VER
Vascular & Endovascular Review

An Enhanced Bayesian Optimization Method for Tuning of Hybrid and High-dimensional Hyperparameters of CNNs in Brain Tumor Detection and Classification

Kalpana Devi[1]*, Aman Kumar Sharma[1]

[1]Research Scholar, Department of Computer Science, Himachal Pradesh University, Summer Hill, Shimla 171005, India
Email ID: kalpana.hpu@gmail.com
ORCID: 0009-0005-4646-8789
[2]Professor, Department of Computer Science, Himachal Pradesh University, Summer Hill, Shimla 171005, India
Email ID: sharmaas1@gmail.com
ORCID: 0009-0009-6815-2602

ABSTRACT

Background and Objective: Bayesian Optimization (BO) is a powerful strategy for optimizing complex black-box functions and is widely used for hyperparameter tuning in machine learning. Traditional BO methods commonly use Gaussian Processes (GPs), Random Forests (RF), or Bayesian Neural Networks (BNNs) as surrogate models. However, they struggle to scale in hybrid and high-dimensional search spaces and often fail to capture the spatial hierarchies required for image-based tasks such as those handled by Convolutional Neural Networks (CNNs). This study aims to overcome these limitations by proposing an Enhanced Bayesian Optimization (EBO) framework specifically designed to optimize CNN hyperparameters for brain tumor detection and classification using MRI data. Methods: A Bayesian Convolutional Neural Network (BCNN) is introduced as a novel surrogate model to address the hybrid and high-dimensional hyperparameter search spaces of CNNs. Its performance is benchmarked against GP, RF, and BNN, each paired with five acquisition functions: Expected Improvement (EI), Upper Confidence Bound (UCB), Probability Improvement (PI), Entropy Search (ES), and Knowledge Gradient (KG). Experiments on two MRI datasets - binary (tumor vs. non-tumor) and three-class (glioma, meningioma, pituitary), show BCNN consistently outperforms other surrogates. To further improve validation accuracy, the two best acquisition functions are hybridized with Bayesian CNN to form the EBO framework. Results: The Bayesian CNN surrogate outperformed GP, RF, and BNN across acquisition functions, with ES and KG showing the best mean performance. The proposed hybrid BCNN_ES+KG (EBO) achieved the highest validation accuracies of 97.0% (Dataset D1) and 92.13% (Dataset D2), surpassing single acquisition functions. Using the optimized hyperparameters, Optimized_CNN 1 reached 98.0% accuracy and Optimized_CNN 2 achieved 95.79% accuracy, both outperforming existing state-of-the-art methods. Conclusions: The proposed EBO framework, using Bayesian CNN as a surrogate model combined with a hybrid ES+KG acquisition strategy, effectively optimizes high-dimensional CNN hyperparameters. The optimized CNNs achieved superior performance, validating the effectiveness and generalizability of EBO for brain tumor detection and classification using MRI data.

KEYWORDS: Bayesian-optimization; Surrogate model; Acquisition function; Hyperparameters; CNN.

How to Cite: Kalpana Devi, Aman Kumar Sharma, (2024) An Enhanced Bayesian Optimization Method for Tuning of Hybrid and High-dimensional Hyperparameters of CNNs in Brain Tumor Detection and Classification, Vascular and Endovascular Review, Vol.7, No.2, 100-126

INTRODUCTION

Optimization plays a pivotal role across diverse areas of scientific and engineering disciplines, including operations research, control systems, artificial intelligence, and computational biology. In recent years, its significance has grown substantially in the domains of Machine Learning (ML) and Deep Learning (DL), where model performance is often governed by the selection of optimal hyperparameters and architectural configurations. Many modern ML models involve objective functions that are non-convex, non-differentiable, noisy, or expensive to evaluate, making them difficult to optimize using classical gradient-based methods. As a result, researchers have increasingly relied on metaheuristic and probabilistic optimization techniques that can efficiently search large and complex parameter spaces without relying on explicit gradient information. Within the domain of deep learning, especially with CNNs, effective optimization of architectural and training hyperparameters can significantly improve accuracy, generalization, and computational efficiency across various tasks, including image classification, object detection, and medical diagnosis [1 - 4]. Consequently, robust optimization techniques have become a cornerstone in the development and deployment of high-performing ML models.

CNNs have emerged as a powerful deep learning architecture, particularly effective in processing and analyzing image data due to their capacity to represent spatial hierarchies and patterns. In the domain of medical imaging, CNNs have been widely adopted for applications including classification, detection, and segmentation of abnormalities, including brain tumors. Brain tumor detection and classification using MRI are critical tasks, where accurate and timely diagnosis greatly enhances treatment effectiveness and patient survival. CNN-based models have shown remarkable performance in these applications through automatically learning discriminative characteristics from raw imaging data, thereby minimizing reliance on handcrafted feature engineering [5, 6]. However, the effectiveness of CNNs strongly depends on the selection of optimal hyperparameters such as

the number of convolutional blocks, filter sizes, learning rate, dropout rate, batch size, and the choice of optimizer. These hyperparameters influence both the architectural design and the training dynamics of the network, thereby playing a critical role in determining its generalization ability. The optimization of these parameters becomes significantly more challenging in high-dimensional and hybrid search spaces, which may include a mix of continuous, discrete, and categorical variables. In such scenarios, the interaction between hyperparameters is often highly interdependent and dataset-specific, making the search for optimal configurations more difficult. Manual tuning or traditional approaches like grid and random search are inefficient and computationally expensive, especially when navigating these complex search spaces. This limitation highlights the urgent need for more advanced, scalable, and data-efficient optimization strategies capable of effectively exploring hybrid and high-dimensional hyperparameter spaces [2, 3].

The interest in the hyperparameter optimization is strongly linked to the interest in developing dependable deep learning models for a wide range of applications [7-13], and as the latest studies [14-18] show, the adjustment of hyperparameters can significantly improve CNN's performance. Numerous techniques exist for optimizing over hyperparameter settings, ranging from straightforward processes like grid or random search [1] to more complex metaheuristic approaches like evolutionary algorithms or Genetic Algorithms [19-23]. However, these methods typically demand a substantial number of evaluations, which can be computationally prohibitive. In contrast, Bayesian Optimization provides an effective framework for global optimization, particularly when dealing with costly, noisy, and black-box functions. By modeling uncertainty in a principled way, it enables a natural trade-off between exploring novel regions of the search space and exploiting known promising regions. Bayesian optimization [24, 25] has been broadly applied across various areas such as chemical design [26, 27], material science [28], aerospace engineering [29], civil engineering [30], and hyperparameter optimization [31, 32]. BO is highly data-efficient, as it incorporates prior knowledge about the objective function and strategically balances exploration (searching unknown areas) and exploitation (refining known good areas). Formally, consider the maximization of an unknown, expensive-to-evaluate function:

$$\boldsymbol{x}^* = \arg\max \boldsymbol{f}(\boldsymbol{x})$$

$x \in X$ indicate the decision/search space of interest, and $x*$ is the global maximum

BO builds a probabilistic surrogate model that estimates the true objective function and utilizes this model to guide the search for optimal configurations. BO begins with a prior distribution over the objective function and updates it using Bayes' theorem as new data is collected. This generates a posterior distribution that reflects an improved understanding of the function. An acquisition function then evaluates where the next evaluation should occur by balancing exploration of uncertain regions and exploitation of high-performing areas. This enables BO to make intelligent decisions about which configurations to test next, even with a limited number of evaluations [33, 34]. In the context of tuning hyperparameters in convolutional neural networks, where the search space is often hybrid and high-dimensional, BO has shown great promise. It efficiently navigates the complex relationships between parameters such as filter sizes, dropout rate, learning rate, and optimizer type. By capturing these dependencies and reducing the computational burden, BO enables the discovery of near-optimal architectures that generalize well to unseen data [2, 3].

Despite the effectiveness of Bayesian Optimization in hyperparameter tuning, traditional BO methods face several limitations. Gaussian Processes, commonly used as surrogate models, struggle to scale with larger datasets due to their high computational cost. They also rely on assumptions that may not hold in complex search spaces involving both discrete and continuous parameters. These issues lead to poor generalization and reduced efficiency in modeling the objective function [35, 36]. Another challenge lies in the acquisition functions used to guide the search. A proper balance between exploration and exploitation is critical. Over-exploitation can cause the model to get stuck in local optima, while over-exploration can waste computational resources [33]. Moreover, standard BO lacks spatial and structural awareness, making it less effective in hybrid, hierarchical, and high-dimensional search spaces often seen in many applications [37, 38]. These gaps highlight the need for advanced optimization approaches that can scale efficiently, model complex functions accurately, and balance the exploration-exploitation trade-off more effectively.

Hyperparameter optimization in CNNs becomes particularly challenging in high-dimensional and hybrid search spaces. As Frazier [34] states that "*developing Bayesian optimization methods that work well in high dimensions is of great practical and theoretical interest*". As more complicated the models, the quantity of hyperparameters and the dimension of the search area in hyperparameter optimization of machine learning models also grow [39]. Addressing the limitations of traditional Bayesian Optimization is essential for advancing the effectiveness of convolutional neural networks in medical imaging. In brain tumor classification, where accurate and timely diagnosis can significantly influence patient outcomes, even marginal improvements in model performance are valuable. However, real-world datasets in this domain are often small, imbalanced, and high in dimensionality, which makes the hyperparameter optimization task even more challenging. Efficient and scalable optimization methods can enhance model generalization and reduce training time, enabling the deployment of more accurate and reliable diagnostic tools in clinical settings.

The purpose of this study is to address the challenges of traditional BO such as scalability, limited generalization in hybrid/high-dimensional spaces, and imbalance in exploration and exploitation when tuning CNN hyperparameters for brain tumor classification across two publicly available datasets. The main contributions are:

- Introduce BCNN as a novel surrogate model for optimizing CNN hyperparameters in complex search spaces.
- Conducted comprehensive benchmarking of BCNN against GP, RF, and BNN across multiple acquisition functions.
- Introduced the Enhanced Bayesian Optimization (EBO) framework, which integrates BCNN with a hybrid ES+KG acquisition strategy.

- Developed optimized CNN architectures using hyperparameters identified by EBO for improved model design.
- Performed a comparison of the optimized CNN architectures with existing state of the art methods.

The rest of this paper proceeds the follows sections. The Background and Related Work section provides the theoretical foundation, covering surrogate models, acquisition functions, and prior studies on BO, including its applications in CNN hyperparameter optimization for medical imaging. The Proposed Methodology describes the dataset, objective function, search space, and details of the EBO algorithm. The Results and Discussions section presents the experimental findings, including the selection of the optimal surrogate model and acquisition functions, evaluation of the proposed BCNN_ES+KG framework, construction of optimized CNN models for dataset_1 and dataset_2, and comparative analysis with existing methods. Finally, the Conclusion and Future Work section summarizes the main findings and proposes directions for further study.

BACKGROUND AND RELATED WORK

1.1. Background

The roots of Bayesian optimization trace back to the work of Harold Kushner [25], who utilized Wiener processes for unconstrained one-dimensional optimization tasks and maximizing the likelihood of improvement when choosing the subsequent sample. Mockus [40] introduced an innovative acquisition function named Expected Improvement (EI), which was subsequently applied in additional research by Zilinskas [41]. Perttunen [42], Stuckman [43], and Elder [44] extended Kushner's approach to address optimization problems in higher-dimensional spaces. In recent decades, Bayesian optimization has seen significant growth and has been effectively used to address a variety of real-world issues, such as materials design and discovery [45], sensor networks [46], the financial sector [47], and experimental design [48]. Recently, it is increasingly more popular within the field of machine learning, especially in reinforcement learning [49], neural architecture search [50], and hyperparameter tuning [51].

1.1.1. Surrogate models

In BO, a surrogate model is a probabilistic approximation of the true objective function. Instead of querying the real function directly at every iteration, BO builds and updates a surrogate model to predict outcomes, enabling the algorithm to decide where to sample next based on knowledge. Regarding the issue of optimizing a black-box (objective) function that is costly to evaluate, $f = R^d \rightarrow R$ takes an input vector $x \in R^d$ (with d dimensions) and outputs a scalar value $f(x) \in R$. Suppose the collected data with n observations is $D = \{(x_i,\ y_i)\}_{i=1}^{n}$, where each output y_i is a disruptive analysis of the actual function:

$$y_i = f(x_i) + \epsilon,\ \epsilon \sim N(0, \sigma^2)$$

Here, ϵ represents Gaussian noise with zero mean and variance σ^2, which models measurement uncertainty or randomness.

- **Gaussian Process (GP):** A GP [52] is not parametric, and commonly used Bayesian approach to regression. It establishes a prior over functions, and after data observation, it produces a posterior that represents an updated belief about the function. Assume a Gaussian Process prior over the function $f(x)$:

$$f(x) \sim GP(\mu(x), k(x, x'))$$

Which means that any limited collection of function values $[f(x_1), f(x_2), \ldots\ldots, f(x_n)]$ follows a normal multivariate distribution with a mean function, $\mu(x) = 0$ for ease of use and the covariance function, $k(x, x')$ calculates how similar two inputs are x and x'. For the given set of prior observations D, the Gaussian posterior distribution of the function value at a new location x_* is given by:

$$f(x_*) \mid D, x_* \sim N(\mu_*(x_*), {\sigma^2}_*(x_*))$$

where the posterior mean $\mu_*(x_*)$ and variance ${\sigma^2}_*(x_*)$ are analytically derived using the kernel function. The mean is determined as $\mu_*(x_*) = k(x_*, X)^T[K(X,X) + \sigma^2 I]^{-1} y$ while the variance is ${\sigma^2}_*(x_*) = k(x_*, x_*) - k(x_*, X)^T[K(X,X) + \sigma^2 I]^{-1} k(x_*, X)$. Here, $X = [x_1, x_2, \ldots\ldots, x_n]^T$ denotes the matrix of training inputs and $y = [y_1, y_2, \ldots\ldots, y_n]^T$ the vector of corresponding outputs. The kernel matrix $K(X,X) \in R^{n \times n}$ captures pairwise covariances between training points using a predefined kernel function $k(.,.)$ and $k(x_*, X) \in R^n$ is the covariance vector among the new input x_* and all training inputs. The term $\sigma^2 I$ accounts for the observation noise and ensures numerical stability during matrix inversion.

- **Bayesian Neural Network (BNN):** A (BNN) [53] is a neural network framework that applies Bayesian principles to capture uncertainty in its parameters. It extends conventional neural networks through the introduction of a probabilistic framework with respect to the network's weights and biases of the network. Instead of learning fixed point estimates of the parameters, BNNs learn distributions, typically overlaying the network weights w with a prior p(w). For dataset $D = \{(x_i,\ y_i)\}_{i=1}^{n}$, the objective is to determine the posterior distribution $p(w \mid D)$ which measures the model parameters' degree of uncertainty. However, due to the intractability of the exact posterior, approximate inference methods like variational inference or Markov Chain Monte Carlo (MCMC) [54] are employed. In variational inference [55, 56], for instance, a simpler distribution $q(w)$ is optimized to minimize the Kullback–Leibler divergence to approximate the true posterior, $KL(q(w)||p(w|D))$. Once the approximate posterior is learned, predictions at a new input x_* are made by marginalizing over the weight distributions, yielding $p(y_*|x_*, D) = \int p(y_* \mid x_*, w) q(w) dw$. This integral is typically approximated using Monte Carlo sampling [57], by averaging outputs over multiple of the network's stochastic forward passes. Mathematically, the predictive mean and variance might be estimated as:

$$\mu_*(x_*) \approx \frac{1}{T}\sum_{t=1}^{T} f(x_*; w^{(t)}), \quad {\sigma^2}_*(x_*) \approx \frac{1}{T}\sum_{t=1}^{T} f(x_*; w^{(t)})^2 - \mu_*(x_*)^2$$

where $w^{(t)} \sim q(w)$ are sampled weights from the learned posterior.

- **Bayesian Convolutional Neural Network (Bayesian CNN):** In contrast to standard BNNs, which require explicit variational distributions and complex inference methods like MCMC, Bayesian CNNs leverage dropout as an implicit Bernoulli variational distribution and utilize efficient Monte Carlo sampling via stochastic forward passes for posterior estimation, making

them highly practical for large-scale deep learning tasks [57]. This allows scalable uncertainty estimation and effective surrogate modeling for high-dimensional hyperparameter spaces, especially in convolutional architectures where full BNN inference is computationally impractical. Thus, the approximate posterior $q(\theta)$ is implicitly a distribution induced by dropout masks:

$$q(\theta) = \prod_i p_i^{z_i}(1-p_i)^{1-z_i}$$

Where, $z_i \in \{0,1\}$ is a random dropout mask and p_i is the dropout probability for the i^{th} neuron. Mathematically, the predictive mean $\mu_*(x_*)$ and predictive variance $\sigma^2{}_*(x_*)$ are computed over T stochastic forward passes, following the expressions:

$$\mu_*(x_*) \approx \frac{1}{T}\sum_{t=1}^{T} f_{BCNN}(x_*; w^{(t)}), \quad \sigma^2{}_*(x_*) \approx \frac{1}{T}\sum_{t=1}^{T} f_{BCNN}(x_*; w^{(t)})^2 - \mu_*(x_*)^2$$

Here, $w^{(t)}$ represents weights randomly masked through dropout in convolutional and dense layers, and input $\boldsymbol{x}_*$ is typically a 2D or 3D spatial structure (image, not flattened vector).

1.1.2. Acquisition Functions

BO starts by collecting an initial set of samples from the unknown objective function. Based on these samples, a GP model is constructed to estimate the function. An acquisition function for every iteration is computed using the GP and utilized to choose the next most promising point for sampling. After evaluating the actual objective function at this point, the latest observation is included in the training dataset in order to update the model. Until a termination condition is met, this cycle is repeated. Essential utility functions are acquisition functions in guiding the search process toward the optimum of the objective function in BO. They help ascertain the next point to sample by balancing exploration and exploitation. This means selecting points not only from areas that are expected to give good results but also from regions that are still uncertain or less explored. Below, we briefly review some of the frequently employed acquisition functions.

- **Probability of Improvement (PI):** PI [25] is a common acquisition function in BO, particularly valued for its simplicity and intuitive approach. In mathematical terms, the PI function calculates the probability that the value of the unknown objective function at a novel location x_* will exceed the best function value observed so far, denoted as $f(x^+)$, by at least a small positive threshold ξ. This threshold allows the optimization process to maintain a degree of exploration. The formula for PI is expressed as:

$$PI(x_*) = \Phi\left(\frac{\mu(x_*) - f(\mathrm{x}^+) - \xi}{\sigma(x_*)}\right)$$

where $\mu(x_*)$ and $\sigma(x_*)$ symbolize the predicted mean and standard deviation of the function value at point x_* as estimated by the surrogate model, and Φ is the standard normal distribution's cumulative distribution function (or CDF), which is equal to $\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}\exp(-t^2/2)dt$.

- **Expected Improvement (EI):** EI [40] extends the notion of surpassing the best observed value by considering both the probability and the anticipated magnitude of improvement, making it more informative and effective in complex, uncertain search spaces. It calculates as the expected value of improvement over $f(x^+)$, by:

$$EI(x_*) = (\mu(x_*) - f(x^+) - \xi)\cdot\Phi(Z) + \sigma(x_*)\cdot\phi(Z)$$

Where $Z = \left(\frac{\mu(x_*) - f(\mathrm{x}^+) - \xi}{\sigma(x_*)}\right)$, Φ and ϕ are the Gaussian cumulative distribution function (CDF) and probability density function (PDF), respectively. EI, originally introduced in 1975 [40] and later popularized by Jones et al. [58], has been extensively studied and applied across diverse optimization scenarios. These include parallel, high-dimensional, noisy, constrained, multi-objective, and multi-fidelity optimization problems.

- **Knowledge Gradient (KG):** PI is the simpler acquisition function, which only considers the likelihood of improving over the current best, or EI, which incorporates the magnitude of that improvement. KG [59] goes further by estimating how much the new observation at a given point will enhance the overall decision-making quality. KG is mathematically intended to choose the subsequent query point x_* by maximizing the expected increase in the maximum posterior mean of the objective function following the observation of its value at x_*. Let $\mu(x)$ denote the current posterior mean of the surrogate model, and let $\mu^+(x)$ represent the updated posterior mean after including the observation at x_*. The KG at the point x_* is then defined as:

$$KG(x_*) = E[max\ \mu^+(x)] - max\ \mu(x)$$

- **Upper Confidence Bound (UCB):** The confidence bound approach, known as Upper Confidence Bound (UCB) for tasks involving maximization and Lower Confidence Bound (LCB) for tasks involving minimization, is formulated to minimize regret in the context of multi-armed bandit problems. It does so by strategically combining the predicted reward with the associated uncertainty, effectively balancing exploration and exploitation during the search process [60]. Mathematically, UCB is described as:

$$UCB(x_*) = \mu(x_*) + \beta\cdot\sigma(x_*)$$

where $\beta > 0$ is a metric to manage the trade-off between exploitation and exploration.

- **Entropy Search (ES):** ES [61, 62] chooses the subsequent assessment point by seeking to maximize the expected decrease in uncertainty where the global optimum is located. Mathematically, the next point x_* is chosen by maximizing the expected decrease in the distribution's entropy over the minimum's location. This is expressed as:

$$ES(x_*) = \arg max\ E_{y(x)}[\,H\,[\,p(x_{min}\mid D)] - H\,[\,p(x_{min}\mid D\cup\{(x,y(x))\}]]$$

$H[\cdot]$ denotes entropy, $p(x_{min}\mid D)$ is the current posterior distribution of the minimum location given the observed data D, and $y(x)$ represents a possible observation at the point x. The expectation is taken over the predictive distribution of the surrogate model. In essence, Entropy Search evaluates each candidate point based on how informative it is expected to be in reducing uncertainty about the global optimum, rather than just focusing on improvement or confidence bounds.

1.2. Related Work

The development of effective Bayesian Optimization methods for complex, high-dimensional, and hybrid hyperparameter spaces requires a thorough understanding of existing approaches. This section reviews prior work on standard BO, alternative surrogate models, acquisition function innovations, high-dimensional optimization strategies, and hyperparameter tuning for deep learning models, particularly CNNs.

1.2.1. Standard Bayesian Optimization

BO [63] is a well-known strategy for the worldwide optimization of expensive, noisy black-box functions. The literature on GP optimization is extensive. In GP optimization, several heuristics for balancing exploration and exploitation have been put forth. Recent advancements have significantly enhanced Bayesian optimization by improving surrogate modeling techniques, acquisition functions, and scalability. GP-UCB was proposed in [60], which optimizes unknown, noisy functions using Gaussian Processes and achieves sublinear regret bounds by leveraging the concept of maximal information gain, but de Freitas et al in [64] showed sublinear regret for GP bandits with noisy observations, and analyzed the deterministic case and proved a significantly faster exponential convergence rate under certain regularity conditions. EI algorithms in efficient global optimization converge to the minimum of functions in the replicating-kernel Hilbert space defined by a fixed GP prior, with proven convergence rates [65]. A novel surrogate-based collaborative tuning (SCoT) method that leverages knowledge from prior experiments to enhance hyperparameter optimization across multiple learning problems, outperforming traditional single-task tuning approaches by Bardenet et al. [66]. Mahendran et al. [67] present a randomized Bayesian optimization strategy for adaptive MCMC that efficiently tunes proposal parameters for sampling from complex probabilistic graphical models with minimal objective function evaluations. Because of GP's adaptability, precisely calibrated uncertainty, and analytical qualities, they are typically utilized to build the distribution over functions used in BO [68, 69].

1.2.2. Alternative Surrogate Models

BO traditionally uses GPs to model expensive black-box functions due to their accurate uncertainty estimation, but the number of observations causes GPs to scale cubically, limiting their use in large-scale settings. To address this, Snoek et al. [54] in 2015 proposed using neural networks for adaptive basis function regression as a surrogate model, achieving linear scalability and enabling massively parallel hyperparameter optimization for deep learning tasks. Springenberg et al. [70] introduce a scalable Bayesian optimization approach using flexible neural network surrogates combined with stochastic gradient Hamiltonian Monte Carlo, enabling efficient optimization across high-dimensional, multi-task, and deep learning settings. To address the scalability limitations of GP, study [71] proposes Neural Process for Bayesian Optimization (NPBO), which outperforms or matches benchmark methods across power system and standard optimization tasks. Kerleguer et al. [72] suggest GPBNN, a hybrid surrogate model combining GP and BNN for hierarchical multi-fidelity modeling, effectively capturing predictive uncertainty across fidelity levels. Other surrogate models for Bayesian optimization, like random forests [73] and tree-structured Parzen estimators [74, 75], have been the subject of some earlier research. A novel Bayesian Neural Network architecture and algorithm that reduces storage complexity and robustly handles predictive uncertainty, avoiding local optima in non-convex settings, was introduced in [76].

1.2.3. Optimization in High-Dimensional and Hybrid Spaces

Optimization of high-dimensional (dimensionality of the search area is calculated by multiplying the number of possible choices for each hyperparameter) and hybrid (combination of discrete and continuous inputs) search spaces is highly challenging and frequently observed in a few applications. Munteanu et al. [37] introduce a Hashing-enhanced Subspace BO (HeSBO) method for high-dimensional BO utilizing subspace embeddings with low dimensions, showing tight GP error bounds and significantly improved performance over previous projection-based approaches. Wang et al. [38] recommended Ensemble Bayesian Optimization (EBO), a scalable framework that handles high-dimensional inputs, large observation budgets, and batch query selection using randomized partitions and a novel TileGP model, enabling efficient BO with tens of thousands of findings. Liu et al. [77] present a comprehensive review of scalable GPs, categorizing global and local approximation techniques to address GP's cubic complexity and enhance scalability for big data. Bayesian optimization using GP over sparse axis-aligned subspaces (SAASBO) as proposed in [78] enables efficient high-dimensional black box optimization by leveraging Hamiltonian Monte Carlo to identify relevant subspaces without problem-specific tuning. Bayesian Optimization has gained popularity for optimizing expensive functions with many parameters [79]. To address scalability issues in high-dimensional settings, various structural assumptions such as low-dimensional embeddings, additive decomposition, and variable selection have been proposed, often requiring tailored acquisition strategies.

1.2.4. Hyperparameter Optimization of CNNs in Medical Imaging

The comparative study in Table 1 reveals several important limitations in the existing literature. Firstly, none of the selected papers provide a detailed explanation of the surrogate models and acquisition functions utilized in their BO frameworks, which limits methodological transparency and reproducibility. Specifically, [80, 81, 82, 85, 88, 89, 92] do not specify the search space of CNN hyperparameters, leaving unclear the range and structure of the parameters being optimized. Although [83, 84, 86, 87, 90, 91, 93 - 97] define the search space, they all rely on the default settings of BO using the GP as the surrogate model and EI as the acquisition function. Furthermore, none of these studies addresses the complexity introduced by the high-dimensional and hybrid search space of CNN hyperparameters, which often includes a mix of continuous, categorical, and conditional parameters. Effective optimization in such spaces requires careful adaptation or replacement of both the surrogate model and acquisition function, which remains an unexplored direction in these works.

Table 1 A Comparative study of Bayesian Optimization Approaches for CNN Hyperparameter Tuning Across Medical Imaging Domains.

References (Year)	Domain	Surrogate Model	Acquisition Function	Hyperparameters Tuned	Key Findings	Research Gaps
[80] (2024)	Alzheimer's Disease Detection and Classification using ML with CNN Features	Not explicitly stated	Not specified	KNN: Number of neighbors MSVM: Kernel type, C, gamma DT: Max depth, min samples split	AlexNet's fully connected deep features gave best performance; KNN achieved 98.45% accuracy on the Kaggle Alzheimer's dataset; MSVM was best on the ADNI dataset. BO efficiently optimized model performance by searching for optimal hyperparameters.	- Used full deep feature vectors, leading to high dimensionality - No feature selection was applied - No surrogate or acquisition function details
[81] (2024)	MRI Brain Tumor Detection (Feature Extraction + ML Classification)	Gaussian Process (GP)	Not specified (default Expected Improvement (EI))	Hyperparameters of ML classifiers (SVM, KNN, DT). not specified.	Combining features of shallow and deep ResNet18 layers + BA-optimized SVM achieved 99.11% detection accuracy and 97.31% classification accuracy. Enhanced performance in sensitivity, specificity, F1, MCC, and Kappa.	- Surrogate model and acquisition function type not explicitly mentioned. - Search space limited to ML classifier hyperparameters (SVM, KNN, DT), no tuning for CNN layers. - Hybrid and high-dimensional search space unexplored.
[82] (2023)	MRI Subsampling and Reconstruction	Empirical Gaussian Process	Entropy-based exploration (inferred)	Kernel length scale	Achieved 96.3% SSIM and < 0.003 NMSE with only 12.5% k-space sampling; open-loop, generalized concentric ring subsampling paths work efficiently even on pathological brains without retraining.	- Only one hyperparameter tuned (length scale); does not explore high-dimensional CNN hyperparameter space. - Uses a fixed GP from prior data -Acquisition function not optimized for exploration–exploitation trade-off
[83] (2023)	MRI Brain Tumor Detection	Gaussian Process (GP)	Not explicitly stated	Conv layer size: 5, 7, 9, 11; Kernel size: 3×3, 5×5; Filters: 16–256 (step 16); Dropout rate: 0.0–0.6 Optimizer: Adam, SGD (Nesterov); Learning rate: 0.001, 0.0001	Achieved 98.01% accuracy/F1 on dataset 1 and 99.62% accuracy/F1 on larger dataset 2 using BO-optimized CNNs; effective automated design of depth/width parameters for tumor classification.	- Default surrogate and acquisition functions are used. -Key hyperparameters like batch size, activation function, and batch normalization were not included.
[84] (2022)	CNN hyperparameter optimization for brain tumor classification using CE-MRI	Gaussian Process (GP)	Not specified	Activation function: ReLU, ELU, Sigmoid, SELU, Tanh; Batch size: 1–128; Dropout rate: 0.1–0.5; Number of dense nodes: 32–1024; Optimizer: Adam, Nadam, AdaMax, RMSProp, SGD	Bayesian Optimization significantly improved CNN validation accuracy to 98.70% without data augmentation, outperforming pretrained models like VGG16 (97.08%), VGG19 (96.43%), ResNet50 (89.29%), InceptionV3 (92.86%),	- Search space limited to dense layers and optimizer, no tuning of convolutional layers (depth, width, filter size, etc.). - Surrogate model and acquisition function settings not customized for high-dimensional and hybrid search.

					and DenseNet201 (94.81%).	
[85] (2021)	**CNN-based tumor classification with concatenated CNN**	**Gaussian Process (GP**	**Not specified**	**Not clearly specified**	**The proposed method achieved 97.37% classification accuracy, outperforming previous works (accuracy range 84.19%–96.13%) on the same dataset.**	**- No exploration of search space for architectural hyperparameters (depth, width, kernel size, etc.). - Default surrogate and acquisition settings of BO for complex CNN concatenation architectures.**
[86] (2024)	**CNN architecture search for ultrasound image-based breast lesion classification**	**Gaussian Process (GP**	**Not specified**	**Network depth, learning rate, batch size, possibly dropout, and optimizer configuration**	**Combining ENAS with BO yields robust, efficient CNNs; Optimizing both cell structure and trainable hyperparameters improves performance, achieving low error rates on internal (≤20.6%) and external (avg. 17.3%) datasets**	**- No mention of acquisition function types (e.g., UCB, EI, PI) - Limited hyperparameter diversity - Surrogate model characteristics were not deeply analyzed**
[87] (2023)	**Classification and segmentation of breast tumors in ultrasound (US) images using CNN**	**Gaussian Process**	**Not specified**	**Learning rate, regularization factor, momentum, section depth, number of convolution filters**	**Optimized CNN outperforms state-of-the-art and shallow CNNs by ≥3% and ≥5% respectively; Outperforms U-Net and FCN in segmentation metrics (SSIM: 0.98 vs 0.93/0.92, MSE: 0.01 vs 0.21/0.28)**	- Not explicitly mentioned the surrogate and acquisition functions of BO. **- Lack of standardized benchmarks for BO in CNN hyperparameter tuning**
[88] (2021)	**Breast histopathological image classification**	**Bayesian CNN**	**Not specified**	**Not specified (focus is on architecture and uncertainty quantification)**	**Bayesian–CNN improved accuracy by 1.2% over TL-CNN; Reduced false negatives by 11% and false positives by 7.7%; Further improved by modified Bayesian–CNN with stochastic adaptive activation; 6% accuracy boost on 77% of test data using uncertainty thresholding**	**- BO not used for hyperparameter optimization - Not explicitly mentioned the acquisition functions - No exploration of high-dimensional or hierarchical search spaces**
[89] (2022)	**Diabetic Maculopathy detection in OCT and fundus images**	**Not explicitly stated**	**Not specified**	**CNN architecture and hyperparameters**	**BO significantly improved CNN performance for both fundus and OCT images; Proposed CNNs outperformed several pre-trained models (AlexNet, VGG16/19, GoogleNet, ResNet-50); Statistical tests (ANOVA, ROC, histograms) validated results**	- Surrogate model and acquisition functions are not explicitly stated - Unspecified hyperparameters, which ones exactly **- BO applied in a basic form without integration of uncertainty modeling**

[90] (2022)	**Retinal Disease Detection using OCT**	**Not explicitly mentioned**	**Not specified**	**Optimizer: Adam, RMSProp, SGD, AdaDelta; Learning rate: 1e−3, 1e−4, 1e−5, 1e−6; Activation functions: ReLU, ELU, Tanh, Leaky ReLU; Neurons in custom layers: 64, 128, 256, 512, 1024 Batch size: 32, 64, 128**	**DenseNet201 achieved >99% accuracy; Transfer learning with Bayesian optimization significantly improved performance; Image augmentation helped increase generalization**	**- Surrogate model and acquisition function not explicitly explored - Trade-off between hyperparameter tuning and training time not fully addressed**
[91] (2022)	**Tuberculosis Detection from Chest X-rays**	**Not explicitly stated**	**Not specified**	**Kernel Size: [3 to 11]; Number of Filters: [16 to 128]; Kernel Stride: [1 to 5]; Pooling Method: [Max, Average, GlobalMax]; Dense Layer Units: [128 - 1024]; Learning Rate: [0.1 to 0.001]; Optimizer: [Adam, AdaGrad, AdaDelta, SGD]**	**Shallow-CNN achieved peak accuracy and F1-score of 0.95, outperforming modified DenseNet (0.91); AUC of 0.976 with ROC; Shallow-CNN is simpler, more interpretable, and robust to noise compared to DenseNet; CAM and LIME confirmed lower lung regions as key for TB detection**	- Surrogate model and acquisition functions not specified **- Trade-off between hyperparameter tuning and training time not fully addressed**
[92] (2024)	**COVID-19 and Lung Disease Diagnosis from Chest X-rays**	**Not explicitly stated**	**EI**	**SVM kernel function: [Gaussian, Linear, Polynomial]; Box Constraint, Kernel Size, Standardization, Polynomial Order**	**Hybrid CNN + SVM kernel with Bayesian Optimization achieved 98.7% accuracy, 97.89% sensitivity, 98.2% precision, and 97.89% F1-score; SVM (Gaussian kernel) outperformed other kernel function; Five-class classification including COVID-19, Pneumonia (bacterial/viral), Normal, and TB was successfully performed**	**- Surrogate model not disclosed - CNN hyperparameter tuning details are sparse**
[93] (2023)	**Tuberculosis Diagnosis from Chest X-rays**	**Not explicitly stated**	**Not specified**	**Number of hidden layers: 1–10; Number of nodes per layer: 5–512; Learning rate: 0.000001 to 0.1 Activation functions: ReLU, Sigmoid, Tanh, Linear Optimizers: SGD, Adam, Adadelta, Adagrad, RMSprop**	**Features extracted from VGG16, EfficientNetB0, ResNet101, and DenseNet201; EfficientNetB0 + DNN (optimized with Bayesian method) achieved 99.29% accuracy**	- Surrogate model and acquisition functions not disclosed **- No exploration of high-dimensional or hierarchical search spaces**
[94] (2021)	**COVID-19 Detection from Chest X-rays**	**Not explicitly stated**	**Not specified**	**Batch Size (5 to 100); Momentum (0.01 to 0.1); Learning Rate (log scale between 0.01 and 0.02); Optimizer (SGD, Adam, RMSprop, Adagrad, Adadelta, Adamax); Pre-trained CNNs (ResNet18, ResNet50, GoogleNet, VGG16, SqueezeNet, DenseNet)**	**Proposed MKCovid-19 workflow uses transfer learning + Bayesian Optimization to fine-tune models; Accuracy of 98% on test data achieved**	- Surrogate model and acquisition functions not described **- Limited hyperparameter diversity**
[95]	**COVID-19**	**GP**	**EI**	**Number of Dense**	**Introduced**	**- A limited number of**

(2022)	**Detection from Chest X-rays**			**Layers; Number of Dense Nodes; Learning Rate; Activation Function**	**COVIDXception-Net, an enhanced Xception-based CNN tuned using Bayesian Optimization; Achieved Accuracy: 94%, Precision: 95%, Recall: 94%, Specificity: 99.7%, F1-score: 94%, MCC: 0.992**	**hyperparameters were optimized - Search space was not well-defined or extensive - Hyperparameter ranges were not described, limiting reproducibility and generalizability**
[96] (2022)	**COVID-19 Detection from Chest X-rays**	**Not explicitly stated**	**Not specified**	**Initial Learning Rate [0.001 – 1]; SGD Momentum [0.8 – 1]; Depth of Network [15]; L2 Regularization [1e-10 – 0.001]**	**Suggested a CNN model optimized with Bayesian optimization for COVID-19, Normal, and Pneumonia X-ray classification; Bayesian optimization outperformed three ablation scenarios with 96% accuracy**	**- Surrogate model and acquisition function not described - Limited detail on architectural components beyond depth - Lacked broader search over hyperparameter types (e.g., activation, batch size)**
[97] (2022)	**COVID-19 Diagnosis from Chest CT Scans**	**Not explicitly stated**	**Not specified**	**Initial Learning Rate [1e-2 – 1]; Momentum [0.8 – 0.98]; L2 Regularization [1e-10 – 1e-2]**	**BO was used to optimize hyperparameters for MobileNetV2 and ResNet-50; Achieved 99.37% accuracy, 99.36% recall, and 99.37% F1-score on mixed international datasets**	**- Surrogate model and acquisition function were not described - Hyperparameter search was limited to only three optimizer parameters - No architectural tuning of DNN models (e.g., layer configuration, activation)**

PROPOSED METHODOLOGY

The proposed methodology optimizes CNNs for MRI brain tumor detection and classification using an EBO framework, as depicted in Fig. 1. A hybrid and high-dimensional search space of CNN hyperparameters, comprising categorical, discrete, and continuous parameters, is defined, with the optimization objective being the maximization of validation accuracy. Two benchmark MRI datasets are used for evaluation: dataset_1, containing 3,000 images for binary classification of tumors versus non-tumors, and dataset_2, comprising 3,064 images for three-class classification of meningioma, glioma, and pituitary tumors. In the initial stage, BO is applied to both datasets using surrogate models such as GP, RF, BNN, and Bayesian CNN, each tested with acquisition functions including EI, ES, PI, UCB, and KG over 30 trials. The outputs, consisting of optimized hyperparameters, validation accuracy, and corresponding mean and standard deviation, are analyzed to determine the best surrogate model and the two most effective acquisition functions, forming the foundation of the proposed EBO. This enhanced approach combines the identified surrogate model with hybrid acquisition functions to address the challenges of optimizing hybrid and high-dimensional hyperparameters. The EBO is then executed on both datasets for 30 trials, with results measured in terms of best validation accuracy, mean, and standard deviation, and compared against baseline BO to validate performance gains. Subsequently, CNN architectures are constructed using the optimized hyperparameters and evaluated on both datasets using accuracy, recall, precision, F1-score, and specificity as performance metrics. Finally, the optimized CNN models are compared with existing state-of-the-art methods, demonstrating that the proposed EBO framework achieves superior performance and offers a robust strategy for MRI brain tumor detection and classification.

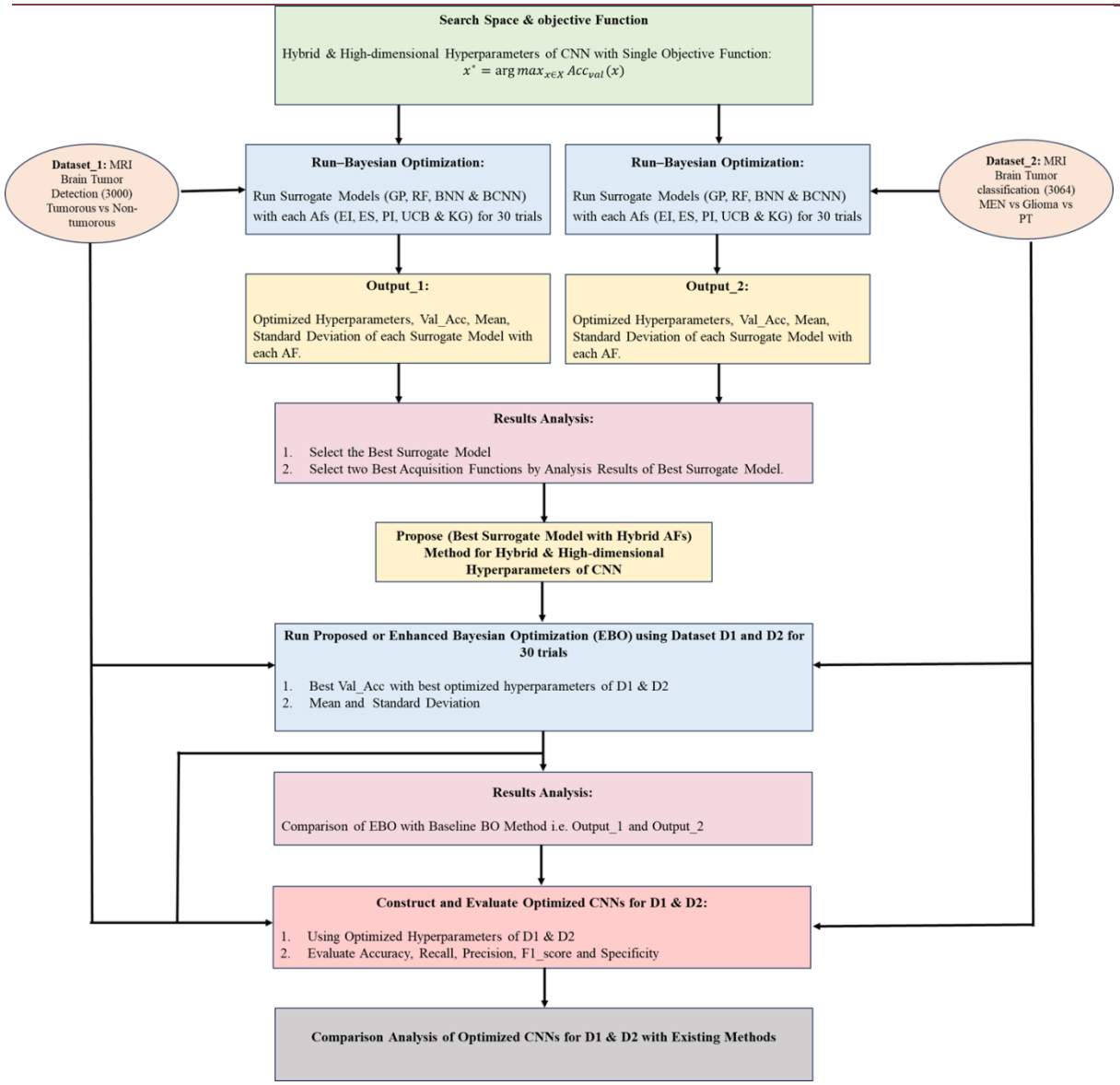


Fig. 1. Workflow of the Proposed Methodology of Enhanced Bayesian Optimization Method (EBO).

1.3. Dataset Description

This study employs the publicly available datasets on Kaggle. The first dataset (dataset_1) [98] "Brain Tumor Detection" was made available by Ahmed Hamada on Kaggle. The dataset_1 consists of a total of 3,000 MRI images, evenly distributed across two binary classes: 1,500 images labeled "Yes" (demonstrating the existence of a tumor) and 1,500 images labeled "No" (demonstrating the absence of a tumor). The second dataset (dataset_2) [99] “Brain Tumor Image Dataset” consists of 3064 T1-weighted contrast-enhanced images from 233 patients with three kinds of brain tumor: meningioma (708 slices), glioma (1426 slices), and pituitary tumor (930 slices). These files were converted by “Deniz Kavi” from the “mat” file format to the ".png" format. Every image is resized and normalized to a standard dimension of 224 × 224 × 3, consistent with the input requirements of deep CNNs and common practices in medical imaging studies. The normalization ensures consistency in feature representation and computational efficiency during training. Following established conventions in the literature for dataset partitioning [100, 101], the dataset is separated at random into 80% training, 10% validation, and 10% testing subsets, ensuring class balance in each split. This stratified division supports reliable model evaluation and mitigates the risk of data leakage or overfitting. The dataset offers sufficient variability in tumor size, location, and intensity, making it suitable for training deep learning models aimed at brain tumor detection in real-world clinical environments.

1.4. Objective Function

The principal objective of the proposed BO algorithm is to identify the ideal hyperparameter configuration $x^* \in X$ that optimizes the performance of a CNN on a given validation dataset. Specifically, the goal is to maximize the validation accuracy $y = Acc_{val}(x)$ obtained after training the CNN using a candidate configuration x. Formally, the objective function can be expressed as:

$$x^* = \arg max_{x \in X} Acc_{val}(x)$$

In which, X denotes the hyperparameter search space, $Acc_{val}(x)$ is the validation accuracy obtained using the configuration x, and x^* is the hyperparameter configuration yielding the highest validation accuracy.

1.5. Search Space

The hyperparameter search space designed for optimizing the CNN is high-dimensional, hybrid, and with a hierarchical structure. Table 2 provides an overview of the hyperparameters considered, their possible values, categories, data types, parameter types, and their roles within the overall architecture. The space includes ten key hyperparameters spanning architectural design (e.g., number of convolutional blocks, filters, filter sizes), regularization methods (e.g., dropout rate, batch normalization), activation functions, optimization strategies (e.g., learning rate, optimizer), and training configurations (e.g., batch size).

Each convolutional block is treated as a structural unit and is composed of a combination of filters, filter sizes, dropout layers, activation functions, pooling types, and optionally batch normalization. This hierarchical organization adds a layer of conditional dependency among hyperparameters, where multiple parameters jointly define the structure and function of a single convolutional block. This characteristic introduces a nested search structure, which is a typical hallmark of hierarchical spaces, where the effectiveness of one hyperparameter (e.g., activation function) is contextually dependent on the values of others (e.g., filter size or pooling type).

Furthermore, the search space is hybrid in nature, comprising categorical (e.g., activation functions, optimizers), discrete (e.g., number of convolutional blocks, filters, batch sizes), and continuous parameters (e.g., learning rate, dropout rate). Such diversity in data types increases the complexity of the search and necessitates a robust optimization strategy. The combination of high dimensionality, heterogeneity in parameter types, and hierarchical organization poses a non-trivial challenge for optimization, reinforcing the suitability of BO using BCNN with a hybrid acquisition function as proposed in this study.

Table 2 Structured and Hybrid Search Space of CNN Hyperparameters for Optimization.

Hyperparameter	Possible Values	Hyperparameter Category	Data Type	Parameter Type	Search Space Role
Number of Conv Blocks	[2, 3, 4, 5]	Architecture	Integer	Discrete	Top-level / Structural
Filters per Block	[32, 64, 128, 256]	Architecture	Integer	Discrete	Nested in Conv Blocks
Filter Size	[(2,2), (3,3)]	Architecture	Tuple (int, int)	Categorical	Nested in Conv Blocks
Dropout Rate	[0.2, 0.3, 0.4]	Regularization	Float	Discrete	Nested in Conv Blocks
Activation Function	["relu", "leakyrelu", "prelu", "swish", hard_swish.]	Architecture	String	Categorical	Nested in Conv Blocks
Pooling Type	["maxpooling", "average_pooling"]	Architecture	String	Categorical	Nested in Conv Blocks
Batch Normalization	["yes", "no"]	Regularization	String (Boolean)	Categorical	Nested in Conv Blocks
Learning Rate	[1e-3, 1e-2]	Optimization	Float	Continuous	Global
Optimizer	["SGD", "RMSprop", "Adam", "Adadelta" "Nadam"]	Optimization	String	Categorical	Global
Batch Size	[8, 16, 32]	Training	Integer	Discrete	Global

1.6. Enhanced BO Algorithm

To address the challenges of achieving optimal hyperparameter tuning in CNNs, particularly for brain tumor classification tasks, we propose an Enhanced BO framework. This enhancement integrates a powerful surrogate model with a hybrid acquisition strategy to better navigate the trade-off between exploration and exploitation. The objective is to improve the quality of suggested hyperparameter configurations by leveraging both uncertainty quantification and expected performance gains in a more balanced manner.

To successfully balance the exploration versus exploitation trade-off during hyperparameter optimization, we introduce hybrid acquisition functions that combine ES and KG. This formulation is specifically designed to complement the strengths of the best surrogate model. ES chooses the point that maximally reduces the uncertainty (entropy) regarding the position of the global optimum. It chooses the point x that maximally reduces the uncertainty about where the minimum lies, not necessarily where the minimum value is expected to be (sec. 2.1.2). Hence, ES is designed to explore uncertain regions. Where KG chooses the point x where the expected improvement in the maximum utility (e.g., accuracy) after sampling is the highest (sec. 2.1.2). It evaluates how much better we expect the best predicted value to be after sampling at x. Thus, KG is designed to exploit promising areas of the search space. Motivated by this observation, we combine both in a weighted formulation:

$$(ES + KG)x = (1 - \lambda) \cdot KG(x) + \lambda \cdot ES(x)$$

Where $\lambda \in [0,1]$ dynamically controls the trade-off. To implement this dynamic balance, we utilize a two-phase search schedule. During the first half of the trials, we set $\lambda = 1$ for full exploration (ES), and in the second half, $\lambda = 0$ for full exploitation (KG). The hybrid score at each trial is computed and used to guide the selection of new hyperparameters. Hence, the combination of ES and KG enables a principled and effective trade-off between global search and local refinement, especially suitable for high-

dimensional CNN hyperparameter spaces. Algorithm 1shows the implementation of the proposed EBO.

Algorithm 1: Proposed EBO

Input:
CNN training dataset $\boldsymbol{D_{train}}$, $\boldsymbol{D_{val}}$: $\boldsymbol{D} = \{(\boldsymbol{x_i}, \boldsymbol{y_i})\}$;
Search space of hyperparameters: $\boldsymbol{X}$;
Surrogate model: Best Model (in sec 4.1);
Acquisition function: Hybrid of ES and KG;
Acquisition parameters: λ, $\in$**;**
Number of trials: $\boldsymbol{N}$;
MC Samples: $\boldsymbol{T}$**;**
Output:
Optimal hyperparameter configuration $\boldsymbol{x}^*$
Validation accuracy $\boldsymbol{y}^*$

Initialize trial count $\boldsymbol{t = 0}$
Initialize dataset $\boldsymbol{D} = \emptyset$
For each trial i= $\boldsymbol{1\ to\ N}$: do
  Sample a candidate hyperparameter configuration $\boldsymbol{x_i} \in \boldsymbol{X}$
  Train the surrogate model on training data using $\boldsymbol{x_i}$
  For each $\boldsymbol{t = 1\ to\ T}$:
    Forward pass: compute $\boldsymbol{f(x_*; w^{(t)})}$
  End for
  Estimate predictive mean and uncertainty:
  $\boldsymbol{\mu(x_i) \approx \frac{1}{T}\sum_{t=1}^{T} f(x_i; w^{(t)})}$,
  $\boldsymbol{\sigma^2(x_i) \approx \frac{1}{T}\sum_{t=1}^{T} f(x_i; w^{(t)})^2 - \mu(x_i)^2}$
  Set λ dynamically:
  If $\boldsymbol{i < \frac{N}{2}}$: $\lambda = \boldsymbol{1.0}$ (exploration phase)
    Else: $\lambda = \boldsymbol{0.0}$ (exploitation phase)
  End if
  Compute acquisition score:
  ES: $\boldsymbol{ES(x_i) = -\log(\sigma(x_i) + \in)}$
  KG: $\boldsymbol{KG(x_i) = \mu(x_i) + \gamma \cdot \sigma(x_i)}$
  Hybrid:
  $\boldsymbol{(ES + KG)x = (1 - \lambda) \cdot KG(x) + \lambda \cdot ES(x)}$
  Evaluate the model using $\boldsymbol{x_i}$, get validation accuracy
End For
Return $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$

RESULTS AND DISCUSSIONS

1.7. Performance Analysis of Novel BCNN Surrogate Model with Acquisition Functions

To identify the most effective surrogate model and the two best acquisition functions with the best surrogate model, for CNN hyperparameter optimization, we evaluated four surrogate models: GP, RF, BNN, and BCNN, in combination with five commonly used acquisition functions: EI, PI, ES, KG, and UCB. Each surrogate and acquisition function combination was run for 30 optimization trials to maximize the validation accuracy of CNN models. All experiments were implemented using Python on Google Colab with GPU support. The evaluation was carried out on two datasets: dataset_1, a binary classification dataset for tumor detection (tumor versus no tumor), and dataset_2, a multi-class dataset for tumor classification (glioma, meningioma, pituitary). This comprehensive setup enabled a fair comparison across different configurations, allowing us to determine the surrogate model that consistently yields high-performing CNN architectures for both classification tasks.

1.7.1. Selection of Optimal Surrogate Model

On dataset_1, the performance of surrogate models varies across acquisition functions shown in Fig. 2. The GP performs strongly under EI and UCB, reaching close to 0.8 accuracy early, but its performance stabilizes at a lower level compared to BCNN in later trials. RF shows steady yet limited improvement, generally stabilizing around 0.72 - 0.75 across acquisitions, with slightly better outcomes under KG and PI. The BNN performs moderately well, achieving accuracies between 0.75 and 0.78 with EI, ES, and UCB, though it remains less consistent and weaker than BCNN. In contrast, the Bayesian CNN (BCNN) outperforms all other surrogates, achieving ~0.81 with EI, ~0.82 with UCB and ES, and peaking at ~0.85 with KG, which demonstrates its superiority in balancing exploration and exploitation.

Table 3 also shows the comparative performance of different surrogate models with various acquisition functions using dataset_1 and computes the validation accuracy, mean, and standard deviation. Across all surrogate models, the BCNN consistently demonstrated superior performance in terms of both mean validation accuracy and overall reliability. Specifically, BCNN paired with ES and KG achieved the highest validation accuracy of 0.8750, with corresponding mean accuracies of 0.5940 and 0.5723,

respectively. These results highlight the strong capability of BCNN in modeling the complex, high-dimensional search space of CNN hyperparameters, benefiting particularly from acquisition strategies that balance exploration and exploitation.

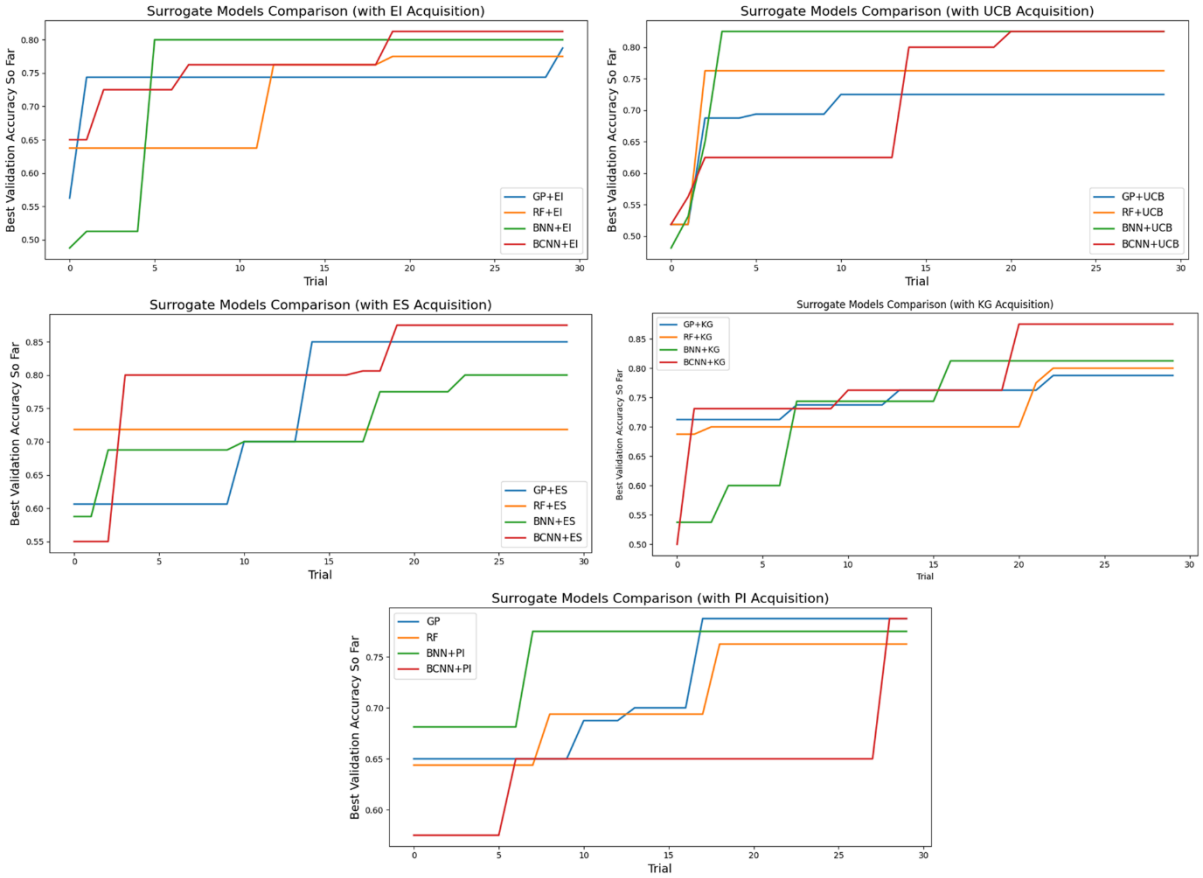


Fig. 2. Performance of Surrogate Models (GP, RF, BNN, BCNN) with Acquisition Functions (EI, PI, ES, KG, UCB) in CNN Hyperparameter Optimization using binary dataset_1

Table 3 Performance Comparison of Surrogate Models with various Acquisition Functions on Validation Accuracy, Mean, and Standard Deviation for Dataset_1

	GP			RF			BNN			BCNN		
	Val Acc.	Mean	Std.	Val Acc.	Mean	Std.	Val Acc.	Mean	Std.	Val Acc.	Mean	Std.
EI	0.7875	0.542	0.100	0.7750	0.580	0.101	0.8000	0.543	0.092	0.8125	0.567	0.120
ES	0.8500	0.554	0.118	0.7188	0.550	0.088	0.8000	0.559	0.106	0.8750	0.594	0.116
UCB	0.7250	0.547	0.084	0.7625	0.582	0.092	0.8250	0.558	0.112	0.8250	0.568	0.099
KG	0.7875	0.555	0.800	0.8000	0.569	0.10	0.8125	0.542	0.110	0.875	0.572	0.109
PI	0.7875	0.567	0.762	0.7625	0.529	0.88	0.7750	0.562	0.107	0.7875	0.519	0.089

The comparative results of surrogate models on dataset_2 across different acquisition functions are illustrated in Fig.3. Each plot represents the progression of the best validation accuracy achieved so far as the number of trials increases. Among the surrogate models, the BCNN consistently outperforms GP, RF, and BNN. Under the EI acquisition, GP and RF show gradual improvement but plateau around 0.6 - 0.7, while BNN achieves close to 0.75. In contrast, BCNN quickly surpasses 0.8 and stabilizes, highlighting its superior learning efficiency. A similar trend is observed with the UCB, where BCNN achieves the highest accuracy (~0.86), while the other surrogates remain below 0.72. When the ES acquisition is used, GP progresses slowly and stabilizes below 0.7, RF performs better, reaching around 0.75, and BNN achieves 0.75. BCNN, however, stands out by reaching nearly 0.88, demonstrating its robustness. The KG function shows competitive performance by RF (~0.77) and BNN (~0.75), but again BCNN dominates with an accuracy close to 0.88. With the PI, GP shows limited improvement (below 0.65), RF reaches ~0.7, BNN stabilizes around 0.67, while BCNN once again leads with ~0.81. Across all acquisition functions, BCNN consistently shows faster convergence and higher accuracy, indicating its effectiveness as a surrogate model for hyperparameter optimization on dataset_2.

The tabulated results in Table 4 further reinforce the observations from the graphs. BCNN records the highest validation accuracies across all acquisition functions, with values of 0.8098 for EI, 0.8721 for ES, 0.8623 for UCB, 0.8787 for KG, and 0.8098 for PI. This indicates that BCNN not only converges faster but also maintains superior predictive accuracy. BNN performs moderately, achieving up to 0.75, though with higher variability in standard deviation. RF demonstrates competitive results under the KG acquisition (0.7749) but still lags behind BCNN. GP, on the other hand, shows the weakest performance overall, failing to exceed 0.71 even with UCB. Importantly, BCNN also shows relatively lower standard deviations across acquisitions, suggesting that its performance is both accurate and stable.

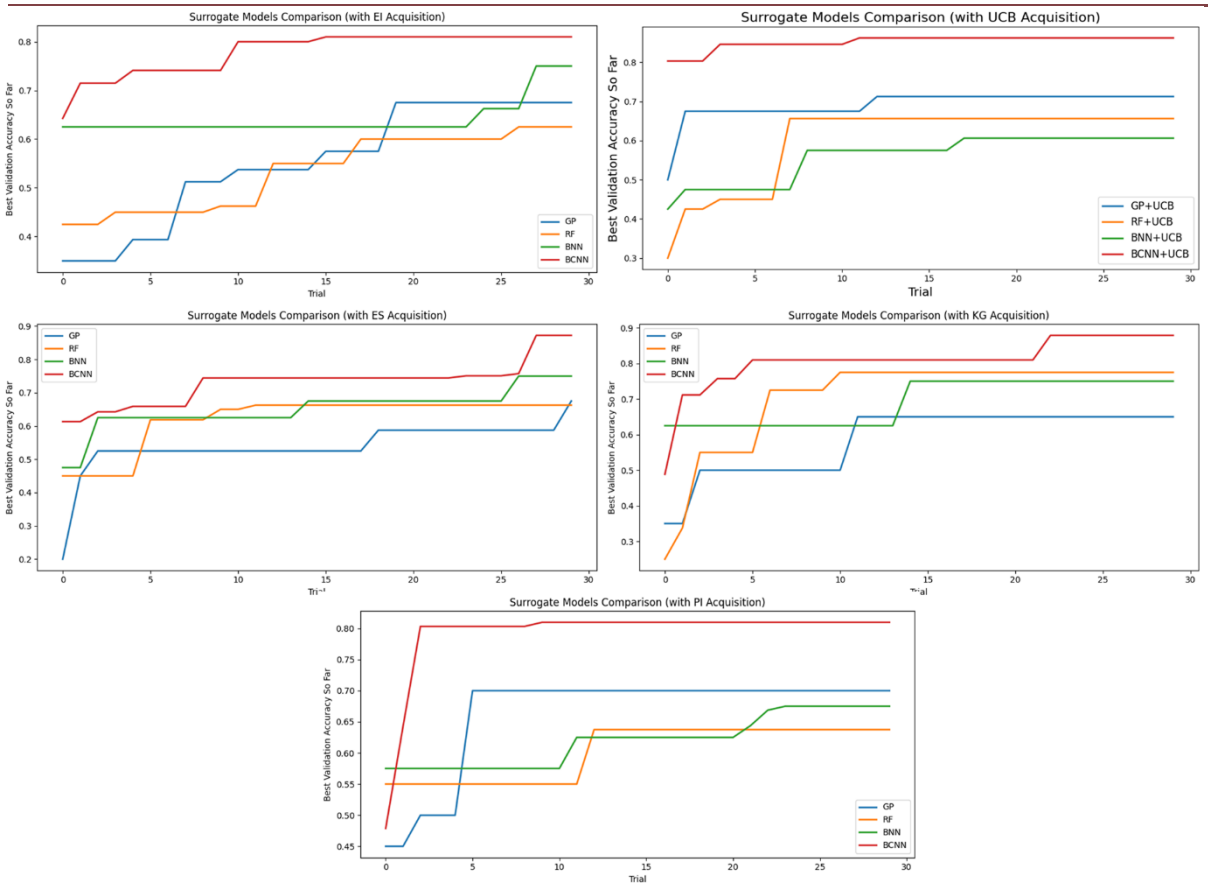


Fig. 3. Performance of Surrogate Models (GP, RF, BNN, BCNN) with Acquisition Functions (EI, PI, ES, KG, UCB) in CNN Hyperparameter Optimization using binary dataset_2

Table 4 Performance Comparison of Surrogate Models with various Acquisition Functions on Validation Accuracy, Mean, and Standard Deviation for dataset_2

	GP			RF			BNN			BCNN		
	Val Acc.	Mean	Std.	Val Acc.	Mean	Std.	Val Acc.	Mean	Std.	Val Acc.	Mean	Std.
EI	0.6750	0.348	0.113	0.625	0.386	0.108	0.75	0.442	0.147	0.8098	0.597	0.152
ES	0.6750	0.391	0.111	0.6625	0.402	0.142	0.75	0.449	0.149	0.8721	0.615	0.115
UCB	0.7124	0..440	0.129	0.6652	0.389	0.114	06062	0.412	0.110	0.8623	0.618	0.163
KG	0.6499	0.378	0.121	0.7749	0.427	0.161	0.75	0.394	0.134	0.8787	0.633	0.148
PI	0.6374	0.415	0.120	0.6999	0.405	0.126	0.6750	0.436	0.143	0.8098	0.624	0.147

Finding: The comparative analysis of surrogate models on both dataset-1 and dataset_2 indicates that the BCNN is the most effective and reliable surrogate for CNN hyperparameter optimization. The GP, while demonstrating strong initial exploration by achieving higher accuracy in the early stages, failed to maintain its performance during later trials, thereby limiting its overall reliability. RF provided stable outcomes, but its improvements remained restricted, converging at comparatively lower accuracy levels, which reduces its suitability for modeling the complexity of CNN hyperparameter spaces. The BNN offered moderate performance and, in certain instances, showed competitive results, but its relatively high variability and lack of consistency diminished its robustness across different scenarios. In contrast, BCNN consistently delivered superior validation accuracy across acquisition strategies and datasets, exhibiting not only faster convergence but also lower variability, which highlights its ability to capture intricate patterns within high-dimensional search spaces. These findings indicate that BCNN is more capable of achieving accurate and stable performance compared to the other surrogates, thereby establishing it as the most reliable and powerful surrogate model for guiding BO in CNN hyperparameter tuning.

1.7.2. Selection of acquisition functions with Optimal Surrogate model (BCNN)

The comparative analysis of acquisition functions for BO in tuning a BCNN demonstrates notable performance differences across methods on dataset_1 (Fig. 4). Among the five acquisition functions evaluated, EI, ES, UCB, KG, and PI, ES and KG were found to be the most effective. Both functions achieved the maximum validation accuracy (0.8750), with ES exhibiting the highest mean accuracy (0.594) and moderate variability (standard deviation = 0.116), indicating reliable and consistent performance. KG followed closely, maintaining competitive stability with a standard deviation of 0.109. In contrast, EI, despite achieving a reasonable accuracy of 0.8125, demonstrated the highest variability (standard deviation = 0.120), suggesting less predictable behavior across iterations. PI, although exhibiting the lowest variability (standard deviation = 0.089), recorded the lowest validation accuracy (0.7875), reflecting a trade-off between stability and effectiveness. UCB achieved moderate performance, striking a balance between accuracy and consistency but falling short compared to ES and KG. These findings

highlight ES as the most robust acquisition strategy for BCNN-based hyperparameter optimization, attributable to its ability to balance exploration and exploitation effectively. The superior performance of ES and KG may be explained by their capability to incorporate uncertainty information in a principled manner, guiding the optimization process toward regions with higher potential for improvement while maintaining consistency. Conversely, EI and PI, being more myopic in nature, fail to leverage uncertainty efficiently, leading to suboptimal and unstable performance. This suggests that advanced acquisition functions prioritizing information gain, such as ES and KG, are more suitable for complex, high-dimensional optimization tasks like CNN hyperparameter tuning.

The bar graph in Fig. 5 on dataset_2 illustrates the validation accuracy of a BCNN optimized using different acquisition functions under the Bayesian Optimization framework. The acquisition functions compared include EI, ES, UCB, KG, and PI. Each bar represents the validation accuracy for a given acquisition function, while error bars indicate the standard deviation. Additionally, the mean accuracy of each acquisition function is represented as connected points forming a trend line to show overall performance consistency. From the analysis, the KG function demonstrated the highest validation accuracy of 0.8787, with a mean accuracy of 0.633 and a standard deviation of 0.148, demonstrating strong and relatively stable performance. ES achieved a validation accuracy of 0.8721, with a mean of 0.615 and the lowest standard deviation (0.115), signifying the most consistent results among all methods. The UCB method obtained a validation accuracy of 0.8623 with a mean of 0.618; however, it exhibited the highest variability (standard deviation of 0.163). In contrast, EI and PI achieved comparatively lower validation accuracies (0.8098 each), with mean values of 0.597 (EI) and 0.624 (PI) and standard deviations of 0.152 and 0.147, respectively. These findings suggest that KG is the most effective acquisition function for BCNN-based hyperparameter optimization, while ES offers the highest consistency. EI and PI showed relatively weaker performance, indicating limited suitability for this specific setup.

Findings: The analysis across dataset_1 and dataset_2 indicates that ES and KG are the best acquisition functions for balancing exploration and exploitation in BCNN-based Bayesian Optimization. KG achieved the highest validation accuracy (0.8787 on dataset_2 and 0.8750 on dataset_1) with strong mean accuracy, while ES showed slightly lower accuracy but the lowest variability (std = 0.115), making it the most consistent method. Both functions effectively utilize uncertainty information, enabling robust optimization compared to EI, PI, and UCB, which showed lower stability or accuracy. Thus, KG is preferred for maximum accuracy, while ES is ideal for consistent performance.

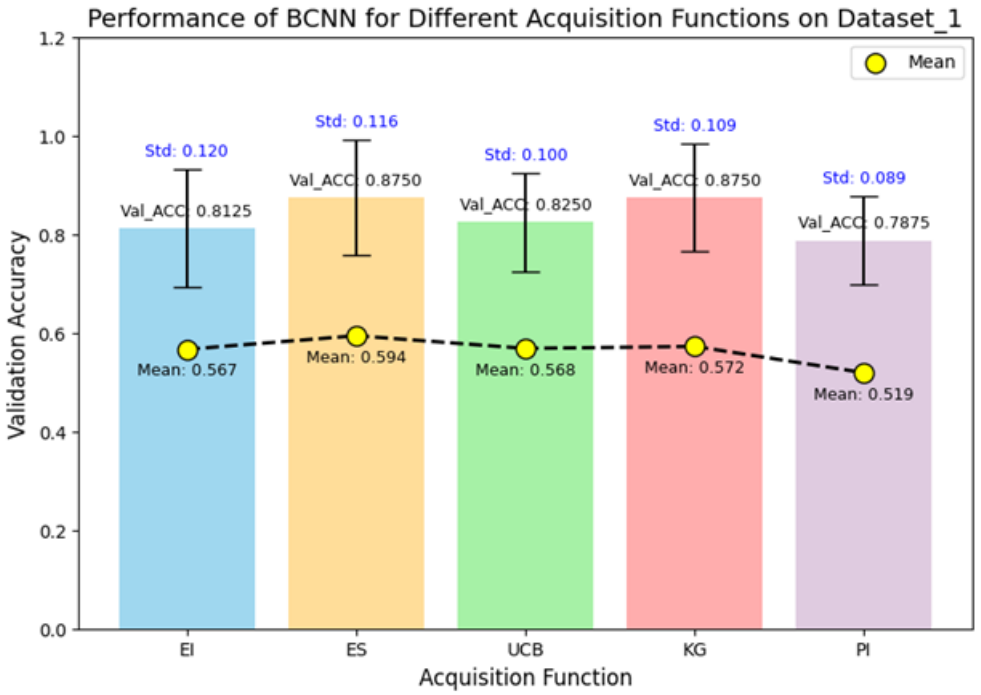
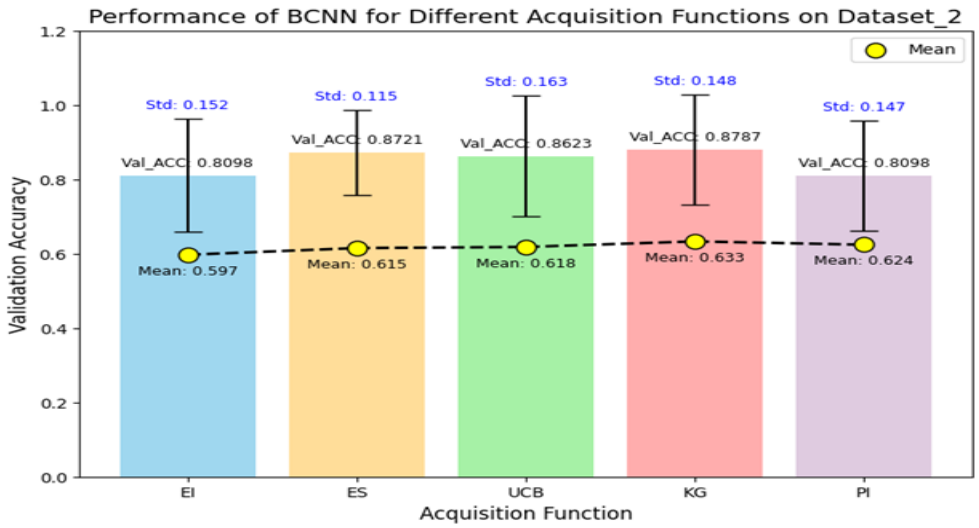


Fig. 4. Validation Accuracy with Error Bars and Mean Trend for BCNN-based Surrogate model with Acquisition functions (EI, ES, UCB, KG, and PI) on dataset_1.



c

Fig. 5. Validation Accuracy with Error Bars and Mean Trend for BCNN-based Surrogate model with Acquisition functions (EI, ES, UCB, KG, and PI) on dataset_2.

1.8. Evaluation of Proposed EBO (BCNN_ES+KG)

Fig. 6 presents the optimization history of BCNN using various acquisition functions, including EI, UCB, ES, KG, PI, and the proposed hybrid ES+KG approach on dataset_1. Among the individual strategies, ES achieved the highest mean accuracy (0.594), followed by KG (0.572), while PI recorded the lowest (0.519). EI and UCB demonstrated moderate performance with mean accuracies of 0.566 and 0.568, respectively. In contrast, the proposed BCNN_ES+KG hybrid approach significantly outperformed all others, achieving the highest mean accuracy (0.648) and demonstrating faster convergence and better stability across trials. This superior performance can be attributed to the combined advantages of ES for exploration and KG for informed exploitation, enabling more effective hyperparameter optimization for BCNN.

Fig. 7 presents the optimization history of the BCNN using six acquisition strategies: EI, UCB, ES, KG, PI, and the proposed ES+KG hybrid on dataset_2. Each subplot shows the progression of validation accuracy across 30 trials, with the blue line representing objective values, the red dashed line indicating the best-so-far accuracy, and the green dashed line showing the mean accuracy. Among the individual methods, KG achieved the highest mean accuracy (0.633), followed by PI (0.624) and ES (0.615), whereas EI recorded the lowest performance (0.597). The ES+KG hybrid approach demonstrated a clear advantage, attaining the highest mean accuracy (0.746) with relatively stable convergence, outperforming all other strategies. This superior performance can be attributed to the combined strengths of ES in promoting exploration and KG in leveraging knowledge-based exploitation, leading to more efficient and robust hyperparameter optimization.

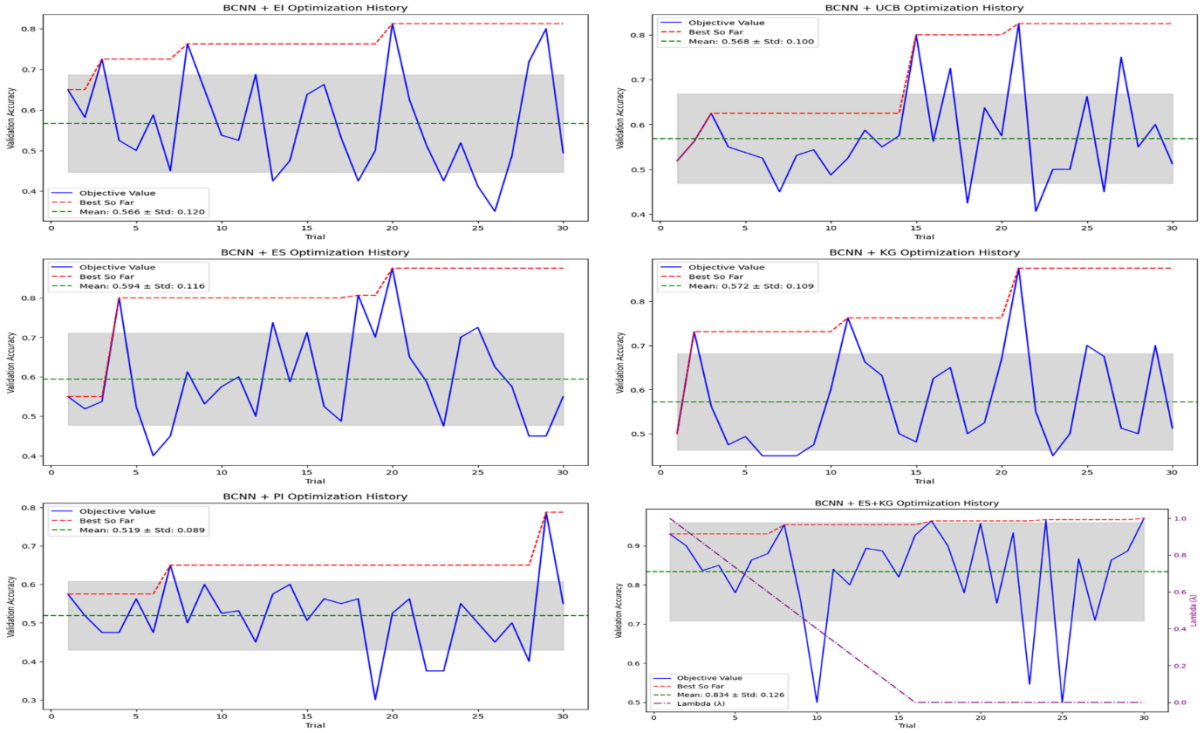


Fig. 6. Comparison of optimization history of proposed BCNN_ES+KG method with BCNN and individual acquisition functions (EI, UCB, ES, PI, and KG) on dataset_1

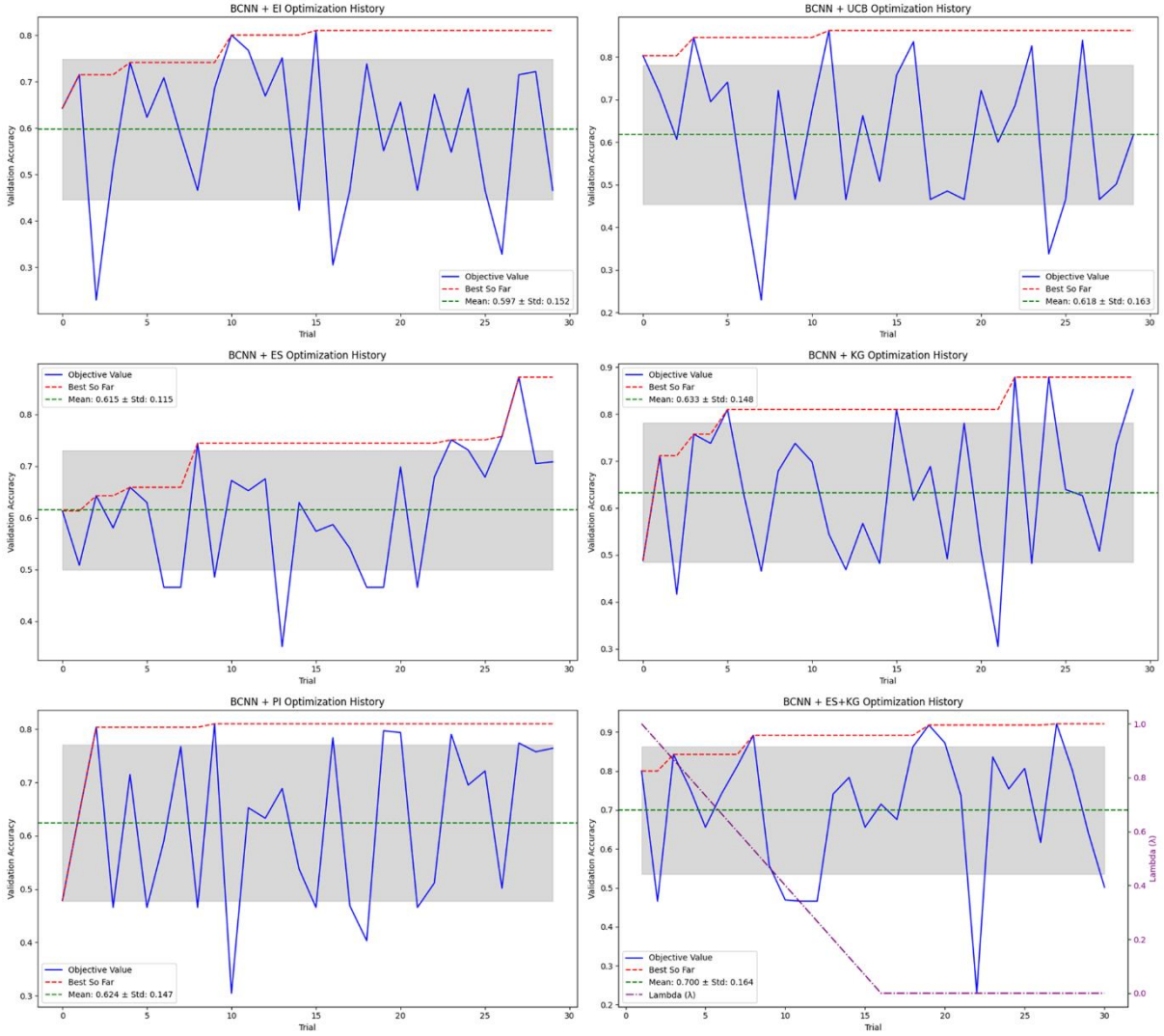


Fig. 7. Comparison of optimization history of proposed BCNN_ES+KG method with BCNN surrogate model and individual acquisition functions (EI, UCB, ES, PI, and KG) on dataset_2

The comparative analysis, as depicted in Table 5, of acquisition functions for BCNN-based BO reveals a clear performance advantage for the proposed hybrid ES+KG approach across both datasets. For dataset_1, the proposed method achieved the highest validation accuracy (0.970) and mean value (0.834), with a low standard deviation (0.126), indicating both superior predictive performance and consistency. Similarly, in Dataset D2, ES+KG recorded the best validation accuracy (0.9213) and mean (0.700) while maintaining competitive stability (standard deviation 0.164). In contrast, individual acquisition functions such as EI, UCB, ES, KG, and PI demonstrated moderate improvements, but none matched the overall effectiveness of the hybrid strategy. These findings confirm that integrating ES with KG achieves an improved balanced trade-off between exploration and exploitation, resulting in improved convergence and robustness for hyperparameter optimization in BCNN-based models.

Table 5 Comparative analysis of different acquisition functions (EI, UCB, ES, KG, PI, and the proposed ES+KG) used with a BCNN surrogate model for hyperparameter optimization across two datasets: dataset_1 and dataset_2.

	Dataset_1			Dataset_2		
	Val Acc	Mean	Standard Deviation	Val Acc	Mean	Standard Deviation
BCNN_EI	0.8125	0.5665	0.1201	0.8098	0.597	0.152
BCNN_UCB	0.8250	0.5683	0.0999	0.8623	0.618	0.163
BCNN_ES	0.8750	0.5940	0.1164	0.8721	0.615	0.115
BCNN_KG	0.8750	0.5723	0.1089	0.8787	0.633	0.148
BCNN_PI	0.7875	0.5190	0.8930	0.8098	0.624	0.147
Proposed BCNN_ES+KG	**0.970**	**0.834**	**0.126**	**0.9213**	**0.700**	**0.164**

Across both datasets, the proposed ES+KG hybrid consistently outperforms individual acquisition functions, delivering the highest mean and validation accuracies with faster, more stable convergence. By jointly leveraging ES-driven exploration and

KG's informed exploitation, the method achieves a superior balance that yields robust and reliable hyperparameter optimization for BCNN.

1.9. Implementation of CNNs with Optimized Hyperparameters of Dataset_1 and Dataset_2

This section presents the implementation of CNNs using the optimized hyperparameters obtained from the proposed EBO framework for both datasets, dataset_1 (binary classification) and dataset_2 (multi-class classification). The optimized hyperparameters include architectural parameters (such as the number of convolutional layers, filter sizes, kernel dimensions, and dropout rates) and training parameters (such as learning rate, batch size, and optimizer type). For each dataset, the CNN model is constructed based on the optimized architectural hyperparameters, followed by training and evaluation using the corresponding optimized training hyperparameters. The evaluation of the optimized models is conducted using standard performance metrics, including accuracy, precision, recall, F1-score, and specificity, which together provide a comprehensive assessment of classification performance. Accuracy represents the percentage of total samples that are correctly classified, while precision evaluates the ratio of true positive predictions to all positive predictions, reflecting the model's ability to minimize false positives. Recall (or sensitivity) measures the proportion of true positives accurately detected, which is particularly important in medical diagnosis to avoid missed cases. F1-score is the harmonic mean of precision and recall, balancing both metrics into a single measure of performance. Finally, specificity assesses the ability of the model to accurately classify negatives, thereby decreasing the rate of false positives. Subsection 4.3.1 details the optimized_CNN1 implementation for dataset_1, while Subsection 4.3.2 presents the implementation optimized_CNN2 for dataset_2.

1.9.1. Construction and Evaluation of Optimized_CNN1 Model for Dataset_1

The optimized_CNN1 for dataset_1 was constructed using the best hyperparameters obtained through the EBO framework. These are "number_of_conv_blocks: 2; learning_rate: 0.003920476718634066, optimizer: Nadam; batch_size: 16; filters_block_1: 32; filter_size_block_1: (2, 2); dropout_rate_block_1: 0.4; activation_block_1: prelu; pooling_layers_block_1: maxpooling; batch_normalization_block_1: yes; filters_block_2: 32; filter_size_block_2: (3, 3); dropout_rate_block_2: 0.3; activation_block_2: relu; pooling_layers_block_2: average_pooling; batch_normalization_block_2: no". These optimized hyperparameters aim to optimize the trade-off between network depth, regularization, and computational performance for binary classification of brain MRI images. Fig. 8 presents a structured diagram of the optimized CNN architecture, offering a visual depiction of the model's design by using these optimized hyperparameters. The dataset, consisting of 3,000 MRI images of brain tumor detection (Tumorous & Non-tumorous), was resized to 224 × 224 pixels and divided into 80% training, 10% validation, and 10% testing sets.

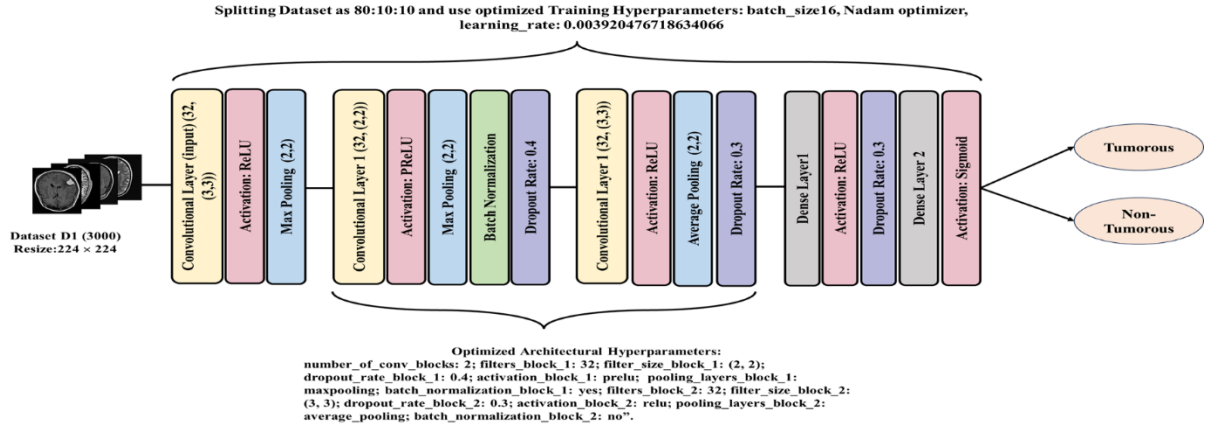


Fig. 8. Architecture of Optimized_CNN1 using optimized hyperparameters for the detection of MRI brain tumor (dataset_1)

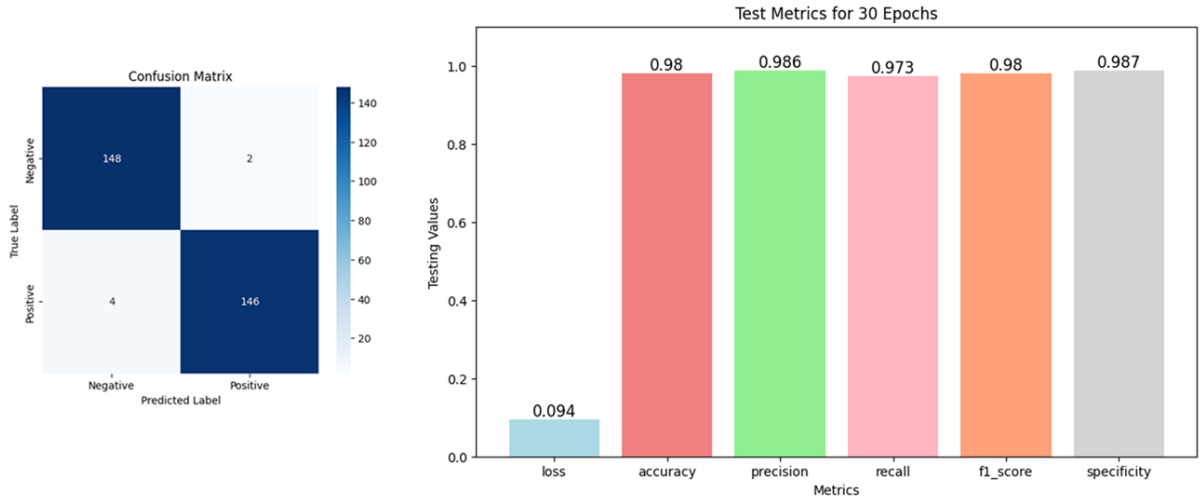


Fig. 9. Confusion matrix and performance metrics of Optimized_CNN1 model on dataset_1

The performance analysis of the optimized CNN model for dataset_1, trained with hyperparameters (learning rate: 0.003920476718634066, optimizer: Nadam, batch size: 16, epochs: 30), demonstrates highly effective classification capability. The confusion matrix in Fig. 9 indicates an almost perfect classification, with 148 true negatives, 146 true positives, and only 2 false positives and 4 false negatives. This indicates that the model preserves strong predictive accuracy for both classes. The bar chart (Fig. 9) of test metrics provides additional insights. The testing loss is 0.094, which signifies minimal error and suggests that the model has achieved strong convergence without significant overfitting. The accuracy is 98%, reflecting highly reliable overall performance on unseen test data. Moreover, precision (98.6%) indicates that false positives are extremely low, while recall (97.3%) confirms the model's strong ability to detect actual positive cases. The F1-score (98%) shows an optimal balance between precision and recall, enabling the model to be robust for real-world deployment. Additionally, specificity (98.7%) highlights the model's ability to correctly detect negative cases, reducing the likelihood of misclassifying normal or non-tumor images as tumors.

These results collectively confirm that the chosen optimization strategy, including the Nadam optimizer and adaptive learning rate, significantly enhances convergence speed and generalization ability. The combination of optimized hyperparameters, regularization techniques (dropout and batch normalization), and effective architecture design has contributed to achieving high classification performance across all evaluation metrics. This level of accuracy and consistency is crucial in medical imaging applications, where diagnostic reliability is of utmost importance.

1.9.2. Construction and Evaluation of Optimized_CNN2 Model for Dataset_2

The best hyperparameters for optimized_CNN2 model for dataset_2 are: learning_rate: 0.009816636240589462; optimizer: SGD; batch_size: 8; number_of_conv_blocks: 3; filters_block_1: 64 filter_size_block_1: (3, 3) dropout_rate_block_1: 0.3 activation_block_1: leakyrelu pooling_layers_block_1: maxpooling batch_normalization_block_1: yes filters_block_2: 64 filter_size_block_2: (2, 2) dropout_rate_block_2: 0.3 activation_block_2: prelu pooling_layers_block_2: maxpooling batch_normalization_block_2: yes filters_block_3: 32 filter_size_block_3: (3, 3) dropout_rate_block_3: 0.2 activation_block_3: leakyrelu pooling_layers_block_3: average_pooling batch_normalization_block_3: no **.** The optimized CNN model for Dataset D2 by using the mentioned optimized hyperparameters is designed for effective brain tumor classification into three classes: Meningioma, Glioma, and Pituitary, as shown in Fig. 10. The dataset, consisting of 3,064 MRI images, was resized to 224 × 224 pixels and divided into 80% training, 10% validation, and 10% testing sets.

The evaluation of the optimized_CNN2 model on dataset_2 is shown in Fig. 11, trained with the hyperparameters (learning rate: 0.009816636240589462, optimizer: SGD, batch size: 8, and 30 epochs), indicates strong performance in multi-class brain tumor classification (Meningioma, Glioma, Pituitary). The confusion matrix shows that the model correctly classified most instances across the three classes, with minor misclassifications. Specifically, the first class achieved 68 correct predictions with only 4 misclassifications, the second class had 134 correct predictions with 10 misclassifications, and the third class was classified with high accuracy, having 93 correct predictions and no errors for other classes. This distribution demonstrates that the model generalizes well across all classes without significant bias toward any particular category. The test metrics further support the model's effectiveness. The loss value of 0.199 indicates efficient convergence with minimal error. The overall accuracy of 95.5% reflects reliable classification across all tumor types. Precision (94.6%) shows that the model maintains a low false-positive rate, while recall (95.8%) suggests strong sensitivity in identifying actual positive cases. The F1-score of 95.1% confirms a well-balanced trade-off between precision and recall, which is critical for medical diagnostic applications. Furthermore, the specificity (97.7%) indicates excellent performance in correctly identifying negative instances, reducing the likelihood of incorrect tumor detection.

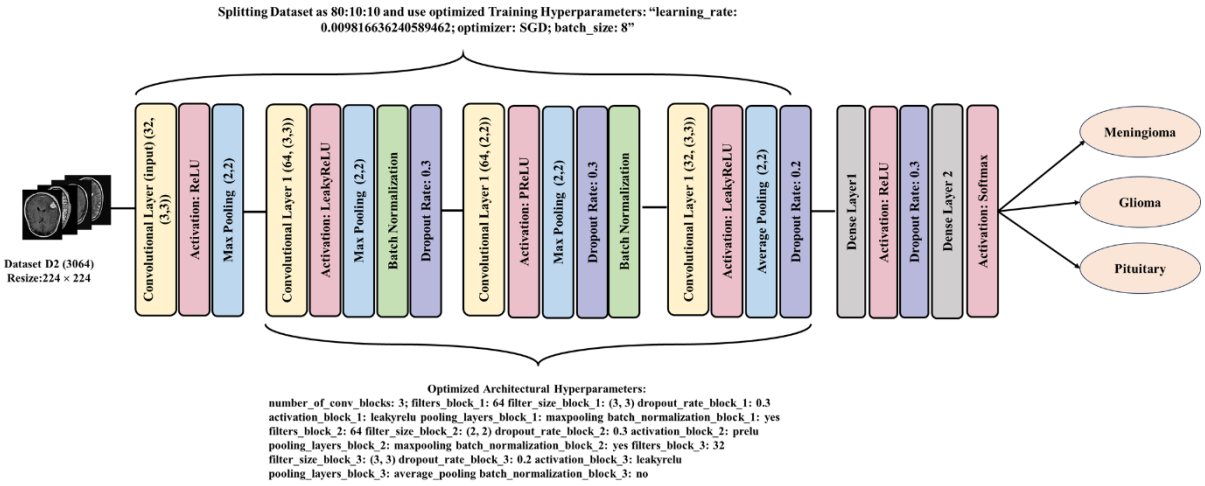


Fig. 10. Architecture of Optimized_CNN2 using optimized hyperparameters for the detection of MRI brain tumor (dataset_2)

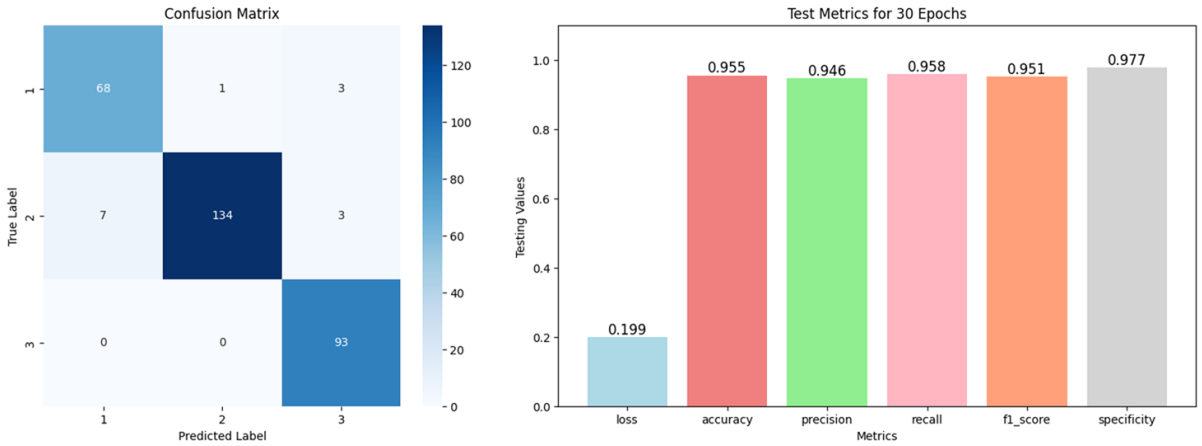


Fig. 11. Confusion matrix and performance metrics of Optimized_CNN2 model on dataset_2

1.10. Comparison Analysis with Existing Methods

The comparative validation of the proposed optimized CNNs in comparison with existing state-of-the-art methods in Table 6 highlights both performance and methodological distinctions. Some prior works, such as [102] and [103], reported very high accuracies above 99% by employing exhaustive GS across extremely large search spaces, ranging from 716,800 to 829,440 configurations with dimensionality between 5 and 9. While such exhaustive strategies ensured strong performance, they required substantial computational resources, making them less practical for real-world clinical deployment. Likewise, studies such as [85] and [84] applied Bayesian optimization over hybrid infinite search spaces, which provided flexibility but suffered from poorly defined configuration boundaries and heavy computational demand. In contrast, the proposed models, trained on Kaggle-based datasets, achieved competitive and balanced results. On the Br35H dataset (dataset_1), the model reached 98.00% accuracy, 97.33% recall, 98.65% precision, and 97.99% F1-score, while on the custom CNN dataset, it achieved 95.79% accuracy, 95.81% recall, 95.11% precision, and 95.44% F1-score. Although slightly lower in accuracy than the top-performing existing approaches, the proposed models exhibit superior balance in recall and F1-score, metrics that are clinically more relevant as they directly reduce the risk of false negatives.

A major strength of the proposed framework lies in its efficient and structured hyperparameter optimization strategy. Unlike previous works that relied on excessively large finite search spaces or undefined infinite ones, the proposed method explores a 10-dimensional hybrid search space that integrates both discrete design parameters and a continuous learning rate. This balance ensures comprehensive exploration of architectural, regularization, and optimization aspects without incurring the computational infeasibility of prior methods. Furthermore, many studies employing BO rely on the default GP surrogate and the EI acquisition function, which, while effective, may limit the diversity and adaptability of the search process. In contrast, the proposed EBO approach employs a BCNN surrogate in combination with ES+KG acquisition, which not only improves search efficiency but also introduces uncertainty quantification, enhancing interpretability and trust in medical applications. This combination of computational efficiency, balanced performance across metrics, and added reliability positions the proposed method as a strong and clinically practical alternative to existing state-of-the-art approaches.

Table 6 Comparison analysis of proposed optimized CNN Models with state-of-the-art optimized CNN methods for MRI Brain Tumor Classification

Reference & Year	Dataset	Model & Classes	Search Space	Dimensionality & Configuration	HPO Method	Acc (%)	Recall (%)	PRE (%)	F1_S (%)	SPE (%)
[85] 2021	Figshare (Cheng et al.) (3064) (Augmented)	Custom CNN MEN (708) vs Glioma (1426) vs PT (930)	Dropout Percentages [0, 0.5]; Conv2D Filters [16, 32, 64, 128, 256, 512]; Conv2D Kernel Size (x, x) [2, 3, 4, 5]; Max Pooling Size (x, x) [2, 3, 4, 5]; Dense Filter [16, 32, 64, 128, 256, 512, 1024, 2048, 4096]	dimensionality = 5 Search-space size = infinite (Hybrid Search-space)	Bayesian Optimization	97.37	97.38	97.4	97.3	98.02
[102] 2021	RIDER + REMBRANDT,	Custom CNN1 Tumor (1640) vs non-tumor (1350)	Number of convolution and max pooling layers [1, 2, 3, 4]; Number of FC layers [1, 2, 3, 4]; Number of filters [16,	dimensionality = 9 Search-space size = 716,800	Grid Search	99.33	99.4	99.25	-	99.4

	TCGA-LGG + Figshare (Cheng et al.) (11,510)	Custom CNN2 Normal (850) vs Glioma (950) vs MEN (700) vs PT (700) vs MTS (750)	24, 32, 48, 64, 96, 128]; Filter size [3, 4, 5, 6, 7]; Activation function [ELU, SELU, ReLU, Leaky ReLU]; Mini-Batch Size [4, 8, 16, 32, 64]; Momentum [0.80, 0.85, 0.9, 0.95]; Learning Rate [0.0001, 0.0005, 0.001, 0.005]; l2 Regularization [0.0001, 0.0005, 0.001, 0.005]			92.66	91.82	92.02	-	98.03
		Custom CNN3 G-II (1676) vs G-III (1218) vs G-IV (1676)				98.14	98.60	98.31	-	99.02
[84] 2022	Figshare (Cheng et al.) (3064)	Custom CNN MEN (708) vs Glioma (1426) vs PT (930)	Activation function [ReLU, ELU, Sigmoid, SELU, Tanh]; Batch size [1 to 128]; Dropout rate [0.1 to 0.5]; Number of dense nodes [32 to 1024]; Gradient descent optimizer [Adam, Nadam, AdaMax, RMSProp, SGD]	dimensionality = 5 Search-space size = infinite (Hybrid Search-space)	Bayesian Optimization	98.70	98.66	98.33	98.66	-
[83] 2023	Kaggle (S. Bhuvaji) (3264)	Custom CNN MEN (937) vs Glioma (926) vs PT (901) vs Normal (500)	Convolutional layer size [5, 7, 9, 11]; Kernel size [3x3, 5x5]; Filters size [min value =16, max value =256, step =16]; Dropout-rate [0.0, 0.2, 0.3, 0.4, 0.5, 0.6]; Optimizer [Adam, SGD with Nesterov]; Learning rate [0.001, 0.0001]	dimensionality = 6 Search-space size = 3,072	Bayesian Optimization	98.01	-	-	98	-
[103] 2024	TCIA (REMBRANDT)	Custom CNN1 Malignant (1743) vs Non-malignant (1422)	Layers of maximum pooling and CNN [1, 2, 3, 4]; Layers that are completely connected [1, 2, 3, 4]; Nnumber of flters [8, 16, 24, 32, 48, 64, 96, 128, 256]; Intensity of fltration [3, 4, 5, 6, 7]; Activation [ReLU, ELU, Leaky ReLU]; Size of minibatch [4, 6, 16, 24, 32, 64]; Rate of change [0.78, 0.77, 0.95, 0.96]; Rate of learning [0.0002, 0.00043, 0.002, 0.004]; R2-regularization [0.0002, 0.00043, 0.002, 0.004]	dimensionality = 6 Search-space size = 829,440	Grid Search	99.53	-	-	-	-
		Custom CNN2 BN (910) vs Glioma (985) vs MEN (750) vs PT (750) vs MTS (800)				93.81				
		Custom CNN3 G-II (1712) vs 4720 (G-III) vs G-				98.56				

		IV (1712)								
Optimized CNNs with Proposed HPO	**Kaggle dataset (Br35H) (3000)**	**Optimized_CNN1 Tumorous (1500) and non-Tumorous (1500)**	**Number of Conv Blocks: [2, 3, 4, 5]; Filters per Block: [32, 64, 128, 256]; Filter Size: [(2,2), (3,3)]; Dropout Rate: [0.2, 0.3, 0.4]; Activation Function: ["relu", "leakyrelu", "prelu", "swish", hard_swish.]; Pooling Type: ["maxpooling", "average_pooling"]; Batch Normalization: ["yes", "no"]; Learning Rate: [1e-3, 1e-2]; Optimizer: ["SGD", "RMSprop", "Adam", "Adadelta", "Nadam"]; Batch Size: [8, 16, 32]**	**dimensionality = 10 Search-space size = infinite (Hybrid Search-space)**	**EBO: BCNN_ES+KG**	**98.00**	**97.33**	**98.65**	**97.99**	**98.67**
	Figshare (Cheng et al.) (3064)	**Optimized_CNN2 MEN (708) vs Glioma (1426) vs PT (930)**				**95.79**	**95.81**	**95.11**	**95.44**	**97.73**

HPO: Hyperparameters Optimization, ACC: Accuracy, PRE: Precision, SPE: Specificity, F1_S: F1_Score, MEN: Meningioma, PT: Pituitary, MST: Metastatic

2. Conclusion and Future Work

This study introduced BCNN as the surrogate model in BO and proposed an Enhanced Bayesian Optimization (EBO) framework for optimizing the hybrid and high-dimensional hyperparameter search space of CNNs in MRI-based brain tumor detection and classification. The framework employed an objective function of maximizing validation accuracy (Val_Acc) and systematically benchmarked four surrogate models (GP, RF, BNN, and BCNN) with five acquisition functions (EI, PI, UCB, ES, and KG) applied across two datasets (detection and classification of brain tumors) and validated over 30 trials. Based on these evaluations, the Bayesian CNN combined with a novel ES+KG hybrid acquisition function, designed to balance exploration and exploitation, was proposed as the EBO framework. The EBO framework was executed for 30 independent trials on both datasets to obtain optimized hyperparameters, which were subsequently used to train two CNN architectures: optimized_CNN1 for dataset_1 (tumor vs. non-tumor) and optimized_CNN2 for dataset_2 (glioma, meningioma, pituitary). Both optimized models consistently outperformed state-of-the-art approaches across multiple performance metrics, including accuracy, recall, precision, F1-score, and specificity. These results demonstrate not only the superior classification ability of the proposed EBO framework but also its robustness, reproducibility, and computational feasibility in high-dimensional hybrid search spaces.

For future work, the framework can be extended to larger and more diverse datasets, multi-modal imaging, and 3D volumetric analysis, as well as adapted for multi-objective optimization. Exploring advanced surrogate models and real-time clinical integration with model compression and clinician-in-the-loop validation will further enhance its robustness and translational impact.

Declarations:

Data Availability Statement: The datasets that support the findings of this study are available online:
https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection
https://doi.org/10.6084/m9.figshare.1512427.v5

Funding: No funding was received for conducting this study.

Competing interests: The authors have no competing interests to declare that are relevant to the content of this article.

Author Contributions:

Kalpana Devi (Corresponding): Conceptualization, Writing- Reviewing, Methodology, Software, Data curation, Writing-Original draft preparation, Visualization, Investigation. **Aman Kumar Sharma***:* Supervision, Software, Validation, and Editing.

REFERENCES

1. J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, The journal of machine learning research, 13(1) (2012) 281-305.
2. J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian optimization of machine learning algorithms, Advances in neural information processing systems, 25 (2012).

3. M. Feurer, F. Hutter, Hyperparameter Optimization, in: Automated machine learning. Cham: Springer. 2019 May 17:113-34.
4. T. Elsken, J.H. Metzen, F. Hutter, Neural architecture search: A survey, Journal of Machine Learning Research, 20(55) (2019), 1-21.
5. G. Litjens, T. Kooi, B.E. Bejnordi, A.A. Setio, F. Ciompi, M. Ghafoorian, J.A. Van Der Laak, B. Van Ginneken, C.I. Sánchez, A survey on deep learning in medical image analysis, Medical image analysis, 42 (2017) 60-88. https://doi.org/10.1016/j.media.2017.07.005
6. A. Younesi, M. Ansari, M. Fazli, A. Ejlali, M. Shafique, J. Henkel, A comprehensive survey of convolutions in deep learning: Applications, challenges, and future trends, IEEE Access 12 (2024) 41180-218. 10.1109/ACCESS.2024.3376441
7. A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, Deep learning for computer vision: A brief review, Computational intelligence and neuroscience, 2018(1) (2018) 7068349. https://doi.org/10.1155/2018/7068349
8. A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, J. Garcia-Rodriguez, A review on deep learning techniques applied to semantic segmentation, arXiv preprint arXiv (2017) 1704.06857. https://doi.org/10.48550/arXiv.1704.06857
9. Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M.S. Lew, Deep learning for visual understanding: A review, Neurocomputing, 187 (2016) 27-48. https://doi.org/10.1016/j.neucom.2015.09.116
10. N. Cummins, A. Baird, B.W. Schuller, Speech analysis for health: Current state-of-the-art and the increasing impact of deep learning, Methods, 151 (2018) 41-54. https://doi.org/10.1016/j.ymeth.2018.07.007
11. M.K.Hasan, M. Shahjalal, M.M. Islam, M.M. Alam, M.F. Ahmed, Y.M. Jang, The role of deep learning in NOMA for 5G and beyond communications, in: 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), vol 19, IEEE, 2020, pp. 303-307. 10.1109/ICAIIC48513.2020.9065219
12. G. Chartrand, P.M. Cheng, E. Vorontsov, M. Drozdzal, S. Turcotte, C.J. Pal, S. Kadoury, A. Tang, Deep learning: a primer for radiologists, Radiographics, 37(7) (2017) 2113-31. https://doi.org/10.1148/rg.2017170077
13. M.H. Hesamian, W. Jia, X. He, P. Kennedy, Deep learning techniques for medical image segmentation: achievements and challenges, Journal of digital imaging, 32 (2019) 582-96. https://doi.org/10.1007/s10278-019-00227-x
14. J. Wang, J. Xu, X. Wang, Combination of hyperband and Bayesian optimization for hyperparameter optimization in deep learning, arXiv preprint arXiv (2018) 1801.01596. https://doi.org/10.48550/arXiv.1801.01596
15. Y. Ozaki, M. Yano, M. Onishi, Effective hyperparameter optimization using Nelder-Mead method in deep learning, IPSJ Transactions on Computer Vision and Applications, 9 (2017) 1-2. https://doi.org/10.1186/s41074-017-0030-7
16. A. Zela, A. Klein, S. Falkner, F. Hutter, Towards automated deep learning: Efficient joint neural architecture and hyperparameter search, arXiv preprint arXiv (2018) 1807.06906. https://doi.org/10.48550/arXiv.1807.06906
17. X. Zhang, X. Chen, L. Yao, C. Ge, M. Dong, Deep neural network hyperparameter optimization with orthogonal array tuning, in: Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, Springer International Publishing, 2019, pp. 287-295. https://doi.org/10.1007/978-3-030-36808-1_31
18. A.H. Victoria, G. Maragatham, Automatic tuning of hyperparameters using Bayesian optimization, Evolving System, 12(1) (2021) 217-23. https://doi.org/10.1007/s12530-020-09345-2
19. M.J. Kochenderfer, T.A. Wheeler, Algorithms for optimization, Mit Press; 2019 Mar 26.
20. S. Falkner, A. Klein, F. Hutter, Practical hyperparameter optimization for deep learning.
21. Q.V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, A.Y. Ng, On optimization methods for deep learning, in: Proceedings of the 28th international conference on international conference on machine learning, 2011, pp. 265-272.
22. R. Sun, Optimization for deep learning: theory and algorithms, arXiv preprint arXiv, (2019) 1912.08957. https://doi.org/10.48550/arXiv.1912.08957
23. R.Y. Sun, Optimization for deep learning: An overview, Journal of the Operations Research Society of China, 8(2) (2020) 249-94. https://doi.org/10.1007/s40305-020-00309-6
24. H.J. Kushner, A versatile stochastic model of a function of unknown and time varying form, J. Math. Anal. Appl., 5(1) (1962) 150–167. https://doi.org/10.1016/0022-247X(62)90011-2
25. H. J. Kushner, A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise, J. Basic Eng., 86(1) (1964) 97–106. https://doi.org/10.1115/1.3653121
26. R.R. Griffiths, J. Hernández-Lobato, Miguel, Constrained Bayesian optimization for automatic chemical design using variational autoencoders, Chem., (2020) 577-586. 10.1039/C9SC04026A
27. K. Wang, A.W. Dowling, Bayesian optimization for chemical products and functional materials, Current Opinion in Chemical Engineering, 36 (2022) 100728. https://doi.org/10.1016/j.coche.2021.100728
28. S. Ament, M. Amsler, D.R. Sutherland, M.C. Chang, D. Guevarra, A.B. Connolly, J.M. Gregoire, M.O. Thompson, C.P. Gomes, R.B. Van Dover, Autonomous materials synthesis via hierarchical active learning of nonequilibrium phase diagrams, Science Advances, 7(51) (2021) eabg4930. 10.1126/sciadv.abg4930
29. R. Lam, M. Poloczek, P. Frazier, K.E. Willcox, Advances in Bayesian optimization with applications in aerospace engineering, in: 2018 AIAA Non-Deterministic Approaches Conference, 2018, p. 1656. https://doi.org/10.2514/6.2018-1656
30. S. Ament, A. Witte, N. Garg, J. Kusuma, Sustainable concrete via Bayesian optimization. arXiv preprint arXiv, (2023) 310.18288. https://doi.org/10.48550/arXiv.2310.18288
31. J. Wang, J. Xu, X. Wang, Combination of hyperband and Bayesian optimization for hyperparameter optimization in deep learning, arXiv preprint arXiv, (2018) 1801.01596. https://doi.org/10.48550/arXiv.1801.01596

32. J. Wu, X.Y. Chen, H. Zhang, L.D. Xiong, H. Lei, S.H. Deng, Hyperparameter optimization for machine learning models based on Bayesian optimization, Journal of Electronic Science and Technology, 17(1) (2019) 26-40. https://doi.org/10.11989/JEST.1674-862X.80904120
33. B. Shahriari, A. Bouchard-Côté A, N. Freitas, Unbounded Bayesian optimization via regularization, in: Artificial intelligence and statistics, PMLR, 51, 2016 pp. 1168-1176. https://doi.org/10.48550/arXiv.1508.03666
34. P.I. Frazier, A tutorial on Bayesian optimization, arXiv preprint arXiv:1807.02811 2018. https://doi.org/10.48550/arXiv.1807.02811
35. Z. Wang, N. de Freitas, Theoretical analysis of Bayesian optimisation with unknown Gaussian process hyper-parameters, arXiv preprint arXiv 1406 (2014) 7758. https://doi.org/10.48550/arXiv.1406.7758
36. K. Swersky, D. Duvenaud, J. Snoek, F. Hutter, M.A. Osborne. Raiders of the lost architecture: Kernels for Bayesian optimization in conditional parameter spaces, arXiv preprint arXiv:1409 (2014).4011. https://doi.org/10.48550/arXiv.1409.4011
37. A. Nayebi, A. Munteanu, M. Poloczek, A framework for Bayesian optimization in embedded subspaces, in: International Conference on Machine Learning 2019 PMLR vol 97 pp. 4752-4761.
38. Z. Wang, C. Gehring, P. Kohli, S. Jegelka, Batched large-scale Bayesian optimization in high-dimensional spaces, in: International Conference on Artificial Intelligence and Statistics 2018 PMLR vol 84 pp. 745-754.
39. M.P. Ranjit G. Ganapathy, K. Sridhar, V. Arumugham,. Efficient deep learning hyperparameter tuning using cloud infrastructure: Intelligent distributed hyperparameter tuning with Bayesian optimization in the cloud, in: 2019 IEEE 12th international conference on cloud computing (CLOUD) 2019 IEEE pp. 520-522. 10.1109/CLOUD.2019.00097
40. J. Močkus, On Bayesian methods for seeking the extremum, in: Optimization Techniques IFIP Technical Conference Novosibirsk, vol 6, Springer Berlin Heidelberg, 1975, pp. 400-404. https://doi.org/10.1007/3-540-07165-2_55
41. A. Zilinskas, Optimization of one-dimensional multimodal functions, Journal of the Royal Statistical Society, Series C (Applied Statistics), 27(3) (1978) 367-75. https://doi.org/10.2307/2347182
42. C.D. Perttunen, B.E. Stuckman, The rank transformation applied to a multivariate method of global optimization, IEEE Transactions on Systems, Man, and Cybernetics, 20(5) (1990) 1216-20. 10.1109/21.59984
43. B.E. Stuckman, A global search method for optimizing nonlinear systems, IEEE Transactions on Systems, Man, and Cybernetics, 18(6) (1988) 965-77. 10.1109/21.23094
44. J.F. Elder, Global r/sup d/optimization when probes are expensive: the grope algorithm, in [Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, 1992, pp. 577-582. 10.1109/ICSMC.1992.271711
45. P.I. Frazier, J. Wang, Bayesian optimization for materials design, Information science for materials discovery and design, (2016) 45-75. https://doi.org/10.1007/978-3-319-23871-5_3
46. R. Garnett, M.A, Osborne, S.J. Roberts, Bayesian optimization for sensor set selection, in Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks, 2010 pp. 209-219. https://doi.org/10.1145/1791212.1791238
47. J. Gonzalvez, E. Lezmi, T. Roncalli, J. Xu, Financial applications of Gaussian processes and Bayesian optimization, arXiv preprint arXiv, (2019) 1903.04841. https://doi.org/10.48550/arXiv.1903.04841
48. R. Lorenz, L.E. Simmons, R.P. Monti, J.L. Arthur, S. Limal, I. Laakso, R. Leech, I.R. Violante, efficiently searching through large tACS parameter spaces using closed-loop Bayesian optimization, Brain stimulation, 12(6) (2019) 1484-9. https://doi.org/10.1016/j.brs.2019.07.003
49. M. Turchetta, A. Krause, S. Trimpe, Robust model-free reinforcement learning with multi-objective Bayesian optimization, in: 2020 IEEE international conference on robotics and automation (ICRA), IEEE, 2020, pp. 10702-10708. 10.1109/ICRA40945.2020.9197000
50. K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, E.P. Xing, Neural architecture search with bayesian optimisation and optimal transport, Advances in neural information processing systems, (2018) 31. https://doi.org/10.48550/arXiv.1802.07191
51. J. Wu, S. Toscano-Palmerin, P.I. Frazier, A.G. Wilson, Practical multi-fidelity Bayesian optimization for hyperparameter tuning, in Uncertainty in Artificial Intelligence, PMLR, 2020, pp. 788-798. https://doi.org/10.48550/arXiv.1903.04703
52. C.E. Rasmussen, Gaussian processes in machine learning, in: summer school on machine learning, Berlin, Heidelberg, Springer Berlin Heidelberg, 2003, pp. 63-71. https://doi.org/10.1007/978-3-540-28650-9_4
53. L.V. Jospin, H. Laga, F. Boussaid, W. Buntine, M. Bennamoun, Hands-on Bayesian neural networks—A tutorial for deep learning users, IEEE Computational Intelligence Magazine, 17(2) (2022) 29-48. 10.1109/MCI.2022.3155327
54. J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Prabhat, R. Adams, Scalable bayesian optimization using deep neural networks, in: international conference on machine learning, PMLR, 2015 vol 37 pp. 2171-2180. https://doi.org/10.48550/arXiv.1502.05700
55. A. Graves, Practical variational inference for neural networks, Advances in neural information processing systems, (2011), 24.

56. C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural network, in: International conference on machine learning, PMLR, 2015, pp. 1613-1622. https://doi.org/10.48550/arXiv.1505.05424
57. R.M. Neal, Bayesian learning for neural network, Springer Science & Business Media, 2012 Dec 6. https://doi.org/10.1007/978-1-4612-0745-0
58. D.R. Jones, M. Schonlau, W.J. Welch, Efficient global optimization of expensive black-box functions, Journal of Global optimization, 13 (1998) 455-92. https://doi.org/10.1023/A:1008306431147
59. J.R. Gardner, M.J. Kusner, Z.E. Xu, K.Q. Weinberger, J.P. Cunningham, Bayesian optimization with inequality constraints, in: ICML, 2014 vol 2014 pp. 937-945.
60. N. Srinivas, A. Krause, S.M. Kakade, M. Seeger, Gaussian process optimization in the bandit setting: No regret and experimental design, arXiv preprint arXiv, (2009) 0912.3995. https://doi.org/10.48550/arXiv.0912.3995
61. P. Hennig, C.J. Schuler, Entropy search for information-efficient global optimization, The Journal of Machine Learning Research, 13(1) (2012) 1809-37.
62. J.M. Hernández-Lobato, M.W. Hoffman, Z. Ghahramani, Predictive entropy search for efficient global optimization of black-box functions, Advances in neural information processing systems, arXiv preprint arXiv: 27 (2014) 406.2541. https://doi.org/10.48550/arXiv.1406.2541
63. J. Mockus J, The application of Bayesian methods for seeking the extremum, Towards global optimization, 2 (1998) 117.
64. N. De Freitas, A. Smola, M. Zoghi, Exponential regret bounds for Gaussian process bandits with deterministic observations, arXiv preprint arXiv, (2012) 1206.6457. https://doi.org/10.48550/arXiv.1206.6457
65. A.D. Bull, Convergence rates of efficient global optimization algorithms, Journal of Machine Learning Research, 12(10) (2011) 2879-2904. https://doi.org/10.48550/arXiv.1101.3501
66. R. Bardenet, M. Brendel, B. Kégl, M. Sebag, Collaborative hyperparameter tuning, in: International conference on machine learning, PMLR, vol 28(2) 2013 pp. 199-207.
67. N. Mahendran, Z. Wang, F. Hamze, N. De Freitas, Adaptive MCMC with Bayesian optimization, in: Artificial Intelligence and Statistics, PMLR, 2012 vol 22 pp. 751-760. https://doi.org/10.48550/arXiv.1110.6497
68. D.R. Jones, A taxonomy of global optimization methods based on response surfaces, Journal of global optimization, 21 (2001) 345-83. https://doi.org/10.1023/A:1012771025575
69. M.A. Osborne, R. Garnett, S.J. Roberts, Gaussian processes for global optimization, (2009).
70. J. T. Springenberg, A. Klein, S. Falkner, F. Hutter, Bayesian optimization with robust Bayesian neural networks, in: Proceeding of the 30th International Conference on neural information processing systems, vol 29 2016 pp. 4141 - 4149.
71. Z. Shangguan, L. Lin, W. Wu, B. Xu, Neural process for black-box model optimization under bayesian framework arXiv preprint arXiv:2104.02487 (2021). https://doi.org/10.48550/arXiv.2104.02487
72. B. Kerleguer, C. Cannamela, J. Garnier, A Bayesian neural network approach to multi-fidelity surrogate modeling, International Journal for Uncertainty Quantification, 14(1) (2024) 43-60. 10.1615/Int.J.UncertaintyQuantification.2023044584
73. F. Hutter, H.H. Hoos, K. Leyton-Brown, Sequential model-based optimization for general algorithm configuration, in: Learning and intelligent optimization: 5th international conference, LION 5, Rome, Italy, Springer Berlin Heidelberg, 2011 pp. 507-523. https://doi.org/10.1007/978-3-642-25566-3_40
74. J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, Advances in neural information processing systems, 24 (2011).
75. J. Bergstra, D. Yamins, D.D. Cox, Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms, in: Proceedings of the 12th Python in Science Conference, Vol 13 2013 Citeseer, 20. https://doi.org/10.25080/Majora-8b375195-003
76. S. Ganguly, S. Bhattacharjee, Restricted Bayesian Neural Network, arXiv preprint arXiv:2403.04810 (2024). https://doi.org/10.48550/arXiv.2403.04810
77. H. Liu, Y.S. Ong, X. Shen, J. Cai, When Gaussian process meets big data: A review of scalable GPs, IEEE transactions on neural networks and learning systems, 31(11) (2020) 4405-23. 10.1109/TNNLS.2019.2957109
78. D. Eriksson, M. Jankowiak, High-dimensional Bayesian optimization with sparse axis-aligned subspaces, in: Uncertainty in Artificial Intelligence 2021 PMLR vol 161 pp. 493-503.
79. M. Binois, N. Wycoff, A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization, ACM Transactions on Evolutionary Learning and Optimization, 2(2) (2022) 1-26. https://doi.org/10.1145/3545611
80. M. Saim, A. Feroui, Automated alzheimer's disease detection and diagnosis method based on Bayesian optimization and CNN-based pre-trained features, Multimedia Tools and Applications, 23 (2024) 1-41. https://doi.org/10.1007/s11042-024-20140-y
81. S.A. Kumar, S. Sasikala, Automated brain tumour detection and classification using deep features and Bayesian optimised classifiers, Current medical imaging, 20(1) (2024) E280323215032. https://doi.org/10.2174/1573405620666230328092218
82. Y. Xu, C.W. Farris, S.W. Anderson, X. Zhang, K.A. Brown, Bayesian reconstruction of magnetic resonance images using Gaussian processes, Scientific Reports, 13(1) (2023) 12527. https://doi.org/10.1038/s41598-023-39533-4

83. M. Kaya, Bayesian Optimization-based CNN Framework for Automated Detection of Brain Tumors, Balkan Journal of Electrical and Computer Engineering 11(4) (2023) 395-404. https://doi.org/10.17694/bajece.1346818
84. M. Ait Amou, K. Xia, S. Kamhi, M. Mouhafid, A novel MRI diagnosis method for brain tumor classification based on CNN and Bayesian Optimization, InHealthcare, MDPI, 10(3) (2022) 494. https://doi.org/10.3390/healthcare10030494
85. M. Alshayeji, J. Al-Buloushi, A. Ashkanani, S.E. Abed, Enhanced brain tumor classification using an optimized multi-layered convolutional neural network architecture, Multimedia Tools and Applications, 80(19) (2021) 28897-917. https://doi.org/10.1007/s11042-021-10927-8
86. M. Ahmed, H. Du, A. AlZoubi, ENAS-B: Combining ENAS with Bayesian Optimization for Automatic Design of Optimal CNN Architectures for Breast Lesion Classification from Ultrasound Images, Ultrasonic Imaging, 46(1) (2024) 17-28. https://doi.org/10.1177/01617346231208709
87. R. Sivanandan, Bayesian optimized novel CNN for improved diagnosis from ultrasound breast tumor images, Multimedia Tools and Applications, 82(15) (2023) 22815-33. https://doi.org/10.1007/s11042-023-14468-0
88. P. Thiagarajan, P. Khairnar, S. Ghosh, Explanation and use of uncertainty quantified by Bayesian neural network classifiers for breast histopathology images, IEEE transactions on medical imaging, 41(4) (2021) 815-25. 10.1109/TMI.2021.3123300
89. G. Atteia, N. Abdel Samee, E.S. El-Kenawy, A. Ibrahim, CNN-hyperparameter optimization for diabetic maculopathy diagnosis in optical coherence tomography and fundus retinography, Mathematics, 10(18) (2022) 3274. https://doi.org/10.3390/math10183274
90. M. Subramanian, M.S. Kumar, V.E. Sathishkumar, J. Prabhu, A. Karthick, S.S. Ganesh, M.A. Meem, Diagnosis of retinal diseases based on Bayesian optimization deep learning network using optical coherence tomography images, Computational Intelligence and Neuroscience, 2022(1) (2022) 8014979. https://doi.org/10.1155/2022/8014979
91. B.U. Maheswari, D. Sam, N. Mittal, A. Sharma, S. Kaur, S.S. Askar, M. Abouhawwash, Explainable deep-neural-network supported scheme for tuberculosis detection from chest radiographs, BMC Medical Imaging, 24(1) (2024) 32. https://doi.org/10.1186/s12880-024-01202-x
92. P. Ramadevi, R. Das, An extensive analysis of machine learning techniques with hyper-parameter tuning by Bayesian optimized SVM kernel for the detection of human lung disease, IEEE Access, 12 (2024) 97752 - 97770. 10.1109/ACCESS.2024.3422449
93. M. Uçar, Deep neural network model with Bayesian optimization for tuberculosis detection from X-Ray images, Multimedia Tools and Applications, 82(24) (2023) 36951-72. https://doi.org/10.1007/s11042-023-15212-4
94. M. Miranda, K. Valeriano, A detailed study on the choice of hyperparameters for transfer learning in Covid-19 image datasets using Bayesian optimization, International Journal of Advanced Computer Science and Applications, 12(4) (2021). 10.14569/IJACSA.2021.0120441
95. S.E. Arman, S. Rahman, S.A. Deowan, COVIDXception-Net: A Bayesian optimization-based deep learning approach to diagnose COVID-19 from X-Ray images, SN Computer Science, 3(2) (2022) 115. https://doi.org/10.1007/s42979-021-00980-3
96. M. Loey, S. El-Sappagh, S. Mirjalili, Bayesian-based optimized deep learning model to detect COVID-19 patients using chest X-ray image data, Computers in Biology and Medicine, 142 (2022) 105213. https://doi.org/10.1016/j.compbiomed.2022.105213
97. M. Canayaz, S. Şehribanoğlu, R. Özdağ, M. Demir, COVID-19 diagnosis on CT images with Bayes optimization-based deep neural networks and machine learning algorithms, Neural Computing and Applications, 34(7) (2022) 5349-65. https://doi.org/10.1007/s00521-022-07052-4
98. Ahmed Hamada Br35H: Brain Tumor Detection 2020. Available online: https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection
99. Cheng, Jun (2017): brain tumor dataset. figshare. Dataset. https://doi.org/10.6084/m9.figshare.1512427.v5
100. K. Devi, A.K. Sharma, Detection of Brain Tumor Using Novel Convolutional Neural Network with Magnetic Resonance Imaging, in: 2023 Seventh International Conference on Image Information Processing (ICIIP) 2023 IEEE pp. 57-62. 10.1109/ICIIP61524.2023.10537668
101. M.H. Arzmi, A. PP Abdul Majeed, R. Muazu Musa, M.A. Mohd Razman, H.S. Gan, I. Mohd Khairuddin, A.F. Ab. Nasir, The Classification of Lung Cancer: A DenseNet Feature-Based Transfer Learning Evaluation, in: Deep Learning in Cancer Diagnostics: A Feature-based Transfer Learning Evaluation 2023 Springer Nature Singapore pp. 21-26. https://doi.org/10.1007/978-981-19-8937-7_4
102. E. Irmak, Multi-classification of brain tumor MRI images using deep convolutional neural network with fully optimized framework. Iranian Journal of Science and Technology, Transactions of Electrical Engineering. 2021; 45(3):1015-36. https://doi.org/10.1007/s40998-021-00426-9
103. S. Srinivasan, D. Francis, S.K. Mathivanan, H. Rajadurai, et al. A hybrid deep CNN model for brain tumor image multi-classification. BMC Medical Imaging. 2024; 24(1):21. https://doi.org/10.1186/s12880-024-01195-7