Classification of Carcinoma in Images using EfficientNet

Jorge Enriquez
Department of Computer Science
California State Univeristy
Northridge, CA, USA
jorge.enriquez.571@my.csun.edu

Abhishek Verma
Department of Computer Science
California State University
Northridge, CA, USA
abhishek.verma@csun.edu

***Abstract*—Early detection of carcinoma has the potential to improve patient care and thereby save the life of a patient. The use of deep neural nets to automatically detect carcinoma in MRI images can speed up and help with early diagnosis. However, the application of deep learning to analyze carcinoma images remains a challenging task due to inherent complexities present in a MRI image such as fine-grained features, varying resolution, image quality, slices, occlusion, and lack of available data to sufficiently train the model. Recent advancements in CNN architecture, such as variants of EfficientNet, offer new opportunities to improve carcinoma detection. We train a compact deep learning model named EfficientNetV2-M to classify four stages of carcinoma in images. Results show our carcinoma classification model achieves an overall high validation accuracy of 97%. The validation accuracy of four individual stages lies between 95.7% and 99.1%. Our research underscores the delicate balance between optimizing model performance and mitigating overfitting.**

***Index Terms*—carcinoma detection, DICOM, EfficientNet, deep learning, medical imaging**

I. Introduction

The integration of machine learning (ML) into medical practices has significantly advanced diagnostics and patient care, particularly in the fields of medical imaging and disease detection. These innovations have enhanced processes, improved accuracy, and enabled earlier detection of conditions that were previously challenging to diagnose. However, the application of ML to medical imaging, particularly in analyzing carcinoma images, remains a complex task. Factors such as tumor heterogeneity, variations in imaging protocols, and inconsistencies in data annotation contribute to these challenges, limiting the generalization and reliability of ML models [1].

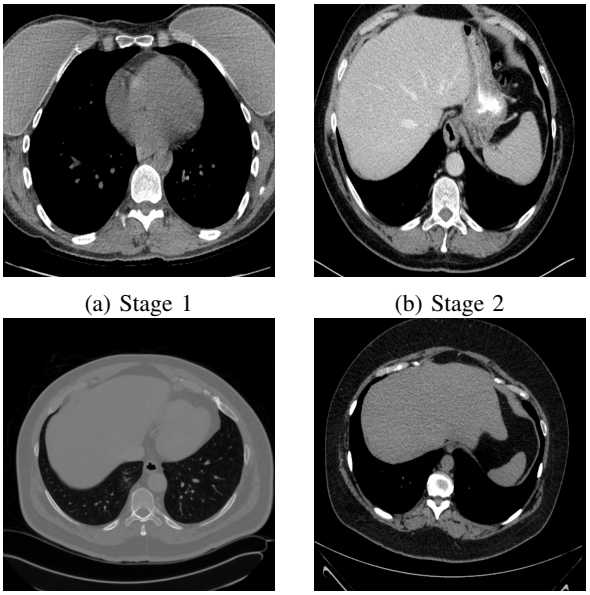
Deep Learning (DL), a specialized subset of ML, has demonstrated considerable potential in overcoming these complexities by learning hierarchical feature representations directly from raw data [2]. Among DL methodologies, Convolutional Neural Networks (CNNs) have emerged as a cornerstone in image analysis, excelling in tasks such as object detection, image classification, and segmentation. However, achieving robust and reliable performance in carcinoma image analysis remains hindered by issues such as overfitting due to limited dataset sizes and the inherent complexity of histopathological images [3].

While existing research has demonstrated the potential of CNNs for medical image classification, many studies rely heavily on large, curated datasets that may not reflect the diversity and variability of real-world medical data. This limitation underscores a critical gap in ensuring the generalizability of these models across diverse patient populations and imaging conditions. Moreover, practical considerations, such as computational efficiency and the interpretability of model predictions, remain significant barriers to the widespread clinical adoption of deep learning systems. By employing the EfficientNetV2-M model [4] and emphasizing advanced augmentation techniques, this study not only seeks to improve classification performance but also aims to address these broader challenges. The outcomes of this work could provide valuable insights for developing scalable and interpretable deep learning solutions that are both clinically applicable and resource-efficient, setting a foundation for future advancements in automated carcinoma detection.

Recent advancements in CNN architectures, such as the EfficientNet [5] series, offer new opportunities to improve these challenges. These architectures prioritize efficient scaling and parameter optimization, offering a pathway to achieving higher diagnostic accuracy while utilizing fewer computational resources. This study builds on this progress by evaluating the state-of-the-art EfficientNetV2-M [4] model's performance in the context of carcinoma image analysis. The study aims to address longstanding challenges by leveraging the strengths of this architecture while proposing enhancements to mitigate issues like data scarcity and overfitting.

To enhance the model's effectiveness, this research prioritizes the expansion of dataset size and diversity through advanced augmentation techniques, allowing the model to generalize effectively across varied cases. Furthermore, the implementation of strategies such as regularization techniques and hyperparameter tuning ensures consistent performance on unseen data. By advancing prior research, this study seeks to contribute to the development of reliable and efficient methodologies for carcinoma detection, with the ultimate goal of improving patient outcomes and streamlining diagnostic workflows.

In section II we describe previous research on carcinoma classification, relevant deep learning methodologies and



(a) Stage 1 (b) Stage 2

(c) Stage 3 (d) Stage 4

Fig. 1: Sample images from four stages of the Carcinoma image dataset [6].

its applications to disease classification in medical imaging. In section III Carcinoma image dataset is discussed including the unique modeling challenges of this dataset. Section IV presents the pipeline of our methodology that uses EfficientNetV2-M. Section V discusses the hardware and software environment, performance metrics, and training parameters. Section VI discusses the results of deep learning methodology in terms of training and validation performance, confusion matrix, and model performance at several different epochs of training. Section VII provides a final perspective on success of research, challenges encountered and sets up directions for future work.

II. Related Work

The classification of medical images has been extensively explored, with various studies employing convolutional neural networks (CNNs) for this purpose. Pham et al. [7] proposed the DICOM Imaging Router, an open deep learning framework that employs CNNs to classify DICOM X-ray images into anatomical categories, including abdominal, adult chest, pediatric chest, spine, and others. This approach addresses the challenge of non-standardized imaging data in clinical practice, facilitating the deployment of artificial intelligence solutions for medical image analysis.

In the realm of disease detection, Reshi et al. [8] developed an efficient CNN model for detecting COVID-19 using X-ray images. Their model exhibited high accuracy in identifying COVID-19 cases, highlighting the potential of CNNs in diagnosing infectious diseases through radiographic imaging.

Beyond X-rays, CNNs have also been utilized for MRI data analysis. Helm et al. [9] introduced an automated approach using a 3D DenseNet-121 model to classify MRI sequences of the chest, abdomen, and pelvis. Their model achieved an impressive F1 score of 99.5% in distinguishing sequences in MRI, demonstrating the effectiveness of deep learning in managing complex imaging modalities.

Focusing on lung cancer, Binani and Satapathy [10] used machine learning for the automated classification of pathological types of lung cancer. Their study aimed to improve diagnostic accuracy and efficiency, thereby reducing the workload of radiologists.

Mandava et al. (2022) [11] explored machine learning techniques to classify cancer patients based on gene mutations, achieving a 67% accuracy with a stacking classifier combined with TF-IDF for feature extraction. Their study highlights the utility of NLP for clinical data preprocessing and encoding strategies like one-hot encoding. While their work provided a foundation, our research extends these efforts by employing [e.g., a deep learning-based approach, larger datasets, advanced architectures like EfficientNetV2-M, etc.] to improve classification accuracy and scalability.

Authors in [12] focus on building a compact CNN model based on mobile net to recognize skin cancer in real time from images, compact CNN models are useful in deploying in resource constrained environments such as a mobile phone. Vo and Verma [13] propose two deep neural nets for classification of diabetic retinopathy from retina images leveraging fine grained texture information.

These studies collectively demonstrate the versatility and efficacy of CNN based models in medical image classification across various imaging modalities and disease conditions. Our research builds upon this foundation by focusing on state-of-the-art CNN models to enhance diagnostic capabilities in medical imaging.

III. Description of Carcinoma Dataset

The Carcinoma image dataset [6] used in this research centers on the adrenal gland, with a particular emphasis on Adrenocortical Carcinoma (ACC). ACC is a rare but aggressive malignancy originating from the adrenal cortex, the outer layer of the adrenal gland, which is responsible for producing steroid hormones. This type of carcinoma is known for its diverse clinical manifestations due to the excessive hormone secretion it often causes, including cortisol, androgens, estrogens, or aldosterone. Given its rarity and the challenges associated with early detection, ACC represents a critical area for research and advanced diagnostic methodologies. The dataset is designed to facilitate studies in the accurate identification, classification, and potential prognostic modeling of ACC through imaging techniques, thereby contributing to improved clinical outcomes.

The Carcinoma image dataset [6] contains 18,204 DICOM (DCM) files, categorized into four stages numbered 1-4, each one showing advanced stage of carcinoma in that order. The dataset also includes a CSV manifest file that provides detailed information about each image, including patient IDs, cancer stage, and study descriptions. Figure 1 shows sample images from four stages of the Carcinoma image dataset.

TABLE I: Distribution of Images in the Carcinoma Dataset

Stage 1	Stage 2	Stage 3	Stage 4
1,180	5,456	7,475	4,093

A. DICOM

Digital Imaging and Communications in Medicine (DICOM) files are unique because they combine medical images and metadata within a single .dcm file [14]. To extract these images, we utilized the Pydicom library. The images were processed into 16-bit unsigned NumPy [15] integer arrays. The target categorization was achieved by extracting the patient's name from the DICOM file and matching it with entries in the manifest. From the manifest, the correct target was assigned and delivered as a tuple.

B. Image Quantity

The dataset displayed an uneven distribution of images across categories, presenting a challenge in creating an unbiased model. Table I show the distribution of images per stage in the Carcinoma dataset. As can be observed, stage 1 has 1,180 images, which is far fewer than in stages 2-4. To mitigate this issue, we applied sampling weights to emphasize the underrepresented stage 1 dataset. We calculated the proportion of images for each category relative to the total number of images and then took the inverse of these proportions to assign weights. These inverse proportions were applied in the weighting function to ensure that each category contributed more evenly during model training.

IV. Research Methodology

This paper aims to measure the effectiveness of deep learning models — specifically CNN — on medical data types. To this end, we consider an EfficientNet architecture known for their robust performance in image classification tasks. EfficientNetV2-M [4] was chosen due to its high performance on image classification using ImageNet [16], which offers a challenging benchmark to test modern deep learning models. In addition, EfficientNetV2-m has a low number of parameters: 55 million, which makes it quicker to train. We chose a pretrained model built using PyTorch [17], aiming to reduce training time without compromising accuracy. Using a pretrained model also allows us to more effectively incrementally train it on a smaller carcinoma dataset.

A. EfficientNet

EfficientNet, the foundational version of the EfficientNet family, employs a compound scaling method to balance network depth, width, and resolution effectively [5]. This approach optimizes model performance while maintaining parameter efficiency. The base EfficientNet-B0 achieves state-of-the-art accuracy on the ImageNet dataset while using significantly fewer parameters compared to contemporary models.

For instance, EfficientNet-B0 contains only 5.3 million parameters, requiring approximately 0.39 billion FLOPs per inference [18]. This compactness makes it particularly appealing for tasks where computational resources are limited. EfficientNet's scaling principle ensures that larger variants (e.g., B1 through B7) maintain a consistent balance of accuracy and efficiency, offering flexibility depending on the dataset size and computational constraints.

The EfficientNet series laid the groundwork for subsequent advancements in neural architecture, leading to the development of EfficientNetV2, which retains the scaling principles but introduces training-aware optimizations for improved performance and speed.

B. EfficientNetV2-M

The selected EfficientNetV2-M model is an advanced convolutional neural network (CNN) optimized for high accuracy, faster training, and enhanced parameter efficiency. EfficientNetV2 models leverage a blend of training-aware neural architecture search and scaling techniques to enhance both performance and efficiency.

In terms of performance, EfficientNetV2-M achieves a top-1 accuracy of approximately 85.1% on the ImageNet dataset. It has around 54 million parameters and requires about 24.7 billion FLOPs (floating-point operations) for inference. Compared to its predecessor, EfficientNetV1, EfficientNetV2-M offers faster training times. For instance, EfficientNetV2-M trains significantly faster than EfficientNetV1-B7 while achieving comparable or better accuracy.

C. Transfer Learning using Pre-Trained Weights

Transfer learning involves reusing a pre-trained model for a new problem. By utilizing the general knowledge, a model has gained from a large dataset like ImageNet, we can refine that learning through additional training on the Carcinoma image dataset.

This approach is widely used in deep learning as it allows deep neural networks to be trained with relatively small datasets. This is particularly beneficial in medical image classification, such as with the Carcinoma image dataset, where labeled data can be scarce to train a large and complex deep learning model such as EfficientNet-V2-M. Another advantage of using transfer learning is to save training time particularly when training large and complex models.

D. System Pipeline for Training and Validation on Carcinoma Dataset

Figure 2 shows the system pipeline for training and validation on the Carcinoma image dataset. We preprocess training images by applying resizing, augmentation, and normalization. Specifically, dataset imbalance is addressed using a weighted random sampler from PyTorch, ensuring that minority classes are adequately represented during training. The images are resized to a resolution of 512 x 512 pixels to maintain uniform input dimensions for the model. Augmentation techniques include random horizontal and vertical flips applied with a probability of 50%, and random rotations within a range of -30 to +30 degrees to enhance model robustness against positional

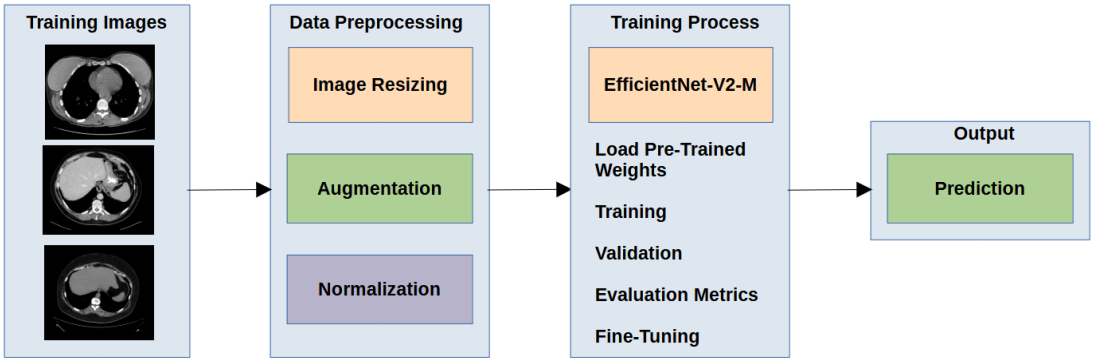


Fig. 2: System pipeline for training and validation on Carcinoma image dataset.

variance. Additionally, images are normalized using a mean value of 0.485 and a standard deviation of 0.229, aligning with the standard preprocessing practices for EfficientNet-based architectures.

The training process using EfficientNetV2-M consists of loading pretrained weights, performing validation at each epoch, and fine-tuning the model performance based on the evaluation metric. Finally, the model outputs a classification score on the validation image data.

V. Experiment Setup

A. Software and Hardware Setup

This model was set up and ran on Amazon Sage Maker via Amazon Web Services. In addition, a S3 Bucket was utilized to hold all the DICOM images needed. The model ran on a CPU instance with 16 virtual CPUs and 64 gibibytes (GiB) of memory [19]. We used Jupyter Notebook [20] as our IDE, the deep learning framework used was PyTorch version 2.2.2 and TorchVision 0.17.2. We use Pandas [21] and NumPy [15] for handling the data, and Matplotlib [22] to plot the results.

B. Metrics

1) Cross-Entropy Loss: Cross-Entropy loss was used as the main loss function to measure the discrepancy between the predicted probability distribution and the actual class labels. We define it as below:

$$l(x, y) = -\log\left(\frac{\exp x[y]}{\sum_{j=1}^{c} \exp x[j]}\right)$$

where x is the input tensor (logits) of size C (number of classes), y is the target class index, an integer in the range $[0, C-1]$, and $x[y]$ denotes the logit corresponding to the target class. This metric guided the optimization process during training.

2) Training and Validation Process: During the training process, both accuracy and loss were tracked for the training and validation splits. Accuracy was determined as the percentage of correctly classified images, while loss was calculated using the Cross-Entropy loss function. Tracking these metrics across epochs provided insights into model convergence and overfitting. Validation metrics, in particular, were used to evaluate generalization performance.

3) Confusion Matrix: To further assess the model's classification performance, a confusion matrix was created for the validation set. This matrix illustrates the number of true positive, true negative, false positive, and false negative predictions, providing a comprehensive evaluation of performance for each class. This analysis also revealed potential biases in the model toward specific classes and informed adjustments to the training process, if necessary.

C. Training Setup

1) Optimizer - Adam: The Adam optimizer was chosen since it is able to adapt the learning rate and handles the sparse gradients effectively. It adapts the learning rate based on individual parameters and maintains a running average of both the gradients and their squared values. The update rule for parameter θ_t at iteration t is:

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t$$

where $\hat{m}_t$ and $\hat{v}_t$ are estimates of the first and second moment respectively, which are corrected for bias. For numerical stability a small constant ϵ is used. The learning rate is represented as η. This method combines the advantages of both AdaGrad and RMSProp algorithms, providing an effective optimization strategy for a variety of real world problems [23].

2) Learning Rate Scheduler - Cyclical Learning Rate (CyclicLR): To enhance training efficiency and potentially escape local minima, a cyclical learning rate (CLR) policy was implemented using PyTorch's CyclicLR scheduler. The learning rate cyclically varies between a lower boundary (base learning rate) and an upper boundary (maximum learning rate) within each cycle. Values of the parameters are: Base learning rate: 0.01. Maximum learning rate: 0.001. The CyclicLR scheduler adjusts the learning rate after each batch, following a triangular policy by default. This approach can lead to improved convergence rates and model performance.

When combining the Adam optimizer with the CyclicLR scheduler, it's important to set the `cycle momentum` parameter to False in the CyclicLR scheduler. This is because the Adam optimizer does not utilize the momentum parameter in the same way as optimizers like stochastic gradient descent

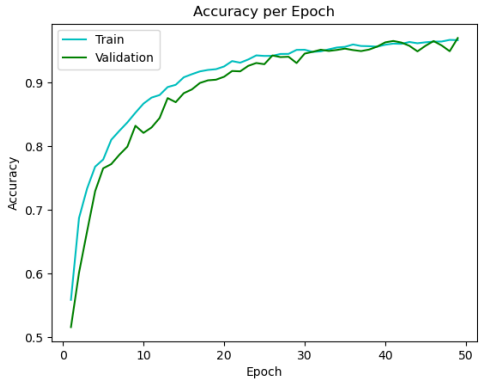


Fig. 3: Training and Validation accuracy at various epochs of EfficientNetV2-M on the Carcinoma image dataset.

(SGD). Failing to set `cycle momentum=False` may result in errors during training.

VI. Results and Discussion

A. Training and Validation Performance Across Epochs

The data was divided into 80%-20% splits respectively for training and validation. We trained the model over 50 epochs. The model accuracy at different epochs of EfficientNetV2-M are shown in Figure 3, while model loss at various epochs on the Carcinoma image dataset are displayed in Figure 4. As shown in both figures, the validation performance has converged within the 50 epochs. The training and validation performance stays close to each other from epoch 20 onwards highlighting the fact that parameter tuning, dropout, regularization, and data preprocessing have helped remove overfitting. A mostly smooth curve across various epochs highlights the consistency in the learning process. In Table II, we show the training and validation accuracy and loss at five different epochs of importance. We decided to choose model weights at epoch 49 since it has the minimal difference between training and validation performance in terms of loss and accuracy and has the highest accuracy. Therefore, our research underscores the delicate balance between optimizing model performance and mitigating overfitting.

TABLE II: Training and Validation Accuracy and Loss at Five different Epochs of Importance using EfficientNetV2-M on the Carcinoma Image Dataset

Epoch	Training Loss	Val. Loss	Training Acc.	Val. Acc.
26	0.15870	0.15609	0.94227	0.94285
31	0.13958	0.13615	0.94846	0.94858
39	0.11495	0.11841	0.95669	0.95682
46	0.10014	0.08945	0.96457	0.96548
49	0.09607	0.07847	0.96701	0.97029

B. Validation Performance Across Four Stages

The Figure 5 shows the confusion matrix at epoch 49. As can be observed, stages 1 and 4 have close to zero confusion with other stages in terms of classification performance of the model. There is a high accurate classification of images in stages 2 and 3 with very minimal confusion with other stages. The validation accuracy of each of the four stages of carcinoma at epoch 49 is shown in Table III. Results show our carcinoma classification model achieves an overall high validation accuracy of 97%. The validation accuracy of four individual stages lie between 95.7% and 99.1%, which is within a narrow range of 3.4% and highlights the consistency of the model performance across four stages of carcinoma.

Additionally, correctly classified carcinoma images for three distinct stages further illustrate the model's capacity to accurately differentiate between varying levels of progression. Figure 6a corresponds to Stage 1, where the model precisely identifies early-stage carcinoma with minimal structural abnormalities. Similarly, Figure 6b and Figure 6c show accurate classifications for Stage 3 and Stage 4, respectively, capturing advanced morphological changes characteristic of later stages.

VII. Conclusion and Future Work

Our study underscores the delicate balance between optimizing model performance and mitigating overfitting. Despite leveraging EfficientNetV2-M pretrained on the ImageNet

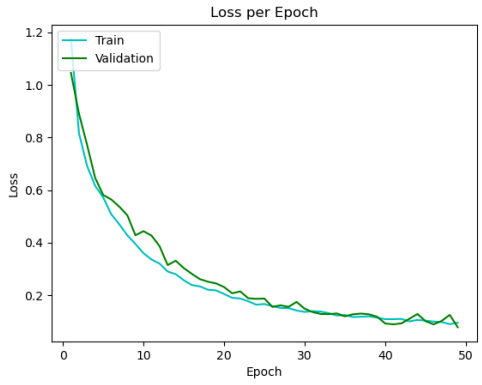


Fig. 4: Training and Validation loss at various epochs of EfficientNetV2-M on the Carcinoma image dataset.

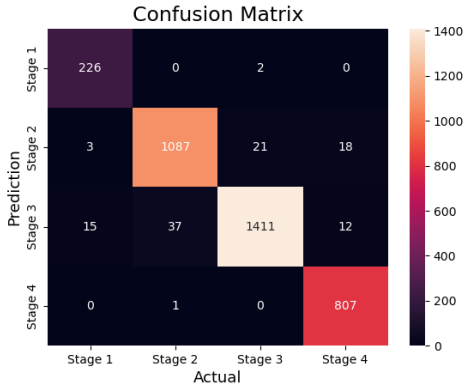
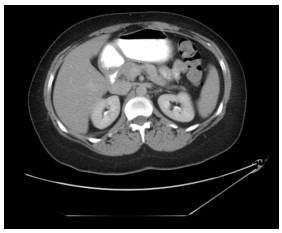


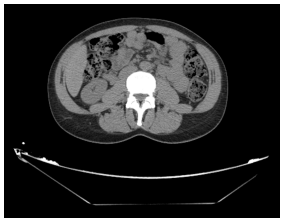
Fig. 5: Confusion matrix at epoch 49.

TABLE III: Validation Accuracy of Each of the Four Carcinoma Stages at Epoch 49

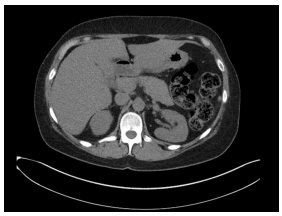
Carcinoma Stage	Stage 1	Stage 2	Stage 3	Stage 4
Validation Accuracy	0.991228	0.962799	0.95661	0.998762



(a) Stage 1



(b) Stage 3



(c) Stage 4

Fig. 6: Example images of correctly classified stage of carcinoma.

dataset, our model necessitated a diverse array of images to achieve robust generalization. This highlights the importance of extensive and varied medical image datasets in training deep neural nets for classification.

The accuracy of our model could be further enhanced by increasing the volume of patient data, which would provide a broader spectrum of examples for the model to learn from. Additionally, exploring alternative architectures, such as transformer based models, presents a promising avenue. Transformers have demonstrated significant success in capturing long-range dependencies and contextual information in imaging tasks. Integrating these models to medical imaging could potentially improve classification performance.

Another prospective direction involves developing low-latency models to facilitate real-time diagnostic applications, thereby enhancing clinical workflow efficiency. Moreover, adapting our model for object detection tasks could enable the precise localization and identification of carcinoma tumors within images, providing detailed spatial information crucial for treatment planning. While our current model offers a solid foundation, future work should focus on expanding the dataset, experimenting with advanced architectures like transformers, optimizing for low-latency scenarios, and extending capabilities to include object detection for comprehensive tumor analysis.

REFERENCES

[1] A. Kosvyra, D. T. Filos, D. T. Fotopoulos, O. Tsave, and et al., "Toward ensuring data quality in multi-site cancer imaging repositories," *Information*, vol. 15, no. 9, p. 533, 2024.

[2] P. Wang, X. Li, C. Yaras, Z. Zhu, and et al., "Understanding deep representation learning via layerwise feature compression and discrimination," 2024. [Online]. Available: https://arxiv.org/abs/2311.02960

[3] M. Unger and J. N. Kather, "Deep learning in cancer genomics and histopathology," *Genome Medicine*, vol. 16, no. 44, 2024.

[4] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," in *Proc. of the 38th Int. Conf. on Machine Learning*, 2021, pp. 10 096–10 106.

[5] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. of the 36th Int. Conf. on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97, 09–15 Jun 2019, pp. 6105–6114.

[6] The Cancer Imaging Archive, "Adrenal acc ki-67 segmentation collection," 2025, accessed: 2025-01-08. [Online]. Available: https://www.cancerimagingarchive.net/collection/adrenal-acc-ki67-seg/

[7] H. H. Pham, D. V. Do, and H. Q. Nguyen, "Dicom imaging router: An open deep learning framework for classification of body parts from dicom x-ray scans," *arXiv preprint arXiv:2108.06490*, 2021.

[8] A. Reshi, F. Rustam, A. Mehmood, A. Alhossan, and et al., "An efficient cnn model for covid-19 disease detection based on x-ray image classification," *Complexity*, vol. 2021, pp. 1–10, 2021.

[9] K. Helm, T. S. Mathai, B. Kim, P. Mukherjee, and et al., "Automated classification of body mri sequence type using convolutional neural networks," *arXiv preprint arXiv:2402.08098*, 2024.

[10] S. Binani and C. Satapathy, "Automated classification of pathological types of lung cancer - a machine learning approach," in *2023 Int. Conf. on Computational Intelligence for Information, Security and Communication Applications (CIISCA)*, 2023, pp. 178–183.

[11] J. S. Mandava, A. Verma, F. Kocaman, M. S. Nistor, and et al., "Machine learning for classification of cancer dataset for gene mutation based treatment," in *ITNG 2022 19th Int. Conf. on Information Technology-New Generations*. Springer, 2022, pp. 229–233.

[12] P. Ly, D. Bein, and A. Verma, "New compact deep learning model for skin cancer recognition," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conf. (UEMCON)*. IEEE, 2018, pp. 255–261.

[13] H. H. Vo and A. Verma, "New deep neural nets for fine-grained diabetic retinopathy recognition on hybrid color space," in *2016 IEEE Int. Symposium on Multimedia (ISM)*. IEEE, 2016, pp. 209–215.

[14] National Electrical Manufacturers Association, "DICOM Standard," https://www.dicomstandard.org/current, 2024, accessed: 2024-10-08.

[15] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, and et al., "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, and et al., "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conf. on Computer Vision and Pattern Recognition*. Ieee, 2009, pp. 248–255.

[17] A. Paszke, S. Gross, S. Chintala, G. Chanan, and et al., "Automatic differentiation in pytorch," 2017.

[18] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2020. [Online]. Available: https://arxiv.org/abs/1905.11946

[19] Amazon Web Services, "Instance types available for use with studio classic," 2024, accessed: 2024-11-22. [Online]. Available: https://docs.aws.amazon.com/sagemaker/latest/dg/notebooks-available-instance-types.html

[20] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, and et al., "Jupyter notebooks – a publishing format for reproducible computational workflows," in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt, Eds. IOS Press, 2016, pp. 87 – 90.

[21] W. McKinney, "Data structures for statistical computing in python," in *Proc. of the 9th Python in Science Conf.*, S. van der Walt and J. Millman, Eds., 2010, pp. 51 – 56.

[22] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. [Online]. Available: https://arxiv.org/abs/1412.6980