Efficient Melanoma Detection Using Grey Wolf Optimized Convolutional Neural Networks

Raam Pravin
Department of Biostatistics
University of Michigan
Ann Arbor, Michigan
rpravin@umich.edu

***Abstract*—This paper presents an approach to improve the accuracy of convolutional neural networks (CNNs) beyond traditional regularization methods such as early stoppage and dropout rate adjustment. Using the Grey Wolf Optimizer (GWO) for hyperparameter optimization; we apply GWO to tune key hyperparameters of the pre-trained CNN model EfficientNet-B7 and assess its performance on image classification of melanomas. When used in addition to early stoppage, GWO demonstrates increased accuracy on a held-out test set, highlighting its potential for mitigating the effects of overfitting and efficient neural architecture search.**

***Index Terms*—Transfer Learning, Hyperparameter tuning, Grey Wolf Optimizer, CNN, EfficientNet, Melanoma Classification.**

I. Introduction

Melanoma is a form of skin cancer that occurs when pigment-producing cells called melanocytes become cancerous. It typically starts on sun-exposed skin, such as the arms, back, face, or legs. Although the exact causes of melanoma are not fully understood, exposure to ultraviolet light is a known risk factor. In the US, melanoma rates are rapidly increasing, particularly among individuals under 40 years of age, with more than 200,000 cases reported annually [6]. Due to its aggressive nature, early detection of melanoma is crucial for effective treatment. This paper investigates the application of the Grey Wolf Optimizer (GWO) [1] for hyperparameter tuning of convolutional neural networks (CNNs) for melanoma classification evaluating its effect in transfer learning on melanoma classification.

II. Background

A. Concolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for processing structured grid data, such as images. Unlike traditional fully connected networks, CNNs leverage local connectivity and parameter sharing through convolutional layers, enabling efficient and hierarchical feature extraction [2].

Typical CNN architecture consists of sequential layers, including convolutional layers that apply learnable filters to capture spatial features, followed by nonlinear activation functions that introduce non-linearity. Pooling layers are interspersed to reduce the spatial dimensions and control overfitting by retaining the most salient features. Deeper layers capture increasingly abstract representations, allowing the network to learn complex patterns such as edges, textures, and object parts. The final layers are usually fully connected and culminate in a softmax or sigmoid activation for classification tasks. CNNs are trained using gradient-based optimization(ex:Stochastic Gradient Descent) with backpropagation used to compute weight updates.

CNNs achieve high performance in a variety of vision-related tasks including image classification, object detection, and melanoma detection [5], [7], [8].

B. Hyperparameter Optimization in Deep Learning

The performance of CNNs heavily depends on hyperparameters such as learning rate, batch size, dropout rate, number of neurons per layer, and the number of filters or layers. Optimal selection is nontrivial since hyperparameters interact in complex ways and influence convergence and generalization; making tuning or attaining optimal hyperparameters extremely non-trivial and computationally intensive [3].

C. Grey Wolf Optimizer (GWO)

GWO is a swarm intelligence algorithm modeled after the social hierarchy and hunting strategies of grey wolves [1]. The three best solutions are classified as alpha, beta, and delta wolves, while the remaining are classified as omega wolves. Each wolf corresponds to a candidate solution vector in the optimization problem's search space. Positions are updated iteratively according to the leadership hierarchy.

Position Update

Let the position of a grey wolf at iteration t be:

$$\vec{X}(t) = [x_1(t), x_2(t), \ldots, x_D(t)]$$

The position is updated according to:

$$\vec{X}(t+1) = \frac{1}{3}\left(\vec{X}_1 + \vec{X}_2 + \vec{X}_3\right)$$

where

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \left|\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}(t)\right|$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \left|\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}(t)\right|$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \left|\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}(t)\right|$$

Coefficient Vectors

The coefficient vectors $\vec{A}$ and $\vec{C}$ are computed as:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}, \quad \vec{C} = 2 \cdot \vec{r}_2$$

where:

- $\vec{a}$ linearly decreases from 2 to 0 over the course of iterations, controlling the exploration–exploitation tradeoff. A larger value of $\vec{a}$ encourages exploration by allowing wolves to diverge from the leading solutions, while smaller values promote exploitation by focusing the search near the alpha, beta, and delta wolves. The value of $\vec{a}$ at iteration t is given by:

$$\vec{a}(t) = 2 - \frac{2t}{T}$$

where t is the current iteration number and T is the maximum number of iterations.
- $\vec{r}_1$ and $\vec{r}_2$ are random vectors where each component is drawn independently from a uniform distribution over the interval $[0, 1]$.

Exploration and Exploitation

The exploration and exploitation phases are controlled by the magnitude of $\vec{A}$:

- If $|\vec{A}| > 1$, the wolves diverge to explore the search space.
- If $|\vec{A}| < 1$, the wolves converge toward the best solutions.

This adaptive mechanism enables GWO to effectively balance exploration and exploitation throughout the optimization process, returning the alpha wolf, which represents the optimal parameter vector.

D. Early Stoppage

Early stoppage is a widely adopted regularization technique in deep learning aimed at preventing overfitting and reducing computational overhead. During training, the model's performance is continuously evaluated on a separate validation set. If the validation loss fails to improve for a specified number of consecutive epochs training is terminated early. A rising validation loss indicates the onset of overfitting, when the model begins to memorize training data rather than generalize to unseen data. By stopping training at the point of optimal validation performance early stopping enhances the model's generalization capability while avoiding unnecessary epochs that do not contribute to improved performance.

III. Methodology

A. Problem Formulation

Learning Problem Formulation: We treat melanoma detection as a binary classification task. The model receives an input of a processed image of a skin lesion and outputs a prediction of whether the lesion is malignant (1) or benign (0).

Dataset: The dataset used in this study consists of 11,879 images of skin lesions, each with a resolution of 224 × 224 pixels. Each image is labeled indicating whether the lesion is *malignant* or *benign*. Prior to model training, the dataset is pre-processed by resizing all images to 600 × 600 pixels and applying normalization techniques to standardize pixel intensity distributions.

B. Model Formulation

The model uses the EfficientNet-B7 architecture, a convolutional neural network known for balancing model efficiency and performance. The model's base architecture is configured for image classification with 64 hidden layers, a width coefficient of 2.0, and a depth coefficient of 3.1. It accepts RGB input images of size 600×600 pixels and uses the Swish activation function throughout the network and applies a dropout rate of 0.5 for regularization. The model architecture which outputted probabilities over 1,000 ImageNet classes using a softmax activation was adapted for binary classification by replacing the final fully connected layer with a single output neuron followed by the sigmoid activation function. The model is trained using a binary cross-entropy loss function and the Adam optimizer is used for optimization.

C. Data Pipeline and Model Setup

The data pipeline involves loading the dataset, performing pre-processing and then splitting the data into training and validation sets. Early Stoppage is employed to prevent the model from overfitting the training data. The produced model is tested on the held-out test set to evaluate its generalization ability.

D. GWO for Hyperparameter Tuning

The objective function for GWO was the validation accuracy. The hyperparameter search space included:

- Learning Rate: [1e-5, 1e-2]
- Dropout Rate: [0.2, 0.5]
- Number of Neurons in Second-to-Last Layer: {64, 512}

GWO was implemented using a population of 25 wolves over 50 iterations. Each wolf represented a candidate solution (hyperparameter vector), and fitness was evaluated by the validation accuracy.

IV. Results

Compared to the base EfficientNet-B7 model, which achieved an accuracy of 80.40%, the EfficientNet-B7 model with optimal hyperparameters tuned by the Grey Wolf Optimizer (as shown in Table 1) achieved a significantly higher accuracy of 85.90% and outperformed the base model across all evaluated metrics (see Table 2). As shown in Figure 1, early stopping was triggered at epoch 23, as the validation loss began to increase thereafter, indicating the onset of overfitting.

V. Discussion

GWO provides a robust mechanism for efficiently searching large hyperparameter spaces. Unlike a random search, GWO dynamically guides the search based on relative fitness and historical performance, reducing the number of unproductive evaluations.

Incorporating GWO into a deep learning workflow enables adaptive, intelligent hyperparameter tuning that scales well with increasing model complexity. Its population-based nature also allows for parallelization, improving computational efficiency.

VI. Conclusion

This study demonstrates that the Grey Wolf Optimizer is a viable alternative for manual or random search hyperparameter tuning in CNNs. It effectively improves model accuracy while minimizing computational cost. Although our final model slightly outperformed MobileNetV2's reported accuracy of 85% in the study by Moturi, Surapaneni, and Avanigadda [7], we were constrained by limited GPU compute hours and therefore could not apply additional data augmentation techniques such as rotations and vignettes, which may have increased our model's overall performance. Future work should include applying these additional processes prior to model training and investigating applying a scheduler to adjust learning rate after about 20 epochs to potentially mitigate the sudden increase in training loss.

Acknowledgments

This research was conducted as part of study in deep learning optimization. The author thanks Professors *Xian Zhang* and *Jeffery Reiger* for guidance.

Figures and Tables

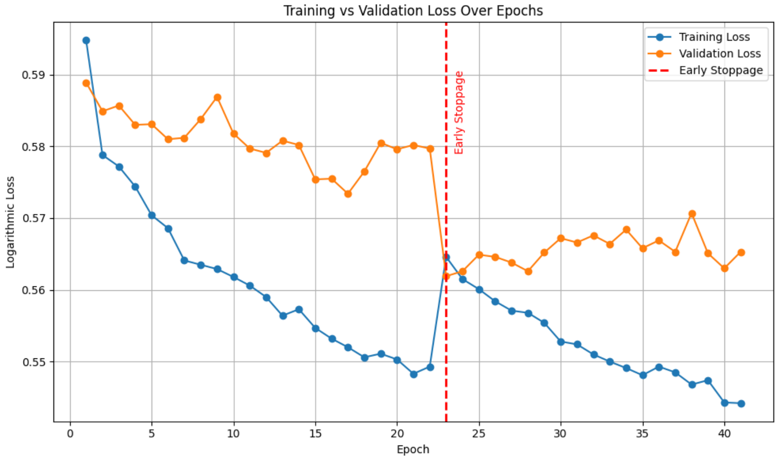
TABLE I
Optimal Hyper-parameters Found by Grey Wolf Optimizer

Learning Rate	**Dropout Rate**	**Final Classifier Neurons**
0.00184	0.3154	358

TABLE II
Performance Comparison Across Models

Model	**Accuracy**	**Sensitivity**	**Specificity**
EfficientNet-B7 base model	80.40%	66.80%	93.36%
EfficientNet-B7 post GWO	85.90%	78.31%	93.43%

Fig. 1. Training vs. Validation Accuracy and Loss over Epochs.



Note: Although the sudden increase in training loss and simultaneous drop in validation loss around epoch 23 is unexpected, it is followed by a plateau and slight upward trend in validation loss, suggesting that epoch 23 is an optimal early stopping point to prevent overfitting.

References

[1] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[2] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. ICML*, 2019.

[3] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *JMLR*, vol. 13, pp. 281–305, 2012.

[4] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated Machine Learning: Methods, Systems, Challenges*. Springer, 2019.

[5] R. Kaur, H. GholamHosseini, R. Sinha, and M. Lindén, "Melanoma classification using a novel deep convolutional neural network with dermoscopic images," *Sensors (Basel, Switzerland)*, vol. 22, no. 3, p. 1134, 2022.

[6] Mayo Clinic, "Melanoma," Mayo Foundation for Medical Education and Research, Dec. 30, 2023. [Online]

[7] D. Moturi, R. K. Surapaneni, and V. S. G. Avanigadda, "Developing an efficient method for melanoma detection using CNN techniques," *J. Egypt Natl. Canc. Inst.*, vol. 36, no. 6, 2024.

[8] R. Sabir and T. Mehmood, "Classification of melanoma skin cancer based on image data set using different neural networks," *Sci. Rep.*, vol. 14, p. 29704, 2024.

[9] N. Saleh, M. A. Hassan, and A. M. Salaheldin, "Skin cancer classification based on an optimized convolutional neural network and multicriteria decision-making," *Sci. Rep.*, vol. 14, p. 17323, 2024.