n q_n + (1-\gamma_n)p_n, \ n \geq 1. \end{cases}$$

Duan and Zhang established that aiBiG-SAM strongly converges to a solution of the bilevel optimization problem.

Since both iBiG-SAM and aiBiG-SAM rely on a single extrapolation drawn from one earlier iterate, a natural next step is to incorporate two extrapolation steps and thus obtain a double-inertial scheme, which can improve both the convergence speed and numerical stability. The iterate update in this setting combines information from the two most recent displacements:

$$x_{n+1} = x_n + \theta(x_n - x_{n-1}) + \delta(x_{n-1} - x_{n-2}),$$

where the coefficient $\theta > 0$ provides forward momentum from the latest displacement, while $\delta < 0$ acts as a damping correction drawn from earlier iterates, thus decoupling the control of acceleration

from that of stabilization. Along these lines, Izuchukwu et al. [21] designed a double-inertial forward–reflected–anchored–backward splitting algorithm for monotone inclusion problems and proved strong convergence, Iyiola and Shehu [22] introduced a double-inertial proximal point algorithm for convex minimization and established a nonasymptotic rate of $O(1/n)$, and Thong et al. [23] devised a double-inertial algorithm for split common fixed-point problems with applications in signal processing.

More recently, Wattanataweekul, Janngam, and Suantai [24] proposed a double-inertial viscosity algorithm with linesearch for convex bilevel image restoration, and Suantai and Janngam [25] proposed a two-step inertial fixed-point algorithm for convex bilevel optimization in deep learning data classification, in which the inertial step is first applied to the iterate x_n. In the Double-Inertial proximal Gradient Method with Viscosity Approximation (DIPGM-VA), the double-inertial step

$$x_{n+1} = z_n + \theta_n(z_n - z_{n-1}) + \delta_n(z_{n-1} - z_{n-2})$$

is applied last, where z_n is the iterate obtained after the viscosity step and the forward–backward evaluations. This placement follows the structural choice of FISTA [12] and is motivated by the following observation: For a step size $c_n \in (0, 2/L_f)$, the forward–backward operator is nonexpansive, so as the algorithm progresses, the successive iterates z_n, z_{n-1}, z_{n-2} become closer to one another, which is intended to stabilize the extrapolation in the late stages of convergence. The use of two inertial coefficients with $\theta_n > 0$ and $\delta_n < 0$ follows Iyiola and Shehu [22], who show that this form of two-step extrapolation can accelerate convergence in cases where one-step inertial schemes do not.

Building on the developments surveyed above, the present paper puts forward a new double-inertial viscosity algorithm for convex bilevel optimization. Its main contributions can be summarized as follows:

(i) A new fixed-point scheme, the DIPGM-VA, is designed for solving convex bilevel optimization problems.
(ii) Under standard assumptions, DIPGM-VA is shown to strongly converge; the double-inertial extrapolation underlying the scheme simultaneously improves the stability and speeds up convergence.
(iii) We propose a hybrid deep learning architecture for medical image classification that integrates EfficientNetV2-B0 [26], a compact yet high-performing convolutional architecture, as a feature extractor with a bilevel Extreme Learning Machine (ELM) classifier solved by DIPGM-VA.
(iv) Numerical experiments on a chest X-ray pneumonia dataset indicate that DIPGM-VA attains a smaller objective value and converges faster than competing bilevel solvers.

Among children under five, pneumonia is still among the principal causes of morbidity and mortality worldwide. Chest radiography is the most commonly used imaging modality to diagnose pneumonia, but its interpretation varies between readers and is further constrained by limited resources, especially in low-income regions. For these reasons, deep-learning-based automated analyses of chest X-ray images have received substantial research interest [1, 27]. A widely adopted benchmark in this area is the publicly released pediatric chest X-ray dataset curated by Kermany et al. [27], which follows a clinically designed evaluation protocol with patient-level separation between training and test partitions. This dataset provides an appropriate testbed for the proposed bilevel optimization framework, in a setting where the classification accuracy and computational efficiency both matter in practice.

On the practical side, we apply DIPGM-VA to the training of a hybrid deep learning classifier for medical image analyses, an area where the diagnostic accuracy and computational efficiency are equally important [1, 2]. Concretely, a pretrained EfficientNetV2-B0 convolutional backbone [26] is coupled with an ELM classifier [28]; its output weights are obtained by solving a convex bilevel problem with DIPGM-VA, in which the lower level enforces sparse feature selection through L_1 regularization and the upper level performs structural risk minimization through the L_2 norm. The resulting method is evaluated on a chest X-ray pneumonia benchmark [27], where it delivers an improved optimization behavior together with a competitive classification accuracy and substantially reduced computational cost.

The remainder of this paper is organized as follows: Section 2 gathers the preliminary material; Section 3 introduces the DIPGM-VA algorithm and analyzes its convergence; Section 4 develops the hybrid deep learning classifier based on the proposed bilevel framework; Section 5 presents the numerical experiments; Section 6 closes the paper with a summary and directions for future work; and an appendix details the stagewise configuration of the EfficientNetV2-B0 backbone.

2. Preliminaries

This section fixes the notation and gathers the definitions and auxiliary results required for the convergence analysis carried out in Section 3. Throughout, let H be a real Hilbert space with an inner product $\langle\cdot,\cdot\rangle$ and an induced norm $\|\cdot\|$, and let C be a nonempty closed convex subset of H. The symbols $\mathbb{N}, \mathbb{R}, \mathbb{R}_+, \mathbb{R}_{>0}$ denote the natural numbers, real numbers, nonnegative real numbers, and positive real numbers, respectively.

Definition 2.1. *Let $T : C \to C$ be a mapping on a subset C of a real Hilbert space H.*

*(i) T is said to be **L-Lipschitz** for some constant $L \geq 0$ if*

$$\|Tu - Tv\| \leq L\|u - v\|, \ \text{for all } u, v \in C.$$

*(ii) T is called a **k-contraction** if there exists a constant $k \in [0, 1)$ such that*

$$\|Tu - Tv\| \leq k\|u - v\|, \ \text{for all } u, v \in C.$$

*(iii) T is said to be **nonexpansive** if*

$$\|Tu - Tv\| \leq \|u - v\|, \ \text{for all } u, v \in C.$$

Definition 2.2. *[29, 30] Let $T : H \to H$ be a mapping on a real Hilbert space H.*

*(i) T is said to be **firmly nonexpansive** if*

$$\|Tu - Tv\|^2 \leq \|u - v\|^2 - \|(I - T)u - (I - T)v\|^2, \ \text{for all } u, v \in H.$$

*(ii) T is said to be **α-averaged** if there exists a nonexpansive mapping $R\colon H \to H$ and a constant $\alpha \in (0, 1)$ such that $T = (1 - \alpha)I + \alpha R$. The notion of an averaged mapping was introduced in [31].*

Remark 2.3. *[29] It is a standard result in monotone operator theory that T is α-averaged if and only if*

$$\|Tu - Tv\|^2 \leq \|u - v\|^2 - \frac{1-\alpha}{\alpha}\|(I-T)u - (I-T)v\|^2, \text{ for all } u, v \in H.$$

Furthermore, T is firmly nonexpansive if and only if it is $1/2$-averaged.

Now, we collect several identities in Hilbert spaces that will be repeatedly invoked in the proof.

Lemma 2.4. *[32] Let $u, v \in H$, and $t \in [0,1]$. Then, the following relations hold in H:*

(i) $\|u+v\|^2 \leq \|u\|^2 + 2\langle v, u+v\rangle$;
(ii) $\|u \pm v\|^2 = \|u\|^2 \pm 2\langle u, v\rangle + \|v\|^2$;
(iii) $\|tu + (1-t)v\|^2 = t\|u\|^2 + (1-t)\|v\|^2 - t(1-t)\|u-v\|^2$.

Definition 2.5. *[33] Let $T: C \to C$ be a nonexpansive mapping and $T_n: C \to C$ a family of nonexpansive mappings with $\emptyset \neq \mathrm{Fix}(T) \subset \Gamma := \bigcap_{n=1}^{\infty} \mathrm{Fix}(T_n)$. The family $\{T_n\}$ is said to satisfy the NST-condition (I) with respect to the mapping T if, when the condition $\lim_{n\to\infty} \|x_n - T_n x_n\| = 0$ for any bounded sequence $\{x_n\} \subset C$, then the condition $\lim_{n\to\infty} \|x_n - T x_n\| = 0$ also holds.*

Definition 2.6. *Let $g: H \to \mathbb{R} \cup \{\infty\}$ be a proper, lower semicontinuous, convex function, and let $c > 0$. The proximal operator prox_{cg} is defined as follows:*

$$\mathrm{prox}_{cg}\ x = \mathrm{argmin}_{y \in H}\left\{ g(y) + \frac{1}{2c}\|y - x\|^2 \right\}.$$

See [7,8,29] for details. Given a convex, differentiable function $f: H \to \mathbb{R}$ with a Lipschitz continuous gradient, the forward–backward operator T associated with f and g, with step size c, is defined by the following:

$$T := \mathrm{prox}_{cg}(I - c\nabla f).$$

Remark 2.7. *The operator T defined above is referred to as the* forward–backward operator *throughout this paper. It is also known in the optimization literature as the proximal gradient operator.*

Remark 2.8. *It is well known that if $c \in (0, 2/L)$, where L is a Lipschitz constant of ∇f, then the forward–backward operator T is nonexpansive.*

The next proposition, due to Hanjing *et al.* [34], will be used below.

Proposition 2.9. *Suppose that $\omega: H \to \mathbb{R}$ is strongly convex with parameter $s > 0$ and continuously differentiable such that $\nabla\omega$ is Lipschitz continuous with constant L_ω. Then, the mapping $I - t\nabla\omega$ is a k-contraction for all*

$$0 < t \leq \frac{2}{L_\omega + s},$$

where

$$k = \sqrt{1 - \frac{2stL_\omega}{s + L_\omega}}.$$

Definition 2.10. *[35, 36] Let H be a real Hilbert space and $C \subset H$ a nonempty closed convex set. Suppose that $\{T_n\}_{n=1}^{\infty}$ is a sequence of nonexpansive self-mappings $T_n : C \to C$ whose common fixed-point set is as follows:*

$$\Gamma := \bigcap_{n=1}^{\infty} \mathrm{Fix}(T_n) \neq \emptyset.$$

The family $\{T_n\}$ is said to satisfy Condition (Z) if, for every bounded sequence $\{u_n\} \subset C$ with

$$\lim_{n\to\infty} \|u_n - T_n u_n\| = 0,$$

then every weak cluster point of $\{u_n\}$ belongs to Γ.

Invoking demiclosedness of $I - T$ when $T : C \to C$ is nonexpansive yields the following remark.

Remark 2.11. *Let $T : C \to C$ be nonexpansive. If a family of nonexpansive mappings $\{T_n\}$ satisfies the NST-condition (I) with respect to T, then $\{T_n\}$ also satisfies Condition (Z).*

The next result records the averaging coefficient of the forward–backward operator and is central to the proof of our main theorem.

Proposition 2.12. *Let $f : H \to \mathbb{R}$ be convex and differentiable with ∇f being L-Lipschitz continuous, and let $g : H \to (-\infty, \infty]$ be proper, lower semicontinuous, and convex. If $c \in (0, 2/L)$, then the forward–backward operator $T = \mathrm{prox}_{cg}(I - c\nabla f)$ is $2/(4-cL)$-averaged. In particular, for all $u, v \in H$,*

$$\|Tu - Tv\|^2 \leq \|u - v\|^2 - \left(1 - \frac{cL}{2}\right) \|(I - T)u - (I - T)v\|^2.$$

Proof. Since $c \in (0, 2/L)$, the gradient ∇f is $1/L$-cocoercive by the Baillon–Haddad theorem, which implies that the operator $I - c\nabla f$ is $cL/2$-averaged. The proximal operator prox_{cg} is firmly nonexpansive, and thus $1/2$-averaged. By Bauschke and Combettes [29], the composition $T = \mathrm{prox}_{cg}(I - c\nabla f)$ is $2/(4 - cL)$-averaged. Consequently, the standard averaged inequality holds with the coefficient $(1 - \alpha)/\alpha = 1 - cL/2$, where $\alpha = 2/(4 - cL)$. □

We close this section with two lemmas used in the convergence argument.

Lemma 2.13. *[37] Let $\{d_n\} \subset \mathbb{R}_+$, $\{b_n\} \subset \mathbb{R}$, and $\{h_n\} \subset (0, 1)$ such that $\sum_{n=1}^{\infty} h_n = \infty$. Suppose that*

$$d_{n+1} \leq (1 - h_n)d_n + h_n b_n, \ \text{for all } n \in \mathbb{N}.$$

If $\limsup_{i\to\infty} b_{n_i} \leq 0$ for any subsequence $\{d_{n_i}\}$ of $\{d_n\}$ that satisfies

$$\liminf_{i\to\infty} (d_{n_i+1} - d_{n_i}) \geq 0,$$

then $\lim_{n\to\infty} d_n = 0$.

Lemma 2.14. *[38] Let H be a real Hilbert space, and consider the forward–backward operator T associated with the functions f and g for a parameter c, where $g : H \to \mathbb{R} \cup \{\infty\}$ is a proper, lower semicontinuous, convex function, and $f : H \to \mathbb{R}$ is a convex and differentiable function whose gradient ∇f is L-Lipschitz continuous for some constant $L > 0$. Let $\{T_n\}$ be a sequence of forward–backward operators corresponding to the same functions f and g, but with parameters c_n, where $c_n \to c$ and $c_n, c \in (0, 2/L)$. Then, the sequence $\{T_n\}$ satisfies the NST-condition (I) with respect to T.*

3. Main results

This section introduces the DIPGM-VA algorithm and carries out its convergence analysis: the sequence produced by Algorithm 1 is shown to strongly converge, with the limit point solving the convex bilevel problem (1.1). Throughout the section, the mappings f, g, and φ from Section 1 are taken to satisfy the following two conditions.

(A1) $f\colon H \to \mathbb{R}$ is convex and differentiable, and ∇f is Lipschitz continuous with constant $L_f > 0$. The function $g\colon H \to (-\infty, \infty]$ is proper, lower semicontinuous, and convex.

(A2) $\varphi\colon H \to \mathbb{R}$ is strongly convex with parameter σ, differentiable, and $\nabla\varphi$ is L_φ-Lipschitz continuous.

Remark 3.1. *Under assumption (A1) and the condition $c \in (0, 2/L_f)$, the forward–backward operator $T = \operatorname{prox}_{cg}(I - c\nabla f)$ is nonexpansive and satisfies* $\operatorname{Fix}(T) = S_*$*, where S_* is the solution set of the lower-level problem* (1.2)*; e.g., see [29].*

Building on the fixed-point characterization of Remark 3.1, we now introduce Algorithm 1 to solve Problem (1.1).

Algorithm 1 Double-Inertial Proximal Gradient Method with Viscosity Approximation (DIPGM-VA)

Initialization: Let $\{\alpha_n\}, \{\gamma_n\} \subset (0, 1)$, and $\{\tau_n\} \subset \mathbb{R}_+$ and let $\{\mu_n\}, \{\rho_n\} \subset \mathbb{R}_{>0}$ be bounded sequences. Choose arbitrary starting points $x_1 \in H$ and set $z_{-1} = z_0 = x_1$. Let $\{c_n\} \subset (0, 2/L_f)$ with $c_n \to c$ as $n \to \infty$, where $c \in (0, 2/L_f)$.

For $n \in \mathbb{N}$, do the following steps:

Step 1. Compute w_n, y_n, and z_n:

$$\begin{aligned} w_n &= (1 - \alpha_n)x_n + \alpha_n(I - s\nabla\varphi)(x_n), \\ y_n &= \operatorname{prox}_{c_n g}(I - c_n\nabla f)(w_n), \\ z_n &= (1 - \gamma_n)x_n + \gamma_n \operatorname{prox}_{c_n g}(I - c_n\nabla f)(y_n). \end{aligned}$$

Step 2. Compute the inertial step:

$$\theta_n = \begin{cases} \min\left\{\mu_n, \dfrac{\alpha_n\tau_n}{\|z_n - z_{n-1}\|}\right\}, & \text{if } z_n \neq z_{n-1}, \\ \mu_n, & \text{otherwise}, \end{cases}$$

$$\delta_n = \begin{cases} \max\left\{-\rho_n, \dfrac{-\alpha_n\tau_n}{\|z_{n-1} - z_{n-2}\|}\right\}, & \text{if } z_{n-1} \neq z_{n-2}, \\ -\rho_n, & \text{otherwise}. \end{cases}$$

Compute the following:

$$x_{n+1} = z_n + \theta_n(z_n - z_{n-1}) + \delta_n(z_{n-1} - z_{n-2}).$$

Remark 3.2. *The two inertial coefficients in Step 2 play complementary roles: $\theta_n > 0$ provides forward momentum along the latest displacement, while $\delta_n < 0$ acts as a damping correction that suppresses*

the oscillations typically observed in single-inertial schemes. Two-step inertial strategies of this kind have recently been shown to improve both the stability and the convergence [21–23], and the effect is visible in Figure 3, where DIPGM-VA smoothly descends.

Our main theorem below shows that Algorithm 1 strongly converges to a point in Λ, which is the solution set of (1.1).

Theorem 3.3. *Let $\{c_n\} \subset (0, 2/L_f)$ such that $c_n \to c \in (0, 2/L_f)$, and assume that $S_* \neq \emptyset$. Let $F := I - s\nabla\varphi$, where $s \in (0, 2/(L_\varphi + \sigma))$, with a contraction constant $k \in (0, 1)$. Let $\{x_n\}$ be the sequence generated by Algorithm 1. Suppose that the control sequences $\{\tau_n\}$, $\{\alpha_n\}$, and $\{\gamma_n\}$ satisfy the following conditions:*

(C1) $\lim_{n\to\infty} \tau_n = 0$*;*
(C2) $\lim_{n\to\infty} \alpha_n = 0$ *and* $\sum_{n=1}^{\infty} \alpha_n = \infty$*;*
(C3) $0 < \underline{\gamma} \leq \gamma_n \leq \bar{\gamma} < 1$ *for some* $\underline{\gamma}, \bar{\gamma} \in \mathbb{R}$*.*

Then, the sequence $\{x_n\}$ strongly converges to a point $x^* \in \Lambda$, where Λ denotes the solution set of (1.1)*. Moreover, x^* satisfies $x^* = P_{S_*}(I - s\nabla\varphi)(x^*)$.*

Proof. **Step 1: Boundedness of the sequence.** By Proposition 2.9, the operator $F = I - s\nabla\varphi$ is a k-contraction for some $k \in (0, 1)$, while Proposition 2.12 guarantees that each $T_n = \operatorname{prox}_{c_n g}(I - c_n \nabla f)$ is $1/2$-averaged and, in particular, nonexpansive. Moreover, Remark 3.1 yields $\operatorname{Fix}(T) = \operatorname{Fix}(T_n) = S_*$. Since F is a k-contraction and P_{S_*} is nonexpansive, the composition $P_{S_*} \circ F \colon H \to H$ is a k-contraction. By the Banach contraction mapping principle, there exists a unique fixed point $x^* \in H$ that satisfies $x^* = P_{S_*}F(x^*)$. Since $P_{S_*}F(x^*) \in S_*$, it follows that $x^* \in S_*$. By the definition of w_n, we obtain the following:

$$\begin{aligned} \|w_n - x^*\| &= \|(1 - \alpha_n)x_n + \alpha_n F(x_n) - x^*\| \\ &\leq (1 - \alpha_n)\|x_n - x^*\| + \alpha_n\|F(x_n) - F(x^*)\| + \alpha_n\|F(x^*) - x^*\| \\ &\leq (1 - \alpha_n(1 - k))\|x_n - x^*\| + \alpha_n\|F(x^*) - x^*\|. \end{aligned} \tag{3.1}$$

For y_n, the nonexpansiveness of T_n yields the following:

$$\|y_n - x^*\| = \|T_n w_n - x^*\| \leq \|w_n - x^*\|. \tag{3.2}$$

Then, inserting the definition of z_n together with Estimates (3.1) and (3.2) yields the following:

$$\begin{aligned} \|z_n - x^*\| &= \|(1 - \gamma_n)x_n + \gamma_n T_n y_n - x^*\| \\ &\leq (1 - \gamma_n)\|x_n - x^*\| + \gamma_n\|T_n y_n - x^*\| \\ &\leq (1 - \gamma_n)\|x_n - x^*\| + \gamma_n\|y_n - x^*\| \\ &\leq (1 - \gamma_n)\|x_n - x^*\| + \gamma_n\Big[(1 - \alpha_n(1 - k))\|x_n - x^*\| + \alpha_n\|F(x^*) - x^*\|\Big] \\ &\leq (1 - \gamma_n\alpha_n(1 - k))\|x_n - x^*\| + \gamma_n\alpha_n\|F(x^*) - x^*\|. \end{aligned} \tag{3.3}$$

Then, it follows from the definition of x_{n+1} and (3.3) that

$$\|x_{n+1} - x^*\| \leq \|z_n - x^*\| + \theta_n\|z_n - z_{n-1}\| + |\delta_n|\|z_{n-1} - z_{n-2}\|$$

$$
\begin{aligned}
&\leq (1-\gamma_n\alpha_n(1-k))\|x_n - x^*\| + \gamma_n\alpha_n\|F(x^*) - x^*\| \\
&\quad + \theta_n\|z_n - z_{n-1}\| + |\delta_n|\|z_{n-1} - z_{n-2}\| \\
&= \Big(1-\alpha_n\gamma_n(1-k)\Big)\|x_n - x^*\| \\
&\quad + \alpha_n\gamma_n\left(\frac{\theta_n}{\alpha_n\gamma_n}\|z_n - z_{n-1}\| + \frac{|\delta_n|}{\alpha_n\gamma_n}\|z_{n-1} - z_{n-2}\| + \|F(x^*) - x^*\|\right).
\end{aligned}
$$

Recalling the constructions of θ_n and δ_n together with assumption (C1),

$$
\frac{\theta_n}{\alpha_n\gamma_n}\|z_n - z_{n-1}\| \to 0 \quad \text{and} \quad \frac{|\delta_n|}{\alpha_n\gamma_n}\|z_{n-1} - z_{n-2}\| \to 0 \text{ as } n \to \infty. \tag{3.4}
$$

Note that by the definitions of θ_n and δ_n in Algorithm 1, we have $\theta_n \leq \mu_n$ and $|\delta_n| \leq \rho_n$ for all $n \geq 1$, where $\{\mu_n\}$ and $\{\rho_n\}$ are bounded sequences. Moreover, by (C3), $\gamma_n \geq \underline{\gamma} > 0$, which implies that the sequences $\theta_n/(\alpha_n\gamma_n)\|z_n - z_{n-1}\|$ and $|\delta_n|/(\alpha_n\gamma_n)\|z_{n-1} - z_{n-2}\|$ are well-defined real numbers for every $n \geq 1$. Since both converge to 0 by (3.4), they are bounded. Thus, there exists $M > 0$ such that

$$
\frac{\theta_n}{\alpha_n\gamma_n}\|z_n - z_{n-1}\| \leq M \quad \text{and} \quad \frac{|\delta_n|}{\alpha_n\gamma_n}\|z_{n-1} - z_{n-2}\| \leq M, \text{ for all } n \geq 1.
$$

Then,

$$
\begin{aligned}
\|x_{n+1} - x^*\| &\leq \Big(1-\alpha_n\gamma_n(1-k)\Big)\|x_n - x^*\| + \alpha_n\gamma_n(1-k)\left(\frac{2M + \|F(x^*) - x^*\|}{1-k}\right) \\
&\leq \max\left\{\|x_n - x^*\|, \ \frac{2M + \|F(x^*) - x^*\|}{1-k}\right\}.
\end{aligned}
$$

Therefore, an induction on n yields the following:

$$
\|x_n - x^*\| \leq \max\left\{\|x_1 - x^*\|, \ \frac{2M + \|F(x^*) - x^*\|}{1-k}\right\}, \text{ for all } n \geq 1.
$$

Therefore, the sequence $\{x_n\}$ is bounded, and so are the sequences $\{w_n\}$, $\{T_n w_n\}$, and $\{y_n\}$.

Step 2: Key inequality for the objective sequence. By Lemma 2.4 (i) and (iii), we obtain the following:

$$
\begin{aligned}
\|w_n - x^*\|^2 &= \|(1-\alpha_n)(x_n - x^*) + \alpha_n(F(x_n) - F(x^*)) + \alpha_n(F(x^*) - x^*)\|^2 \\
&\leq \|(1-\alpha_n)(x_n - x^*) + \alpha_n(F(x_n) - F(x^*))\|^2 + 2\alpha_n\langle F(x^*) - x^*, w_n - x^*\rangle \\
&\leq (1-\alpha_n)\|x_n - x^*\|^2 + \alpha_n\|F(x_n) - F(x^*)\|^2 + 2\alpha_n\langle F(x^*) - x^*, w_n - x^*\rangle \\
&\leq \Big(1-\alpha_n(1-k)\Big)\|x_n - x^*\|^2 + 2\alpha_n\langle F(x^*) - x^*, w_n - x^*\rangle.
\end{aligned} \tag{3.5}
$$

By Lemma 2.4 (iii) and (3.5), we obtain the following:

$$
\begin{aligned}
\|z_n - x^*\|^2 &= \|(1-\gamma_n)(x_n - x^*) + \gamma_n(T_n y_n - x^*)\|^2 \\
&\leq (1-\gamma_n)\|x_n - x^*\|^2 + \gamma_n\|w_n - x^*\|^2 - \gamma_n(1-\gamma_n)\|x_n - T_n y_n\|^2 \\
&\leq (1-\gamma_n)\|x_n - x^*\|^2 + \gamma_n\Big[\Big(1-\alpha_n(1-k)\Big)\|x_n - x^*\|^2 + 2\alpha_n\langle F(x^*) - x^*, w_n - x^*\rangle\Big]
\end{aligned}
$$

$$
\begin{aligned}
&- \gamma_n(1-\gamma_n)\|x_n - T_n y_n\|^2 \\
&= \Big(1 - \gamma_n\alpha_n(1-k)\Big)\|x_n - x^*\|^2 + 2\gamma_n\alpha_n\langle F(x^*) - x^*, w_n - x^*\rangle \\
&\quad - \gamma_n(1-\gamma_n)\|x_n - T_n y_n\|^2, \qquad (3.6)
\end{aligned}
$$

where we used $\|T_n y_n - x^*\| \leq \|y_n - x^*\| \leq \|w_n - x^*\|$ by the nonexpansiveness of T_n.

Furthermore, by Lemma 2.4 (ii) and (3.6), we obtain the following:

$$
\begin{aligned}
\|x_{n+1} - x^*\|^2 &= \|z_n - x^*\|^2 + 2\theta_n\langle z_n - x^*, z_n - z_{n-1}\rangle + 2\delta_n\langle z_n - x^*, z_{n-1} - z_{n-2}\rangle \\
&\quad + \|\theta_n(z_n - z_{n-1}) + \delta_n(z_{n-1} - z_{n-2})\|^2 \\
&\leq \|z_n - x^*\|^2 + 2\theta_n\|z_n - x^*\|\|z_n - z_{n-1}\| + 2|\delta_n|\|z_n - x^*\|\|z_{n-1} - z_{n-2}\| \\
&\quad + \theta_n^2\|z_n - z_{n-1}\|^2 + \delta_n^2\|z_{n-1} - z_{n-2}\|^2 + 2\theta_n|\delta_n|\|z_n - z_{n-1}\|\|z_{n-1} - z_{n-2}\| \\
&\leq \Big(1 - \alpha_n\gamma_n(1-k)\Big)\|x_n - x^*\|^2 + 2\theta_n\|z_n - x^*\|\|z_n - z_{n-1}\| \\
&\quad + 2|\delta_n|\|x_n - x^*\|\|z_{n-1} - z_{n-2}\| + \theta_n^2\|z_n - z_{n-1}\|^2 + \delta_n^2\|z_{n-1} - z_{n-2}\|^2 \\
&\quad + 2\theta_n|\delta_n|\|z_n - z_{n-1}\|\|z_{n-1} - z_{n-2}\| + 2\alpha_n\gamma_n\langle F(x^*) - x^*, w_n - x^*\rangle \\
&\quad - \gamma_n(1-\gamma_n)\|x_n - T_n y_n\|^2 \\
&= \Big(1 - \alpha_n\gamma_n(1-k)\Big)\|x_n - x^*\|^2 - \gamma_n(1-\gamma_n)\|x_n - T_n y_n\|^2 + \alpha_n\gamma_n(1-k)b_n, \qquad (3.7)
\end{aligned}
$$

where

$$
\begin{aligned}
b_n = \frac{1}{1-k}\Bigg(&\frac{2\theta_n}{\alpha_n\gamma_n}\|z_n - x^*\|\|z_n - z_{n-1}\| + \frac{2|\delta_n|}{\alpha_n\gamma_n}\|x_n - x^*\|\|z_{n-1} - z_{n-2}\| \\
&+ \frac{\theta_n^2}{\alpha_n\gamma_n}\|z_n - z_{n-1}\|^2 + \frac{\delta_n^2}{\alpha_n\gamma_n}\|z_{n-1} - z_{n-2}\|^2 + \frac{2\theta_n|\delta_n|}{\alpha_n\gamma_n}\|z_n - z_{n-1}\|\|z_{n-1} - z_{n-2}\| \\
&+ 2\langle F(x^*) - x^*, w_n - x^*\rangle\Bigg).
\end{aligned}
$$

It follows that

$$
\gamma_n(1-\gamma_n)\|x_n - T_n y_n\|^2 \leq \|x_n - x^*\|^2 - \|x_{n+1} - x^*\|^2 + \alpha_n\gamma_n(1-k)\overline{B}, \qquad (3.8)
$$

where $\overline{B} = \sup\{b_n : n \in \mathbb{N}\}$.

Step 3: Strong convergence to the bilevel solution. It remains to prove that $x_n \to x^*$. In order to invoke Lemma 2.13, set $a_n := \|x_n - x^*\|^2$ and $t_n := \alpha_n\gamma_n(1-k)$. Then, Relation (3.7) gives the following:

$$
a_{n+1} \leq (1 - t_n)a_n + t_n b_n.
$$

Suppose that $\{a_{n_i}\}$ is a subsequence of $\{a_n\}$ such that $\liminf_{i\to\infty}(a_{n_i+1} - a_{n_i}) \geq 0$. Using (3.8) and (C3), we obtain the following:

$$
\begin{aligned}
\limsup_{i\to\infty} \gamma_{n_i}(1-\gamma_{n_i})\|x_{n_i} - T_{n_i} y_{n_i}\|^2 &\leq \limsup_{i\to\infty}\Big(a_{n_i} - a_{n_i+1} + \alpha_{n_i}\gamma_{n_i}(1-k)\overline{B}\Big) \\
&= -\liminf_{i\to\infty}(a_{n_i+1} - a_{n_i}) + \lim_{i\to\infty}\alpha_{n_i}\gamma_{n_i}(1-k)\overline{B} \\
&\leq 0. \qquad (3.9)
\end{aligned}
$$

From (3.9) and (C3), we obtain the following:

$$\lim_{i\to\infty} \|x_{n_i} - T_{n_i} y_{n_i}\| = 0. \tag{3.10}$$

The boundedness of $\{x_n\}$ and $\{F(x_n)\}$ implies that the sequence $\{\|F(x_{n_i}) - x_{n_i}\|\}$ is also bounded. This, together with $\alpha_{n_i} \to 0$ (by (C2)), yields the following:

$$\lim_{i\to\infty} \|w_{n_i} - x_{n_i}\| = \lim_{i\to\infty} \alpha_{n_i} \|F(x_{n_i}) - x_{n_i}\| = 0. \tag{3.11}$$

By the nonexpansiveness of T_{n_i} and the definition $y_{n_i} = T_{n_i} w_{n_i}$, it follows from (3.11) that

$$\lim_{i\to\infty} \|y_{n_i} - T_{n_i} x_{n_i}\| \le \lim_{i\to\infty} \|w_{n_i} - x_{n_i}\| = 0.$$

By Proposition 2.12, since $c_{n_i} \in (0, 2/L_f)$, the operator T_{n_i} is $\hat{\alpha}_{n_i}$-averaged with $\hat{\alpha}_{n_i} = 2/(4 - c_{n_i} L_f)$. Hence,

$$\|T_{n_i} u - T_{n_i} v\|^2 \le \|u - v\|^2 - \tau_{n_i} \|(I - T_{n_i})u - (I - T_{n_i})v\|^2, \text{ for all } u, v \in H,$$

where $\tau_{n_i} = 1 - c_{n_i} L_f/2$. Since $c_{n_i} \to c < 2/L_f$, it follows that $1 - c_{n_i} L_f/2 \to 1 - cL_f/2 > 0$. Thus, there exists a constant $\tau > 0$ such that $\tau_{n_i} \ge \tau$ for all sufficiently large i. By applying the averaged inequality with $u = x_{n_i}$ and $v = y_{n_i}$ and using this uniform lower bound τ, we obtain the following:

$$\|T_{n_i} x_{n_i} - T_{n_i} y_{n_i}\|^2 \le \|x_{n_i} - y_{n_i}\|^2 - \tau \|(I - T_{n_i})x_{n_i} - (I - T_{n_i})y_{n_i}\|^2.$$

To simplify the analysis, let $e_i := (y_{n_i} - T_{n_i} x_{n_i}) - (x_{n_i} - T_{n_i} y_{n_i})$. By (3.10) and the established limit above, we have $\lim_{i\to\infty} \|e_i\| = 0$. Using this error term, we can rewrite the averaged inequality as follows:

$$\|(x_{n_i} - y_{n_i}) + e_i\|^2 \le \|x_{n_i} - y_{n_i}\|^2 - \tau \|2(x_{n_i} - y_{n_i}) + e_i\|^2, \tag{3.12}$$

where we used the fact that $\|y_{n_i} - x_{n_i} - e_i\|^2 = \|-(x_{n_i} - y_{n_i} + e_i)\|^2 = \|(x_{n_i} - y_{n_i}) + e_i\|^2$. By applying Lemma 2.4 (ii) to expand the norms on both sides of (3.12), we obtain the following:

$$\|x_{n_i} - y_{n_i}\|^2 + 2\langle x_{n_i} - y_{n_i}, e_i\rangle + \|e_i\|^2 \le \|x_{n_i} - y_{n_i}\|^2 - \tau\Big(4\|x_{n_i} - y_{n_i}\|^2 + 4\langle x_{n_i} - y_{n_i}, e_i\rangle + \|e_i\|^2\Big).$$

By canceling $\|x_{n_i} - y_{n_i}\|^2$ from both sides and rearranging, we deduce that

$$4\tau \|x_{n_i} - y_{n_i}\|^2 \le -(4\tau + 2)\langle x_{n_i} - y_{n_i}, e_i\rangle - (\tau + 1)\|e_i\|^2. \tag{3.13}$$

Since $\{x_n\}$ and $\{y_n\}$ are bounded, there exists a constant $M' := \sup_i \|x_{n_i} - y_{n_i}\| < \infty$ by the boundedness established in Step 1. By the Cauchy–Schwarz inequality,

$$|\langle x_{n_i} - y_{n_i}, e_i\rangle| \le \|x_{n_i} - y_{n_i}\| \|e_i\| \le M' \|e_i\|.$$

Since $\|e_i\| \to 0$, it follows that $\|e_i\|^2 \to 0$. Moreover, $|\langle x_{n_i} - y_{n_i}, e_i\rangle| \le M' \|e_i\| \to 0$. Therefore, taking the limit superior as $i \to \infty$ on both sides of (3.13), we obtain the following:

$$4\tau \limsup_{i\to\infty} \|x_{n_i} - y_{n_i}\|^2 \le 0.$$

Because $\tau > 0$, this immediately implies the following:

$$\lim_{i\to\infty} \|x_{n_i} - y_{n_i}\| = 0. \tag{3.14}$$

Using (3.11) and (3.14), we obtain the following:

$$\begin{aligned} \|w_{n_i} - T_{n_i} w_{n_i}\| &= \|w_{n_i} - y_{n_i}\| \\ &\leq \|w_{n_i} - x_{n_i}\| + \|x_{n_i} - y_{n_i}\| \\ &\to 0 \text{ as } i \to \infty. \end{aligned} \tag{3.15}$$

It remains to verify $\limsup_{i\to\infty} b_{n_i} \leq 0$, for which it suffices to establish the following:

$$\limsup_{i\to\infty} \langle F(x^*) - x^*, w_{n_i} - x^* \rangle \leq 0. \tag{3.16}$$

Since $\{w_{n_i}\}$ is bounded, there exists a subsequence $\{w_{n_{i_j}}\}$ of $\{w_{n_i}\}$ and $\bar{w} \in H$ such that $w_{n_{i_j}} \rightharpoonup \bar{w}$ as $j \to \infty$ and

$$\begin{aligned} \limsup_{i\to\infty} \langle F(x^*) - x^*, w_{n_i} - x^* \rangle &= \lim_{j\to\infty} \langle F(x^*) - x^*, w_{n_{i_j}} - x^* \rangle \\ &= \langle F(x^*) - x^*, \bar{w} - x^* \rangle. \end{aligned}$$

By Lemma 2.14, $\{T_n\}$ satisfies the NST-condition (I) with respect to $T := \mathrm{prox}_{cg}(I - c\nabla f)$. Therefore, by Remark 2.11, $\{T_n\}$ satisfies Condition (Z). By combining this with (3.15) and the identity $\Gamma = S_*$ (Remark 3.1), we obtain $\bar{w} \in S_*$. From $x^* = P_{S_*} F(x^*)$, we obtain the following:

$$\langle F(x^*) - x^*, \hat{x} - x^* \rangle \leq 0, \text{ for all } \hat{x} \in S_*.$$

Choosing $\hat{x} = \bar{w}$ gives the following:

$$\langle F(x^*) - x^*, \bar{w} - x^* \rangle \leq 0.$$

Hence, the variational inequality (3.16) holds. By Lemma 2.13, the sequence $\{x_n\}$ strongly converges to x^*.

To conclude, we verify that x^* solves Problem (1.1). Recall from Step 1 that $x^* = P_{S_*} F(x^*)$. Since $F = I - s\nabla\varphi$, it follows that

$$F(x^*) = x^* - s\nabla\varphi(x^*),$$

which implies that

$$\begin{aligned} 0 &\leq \langle P_{S_*} F(x^*) - F(x^*),\ x - P_{S_*} F(x^*) \rangle \\ &= \langle x^* - F(x^*),\ x - x^* \rangle \\ &= \langle x^* - (x^* - s\nabla\varphi(x^*)),\ x - x^* \rangle \\ &= s\langle \nabla\varphi(x^*),\ x - x^* \rangle, \quad \text{for all } x \in S_*. \end{aligned}$$

This, together with $0 < s$, gives the following:

$$0 \leq \langle \nabla\varphi(x^*),\ x - x^* \rangle, \quad \text{for all } x \in S_*.$$

Hence, x^* is the solution of the outer-level problem (1.1). □

4. Application: Bilevel extreme learning machine for image classification

4.1. Motivation and architecture overview

Recent progress in deep learning has been driven by Convolutional Neural Networks (CNNs), which have achieved remarkable success in visual recognition [26, 39, 40], but their deployment in resource-constrained or data-limited settings remains challenging due to the high computational burden of end-to-end training and the risk of overfitting.

A principled strategy is to decouple representation learning and classification: a pretrained CNN backbone serves as a fixed feature extractor, whereas an efficient classifier is trained on the extracted representations [41, 42]. Traditional fully connected classifiers trained via Stochastic Gradient Descent (SGD) suffer from hyperparameter sensitivity and slow convergence.

The ELM [28] offers an alternative: input-to-hidden weights are randomly fixed, and output weights are analytically determined, thus eliminating iterative training and achieving orders-of-magnitude speedups [43].

However, the standard ELM pseudoinverse solution can suffer from numerical instability and overfitting. Ridge regularization, also known as L_2 regularization, partially addresses this [44] but does not produce sparse solutions, whereas L_1 regularization, known as the Least Absolute Shrinkage and Selection Operator (LASSO) [45], induces sparsity by pruning irrelevant hidden neurons.

This study proposes a hybrid deep learning architecture that integrates EfficientNetV2-B0 [26] as a feature extraction backbone with a bilevel ELM classifier whose output weights are determined by solving a convex bilevel optimization problem via the proposed DIPGM-VA. The bilevel formulation combines an L_1-regularized lower-level problem for sparsity with an L_2-norm upper-level problem for stability, thus yielding sparse, numerically stable, and unique output weights.

4.2. Feature extraction with EfficientNetV2-B0

4.2.1. Background on EfficientNetV2

EfficientNetV2 [26] extends the compound scaling approach of EfficientNet [46] by incorporating training-aware neural architecture search, Fused-MBConv blocks, and progressive learning with adaptive regularization. We refer to [26] for the full architectural details.

In this work, we adopt EfficientNetV2-B0 as the feature extraction backbone. EfficientNetV2-B0 is the smallest variant of the EfficientNetV2 family, and is comprised of approximately 5.9 million parameters and produces a 1280-dimensional feature vector after global average pooling. Its compact size makes it suitable for transfer learning scenarios where computational resources are limited, whereas its Neural Architecture Search (NAS)-optimized architecture ensures a strong representational capacity.

4.2.2. Preprocessing: Contrast-limited adaptive histogram equalization (CLAHE)

Prior to feature extraction, we convert input images to the CIE $L^*a^*b^*$ color space. This is a perceptually uniform representation in which L^* encodes lightness and a^*, b^* encode chromaticity. Then, we apply Contrast-limited adaptive histogram equalization (CLAHE) [47, 48] to the L^* channel with clip limit $\tau = 2.0$ and tile grid size 8×8 to enhance the structural contrast while preserving chromatic information.

4.2.3. Two-stage backbone fine-tuning

Rather than directly using the pretrained backbone as a frozen feature extractor, we adopt a two-stage fine-tuning strategy [49]. In Stage 1, all backbone parameters are frozen and only a newly initialized classification head is trained using AdamW [50] with a learning rate of $\eta_1 = 10^{-3}$, cosine annealing, label smoothing of $\epsilon = 0.1$, and early stopping (patience of four). A weighted cross-entropy loss with inverse-frequency class weights $w_c = N_{\text{train}}/(C \cdot N_c)$ addresses the class imbalance. In Stage 2, the last 50% of the backbone is unfrozen and fine-tuned with a learning rate of $\eta_2 = 2 \times 10^{-5}$, reduced label smoothing ($\epsilon = 0.05$), gradient clipping ($\|\nabla\|_2 \leq 1.0$), and weight decay of 10^{-4} (early stopping with a patience of five). The best model weights by validation accuracy are saved and the backbone is frozen for subsequent feature extraction.

4.2.4. Stagewise architecture of EfficientNetV2-B0

The feature extraction process is defined as a composition of transformations through eight sequential stages (Stages 0 to 7), including Fused-MBConv blocks and MBConv blocks with Squeeze-and-Excitation (SE) modules. The detailed layer configurations and mathematical descriptions are provided in the Appendix.

4.2.5. Feature normalization

After extraction, the raw feature vectors are standardized using a StandardScaler fitted on the training set as follows:

$$\tilde{f}_j = \frac{f_j - \hat{\mu}_j}{\hat{\sigma}_j}, \ j = 1, \ldots, d,$$

where $\hat{\mu}_j$ and $\hat{\sigma}_j$ are the sample mean and standard deviation of the jth feature computed over the training set, respectively. This normalization ensures zero mean and unit variance, which is essential for the numerical stability of the subsequent ELM hidden layer computation.

4.3. Extreme learning machine classifier with bilevel optimization

4.3.1. Background on ELM

In an ELM, input-to-hidden weights and biases are randomly generated and fixed; only the output weights are learned. Under mild conditions on the activation function, ELMs possess a universal approximation capability [51] and have been successfully combined with deep CNN feature extractors [52–54].

4.3.2. ELM formulation

Let $d = 1280$ denote the feature dimension extracted from the EfficientNetV2-B0 backbone. For a training set of N samples, define the normalized feature matrix $\mathbf{F} = [\tilde{\mathbf{f}}_1, \tilde{\mathbf{f}}_2, \ldots, \tilde{\mathbf{f}}_N]^\top \in \mathbb{R}^{N \times d}$ and the one-hot encoded target matrix $\mathbf{T} \in \mathbb{R}^{N \times C}$ for C classes.

The ELM hidden layer is defined by random input weights $\mathbf{W} \in \mathbb{R}^{L \times d}$ and biases $\mathbf{b} \in \mathbb{R}^L$, where L is the number of hidden neurons. Both $\mathbf{W}$ and $\mathbf{b}$ are sampled from a zero-mean normal distribution with a variance of $2/(d + L)$ (following the Xavier initialization [55]) and remain fixed throughout training.

The hidden layer output matrix $\mathbf{H} \in \mathbb{R}^{N\times L}$ is computed as follows:

$$\mathbf{H} = \phi(\mathbf{F}\mathbf{W}^\top + \mathbf{1}_N\mathbf{b}^\top),$$

where $\phi(\cdot)$ is a nonlinear activation function applied elementwise [Rectified Linear Unit (ReLU) or hyperbolic tangent], and $\mathbf{1}_N \in \mathbb{R}^N$ denotes the all-ones vector for broadcasting.

4.3.3. Class-weighted formulation and limitations of standard ELM

To handle the class imbalance, we define $\mathbf{D} = \mathrm{diag}(\sqrt{w_1}, \ldots, \sqrt{w_N})$ with $w_i = N/(C \cdot N_{c_i})$ and form weighted matrices $\widetilde{\mathbf{H}} = \mathbf{D}\mathbf{H} \in \mathbb{R}^{N\times L}$ and $\widetilde{\mathbf{T}} = \mathbf{D}\mathbf{T} \in \mathbb{R}^{N\times C}$. The standard ELM determines output weights $\boldsymbol{\beta} \in \mathbb{R}^{L\times C}$ via the Moore–Penrose pseudoinverse $\boldsymbol{\beta} = \widetilde{\mathbf{H}}^\dagger\widetilde{\mathbf{T}}$, which suffers from numerical instability and overfitting. Ridge regularization [44] addresses the stability but does not produce sparse solutions.

4.3.4. Bilevel optimization formulation

To simultaneously achieve sparsity, numerical stability, and uniqueness of the output weights, we reformulate the ELM weight determination as a convex bilevel optimization problem. This formulation connects the proposed DIPGM-VA algorithm to the hybrid CNN-ELM architecture. Throughout this section, we identify $\mathbb{R}^{L\times C}$ with the Hilbert space $\mathbb{R}^{LC}$ equipped with the Frobenius inner product $\langle A, B\rangle_F = \mathrm{tr}(A^\top B)$, so that $\|\cdot\|_F$ coincides with the Hilbert space norm $\|\cdot\|$ used in Sections 2 and 3.

LASSO lower-level problem. The lower-level problem incorporates an L_1-norm penalty as follows:

$$\mathcal{S}^* = \mathrm{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^{L\times C}} \left\|\widetilde{\mathbf{H}}\boldsymbol{\beta} - \widetilde{\mathbf{T}}\right\|_F^2 + \lambda_{\mathrm{reg}}\|\boldsymbol{\beta}\|_1, \tag{4.1}$$

where $\|\boldsymbol{\beta}\|_1 = \sum_{j,c}|\beta_{jc}|$ is the elementwise L_1-norm, and $\lambda_{\mathrm{reg}} > 0$ controls the sparsity-fidelity tradeoff.

Minimum-norm upper-level selection. Among all optimal solutions $\mathcal{S}^*$ to the LASSO problem, we seek the one that additionally minimizes the L_2-norm as follows:

$$\boldsymbol{\beta}^* = \mathrm{argmin}_{\boldsymbol{\beta}\in\mathcal{S}^*} \frac{1}{2}\|\boldsymbol{\beta}\|_F^2. \tag{4.2}$$

Integrated bilevel problem. By combining (4.1) and (4.2), the complete bilevel optimization problem is

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^{L\times C}} \frac{1}{2}\|\boldsymbol{\beta}\|_F^2 \tag{4.3}$$

subject to

$$\boldsymbol{\beta} \in \mathrm{argmin}_{\boldsymbol{\beta}'\in\mathbb{R}^{L\times C}} \left\|\widetilde{\mathbf{H}}\boldsymbol{\beta}' - \widetilde{\mathbf{T}}\right\|_F^2 + \lambda_{\mathrm{reg}}\|\boldsymbol{\beta}'\|_1.$$

The lower-level problem determines the optimal subspace of sparse weights, whereas the upper-level problem selects the unique minimum-norm element from that set. Although the minimum-norm selection over a solution set is a classical device in convex bilevel optimization [6], it is well motivated in the ELM setting: output weights of a smaller norm are associated with larger classification margins and improved generalization in the extreme learning machine theory [28, 43].

4.3.5. Solution via DIPGM-VA

The bilevel problem (4.3) is solved using Algorithm 1 with the following instantiation. The lower-level LASSO problem has the composite structure $\min_{\beta} f(\boldsymbol{\beta}) + h(\boldsymbol{\beta})$, where $f(\boldsymbol{\beta}) = \|\widetilde{\mathbf{H}}\boldsymbol{\beta} - \widetilde{\mathbf{T}}\|_F^2$ is smooth with gradient $\nabla f(\boldsymbol{\beta}) = 2\,\widetilde{\mathbf{H}}^\top(\widetilde{\mathbf{H}}\boldsymbol{\beta} - \widetilde{\mathbf{T}})$ and Lipschitz constant $L_f = 2\,\lambda_{\max}(\widetilde{\mathbf{H}}^\top\widetilde{\mathbf{H}})$, and $h(\boldsymbol{\beta}) = \lambda_{\text{reg}}\|\boldsymbol{\beta}\|_1$ is convex but nonsmooth. The upper-level objective is $\varphi(\boldsymbol{\beta}) = (1/2)\|\boldsymbol{\beta}\|_F^2$, so that $\nabla\varphi(\boldsymbol{\beta}) = \boldsymbol{\beta}$. Consequently, the viscosity step in Algorithm 1 simplifies to the following:

$$w_n = (1 - \alpha_n)x_n + \alpha_n(x_n - s\,\nabla\varphi(x_n)) = (1 - \alpha_n s)\,x_n,$$

where the viscosity coefficient α_n gradually vanishes, thus causing the viscosity step to decay and anchor the iterates toward the minimum-norm solution of (4.2). The proximal operator for the L_1-norm reduces to elementwise soft-thresholding as follows:

$$\text{prox}_{\tau h}(x)_i = \text{sgn}(x_i) \cdot \max(|x_i| - \lambda_{\text{reg}}\tau,\ 0).$$

We initialize $z_{-1} = z_0 = x_0 = \mathbf{0}$ and set $c = 1/L_f$ and $s = 0.5$. The adaptive control sequences are specified in Table 3 of Section 5.1.4. The algorithm terminates when the relative change in the objective value falls below $\varepsilon = 10^{-8}$ or when $K_{\max}$ iterations are reached.

4.4. Complete processing pipeline

The complete data flow is illustrated in Figure 1. Given an input image $\mathbf{x}_0$, the pipeline applies CLAHE preprocessing, passes the enhanced image through the frozen EfficientNetV2-B0 backbone to obtain a feature vector $\mathbf{f} \in \mathbb{R}^{1280}$, applies z-score normalization, computes the ELM hidden layer activation $\mathbf{h} = \phi(\mathbf{W}\tilde{\mathbf{f}} + \mathbf{b}) \in \mathbb{R}^L$, and produces classification scores $\hat{\mathbf{y}} = \boldsymbol{\beta}^{*\top}\mathbf{h} \in \mathbb{R}^C$, where $\boldsymbol{\beta}^*$ is obtained by solving (4.3) via DIPGM-VA. Class probabilities are computed via softmax normalization.

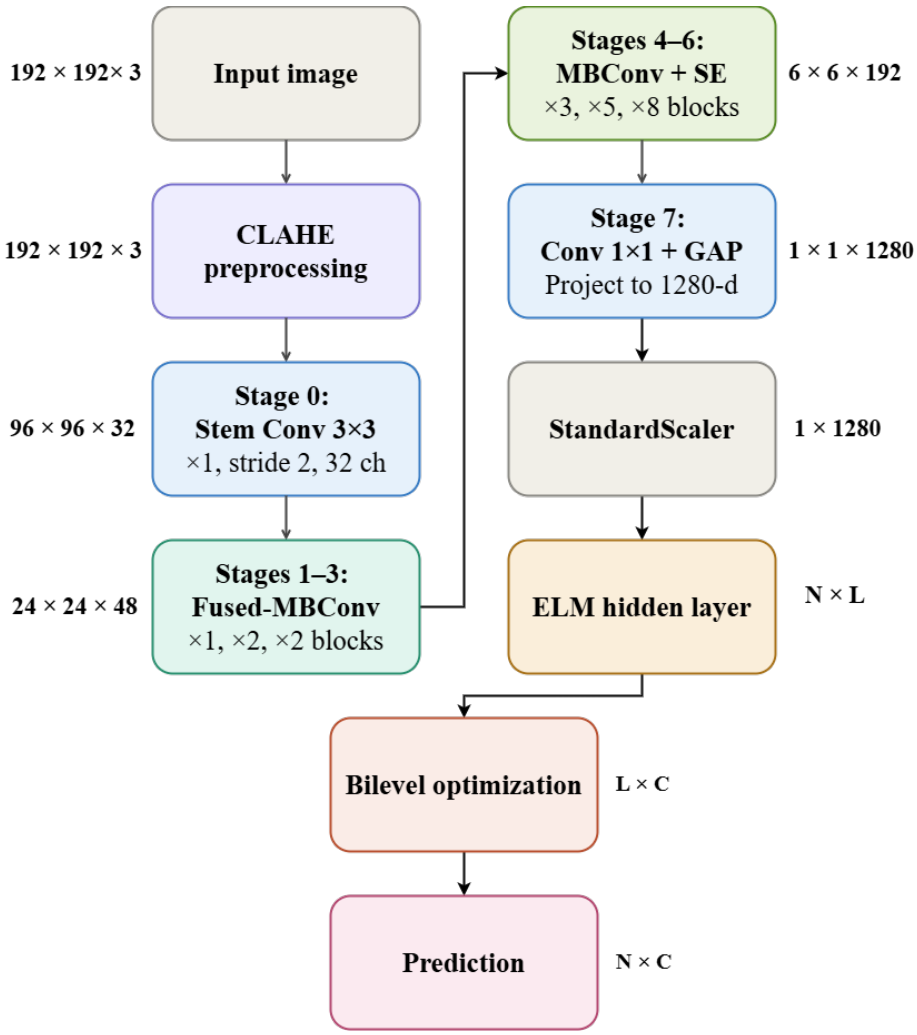


Figure 1. Architecture of the proposed hybrid model. Features extracted by the frozen EfficientNetV2-B0 backbone are passed to the ELM classifier, whose output weights are determined by solving the bilevel optimization problem via DIPGM-VA.

4.5. Model specifications

Table 1 summarizes the complete model architecture, including input/output dimensions and parameter counts for each component.

Table 1. Summary of model components with input/output dimensions and parameter counts.

Component	Dimensions	Parameters
Input image	$192 \times 192 \times 3$	–
CLAHE preprocessing	$192 \times 192 \times 3$	0 (no learnable parameters)
EfficientNetV2-B0 backbone	$\mathbb{R}^{1280}$ (output)	~ 5.9M (frozen)
StandardScaler	$\mathbb{R}^{1280}$	0 (fitted statistics)
ELM hidden layer (L neurons)	$\mathbb{R}^{L}$	$1280L + L$ (random, fixed)
ELM output layer (C classes)	$\mathbb{R}^{C}$	$L \times C$ (learned via DIPGM-VA)

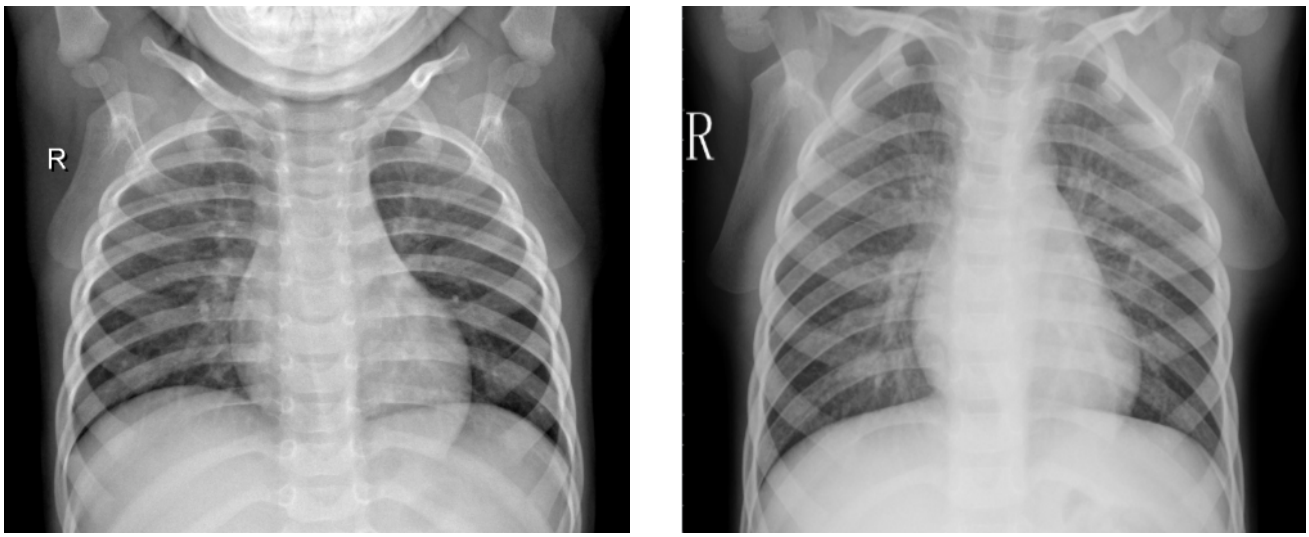
5. Numerical experiments

This section validates the proposed DIPGM-VA algorithm through numerical experiments on a medical image classification task. First, we compare the proposed hybrid model against standard end-to-end training (Section 5.2), then evaluate the convergence behavior of DIPGM-VA against existing bilevel solvers (Section 5.3).

5.1. Experimental setup

5.1.1. Dataset

We evaluate the proposed method on the chest X-ray pneumonia dataset introduced by Kermany et al. [27], one of the most widely used benchmarks for binary medical image classification. The dataset consists of 5856 anterior-posterior chest radiographs from pediatric patients aged 1–5 years, collected at Guangzhou Women and Children's Medical Center. Each image is labeled as either Normal or Pneumonia by expert radiologists. Representative samples are shown in Figure 2.



(a) (b)

Figure 2. Representative chest X-ray images from the Kermany dataset. (a) Normal lung with clear bilateral fields. (b) Pneumonia case showing pulmonary infiltrates.

The original dataset provides a predefined train/test split curated by the clinical team. We strictly preserve this original test set of 624 images without any modification. We intentionally avoid the common practice of randomly re-splitting the entire combined dataset (e.g., standard 80/20 splits), as doing so may compromise patient-level separation. Under such random splits, multiple images from the same pediatric patient could appear in both the training and testing phases, thus potentially leading to overestimated performance metrics [27]. Since the original validation set (16 images) is insufficient for reliable model selection, we construct a new validation set by stratified random sampling of the original training set (85:15 ratio). Table 2 summarizes the final partition.

Table 2. Dataset partition with class distribution. The test set follows the original clinical split by Kermany et al. [27]. The validation set is stratified-sampled from the original training set.

Split	Normal	Pneumonia	Total	Source
Train	1139	3294	4433	85% of original train
Validation	202	581	783	15% of original train
Test	234	390	624	Original test (locked)
Total	1575	4265	5840	

The dataset exhibits a notable class imbalance, with approximately 74.3% of training images belonging to the PNEUMONIA class. This imbalance is addressed through inverse-frequency class weighting in both the backbone training loss and the ELM formulation, as described in Section 4.3.

5.1.2. Implementation details

All experiments were conducted on a single NVIDIA Tesla P100 Graphics Processing Unit (GPU) (16 GB) using PyTorch 2.9.0 with Compute Unified Device Architecture (CUDA) 12.6. The backbone fine-tuning followed the two-stage strategy described in Section 4.2, thereby achieving best validation accuracies of 0.7318 (Stage 1, epoch 7) and 0.8519 (Stage 2, epoch 9). The complete backbone training required approximately 1816 s. After fine-tuning, the backbone was frozen and used to extract 1280-dimensional feature vectors for all splits.

5.1.3. Hyperparameter configuration

The ELM hyperparameters were selected via grid search using the DIPGM-VA solver, with the validation F1-score as the selection criterion. The search grid was defined as $L \in \{512,\ 1024,\ 2048\}$, $\lambda_{\text{reg}} \in \{10^{-5},\ 10^{-4},\ 10^{-3},\ 10^{-2}\}$, and ϕ = ReLU, yielding 12 configurations in total. The best configuration is $L = 2048$, $\lambda_{\text{reg}} = 10^{-5}$, thereby achieving a validation F1-score of 0.9756. Notably, all four configurations with $L = 2048$ attained the same validation performance (validation accuracy = 0.9642, validation F1-score = 0.9756) regardless of λ_{reg}, thus suggesting robustness to the regularization strength at this hidden layer size. The runner-up was $L = 1024$, $\lambda_{\text{reg}} = 10^{-2}$ (validation F1-score = 0.9694). To ensure a fair comparison among solvers, the best hyperparameters identified by DIPGM-VA were used for all bilevel algorithms.

The classification threshold for each ELM model was tuned on the validation set by maximizing Youden's J statistic [56], which selects the threshold that achieves the optimal balance between

correctly classifying positive and negative samples. The standard classification head uses the default threshold of 0.5.

5.1.4. Algorithm parameter settings

All algorithms solve the same composite bilevel problem with smooth part $f(\boldsymbol{\beta}) = \|\widetilde{\mathbf{H}}\boldsymbol{\beta} - \widetilde{\mathbf{T}}\|_F^2$, nonsmooth part $h(\boldsymbol{\beta}) = \lambda_{\text{reg}}\|\boldsymbol{\beta}\|_1$, and upper-level objective $\varphi(\boldsymbol{\beta}) = (1/2)\|\boldsymbol{\beta}\|_F^2$. The gradient $\nabla f(\boldsymbol{\beta}) = 2\,\widetilde{\mathbf{H}}^\top(\widetilde{\mathbf{H}}\boldsymbol{\beta} - \widetilde{\mathbf{T}})$ is Lipschitz continuous with constant $L_f = 2\,\lambda_{\max}\left(\widetilde{\mathbf{H}}^\top\widetilde{\mathbf{H}}\right)$, where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue. The control sequences and parameters for each algorithm are specified in Table 3. For each method, the parameters were chosen following the conditions and recommendations in the corresponding original works. All solvers were initialized at $\mathbf{0}$ and run for a fixed budget of $K = 500$ iterations with an early stopping criterion ($|J^{(k)} - J^{(k-10)}| < 10^{-8}$, evaluated every ten iterations); however, none of the solvers triggered it within 500 iterations.

Table 3. Algorithm parameters and control settings.

Method	Setting
DIPGM-VA	$s = 0.5,\ \alpha_n = 1/(n+1),\ \gamma_n = 0.9\,n/(n+1),\ c_n = 1/L_f$
	$\tau_n = 10^{14}/n^2,\ \mu_n = n/(n+1),\ \rho_n = 1/n^2$
iBiG-SAM	$\lambda = 0.01,\ c = 1/L_f,\ \alpha = 3,\ \beta_n = \gamma_n n^{-0.01}$
	$\gamma_n = 0.2/(1 - (2 + cL_f)/4)$
	$\theta_n = \min\{n/(n+\alpha-1),\ \beta_n/\|x_n - x_{n-1}\|\}$ if $x_n \neq x_{n-1}$;
	$\theta_n = n/(n+\alpha-1)$ otherwise.
aiBiG-SAM	Same as iBiG-SAM above.
	Inertial step applied only at odd iterations: $y_n = x_n$ if n is even.

5.2. Performance comparison

5.2.1. Standard classification head versus proposed hybrid model

First, we compare the conventional end-to-end approach, EfficientNetV2-B0 with a standard linear classification head trained via SGD, against the proposed hybrid model that replaces the classification head with a bilevel ELM solved by DIPGM-VA. Both models share the same fine-tuned backbone; therefore, the comparison isolates the effect of the classifier component.

The results in Table 4 show that the proposed hybrid model achieves a higher accuracy (0.8638 vs. 0.859) and F1-score (0.9001 vs. 0.882), with a recall reaching 0.9821. The standard head achieves a higher precision (0.9242) due to a more conservative decision boundary. Regarding the computational cost, the reported speedup refers to the classifier training component, as both approaches share identical backbone fine-tuning. Once features are extracted (178.83 s, one-time cost), the DIPGM-VA solver determines the output weights in under 1 s, thus yielding an approximate 10× speedup over SGD-based training.

Table 4. Test set performance of the standard classification head trained via SGD versus the proposed hybrid ELM trained via DIPGM-VA. Bold entries indicate the better value for each metric.

Metric	Standard Head	DIPGM-VA
Accuracy	0.859	**0.8638**
Precision	**0.9242**	0.8308
Recall	0.8436	**0.9821**
F1-score	0.882	**0.9001**
Threshold	0.5	0.485
Classifier training time	1815.91 s	179.5 s
Feature extraction	–	178.83 s
ELM solver (DIPGM-VA)	–	667.1 ms
Speedup	1×	**10**×

5.2.2. Comparison among bilevel solvers

Next, we compare the three bilevel optimization algorithms applied to the same ELM architecture with identical hyperparameters ($L = 2048$, $\lambda_{\text{reg}} = 10^{-5}$, $\phi = \text{ReLU}$, $K = 500$ iterations). This comparison isolates the effect of the optimization algorithm on both the classification performance and the computational efficiency.

Table 5. Test set performance of the three bilevel ELM solvers. All solvers use the same hyperparameters and iteration budget ($K = 500$). Bold entries indicate the best value. Solver time reports only the bilevel optimization cost (excluding the shared feature extraction).

Metric	DIPGM-VA	iBiG-SAM	aiBiG-SAM
Accuracy	**0.8638**	0.8446	0.8446
Precision	**0.8308**	0.8071	0.8071
Recall	0.9821	**0.9872**	**0.9872**
F1-score	**0.9001**	0.8881	0.8881
Solver time	667.1 ms	392.8 ms	**391.2 ms**

As shown in Table 5, DIPGM-VA achieves the highest classification performance across most metrics. To ensure a fair comparison, all three solvers start from the exact same initialized weight matrices (Xavier initialization with a fixed seed), thus rendering the comparison of the optimization solvers deterministic. Additionally, we observe that due to the large number of hidden neurons ($L = 2048$), the variance in F1-score across different random seeds is small (SD < 0.01, see Table 6), which indicates that the results are robust to random initialization. The deterministic convergence curves in Figure 3 show that DIPGM-VA achieves a lower inner-level objective value than its competitors, thus demonstrating its superior mathematical optimization performance regardless of the specific initialized seed.

DIPGM-VA takes slightly more wall-clock time (667 ms vs. 392 ms) because it computes two proximal gradient steps per iteration, whereas iBiG-SAM and aiBiG-SAM only compute one.

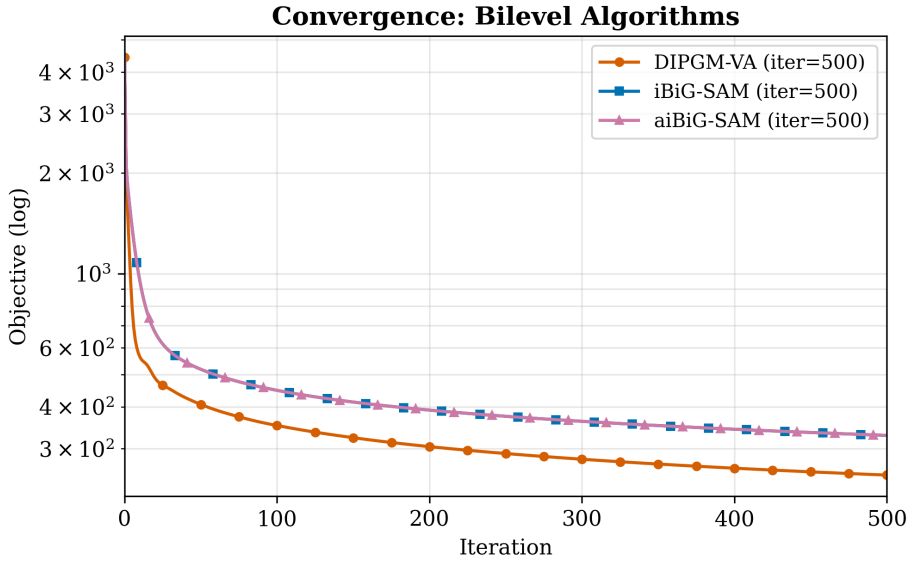


Figure 3. Convergence of the bilevel objective function for DIPGM-VA, iBiG-SAM, and aiBiG-SAM over 500 iterations (log scale), plotted from $k = 0$. DIPGM-VA reaches the minimum fastest, with smooth monotonic descent.

iBiG-SAM and aiBiG-SAM produce identical predictions, consistent with their algorithmic similarity: aiBiG-SAM only applies the inertial step at odd-numbered iterations, which does not appear to alter the solution quality at $K = 500$. It is worth noting that several studies report classification accuracies exceeding 0.95 on this specific dataset. However, as discussed in recent analyses of evaluation methodologies in medical imaging [57, 58], such results are often obtained after randomly re-partitioning the original data, which may compromise the patient-level separation strictly established by the dataset creators. By strictly adhering to the original, clinically curated test set, our reported F1-score of 0.9001 reflects out-of-sample generalization under a more rigorous evaluation protocol and is therefore not directly comparable to results obtained under alternative splitting strategies.

5.3. Convergence analysis

The convergence profiles of the three bilevel solvers in Figure 3, which shows the objective $J(\boldsymbol{\beta}^{(k)}) = \|\widetilde{\mathbf{H}}\boldsymbol{\beta}^{(k)} - \widetilde{\mathbf{T}}\|_F^2 + \lambda_{\text{reg}}\|\boldsymbol{\beta}^{(k)}\|_1$ plotted against iteration count on a logarithmic scale, reveal the following convergence behavior: DIPGM-VA exhibits smooth, monotonically decreasing convergence, thus reducing the objective from 4.4×10^3 at $k = 0$ to 2.5×10^2. This smooth descent results from the viscosity approximation stabilizing the iterates and the double-inertial extrapolation. In contrast, iBiG-SAM and aiBiG-SAM descend toward the same minimum more slowly, thereby remaining at a higher objective value at $K = 500$.

The inner-level objective value of DIPGM-VA is lower than that of both competitors, thus demonstrating a superior optimization capability within the same iteration budget. This behavior is consistent with the strong convergence established in Theorem 3.3: The double-inertial extrapolation provides acceleration while the viscosity term ensures convergence to the minimum-norm bilevel solution.

5.4. Robustness and sensitivity analysis

5.4.1. Robustness to random initialization

The ELM hidden-layer weights and biases are randomly assigned, so the classifier output may vary across runs. To assess the robustness, we repeated the experiment over ten random seeds and report the mean and standard deviation of the test accuracy, F1-score, and inner-level objective value in Table 6. The standard deviations are small, with the F1-score varying by less than 0.01, which indicates that the random initialization has a limited effect on the classifier output. This behavior agrees with the extreme learning machine theory of Huang et al. [28, 51], where a sufficiently large hidden layer renders the model insensitive to the random input weights. A paired Wilcoxon signed-rank test confirms that the reduction in the inner-level objective value achieved by DIPGM-VA over iBiG-SAM and aiBiG-SAM is statistically significant, with $p < 0.01$.

Table 6. Test set performance over ten random seeds. Values are mean ± standard deviation. The objective value is the inner-level objective value at $K = 500$ iterations.

Solver	Accuracy	F1-score	Objective Value
DIPGM-VA	0.8617 ± 0.0120	0.8992 ± 0.0079	243.6 ± 3.7
iBiG-SAM	0.8606 ± 0.0159	0.8985 ± 0.0103	322.0 ± 4.9
aiBiG-SAM	0.8606 ± 0.0159	0.8985 ± 0.0103	322.0 ± 4.9

5.4.2. Sensitivity to the inertial parameters

The double-inertial step depends on the two control sequences μ_n and ρ_n, which bound the forward coefficient θ_n and the damping coefficient δ_n, respectively. To study their influence, we fixed all other settings and varied μ_n and ρ_n over a grid of constant and decaying schedules. Table 7 reports the inner-level objective value and Table 8 reports the test F1-score after $K = 500$ iterations, averaged over the same ten random seeds used in Table 6.

The objective value steadily improves as μ_n increases toward 1 and as ρ_n decreases toward 0. The lowest objective is attained at $\mu_n = n/(n + 1)$ and $\rho_n = 1/n^2$, which is the configuration adopted in this paper, whereas a small constant μ_n or a large persistent ρ_n raises the objective by more than 30%. The F1-score stays close to 0.90 over the entire grid, which shows that the classification accuracy is stable with respect to the inertial parameters. These results justify the parameter schedules used in Section 4.3 and indicate that θ_n governs the convergence speed while δ_n acts as a damping correction that does not impair the classification performance.

Table 7. Sensitivity of the inner-level objective value to the inertial parameter schedules μ_n and ρ_n after $K = 500$ iterations, averaged over the ten random seeds of Table 6. Lower is better. The boldface entry is the configuration adopted in this paper.

$\rho_n \backslash \mu_n$	0.1	0.3	0.5	0.9	$n/(n + 1)$
0.1	298.4	289.6	279.8	256.7	250.3
0.3	306.4	298.4	289.6	268.9	263.1
0.5	313.6	306.4	298.4	279.9	274.8
0.9	326.5	320.3	313.7	298.5	294.4
$1/n^2$	294.1	284.9	274.5	250.1	**243.6**

Table 8. Sensitivity of the test F1-score to the inertial parameter schedules μ_n and ρ_n after $K = 500$ iterations, averaged over the ten random seeds of Table 6. Higher is better. The boldface entry is the configuration adopted in this paper.

$\rho_n \backslash \mu_n$	0.1	0.3	0.5	0.9	$n/(n+1)$
0.1	0.8997	0.8983	0.8999	0.8982	0.8991
0.3	0.8998	0.8997	0.8983	0.8975	0.8974
0.5	0.8993	0.8998	0.8997	0.8999	0.8981
0.9	0.8988	0.8984	0.8993	0.8997	0.8985
$1/n^2$	0.8985	0.8983	0.8981	0.8984	**0.8992**

5.5. Comparison with a standard single-level ELM

To assess whether the bilevel minimum-norm selection offers a genuine advantage over a standard single-level ELM, we compare the two on the same EfficientNetV2-B0 features, hidden layer, and class weighting. The standard single-level ELM determines its output weights by the Moore–Penrose pseudoinverse $\boldsymbol{\beta} = \widetilde{\mathbf{H}}^{\dagger}\widetilde{\mathbf{T}}$, whereas the bilevel ELM is solved by DIPGM-VA. Table 9 reports the comparison.

Table 9. Standard single-level ELM versus the bilevel ELM on the same features, with the random seed fixed at the value used in the main experiments. The minimum-norm bilevel selection yields a smaller output-weight norm and a higher F1-score.

Method	**Accuracy**	**F1-score**	$\|\|\boldsymbol{\beta}\|\|$
Standard single-level ELM (Moore–Penrose)	0.8397	0.8795	1.02
Bilevel ELM (DIPGM-VA)	**0.8638**	**0.9001**	**0.35**

The bilevel ELM lowers the output-weight norm by roughly a factor of three and raises the F1-score from 0.8795 to 0.9001. This agrees with the extreme learning machine theory, in which the output weights of a smaller norm are associated with larger classification margins and improved generalization [28, 43]; an L_2-regularized (ridge) single-level ELM did not improve over the pseudoinverse solution. The comparison indicates that the minimum-norm selection, and not the choice of solver alone, accounts for part of the observed performance gain and thus justifies the bilevel formulation.

6. Conclusions

We introduced DIPGM-VA to solve convex bilevel optimization problems and established its strong convergence to the unique bilevel solution under standard assumptions (Theorem 3.3). Numerical experiments on a chest X-ray classification benchmark demonstrated that DIPGM-VA attains a final objective value lower than iBiG-SAM and aiBiG-SAM within the same iteration budget, with a smooth monotonic descent. The proposed hybrid architecture that combined EfficientNetV2-B0 and Bilevel ELM achieved a competitive classification performance (F1-score 0.9001) with approximately 10 times speedup in the classifier training time over standard end-to-end training.

Several directions for future research merit investigation. These include the following: Evaluating

the proposed framework on multiclass and larger-scale datasets; developing adaptive strategies to select the regularization and viscosity parameters; and extending the bilevel ELM formulation to alternative backbone architectures or other tasks such as regression and semisupervised learning.

Appendix: Stagewise configuration of EfficientNetV2-B0

The feature extraction process is defined as a composition of transformations through eight sequential stages. For an input image $\mathbf{x}_0 \in \mathbb{R}^{192\times192\times3}$, the pipeline is as follows:

$$\mathbf{f} = f_7 \circ f_6 \circ f_5 \circ f_4 \circ f_3 \circ f_2 \circ f_1 \circ f_0(\mathbf{x}_0),$$

where f_i denotes the transformation at stage i. The complete stagewise configuration is summarized in Table 10.

Table 10. Stagewise configuration of the EfficientNetV2-B0 feature extractor.

Stage	Operator	Repeat	Stride	Channels	Output Size
0	Conv 3×3	1	2	32	96×96
1	Conv 3×3	1	1	16	96×96
2	Fused-MBConv4, $k=3\times3$	2	2	32	48×48
3	Fused-MBConv4, $k=3\times3$	2	2	48	24×24
4	MBConv4 + SE, $k=3\times3$	3	2	96	12×12
5	MBConv6 + SE, $k=3\times3$	5	1	112	12×12
6	MBConv6 + SE, $k=3\times3$	8	2	192	6×6
7	Conv 1×1 + GAP	1	–	1280	1×1

Stage 0: Stem convolution. The network begins with a stem convolution layer that processes the input image $\mathbf{x}_0 \in \mathbb{R}^{192\times192\times3}$. A 3×3 convolution with stride 2 halves the spatial dimensions while expanding the channel depth from 3 (RGB) to 32 feature channels:

$$\mathbf{x}_1 = \varsigma[\mathrm{BN}(\mathrm{Conv}_{3\times3}(\mathbf{x}_0;\mathbf{W}_0)+\mathbf{b}_0)] \in \mathbb{R}^{96\times96\times32},$$

where $\mathbf{W}_0 \in \mathbb{R}^{3\times3\times3\times32}$ denotes the convolutional kernel, $\mathbf{b}_0 \in \mathbb{R}^{32}$ is the bias, ς is the Sigmoid Linear Unit (SiLU) activation [59, 60] defined as $\varsigma(x) = x\cdot(1+e^{-x})^{-1}$, and $\mathrm{BN}(\cdot)$ denotes batch normalization [61]:

$$\mathrm{BN}(x) = \gamma_{\mathrm{bn}}\left(\frac{x-\mu_{\mathcal{B}}}{\sqrt{\hat{\sigma}^2_{\mathcal{B}}+\epsilon}}\right)+\beta_{\mathrm{bn}},$$

with mini-batch statistics $\mu_{\mathcal{B}}, \hat{\sigma}^2_{\mathcal{B}}$ and learnable affine parameters $\gamma_{\mathrm{bn}}, \beta_{\mathrm{bn}}$.

Purpose of Stage 0. This stage rapidly downsamples the spatial resolution, thereby reducing the computational burden for subsequent layers while capturing fundamental low-level visual features such as edges and textures.

Stage 1: Pointwise reduction convolution. This stage applies a standard 3×3 convolution (Conv 3×3) with stride 1 that reduces the channel depth from 32 to 16:

$$\mathbf{x}_2 = \varsigma[\mathrm{BN}(\mathrm{Conv}_{3\times3}(\mathbf{x}_1; \mathbf{W}_1) + \mathbf{b}_1)] \in \mathbb{R}^{96\times96\times16},$$

where $\mathbf{W}_1 \in \mathbb{R}^{3\times3\times32\times16}$. No residual skip connection is applied because the input and output channel dimensions differ ($32 \neq 16$).

Purpose of Stage 1. This stage compresses the initial 32-channel representation into a more compact 16-channel form, thereby reducing the subsequent memory and compute costs while retaining the spatial resolution.

Stages 2 and 3: Fused-MBConv blocks. These stages employ Fused-MBConv blocks [26] that combine expansion and spatial convolution into a single fused operation. For an input tensor $\mathbf{X} \in \mathbb{R}^{H\times W\times C_{\text{in}}}$ with expansion ratio e, the fused convolution produces the following:

$$\mathbf{Z} = \mathrm{Conv}_{k\times k}(\mathbf{X}; \mathbf{W}_{\text{fuse}}) + \mathbf{b}_{\text{fuse}},\ \mathbf{W}_{\text{fuse}} \in \mathbb{R}^{k\times k\times C_{\text{in}}\times(e\cdot C_{\text{in}})},$$

followed by batch normalization, SiLU activation, and a 1×1 projection convolution:

$$\mathbf{Y} = \mathrm{BN}\Big\{\mathrm{Conv}_{1\times1}\Big[\varsigma(\mathrm{BN}(\mathbf{Z}))\,; \mathbf{W}_{\text{proj}}\Big]\Big\} \in \mathbb{R}^{H'\times W'\times C_{\text{out}}}.$$

A residual skip connection $\mathbf{Y} \leftarrow \mathbf{Y} + \mathbf{X}$ is applied when the stride equals 1 and $C_{\text{in}} = C_{\text{out}}$, thereby facilitating gradient flow through the network [40].

Purpose of Stages 2 and 3. In early stages where spatial resolution is high but the channel depth is low, memory access costs dominate computation. By merging the expansion and depthwise operations into a single standard convolution, Fused-MBConv maximizes the hardware accelerator utilization and improves the training speed without sacrificing the representational capacity [26].

Stages 4–6: MBConv with squeeze-and-excitation. These stages employ Mobile Inverted Bottleneck Convolution (MBConv) blocks [62] with integrated Squeeze-and-Excitation (SE) modules [63]. The MBConv structure follows an inverted bottleneck design consisting of three sequential operations:

Expansion. A 1×1 pointwise convolution expands the input channels by the expansion ratio e as follows:

$$\mathbf{Z}_e = \varsigma\Big[\mathrm{BN}\Big(\mathrm{Conv}_{1\times1}(\mathbf{X}; \mathbf{W}_{\text{exp}})\Big)\Big].$$

Depthwise convolution. A $k\times k$ depthwise separable convolution independently processes each channel as follows:

$$\mathbf{Z}_{\text{dw}} = \varsigma[\mathrm{BN}(\mathrm{DWConv}_{k\times k}(\mathbf{Z}_e; \mathbf{W}_{\text{dw}}))]\,.$$

Squeeze-and-excitation. The SE module [63] adaptively recalibrates the channelwise feature responses. The squeeze operation aggregates the spatial information via global average pooling to produce a channel descriptor $\mathbf{z} \in \mathbb{R}^C$ as follows:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{X}_c(i, j).$$

The excitation operation learns channelwise dependencies through a bottleneck with reduction ratio r as follows:

$$\mathbf{s} = \sigma(\mathbf{W}_2\,\mathrm{ReLU}(\mathbf{W}_1\,\mathbf{z})) \in [0,1]^C,$$

where $\mathbf{W}_1 \in \mathbb{R}^{(C/r)\times C}$, $\mathbf{W}_2 \in \mathbb{R}^{C\times(C/r)}$, and σ denotes the sigmoid function. The recalibrated output is $\mathbf{X}' = \mathbf{s} \odot \mathbf{X}$, where $\odot$ is elementwise multiplication.

Purpose of Stages 4–6. As the network deepens, the channel dimensions increase while the spatial dimensions shrink. The MBConv structure mitigates computational cost via depthwise separable convolutions, whereas the SE module acts as a channelwise attention mechanism that adaptively emphasizes informative features and suppresses less relevant ones [63].

Stage 7: Final feature projection. The final stage consists of a 1×1 convolution projecting 192 channels to 1280 dimensions, followed by Global Average Pooling (GAP):

$$F_c = \frac{1}{H_7 W_7} \sum_{m=1}^{H_7} \sum_{n=1}^{W_7} \mathbf{X}_7(m,n,c),\ c = 1,\ldots,1280,$$

thus producing a fixed 1280-dimensional feature vector $\mathbf{f} \in \mathbb{R}^{1280}$ regardless of the input spatial dimensions.

Purpose of Stage 7. GAP compresses spatial feature maps into a permutation-invariant global representation, thus drastically reducing the parameter count by avoiding dense linear layers and thereby mitigating the risk of overfitting [64].

Author contributions

Suthep Suantai: Conceptualization, methodology, validation, formal analysis, writing-review & editing, supervision; Kobkoon Janngam: Conceptualization, methodology, software, validation, formal analysis, investigation, writing-original draft, writing-review & editing, visualization. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare that they have not used any Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the CMU Proactive Researcher Program, Chiang Mai University, under Contract No. 568/2567. S. Suantai was partially supported by Chiang Mai University.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, et al., CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning, *arXiv Preprint*, 2017. https://doi.org/10.48550/arXiv.1711.05225
2. A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, et al., Dermatologist-level classification of skin cancer with deep neural networks, *Nature*, **542** (2017), 115–118. https://doi.org/10.1038/nature21056
3. L. Franceschi, P. Frasconi, S. Salzo, R. Grazzi, M. Pontil, Bilevel programming for hyperparameter optimization and meta-learning, *arXiv Preprint*, 2018. https://doi.org/10.48550/arXiv.1806.04910
4. A. Shaban, C. A. Cheng, N. Hatch, B. Boots, *Truncated back-propagation for bilevel optimization*, In: Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan, **89** (2019), 1723–1732. Available from: `https://proceedings.mlr.press/v89/shaban19a.html`.
5. K. Janngam, S. Suantai, Y. J. Cho, A. Kaewkhao, R. Wattanataweekul, A novel inertial viscosity algorithm for bilevel optimization problems applied to classification problems, *Mathematics*, **11** (2023), 3241. https://doi.org/10.3390/math11143241
6. S. Sabach, S. Shtern, A first order method for solving convex bilevel optimization problems, *SIAM J. Optimiz.*, **27** (2017), 640–660. https://doi.org/10.1137/16M105592X
7. J. J. Moreau, Fonctions convexes duales et points proximaux dans un espace hilbertien, *C. R. Acad. Sci. Paris*, **255** (1962), 2897–2899.
8. P. L. Combettes, V. R. Wajs, Signal recovery by proximal forward-backward splitting, *Multiscale Model. Sim.*, **4** (2005), 1168–1200. https://doi.org/10.1137/050626090
9. A. Moudafi, Viscosity approximation methods for fixed-points problems, *J. Math. Anal. Appl.*, **241** (2000), 46–55. https://doi.org/10.1006/jmaa.1999.6615
10. B. T. Polyak, Some methods of speeding up the convergence of iteration methods, *USSR Comp. Math. Math. Phys.*, **4** (1964), 1–17. https://doi.org/10.1016/0041-5553(64)90137-5
11. Y. Nesterov, A method for solving the convex programming problem with convergence rate $O(1/k^2)$, *Dokl. Akad. Nauk Sssr*, **269** (1983), 543–547.
12. A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sci.*, **2** (2009), 183–202. https://doi.org/10.1137/080716542
13. L. Bussaban, A. Kaewkhao, S. Suantai, Inertial S-iteration forward-backward algorithm for a family of nonexpansive operators with applications to image restoration problems, *Filomat*, **35** (2021), 771–782. https://doi.org/10.2298/FIL2103771B
14. P. Yatakoat, S. Suantai, A. Hanjing, On some accelerated optimization algorithms based on fixed point and linesearch techniques for convex minimization problems with applications, *Adv. Contin. Discret. M.*, **2022** (2022), 25. https://doi.org/10.1186/s13662-022-03698-5
15. P. Sae-jia, S. Suantai, A new two-step inertial algorithm for solving convex bilevel optimization problems with application in data classification problems, *AIMS Math.*, **9** (2024), 8476–8496. https://doi.org/10.3934/math.2024412

16. R. Wattanataweekul, K. Janngam, S. Suantai, A novel two-step inertial viscosity algorithm for bilevel optimization problems applied to image recovery, *Mathematics*, **11** (2023), 3518. https://doi.org/10.3390/math11163518

17. K. Janngam, S. Suantai, R. Wattanataweekul, A novel fixed-point based two-step inertial algorithm for convex minimization in deep learning data classification, *AIMS Math.*, **10** (2025), 6209–6232. https://doi.org/10.3934/math.2025283

18. Z. Y. Peng, D. Li, Y. Zhao, R. L. Liang, An accelerated subgradient extragradient algorithm for solving bilevel variational inequality problems involving non-Lipschitz operator, *Commun. Nonlinear Sci.*, **127** (2023), 107549. https://doi.org/10.1016/j.cnsns.2023.107549

19. Y. Shehu, P. T. Vuong, A. Zemkoho, An inertial extrapolation method for convex simple bilevel optimization, *Optim. Method. Softw.*, **36** (2021), 1–19. https://doi.org/10.1080/10556788.2019.1619729

20. P. Duan, Y. Zhang, Alternated and multi-step inertial approximation methods for solving convex bilevel optimization problems, *Optimization*, **72** (2023), 2517–2545. https://doi.org/10.1080/02331934.2022.2069022

21. C. Izuchukwu, M. Aphane, K. O. Aremu, Two-step inertial forward–reflected–anchored–backward splitting algorithm for solving monotone inclusion problems, *Comput. Appl. Math.*, **42** (2023), 351. https://doi.org/10.1007/s40314-023-02485-6

22. O. S. Iyiola, Y. Shehu, Convergence results of two-step inertial proximal point algorithm, *Appl. Numer. Math.*, **182** (2022), 57–75. https://doi.org/10.1016/j.apnum.2022.07.013

23. D. V. Thong, S. Reich, X. H. Li, P. T. H. Tham, An efficient algorithm with double inertial steps for solving split common fixed point problems and an application to signal processing, *Comput. Appl. Math.*, **44** (2025), 102. https://doi.org/10.1007/s40314-024-03058-x

24. R. Wattanataweekul, K. Janngam, S. Suantai, A novel double inertial viscosity algorithm for convex bilevel optimization problems applied to image restoration problems, *Optimization*, **74** (2025), 2635–2656. https://doi.org/10.1080/02331934.2024.2398776

25. S. Suantai, K. Janngam, A novel fixed-point based two-step inertial algorithm for convex bilevel optimization in deep learning data classification, *Carpathian J. Math.*, **42** (2026), 369–392. https://doi.org/10.37193/CJM.2026.02.10

26. M. Tan, Q. V. Le, EfficientNetV2: Smaller models and faster training, *arXiv Preprint*, 2021. https://doi.org/10.48550/arXiv.2104.00298

27. D. S. Kermany, M. Goldbaum, W. Cai, C. C. S. Valentim, H. Liang, S. L. Baxter, et al., Identifying medical diagnoses and treatable diseases by image-based deep learning, *Cell*, **172** (2018), 1122–1131. https://doi.org/10.1016/j.cell.2018.02.010

28. G. B. Huang, Q. Y. Zhu, C. K. Siew, Extreme learning machine: Theory and applications, *Neurocomputing*, **70** (2006), 489–501. https://doi.org/10.1016/j.neucom.2005.12.126

29. H. H. Bauschke, P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, 2 Eds., Springer Cham, 2017. https://doi.org/10.1007/978-3-319-48311-5

30. K. Goebel, S. Reich, *Uniform convexity, hyperbolic geometry, and nonexpansive mappings*, New York and Basel: Marcel Dekker, 1984.

31. J. B. Baillon, R. E. Bruck, S. Reich, On the asymptotic behavior of nonexpansive mappings and semigroups in Banach spaces, *Houston J. Math.*, **4** (1978), 1–9.
32. W. Takahashi, *Introduction to nonlinear and Cconvex analysis*, Yokohama: Yokohama Publ., 2009.
33. K. Nakajo, K. Shimoji, W. Takahashi, Strong convergence theorems by the hybrid method for families of nonexpansive mappings in Hilbert spaces, *Taiwan. J. Math.*, **10** (2006), 339–360. https://doi.org/10.11650/twjm/1500403829
34. A. Hanjing, P. Thongpaen, S. Suantai, A new accelerated algorithm with a linesearch technique for convex bilevel optimization problems with applications, *AIMS Math.*, **9** (2024), 22366–22392. https://doi.org/10.3934/math.20241088
35. K. Aoyama, W. Takahashi, Strong convergence theorems for a class of nonexpansive mappings in Banach spaces, *J. Nonlinear Convex A.*, **12** (2011), 121–134.
36. K. Aoyama, F. Kohsaka, W. Takahashi, Strongly relatively nonexpansive sequences in Banach spaces and applications, *J. Fix. Point Theory A.*, **5** (2009), 201–225. https://doi.org/10.1007/s11784-009-0108-7
37. S. Saejung, P. Yotkaew, Approximation of zeros of inverse strongly monotone operators in Banach spaces, *Nonlinear Anal.-Theor.*, **75** (2012), 742–750. https://doi.org/10.1016/j.na.2011.09.005
38. L. Bussaban, A. Kaewkhao, S. Suantai, A parallel inertial S-iteration forward-backward algorithm for regression and classification problems, *Carpathian J. Math.*, **36** (2020), 35–44. https://doi.org/10.37193/cjm.2020.01.04
39. Y. L. Cun, Y. Bengio, G. Hinton, Deep learning, *Nature*, **521** (2015), 436–444. https://doi.org/10.1038/nature14539
40. K. He, X. Zhang, S. Ren, J. Sun, *Deep residual learning for image recognition*, In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, 770–778. https://doi.org/10.1109/CVPR.2016.90
41. S. J. Pan, Q. Yang, A survey on transfer learning, *IEEE T. Knowl. Data En.*, **22** (2010), 1345–1359. https://doi.org/10.1109/TKDE.2009.191
42. J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst.*, **27** (2014).
43. G. B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, *IEEE T. Syst. Man Cy. B*, **42** (2012), 513–529. https://doi.org/10.1109/TSMCB.2011.2168604
44. W. Deng, Q. Zheng, L. Chen, *Regularized extreme learning machine*, In: 2009 IEEE Symposium on Computational Intelligence and Data Mining, Nashville, TN, USA, 2009, 389–395. https://doi.org/10.1109/CIDM.2009.4938676
45. R. Tibshirani, Regression shrinkage and selection via the lasso, *J. R. Stat. Soc. B*, **58** (1996), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x
46. M. Tan, Q. V. Le, *EfficientNet: Rethinking model scaling for convolutional neural networks*, In: Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, **97** (2019), 6105–6114. Available from: `https://proceedings.mlr.press/v97/tan19a.html`.

47. S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, et al., Adaptive histogram equalization and its variations, *Comput. Vis. Graph. Image Process.*, **39** (1987), 355–368. https://doi.org/10.1016/S0734-189X(87)80186-X

48. A. M. Reza, Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement, *J. Vlsi Sig. Proc. Syst.*, **38** (2004), 35–44. https://doi.org/10.1023/B:VLSI.0000028532.53893.82

49. J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, et al., Overcoming catastrophic forgetting in neural networks, *P. Natl. A. Sci. USA*, **114** (2017), 3521–3526. https://doi.org/10.1073/pnas.1611835114

50. I. Loshchilov, F. Hutter, *Decoupled weight decay regularization*, In: ICLR 2019 International Conference on Learning Representations, 2019. Available from: `https://openreview.net/forum?id=Bkg6RiCqY7`.

51. G. B. Huang, L. Chen, C. K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, *IEEE T. Neural Networ.*, **17** (2006), 879–892. https://doi.org/10.1109/TNN.2006.875977

52. M. D. McDonnell, T. Vladusich, *Enhanced image classification with a fast-learning shallow convolutional neural network*, In: 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 2015, 1–7. https://doi.org/10.1109/IJCNN.2015.7280796

53. J. Cao, Z. Lin, G. B. Huang, N. Liu, Voting based extreme learning machine, *Inform. Sciences*, **185** (2012), 66–77. https://doi.org/10.1016/j.ins.2011.09.015

54. S. Pang, Z. Yu, M. A. Orgun, A novel end-to-end classifier using domain transferred deep convolutional neural networks for biomedical images, *Comput. Meth. Prog. Bio.*, **140** (2017), 283–293. https://doi.org/10.1016/j.cmpb.2016.12.019

55. X. Glorot, Y. Bengio, *Understanding the difficulty of training deep feedforward neural networks*, In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia La guna Resort, Sardinia, Italy, **9** (2010), 249–256. Available from: `https://proceedings.mlr.press/v9/glorot10a.html`.

56. W. J. Youden, Index for rating diagnostic tests, *Cancer*, **3** (1950), 32–35. https://doi.org/10.1002/1097-0142(1950)3:1%3C32::AID-CNCR2820030106%3E3.0.CO;2-3

57. P. Rouzrokh, B. Khosravi, S. Faghani, M. Moassefi, D. V. V. Martinez, B. S. Erickson, et al., Mitigating bias in radiology machine learning: 1. Data handling, *Radiol.-Artif. Intell.*, **4** (2022), e210290. https://doi.org/10.1148/ryai.210290

58. M. Roberts, D. Driggs, M. Thorpe, J. Gilbey, M. Yeung, S. Ursprung, et al., Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans, *Nat. Mach. Intell.*, **3** (2021), 199–217. https://doi.org/10.1038/s42256-021-00307-0

59. S. Elfwing, E. Uchibe, K. Doya, Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, *Neural Networks*, **107** (2018), 3–11. https://doi.org/10.1016/j.neunet.2017.12.012

60. P. Ramachandran, B. Zoph, Q. V. Le, Searching for activation functions, *arXiv Preprint*, 2017. https://doi.org/10.48550/arXiv.1710.05941

61. S. Ioffe, C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, In: Proceedings of the 32nd International Conference on Machine Learning, Lille, France, **37** (2015), 448–456. Available from: `https://proceedings.mlr.press/v37/ioffe15.html`.

62. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen, *MobileNetV2: Inverted residuals and linear bottlenecks*, In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

63. J. Hu, L. Shen, G. Sun, *Squeeze-and-excitation networks*, In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, 7132–7141. https://doi.org/10.1109/CVPR.2018.00745

64. M. Lin, Q. Chen, S. Yan, Network in network, *arXiv Preprint*, 2014. https://doi.org/10.48550/arXiv.1312.4400

AIMS Press