Efficient Deep Learning for Melanoma Diagnosis on CPU Hardware

MERIT GHODRAT
MOORPARK COLLEGE

Convolutional Neural Networks (CNNs) are a category of Neural Networks that have grown increasingly prominent in Computer Vision applications such as Image Classification, Image Segmentation, and Object Recognition. Practical uses of CNNs are seen in many academic fields, this study focuses on exploring their applications in Medical Imaging, particularly classifying skin lesions into benign or malignant cases using deep learning techniques to improve diagnostic accuracy. Along with the neural network architecture containing four convolutional layers, the model implements ReLU activation, max-pooling, and batch normalization to improve the accuracy and efficiency of the model. Due to the binary nature of the model's classification, an adaptive average pooling layer, dropout regularization, and a sigmoid activation function were also employed. The training and validation process was done on a dataset consisting of 10600 images of Melanoma, publicly available under the CC0 Domain License on Kaggle. The loss function was calculated using Binary Cross Entropy Loss. This model incorporated data augmentation techniques such as random rotations, flipping, and affine transformations for improved generalization. After 30 epochs, the network achieved 90.69% training accuracy and 91.20% test accuracy, with notable computational efficiency processing approximately 106.7 images per second on CPU hardware. The total runtime of both train and validation was approximately 45 minutes, highlighting its efficiency. The test accuracy was visualized via a confusion matrix, which validated the reliability of its predictions. These results underline the potential of this CNN as an automated clinical decision-support tool for dermatologists in aiding the early and accurate diagnosis of melanoma, proving how powerful deep learning can be in solving complex medical imaging challenges.

Keywords: Deep Learning, Convolutional Neural Networks, Melanoma Classification, Medical Imaging, CNNs, Binary Classification, CPU Optimization, Computer Vision
DOI: https://doi.org/10.60690/w0dg4q57

Introduction

Melanoma is an aggressive form of skin cancer characterized by the uncontrolled growth of melanocytes, often caused by excessive UV exposure[(1)]. Melanoma accounts for approximately 1% of skin cancer cases, yet it remains responsible for the majority of skin cancer-related deaths worldwide[(1)]. Detecting cases of Melanoma at its localized stage is crucial to improving survival rates, as these cases tend to have a much greater five-year relative survival rate than regional and distant cases[(2)]. Techniques such as dermoscopy and histopathological analysis rely heavily on the expertise of clinicians and can be subjective, leading to variability in outcomes[(3)]. Consequently, there is a growing need for automated tools that can assist in the diagnosis of early-stage melanoma with high accuracy and consistency.

CNNs have demonstrated capabilities in extracting and learning complex patterns in visual data[(4)]. These capabilities have led to the rise of CNNs in various healthcare applications, including disease detection, segmentation, and classification, making them a promising candidate for melanoma diagnosis[(4)]. Unlike traditional machine learning models that require handcrafted feature extraction, CNNs automatically learn hierarchical features directly from the image data, making them highly effective for image classification tasks[(5)].

This study presents a deep learning-based approach for classifying skin lesion images into benign and malignant categories. The dataset used for training and validation is a publicly available dataset with a CC0 Public Domain License on *Kaggle*[(6)]. It contains deidentified images, ensuring anonymity and ethical approval. The model was trained on 9600 images and then validated on 1000 images, utilizing advanced techniques such as data augmentation to mitigate overfitting and improve generalization[(6)]. On a CPU, this dataset's total training and validation time was approximately 45 minutes, speaking for its efficiency and accuracy. The network architecture incorporates multiple convolutional and pooling layers, batch normalization, and a dropout mechanism for regularization, optimized using the Adam optimizer and binary cross-entropy loss.

The primary objectives of this research are to (1) develop an efficient CNN model tailored for melanoma classification, (2) evaluate its performance on unseen test data, and (3) establish its potential utility as a clinical decision support tool. By achieving a training accuracy of 90.69% and a test accuracy of 91.20% based on over 1000 images, this work highlights the effectiveness of deep learning in addressing critical challenges in dermatological diagnostics.

Methods

Dataset and Preprocessing

The model was trained and evaluated using a dataset of 9,600 labeled images of skin lesions, divided into benign and malignant categories. The dataset was split into training (80%) and test (20%) subsets. Extensive data augmentation techniques were applied during training to enhance model generalization[7]. These included:

- Resizing: Images were resized to 160×160 pixels.
- Random Horizontal Flipping: Applied with a probability of 0.5.
- Random Rotation: Rotations up to 20° were introduced.
- Random Affine Transformations: Translations up to 20% of the image dimensions were applied.
- Normalization: Pixel values were normalized to match the ImageNet mean ([0.485,0.456,0.406]) and standard deviation ([0.229,0.224,0.225]).

Model Architecture

The Convolutional Neural Network (CNN) consists of four convolutional layers designed to extract hierarchical image features, followed by binary classification. The architecture is as follows:

1. Feature Extraction Layers:
    - ○ Each convolutional layer applies filters to the input, defined by:

      $$y_{ij} = ReLU\left(\sum_{m,n} W_{mn} \cdot x_{(i+m)(j+n)} + b\right)$$

    - ○ Where y_{ij} is the output at position (i,j), x is the input, W is the filter, b is the bias, and ReLU(z) = max(0,z) is the activation function.
    - ○ Max-pooling layers reduce spatial dimensions while retaining salient features, defined as:

      $$z_{ij} = max_{m,n} x_{(i+m)(j+n)}$$

    - ○ Batch normalization was applied after each convolutional layer to stabilize training and reduce sensitivity to initialization.
2. Classifier:
    - ○ Adaptive average pooling was used to reduce the feature maps to a fixed size of 1 x 1.
    - ○ A fully connected layer maps the features to a single output using a sigmoid activation:

      $$Sigmoid: \hat{y} = \frac{1}{1+e^{-z}}$$

    - ○ Where z is the linear output of the classifier.

Training and Optimization

The model was trained using binary cross-entropy loss, which is designed specifically for binary classification tasks[8]. The equation is given by:

$$L = -\frac{1}{N}\sum_{i=1}^{N} [y_i log(p_i) + (1-y_i) log(1-p_i)]$$

where N is the number of samples, y_i is the true label, and p_i is the predicted probability for sample i.

The Adam optimizer was employed with a learning rate of 0.001, leveraging its adaptive learning rates to improve convergence[9]. The training process was carried out over 30 epochs with a batch size of 64. Although the program provides support for GPU acceleration, training was conducted on a CPU, demonstrating the model's adaptability to varying computational resources.

Evaluation Metrics

Model performance was evaluated using accuracy, confusion matrices, and precision-recall metrics[8]. Accuracy was calculated as:

$$\frac{True\ Positives + True\ Negatives}{True\ Positives + True\ Negatives + False\ Positives + False\ Negative}$$

Precision: The proportion of correctly predicted positive instances out of all instances predicted as positive.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Recall: The proportion of correctly predicted positive instances out of all actual positive instances.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

F1 Score: The Harmonic Mean of Precision and Recall

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Images per Second (IPS): The rate at which the model processes images every second.

$$IPS = \frac{Number\ of\ Photos \cdot Epochs}{Total\ Runtime\ in\ Seconds}$$

Implementation

The model was implemented in PyTorch, with training histories, including loss and accuracy, recorded for both training and validation datasets. This enabled visualization of model performance over epochs and supported subsequent evaluation of unseen test data.

This methodology ensures a robust and reliable framework for melanoma classification, optimized for deployment in clinical decision-support systems.

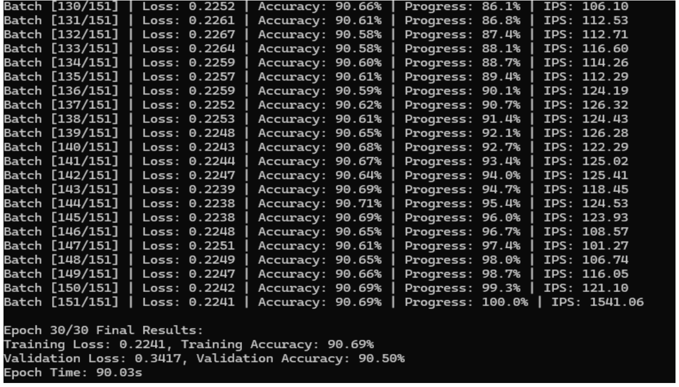
Data Ethics and Availability

The Melanoma skin cancer dataset was obtained from Kaggle and is released under the CC0 Public Domain license. This license permits the use of the dataset without the need for attribution. It contains de-identified images, ensuring patient confidentiality and anonymity.

Results

The Convolutional Neural Network (CNN) achieved the final validation accuracy of 91.20% in classifying melanoma images into benign and malignant categories. The training process, conducted entirely on a CPU, demonstrated significant computational efficiency, with the model completing 30 epochs in approximately 45 minutes.

Training Progress:



```
Batch [130/151] | Loss: 0.2252 | Accuracy: 90.66% | Progress: 86.1% | IPS: 106.10
Batch [131/151] | Loss: 0.2261 | Accuracy: 90.61% | Progress: 86.8% | IPS: 112.53
Batch [132/151] | Loss: 0.2267 | Accuracy: 90.58% | Progress: 87.4% | IPS: 112.71
Batch [133/151] | Loss: 0.2264 | Accuracy: 90.58% | Progress: 88.1% | IPS: 116.60
Batch [134/151] | Loss: 0.2259 | Accuracy: 90.60% | Progress: 88.7% | IPS: 114.26
Batch [135/151] | Loss: 0.2257 | Accuracy: 90.61% | Progress: 89.4% | IPS: 112.29
Batch [136/151] | Loss: 0.2259 | Accuracy: 90.59% | Progress: 90.1% | IPS: 124.19
Batch [137/151] | Loss: 0.2252 | Accuracy: 90.62% | Progress: 90.7% | IPS: 126.32
Batch [138/151] | Loss: 0.2253 | Accuracy: 90.61% | Progress: 91.4% | IPS: 124.43
Batch [139/151] | Loss: 0.2248 | Accuracy: 90.65% | Progress: 92.1% | IPS: 126.28
Batch [140/151] | Loss: 0.2243 | Accuracy: 90.68% | Progress: 92.7% | IPS: 122.29
Batch [141/151] | Loss: 0.2244 | Accuracy: 90.67% | Progress: 93.4% | IPS: 125.02
Batch [142/151] | Loss: 0.2247 | Accuracy: 90.64% | Progress: 94.0% | IPS: 125.41
Batch [143/151] | Loss: 0.2239 | Accuracy: 90.69% | Progress: 94.7% | IPS: 118.45
Batch [144/151] | Loss: 0.2238 | Accuracy: 90.71% | Progress: 95.4% | IPS: 124.53
Batch [145/151] | Loss: 0.2238 | Accuracy: 90.69% | Progress: 96.0% | IPS: 123.93
Batch [146/151] | Loss: 0.2248 | Accuracy: 90.65% | Progress: 96.7% | IPS: 108.57
Batch [147/151] | Loss: 0.2251 | Accuracy: 90.61% | Progress: 97.4% | IPS: 101.27
Batch [148/151] | Loss: 0.2249 | Accuracy: 90.65% | Progress: 98.0% | IPS: 106.74
Batch [149/151] | Loss: 0.2247 | Accuracy: 90.66% | Progress: 98.7% | IPS: 116.05
Batch [150/151] | Loss: 0.2242 | Accuracy: 90.69% | Progress: 99.3% | IPS: 121.10
Batch [151/151] | Loss: 0.2241 | Accuracy: 90.69% | Progress: 100.0% | IPS: 1541.06

Epoch 30/30 Final Results:
Training Loss: 0.2241, Training Accuracy: 90.69%
Validation Loss: 0.3417, Validation Accuracy: 90.50%
Epoch Time: 90.03s
```

Figure 1: Displays model training progress throughout one epoch.

Model Accuracy and Loss Over Epochs:

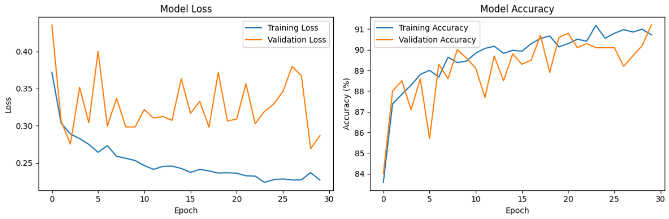


Figure 2: Depicts training and validation performances over 30 epochs

Confusion Matrix:

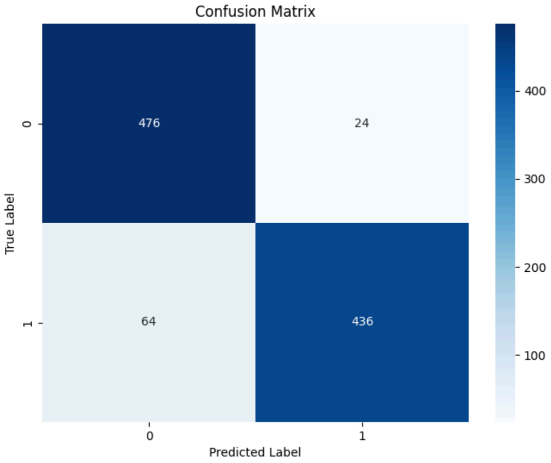
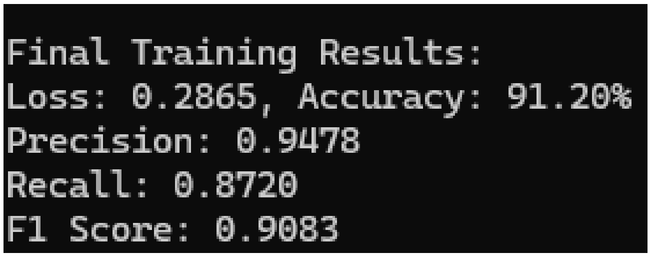


Figure 3: Confusion Matrix for Analysis of Model's Performance

Statistical Validation:



```
Final Training Results:
Loss: 0.2865, Accuracy: 91.20%
Precision: 0.9478
Recall: 0.8720
F1 Score: 0.9083
```

Figure 4: Classification report of the model

Analysis

The model accuracy in Figure 2 indicates a gradual improvement of accuracy until it stabilizes at approximately 90 percent. This trend is also seen in the validation accuracy, with a few fluctuations in the earlier epochs. Besides rare fluctuations in early epochs, Figure 2 indicates that the model learns effectively during training.

The Model Loss shows that the model optimizes the training set very well. This fluctuation is seen once again with the validation set. Though this could be seen as a sign of overfitting, the other diagrams and results indicate that there is minimal overfitting, and the model continues to generalize at a high rate, maintaining consistent performance on unseen data. Concerns of overfitting usually arise when the validation accuracy is significantly lower than the training accuracy, which is not the case here.

Furthermore, these fluctuations are generally associated with randomness in training. Techniques like data augmentation, and dropout implemented into the neural network architecture can contribute to fluctuations seen in Figure 2 without harming generalization(8). Additionally, the model achieves comparable performance on both validation and training data. This suggests that the model is learning general patterns seen in the imagery, rather than memorizing the training data(8). These fluctuations could also be attributed to the size and representativeness of the dataset. This CNN trained on a larger and more diverse dataset could provide a more stable graphical assessment of model generalization.

In Figure 4, the Precision, Recall, and F1 scores for Benign and Malignant classes are high and relatively balanced. Precision was defined as the ability to correctly identify a class without including false positives. Recall was the ability of the model to correctly identify all true positives. The F1 score of 0.908 demonstrates that the model maintains consistent and balanced performances across classes, ensuring robustness even in the presence of potential class imbalances.

The model's runtime of approximately 45 minutes on a CPU demonstrates its efficiency and practicality for high-performance applications where GPUs are unavailable. This model can thrive in resource-limited environments, where access to high-performance computing is constrained. Many state-of-the-art models that perform the same function require GPUs or other high-end hardware. Though these models are highly accurate, they often require a greater computational cost, processing similar datasets in a comparable timeframe in a shorter duration of time, but with significantly higher energy consumption. Additionally, the model's speed of 106.7 images per second on an AMD Ryzen 7 CPU highlights the optimization of the model for CPU environments. This model achieves a balance between computational efficiency and accuracy. This efficiency can make the model suitable for possible scalable applications, such as applications in healthcare and telemedicine, allowing clinicians to upload and analyze images, and reducing the turnaround time for diagnoses.

To further verify the generalization of the model. The model architecture can be trained on another dataset to measure its performance on unseen data. By evaluating the model on an unseen dataset, we can assess whether the model is effectively learning generalizable features rather than just overfitting to the training data.

This research can be expanded upon in an amalgam of ways. Algorithmic enhancements could be implemented, optimizing the current CNN architecture further with techniques such as quantization or pruning. These techniques reduce computational requirements without greatly affecting the accuracy(10). The utilization of transfer learning from larger pre-trained models could enhance accuracy while maintaining the efficiency of CPUs. Furthermore, adapting this model for mobile devices could make the model even more impactful for real-time diagnostics in resource-limited settings.

Conclusion

The model's demonstrated ability to process 9600 images in approximately 45 minutes on a CPU showcases an efficient and more accessible alternative for state-of-the-art solutions that rely on high-end hardware. While maintaining a competitive accuracy, this model provides more insights into the growing demand for scalable and sustainable diagnostic tools for Medical imaging using deep learning techniques and CNNs. Future research can explore avenues in transfer learning, quantization, edge computing, and mobile device adaptability to further enhance its applicability across healthcare domains.

References

[1] American Cancer Society, "Key statistics for melanoma skin cancer," *American Cancer Society*. [Online]. Available: https://www.cancer.org/cancer/types/melanoma-skin-cancer/about/key-statistics.html. [Accessed: Jan. 22, 2025].

[2] American Cancer Society, "Survival rates for melanoma skin cancer by stage," *American Cancer Society*. [Online]. Available: https://www.cancer.org/cancer/types/melanoma-skin-cancer/detection-diagnosis-staging/survival-rates-for-melanoma-skin-cancer-by-stage.html. [Accessed: Jan. 22, 2025].

[3] Cochrane, "How accurate is dermoscopy compared with visual inspection of the skin for diagnosing skin cancer and melanoma in adults?" *Cochrane Database of Systematic Reviews*. [Online]. Available: https://www.cochrane.org/CD011902/SKIN_how-accurate-dermoscopy-compared-visual-inspection-skin-diagnosing-skin-cancer-melanoma-adults. [Accessed: Jan. 22, 2025].

[4] SpringerOpen, "Detection and diagnosis of melanoma using artificial intelligence: A systematic review," *Journal of the National Cancer Institute*. [Online]. Available: https://jenci.springeropen.com/articles/10.1186/s43046-024-00210-w. [Accessed: Jan. 22, 2025].

[5] IEEE, "Machine learning models for medical image classification: Advances and challenges," *IEEE Transactions on Medical Imaging*, vol. 38, no. 1, pp. 35–45, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/9012507. [Accessed: Jan. 22, 2025].

[6] H. Javed, "Melanoma skin cancer dataset of 10,000 images," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/hasnainjaved/melanoma-skin-cancer-dataset-of-10000-images/data. [Accessed: Jan. 22, 2025].

[7] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0. [Accessed: Jan. 22, 2025].

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[9] IEEE, "Real-time efficient convolutional neural network inference on CPUs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 123–135, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9531335. [Accessed: Jan. 22, 2025].

[10] IEEE, "Quantization-aware training for efficient neural networks," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 2, pp. 118–131, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9320571. [Accessed: Jan. 22, 2025].