Optimizing Adrenocortical Carcinoma Cancer Diagnosis and Predictions through Transfer Learning and Hyperparameter-Optimized CNN Models

Sachchit Balamurugan[1]

Abstract

Adrenocortical carcinoma (ACC) is very rare and highly invasive, which makes it necessary to thoroughly diagnose and evaluate clinical and imaging assessment. A combination of MRI imaging and clinical information might help diagnose the disease at its initial stage. Deep learning algorithms have become increasingly popular in the medical field due to their ability to recognize complex patterns which creates new opportunities for improving medical imaging accuracy, especially by finding out anomalies and differentiating benign and malice lesions. Data is the basis of any machine learning, but medical image is rare and deep learning requires a large amount of training data. As a solution, this study aims to utilize transfer learning, which uses the knowledge gained by the model on one dataset to aid in classifying another one. This study evaluates the effectiveness of the transfer learning method along with several existing pre-trained architecture models like VGG16 and ResNet50, and others like MobileNetV2, ConvNeXtTiny and a custom CNN models designed with the assistance of hyperparameter optimization framework Optuna to distinguish the best hyperparameter. This research aims to contribute to the development of effective diagnostic tools for ACC by employing Models capable of comprehending complex textures in MRI images.

Keywords

Adrenocortical carcinoma (ACC), Deep learning, Transfer learning, Image classification, Hyperparameters, VGG16, ResNet18, ShuffleNet, ConvNeXtTiny

Introduction

Adrenocortical carcinoma (ACC) is a rare cancer with an annual incidence of 0.7-2 cases per 100,000 in the United States (1). This is an aggressive cancer that arises in the cortex of the adrenal gland with no prognosis. To determine the malignant status of adrenocortical cancers, careful evaluation of clinical and imaging features is required. Preoperatively, ACC diagnosis requires evaluation of clinical symptoms, hormonal markers, and imaging studies such as computed tomography, which can lead to earlier stage diagnosis. Thus, many features can be extracted from Magnetic resonance imaging (MRI) images for quantitative analysis. MRI is a commonly used noninvasive tool that assists in the diagnosis and determination of tumor morphology due to its merits of multi-parametric and multi-directional imaging with high soft tissue resolution (2).

All artificial intelligence (AI) applications and related deep learning models, clinical information, and picture investigation may have the most potential element for making a positive, enduring effect on human lives in a moderately short measure of time (4). There are significant perceived values of using AI solutions in healthcare at every stage of the clinical workflow. In radiology, this means improvements to the patient diagnostic pathway, from the appropriateness of imaging requests to how actionable findings in radiological reports are followed up (5). Deep learning, for example, can automatically acquire much richer information in a data-driven manner, and these features are typically more discriminative than traditional hand-crafted features. Second, deep learning models are typically trained from beginning to end; thus, feature extraction, feature selection, and classification can be carried out and gradually interactively improved through supervised learning. Therefore, deep learning is promising in a wide variety of applications including cancer detection, such as in brain tumor segmentation, tumor classification, and survival prediction (17). From 2021, the new EU Medical Device Regulations have been enforced, mandating deeper scrutiny of software as a medical device (SaMD) (5). Certification is given following how the software is used and applied within the clinical workflow. The majority of AI software in imaging is being certified as a decision-support tool, that is to say, it should not be used on its own for clinical or patient management.

Machine learning (ML) algorithms, especially deep learning, are powerful tools for MRI image analysis, capable of being trained on large amounts of data to identify patterns and structures necessary for detecting abnormalities in anatomy or focus on detection, which is likely to be invaluable in order to diagnose different types of cancer. The most commonly quoted applications of deep learning in medical imaging applications have been for object detection (eg, location of a lesion), object segmentation (eg, lesion contouring), and object classification (eg, malignant vs. benign lesion).

The object classification can be also for multi-classification such as the work of Ibrahim et al. (18) which was designed for detecting COVID-19, pneumonia, and lung cancer from chest X-ray and CT images. The model was the first attempt to classify the three chest diseases in a single model. It was important to correctly diagnose these diseases early to determine the proper treatment and apply isolation to COVID-19 patients to prevent the virus from spreading (18).

In a recent study, Moawad et al. (3) investigated the ability of ML combined with radiomic feature extraction to distinguish between benign, malignant, and nondetectable adrenal lesions in contrast-enhanced MRI scans intending to overcome complications caused by adrenal insufficiency address, unrelated -Adrenal presence of random lesions during clinical examination, which focused on sizes less than 4 cm and greater.

Kazemi A (6) compared two machine learning models, ResNet18 and ShuffleNet, focusing on their classification accuracy for different types of brain tumors. Both models achieved a high accuracy of 97.86% when trained using a specific method of convolutional basis. Despite the same accuracy, ShuffleNet exhibited faster processing speeds than ResNet18. Notably, high precision and recall scores are important in medical classification tasks. Specificity refers to the correct identification of persons, which is important for the patient to avoid unnecessary treatment. In contrast, recall reflects the ability of the model to recognize most events in a particular tumor, which is particularly important in detecting potentially malignant tumors and for false negatives to decrease and prevent delayed healing. Schirrmacher et al. present a comprehensive review of diverse applications of deep learning in molecular imaging of cancer (17). CNN-based models are most commonly used in these studies and have achieved excellent results. Despite the encouraging performance, further study on model optimization, public database establishment, and unsupervised learning, as well as the correlation between high-level features and clinical cancer characteristics, is required. To solve these problems, clinicians and engineers should collaborate by leveraging complementary expertise and advantages.

It's important to mention that Deep learning has revolutionized medical image analysis but comes with challenges and limitations. In the study of Zhou et al., for example, there were several limitations related to the specific application of renal tumor classification as well as general aspects of deep learning (19). the models could not detect specific subtypes of lesions that radiological experts might diagnose from images, such as clear-cell RCC and papillary RC.

In this study, we assess the effectiveness of several transfer learning models, encompassing pre-trained architectures such as VGG16, ResNet50, MobileNetV2, ConvNeXtTiny, and custom CNN models optimized using the Optuna hyperparameter tuning framework.

Materials and Methods

Dataset

As the adage goes, if AI is the engine that propels technology forward, data is the fuel that powers it. The algorithms used to develop predictive deep learning models require massive amounts of data to function. These data must be accessible, usable, valuable, and frequently validated (8). The dataset used in this study was publicly availabl e on various websites such as Pubmed, radiomedia, The Cancer Imaging Archive, PubMed Central, National Cancer Institute, International Cancer Imaging Society, and DataMed. The dataset includes 1580 adrenal gland MRI images in total, the image of the adrenal gland is from different angles as shown in Figure 1. The images were divided into 940 ACC images and 640 normal adrenal gland images. The images used in this study were collected at Stage II of ACC, according to the ENSAT staging system criteria (ENSAT stages ACC based on tumor size and spread). These images were obtained before chemotherapy treatment. The images are available in various sizes such as .jpeg, .jpg, .bmp, and .png files. The cancer existence labels were converted to numerical values by assigning a value of 0 to normal adrenal glands and a value of 1 to the presence of ACC. The dataset was split into 80% and 20% for training and testing, respectively. While our dataset does not contain detailed descriptors such as “incidental CT”, “preventive CT”, “CT after treatment”, “CT after radiation” or “CT stage=x” this limitation is not significantly detrimental to the performance of our model. The primary objective of our study is to differentiate between cancerous and non-cancerous adrenal gland images using transfer learning model, and custom optimized CNN models. The binary classification task focuses on identifying the presence or absence of adrenocortical carcinoma (ACC), which relies predominantly on the visual and morphological features captured in the MRI images. These features are sufficient for training robust deep learning models for the specific purpose of cancer detection. Furthermore, including detailed clinical descriptors could add complexity without substantially improving the models ability to distinguish between the two classes. Therefore, the absence of these descriptors does not undermine the effectiveness of our model in achieving its primary diagnostic goal.

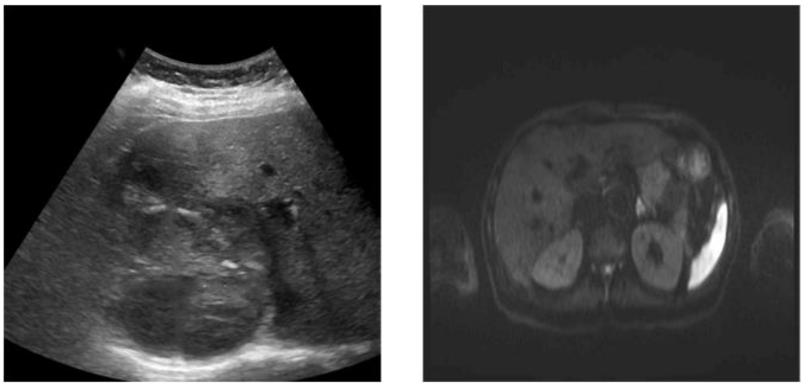

Figure 1: Sample of ACC image

Methodology

This study effectively utilized a variety of advanced techniques to enhance the performance of convolutional neural network models. Regularization Techniques such as dropout, batch normalization, and weight decay were skillfully employed to combat issues like overfitting and increase overall model robustness. In addition, data augmentation techniques were successfully implemented which is a data-space solution to the problem of small data sets (7). Several techniques that enhance the size and variety of training datasets can be implemented to expand the training dataset and improve the model's ability to make accurate predictions like rotation, width, and height shift, zoom, flip ...etc. Model assembling was also explored, leveraging the power of multiple models to boost classification accuracy. Furthermore, pre-trained models were fine-tuned on the target dataset, capitalizing on the wealth of knowledge gained through the training of models on vast datasets. Continuously, we monitored model performance and regularly evaluated their progress. They were adapting model architecture based on deployment requirements (e.g., computational resources, memory constraints) aimed to strike a balance between performance and resource efficiency. The iterative model development process included regular updates based on new insights and incoming data during the deployment phases. Figure 2 shows the sequential processing steps undertaken in this study.

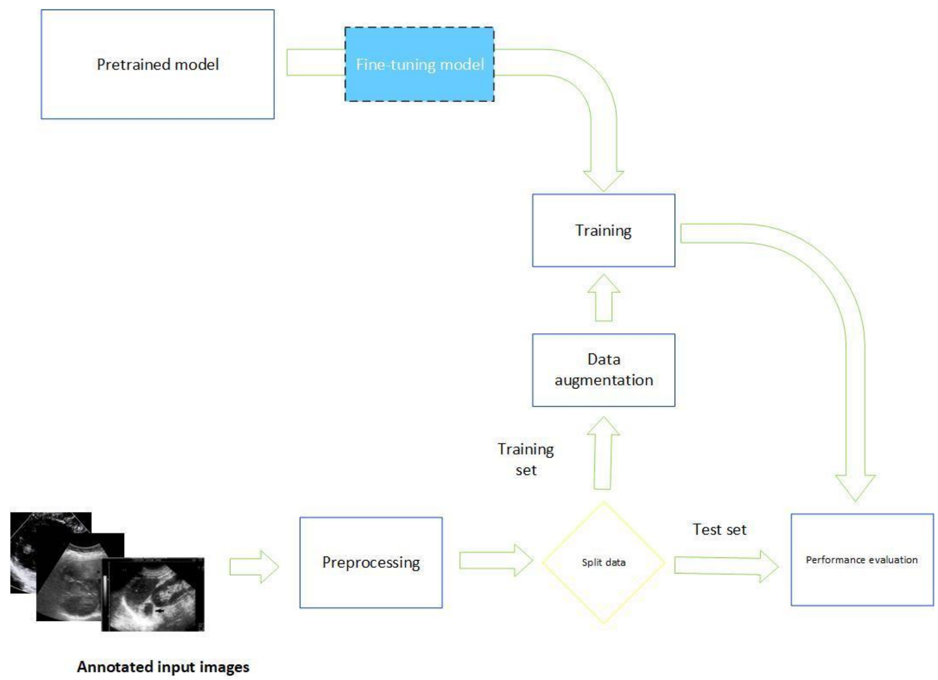

Figure 2: Workflow for the pre-trained mode

The implementation of a classification model involves three phases, training, testing, and validation. The training phase is the one in which the learning of the classification model itself takes place (7). To ensure the model's capacity for generalization, implying its effectiveness with new data, the training dataset needs to be both extensive in size and reflective of the broader population it aims to represent. Specifically, in our context, this pertains to the adrenal gland population, aligning with potential clinical applications. The testing phase is the one in which the model learned during the training phase is used or tested on new samples (7). To improve learning performance, and when the available samples are sufficient in number, a third phase called validation is introduced between the training and testing phases. The model parameters learned during the training phase are tuned and optimized in this phase to maximize a given metric (e.g., classification performance). The number of variables used, or their relative weight are examples of such parameters. It is important to note that the testing performance represents the model's final performance, i.e., the one that demonstrates the learned model's ability to work on the general population (7). for the performance evaluation of the ML models, metrics such as accuracy, precision, recall, and detection of potential overfitting are employed.

Pre-trained machine learning models

Transfer learning is an impactful strategy, especially in scenarios with limited datasets. It involves leveraging a network previously trained on a vast dataset like ImageNet, which boasts 1.4 million images spanning 1000 classes (12). This pre-trained network is repurposed to suit the specific task under study. The concept driving transfer learning is the belief that general features acquired from a sufficiently large dataset can be shared across different datasets, as showcased in Figure 2. This adaptability of learned features is a fundamental strength of deep learning, particularly advantageous for various domain-specific tasks grappling with constrained datasets. In this study, several CNN architectures, initially trained on expansive natural image datasets such as ImageNet, have been repurposed by fine-tuning pre-trained layers to address the challenge of sparse medical data (7).

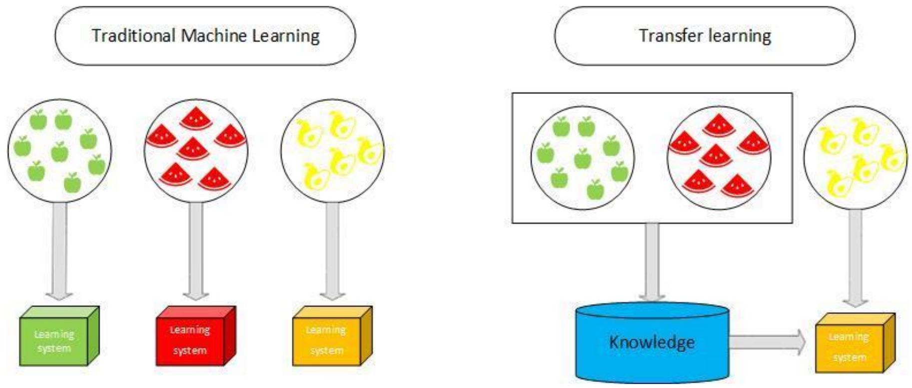

Figure 3: traditional Machine learning vs transfer learning

The pre-trained model has already learned general image features like edges and shapes. Consequently, when trained on the second dataset, it can focus on the more specific features of those images rather than relearning the basics (16). Transfer learning circumvents the necessity for extensive datasets and computational power by avoiding complete model training from scratch. By preserving the weights from the ImageNet training via freezing the convolutional base (as illustrated in Figure 4), this approach significantly reduces both training and testing time, as only the top classification layer necessitates training (6).

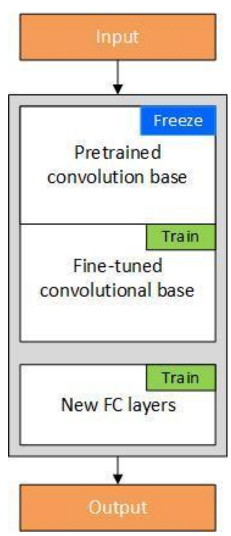

Figure 4: Fine-tuning method

In this work, we had 4 different pre-trained models:

VGG16

VGG is an acronym for Oxford University's Visual Geometric Group, and VGG-16 is a network with 16 layers proposed by the Visual Geometric Group (9). These 16 layers contain the trainable parameters, and there are additional layers, such as the Max pool layer, that do not contain any trainable parameters. In the 2014 ILSVRC challenge, VGG-16 was one of the top-performing architectures. We used transfer learning to fine-tune a previously trained VGG16 model, with a sequence of convolutional and pooling layers organized into blocks, which is known for its simplicity and effectiveness. VGG16 progressively extracts intricate features through convolutional layers and downsampling via max-pooling operations. The classifier segment, consisting of global average pooling and dense layers, refines these features into a final prediction. With over 15 million trainable parameters, VGG16 delivers impressive performance in image classification.

ResNet50

ResNet stands for "Residual Network with 50 layers," and it's a deep convolutional neural network architecture that consists of 50 layers (10). ResNet50 is a variant of the ResNet architecture, known for its exceptional performance in image classification tasks. It addresses the vanishing gradient problem commonly encountered in very deep neural networks by utilizing residual or skip connections (10). These connections enable the network to learn residual functions, making it easier to train extremely deep models while minimizing accuracy degradation as network depth increases.

The architecture is a deep convolutional neural network architecture that consists of 4 stages, each containing multiple convolutional blocks.

Stage 1: Starts with an initial convolutional layer followed by max-pooling. This stage involves multiple convolutional blocks that perform convolutions and identity mappings.

Stage 2: Comprises more convolutional blocks and works on higher-level feature mapping.

Stage 3: Continues feature extraction with convolutional blocks.

Stage 4: Carries out additional feature extraction with convolutional.

Every block in a stage is composed of convolutional layers, batch normalization layers operating as described in the provided equation (1), rectified linear unit activation functions (ReLU) producing outputs according to the pattern illustrated in equation (2), and shortcut connections (`Add`) designed to address vanishing gradient issues encountered during the training process.

$$output = \frac{input - mean}{\sqrt{variance} + epsilon} * scale + offset \quad (1)$$

Where:

- Mean is the mean of the batch.
- Variance is the variance of the batch.
- Epsilon is a small constant to prevent division
    - by zero.
- Scale and offset are learnable parameters

$$Activation\ (ReLU) = max\ (0, input) \quad (2)$$

- **MobileNetV2**

MobileNet is a convolutional neural network

(CNN) architecture designed primarily for mobile and edge devices with limited computational resources. It was developed by Google researchers and introduced as an improvement over the original MobileNet architecture, aiming to provide efficient yet powerful deep learning models for mobile applications (10).

MobileNetV2 is a refined neural network architecture tailored for streamlined and lightweight deep learning applications, specifically targeting mobile and embedded vision tasks. It represents an enhancement over the original MobileNet by bolstering efficiency and elevating overall performance. The architecture begins with an initial Convolutional Layer employing a standard convolution operation featuring 32 filters and a 3x3 kernel. Subsequently, it integrates inverted Residual Blocks followed by an expansion Layer that augments channel numbers through a 1x1 convolution. The Depthwise Convolution step is pivotal, executing spatial filtering without substantially inflating the model's parameters, utilizing the equation (3).

$$output[i,j,n] = \sum_{l,m} depthwise\_kernel[l,m,n] * input|i+l, j+m, n] \quad (3)$$

A Projection Layer is then employed to reduce the channel count via a 1x1 convolution. Certain blocks integrate Skip Connections, facilitating seamless information flow. Bottleneck Layers are strategically incorporated to curtail computational complexity while upholding the model's expressive capacity. The Fully Convolutional Tail culminates in several layers executing global average pooling, diminishing spatial dimensions, and generating the conclusive features for classification. Ultimately, a dense layer is responsible for producing a singular output, serving the purpose of binary classification.

The key innovations in MobileNetV2 include linear bottlenecks between layers, shortcut connections, and using a lightweight block design to improve the network's efficiency.

- **ConvNeXtTiny**

ConvNeXtTiny is a compact convolutional neural network architecture designed for efficient image recognition tasks, especially on resource-constrained devices such as mobile phones or embedded systems. It is based on the ConvNeXt architecture, which aims to strike a balance between model efficiency and performance (11).

we've provided the layer-by-layer architecture description of a ConvNeXtTiny model, including different convolutional blocks, normalization layers, activations, and the flow of data through the network. It follows a pattern where it has multiple blocks of convolutional layers and normalization, utilizing depthwise convolutions and pointwise convolutions for efficient feature extraction.

We based on architecture, which is organized in stages, with each stage having several blocks of convolutional layers. The down sampling process reduces the spatial dimensions of the data while increasing the depth, capturing more complex features as the network progresses through the stages.

Initially, the pre-trained models' architecture, preloaded with ImageNet weights, is instantiated while excluding its fully connected layers and setting the input shape. Then, on top of the base, we add Global Average Pooling and Dense layers to tailor it for the classification task. These layers culminate in a final Dense layer that uses sigmoid activation to predict binary classification. Callbacks for learning rate scheduling, model checkpointing, and Tensor Board visualization are also set up to help with training and monitoring. Finally, each model is compiled with the Adam optimizer, employing binary cross-entropy loss and metrics such as accuracy, precision, and recall for evaluation.

Custom CNN model:

Deep learning models offer the opportunity to automatically extract imaging features to maximize model performance for the task at issue (7). Among different deep learning architectures, CNNs are the most used for medical image processing tasks (14).

These networks are characterized by the presence of convolutional layers between the layers of neurons, convolving an input image with a given kernel function. In CNNs, different convolution layers can be implemented according to the application purpose, since the weights of convolutional layers being learned during training can extract imaging features tailored to the investigated task. The convolution layer output equation is as follows:

$$\text{Output }[i,j,k] = \sum_{l,m,n} kernel[l,m,n,k] * input[i+l,j+m,n] + bias[k] \quad (4)$$

Where:

- $\text{Output}[i,j,k]$ is the value at position (i,j) in the output feature map for channel k.
- $kernel[l,m,n,k]$ is the weight at position (l,m) in the kernel for channel n and the output k.
- $input[i+l,j+m,n]$ is the value at position $(i+l,j+m)$ in the input feature map for channel n.
- $bias[k]$ is the bias term for output channel k.

In CNNs, unlike fully connected neural networks, identical kernel parameters are employed across the entire image. This approach significantly reduces the total number of trainable parameters, leading to enhanced training efficiency (7). Pooling layers are another key component of CNN architecture: they reduce feature map resolution to introduce translational invariance to minor image distortions. Moreover, the combination of convolutional and pooling layers allows for learning spatial hierarchies among feature patterns (12).

Optuna is an open-source hyperparameter optimization framework primarily used for machine learning. It's designed to automate the process of tuning the hyperparameters of machine learning models to improve their performance (13). thus, it facilitates efficient optimization of machine learning model hyperparameters by automatically searching for the best set of hyperparameters within a defined search space. It provides various optimization algorithms such as TPE (Tree-structured Parzen Estimator), random search, and grid search to efficiently explore the parameter space. It’s known for offering flexibility in defining the search space for hyperparameters and can optimize a wide range of parameters, including learning rates, activation functions, dropout rates, batch sizes, etc. The Optuna aims to minimize the validation loss of the CNN model by exploring various hyperparameters.

The custom CNN model, integrated with Optuna for hyperparameter optimization, undergoes a systematic search for optimal configuration to enhance its performance. Optuna explores a predefined search space, adjusting hyperparameters such as learning rate which is suggested in logarithmic space between 1 10^{-5} and 1 $+10^{-2}$, dropout rate, batch size, number of neurons in fully connected layers, kernel size for convolutional layers, activation function, and the number of layers. Through Bayesian Hyperparameter Optimization, Optuna iteratively evaluates these combinations, guiding the model towards the set goal. The best-performing model, determined by the optimization process, is then selected as the Optuna Winner Model. In Figure 5 above, the hyperparameter optimization Optuna is external to the model and the tuning is done before model training. The model will continue to run trials with various and different optimizations until 3 trials in a row exhibit the same or similar results. When the results are the same for 3 trials in a row, the program stops and finds the best results and saves the data in the plots.

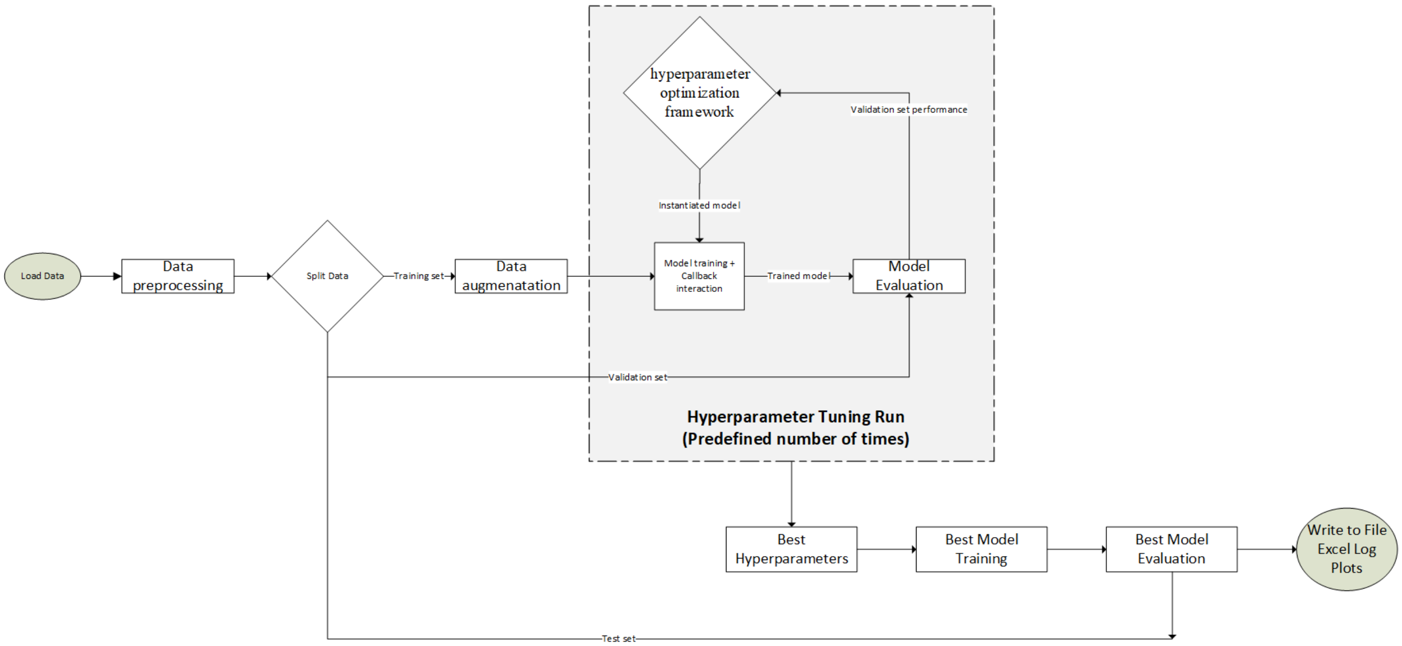

Figure 5: hyperparameter optimization Optuna integration

This approach aims to efficiently fine-tune the CNN architecture, maximizing its predictive capabilities for the given dataset and improving overall classification accuracy by iterating through different hyperparameter combinations, training and evaluating multiple models to find the set of parameters that minimizes the validation loss, resulting in an optimized CNN model architecture for the given task.

Results

Evaluation Metrics

For the performance evaluation of this experiment, the accuracy, Precision, and Recall. True positive (TP) is the correct classification of the positive class and true negative (TN) is the correct classification of the negative class. False positive (FP) is the incorrect prediction of the positives class and false negative (FN) is the incorrect prediction of the negatives class (15).

Precision checks how precise the model works by checking the correct true positives from the predicted ones (5).

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Recall calculates how many actual true positives the model has captured, labeling them as positives (6).

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

Accuracy is the most important performance measure (7). Accuracy determines how many true positives TP, true negatives TN, false positives FP, and false negatives FN were correctly classified.

$$Accuracy = \frac{TP+TF}{TP+TN+FP+FN} \tag{7}$$

Experimental Analysis

The confusion matrix (Figure 6), a fundamental tool in evaluating classification model performance, showcases the distribution of predictions across classes. There are 500 instances correctly classified as positive (True Positives), signifying accurate identifications of the presence of the studied condition. Additionally, 100 instances are accurately classified as negative (True Negatives), representing correct identifications of the absence of the condition. However, the model also misclassified 10 instances as positive when they were negative (False Positives), potentially leading to false alarms. Furthermore, 5 instances were incorrectly classified as negative when they were positive (False Negatives), which could signify missed detections. These metrics serve as the groundwork for calculating crucial performance indicators like accuracy, precision, recall, and specificity, offering a comprehensive view of the model's efficacy in classifying different instances.

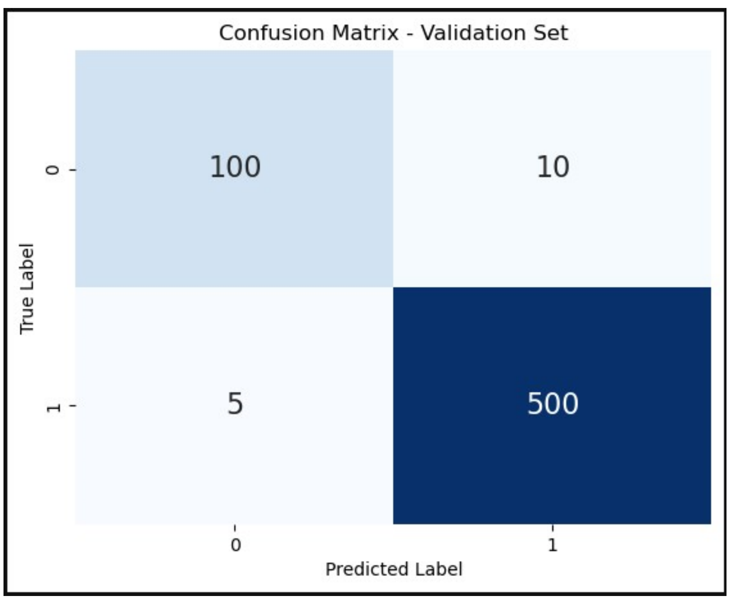

Figure 3: confusion matrix of the validation set

The results of all four pre-trained models and the Optuna-tuned model are shown in Table 1. In the evaluation of pre-trained architectures, VGG16 emerges with a moderate parameter count yet showcases a robust 92.13% accuracy in training. However, this model suffers from significant overfitting, evidenced by a notable drop in validation and test accuracy, reaching 87.16% and 85.94%, respectively. Contrarily, ResNet50, with a higher parameter count compared to VGG16, maintains a commendable 95.38% training accuracy while demonstrating better performance on validation and test datasets, exhibiting a moderate level of overfitting.

MobileNetV2 stands out with relatively fewer parameters but exceptional performance, boasting a 97.31% training accuracy. Impressively, it manages to sustain good accuracy on both validation (83.21%) and test datasets (90.63%) while showcasing minimal signs of overfitting. On the other hand, ConvNeXtTiny, despite a high parameter count and a decent 95.28% training accuracy, suffers from significant overfitting, resulting in a considerable decline in validation (76.03%) and test accuracy (92.19%).

The Optuna Winner Model indeed shows promise within the context of the study's findings. While it doesn't outperform the pre-trained architectures like MobileNetV2 or ResNet50 in terms of accuracy on validation and test datasets, it presents interesting aspects.

With a moderate parameter count, this model achieves a reasonable 90.05% training accuracy. Although its accuracy on validation (67.91%) and test datasets (76.56%) is comparatively lower, it demonstrates less severe overfitting than some of the trial models evaluated.

This indicates the potential for further optimization or refinement. Through additional hyperparameter tuning or architectural adjustments, there might be opportunities to enhance its performance and mitigate overfitting, thereby potentially bridging the gap between its current performance and that of the more successful pre-trained models.

Table 1: Evaluation of pre-trained models and custom model

Metric	**VGG16**	**ResNet50**	**MobileNetV2**	**ConvNeXtTiny**	**Optuna winner model**
Parameters	15,241,025	25,686,913	3,570,753	28,608,609	13,537,047
Train Loss	0.1892	0.1084	0.0732	0.1193	0.2228
Train Accuracy	0.9213	0.9538	0.9731	0.9528	0.9005
Train Precision	0.9094	0.9477	0.9678	0.9557	0.8723
Train Recall	0.9185	0.9509	0.9732	0.9397	0.9152
Validation Loss	0.3727	0.2973	0.3147	0.3306	0.4127
Validation Accuracy	0.8716	0.9128	0.906	0.8807	0.8165
Validation Precision	0.8273	0.8629	0.8321	0.7603	0.6791
Validation Recall	0.7109	0.8359	0.8516	0.8672	0.7109
Test Loss	0.2819	0.2427	0.2851	0.3114	0.395
Test Accuracy	0.9174	0.922	0.922	0.9083	0.8028
Test Precision	0.8593	0.8615	0.841	0.7973	0.6364
Test Recall	0.8594	0.875	0.9063	0.9219	0.7656

Discussion

In order to maximize the performance of our models, in the field of networks (CNN) we conducted several experiments as shown in Figure 7.

We noticed that certain trials, such as Trial_1, Trial_3, Trial_4, Trial_5 and others exhibited performance with no accuracy on validation and test datasets. This implies that there might be some issues, with the convergence of the model or the settings of the hyperparameters.

In another experiment called Trial 2 we introduced data augmentation to expand the training set and enhance our model's ability to generalize. In Trial 3 we focused on evaluating methods by using confusion matrices and ROC curves across trained models. This gave us insights into their performance differences. Based on our observations it became evident that most models faced challenges due to data quality and image related issues. However, MobileNetV2 stood out by achieving a test accuracy level of 73%.

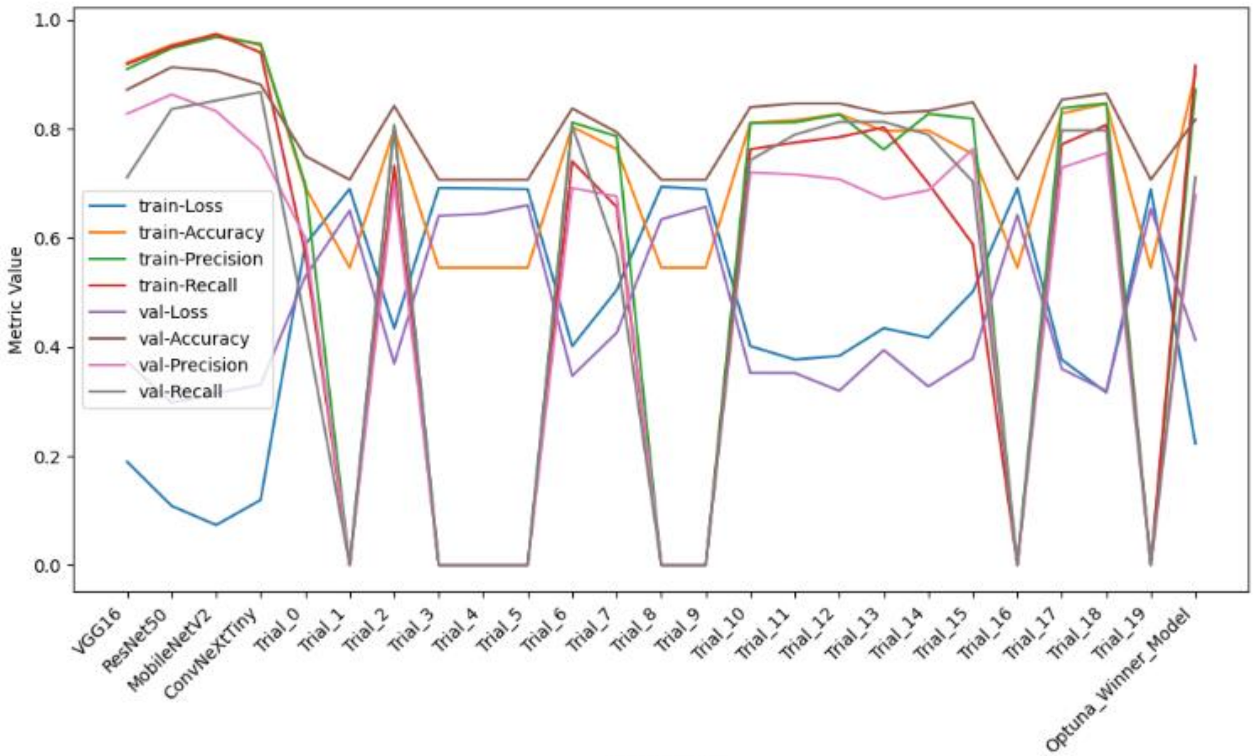

Figure 6: training and validation metrics

Throughout this experiment we faced difficulties where models tended to predict one class for input data. This emphasized the significance of having evaluation tools in place. In Trial 4 we made improvements by introducing performance comparison plots for both the training and validation datasets. Additionally, in the Optuna experiment we employed regularization techniques to address any overfitting problems.
While it is true that currently the Optuna Winner Model does not match up with trained architectures in terms of performance its moderate overfitting and decent accuracy indicate that there is potential for improvement. This makes it an intriguing option worth exploring further for investigation and enhancement. While this study did not use any cross-validation due to constraints like time and resources, since this model had high results, using cross-validation techniques would not significantly change the results of the model.

Conclusion

This study provides insights into the complexities involved in diagnosing ACC using learning models. It highlights the importance of optimizing existing architectures and the potential for further improvement by fine tuning hyperparameters. These discoveries open doors for exploration and enhancement in utilizing AI for prompt diagnosis, in the field of Adrenocortical carcinoma. Although specificities and sensitivities exceeding 97% are achievable with conventional methods such as washout times with iodinated contrast medium delayed CT scans (Korobkin et al., Boland et al.) and a combination of APW and HU values achieving over 93% sensitivity and specificity (Szolar et al.), deep learning models bring several additional advantages that justify their use. Deep learning models have the potential to detect cancers earlier by identifying subtle patterns and features in MRI images that may not be apparent to the human eye or detectable by conventional methods. These models can analyze large datasets efficiently, continuously learning and improving their accuracy over time. This capability is crucial for early diagnosis, which can significantly improve patient outcomes by initiating treatment at earlier stages. Our research focuses on MRI images, which provide superior soft tissue contrast and resolution compared to CT scans. By leveraging advanced deep learning techniques, we aim to enhance the early detection of adrenocortical carcinoma (ACC). The use of pre- trained models and custom CNN architectures, optimized with tools like Optuna, underscores the potential of AI to revolutionize medical imaging analysis. Despite the high performance of traditional methods, the ability of deep learning to detect cancers earlier and integrate various data types makes it a valuable area of research. We can see that the best performing model was MobileNetV2 with the highest accuracy out of all the pre-trained architectures, as well as the custom hyperparameter-optimized CNN model. It had a high accuracy showing that this model has the capability to predict ACC using MRI even classifying accidental detection. The pre-trained models performed better compared to the Hyperparameter optimized CNN model this is largely due to the fact that the pre-trained models already have learned to recognize shapes thus not needing as much data to correctly classify whether or not the MRI data was cancerous. Our approach employed a comprehensive analysis of the entire MRI dataset to isolate regions of interest (ROIs) and visualize feature relationships using advanced deep learning techniques. By leveraging pre-trained convolutional neural network (CNN) architectures like MobileNetV2, which are already trained on large datasets, our models required less data to accurately classify MRI images indicative of adrenocortical carcinoma (ACC). In addition to these pre-trained models, we developed custom CNN architectures optimized with hyperparameter tuning tools like Optuna, designed to capture specific features relevant to ACC diagnosis from MRI data. This methodology enabled the extraction of critical features such as density heterogeneity, spatial relationships, and entropy levels directly from the MRI images without predefined segmentation. The holistic analysis allowed the models to continuously learn and improve, achieving high accuracy in diagnosing ACC. By employing these advanced deep learning techniques and analyzing the full spectrum of MRI data, our study highlights the potential for AI to revolutionize medical imaging analysis, simplifying the diagnostic workflow and enhancing the model's ability to detect subtle and complex patterns indicative of malignancy. The findings underscore the significance of leveraging comprehensive image data and sophisticated machine learning models to improve early detection and diagnosis of adrenocortical carcinoma. Future research will focus on refining these models and exploring additional features to enhance diagnostic accuracy further.

References:

1. Libé, R., & Huillard, O. (2023). Adrenocortical carcinoma: Diagnosis, prognostic classification and treatment of localized and advanced disease. Cancer Treatment and Research Communications, 37. DOI: 10.1016/j.ctarc.2023.100759

2. X. Jiang et al. 2021. MRI Based Radiomics Approach With Deep Learning for Prediction of Vessel Invasion in Early-Stage Cervical Cancer. IEEE/ACM Transactions on Computational Biology and Bioinformatics. vol. 18, no. 3, pp. 995-1002, doi: 10.1109/TCBB.2019.2963867.

3. Moawad, A. W., Ahmed, A., Fuentes, D. T., Hazle, J. D., Habra, M. A., & Elsayes, K. M. (2021). Machine learning based texture analysis for differentiation of radiologically indeterminate small adrenal tumors on adrenal protocol CT scans. Abdominal Radiology, 46 (11), 5529-5538. DOI: 10.1007/s00261-021-03386-0

4. Suganyadevi, S., Seethalakshmi, V. & Balasamy, K. (2022). A review on deep learning in medical image analysis. Int J Multimed Info Retr, 11, 19–38. DOI: 10.1007/s13735-021-00218-1

5. Koh, DM., Papanikolaou, N., Bick, U. et al. (2022). Artificial intelligence and machine learning in cancer imaging. Commun Med, 2 , 133. DOI: 10.1038/s43856-022-00199-0

6. Kazemi, A. (2023). Classification of brain tumors from MRI images using deep transfer learning. Journal of High School Science.

7. Castiglioni, I., Rundo, L., Codari, M., Di Leo, G., Salvatore, C., Interlenghi, M., Gallivanone, F., Cozzi, A., D'Amico, N. C., & Sardanelli, F. (2021). AI applications to medical images: From machine learning to deep learning. Physica Medica, 83 , 9-24. DOI: 10.1016/j.ejmp.2021.02.006

8. Currie, G., Hawk, K. E., Rohren, E., Vial, A., & Klein, R. (2019). Machine Learning and Deep Learning in Medical Imaging: Intelligent Imaging. Journal of Medical Imaging and Radiation Sciences, 50 (4), 477-487. DOI: 10.1016/j.jmir.2019.09.005

9. Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations (ICLR), 14. DOI: 10.48550/arXiv.1409.1556

10. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

11. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. -C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition , 4510-4520. DOI: 10.1109/CVPR.2018.00474

12. Yamashita, R., Nishio, M., Do, R.K.G. et al. (2018). Convolutional neural networks: an overview and

application in radiology. Insights Imaging, 9, 611–629. DOI: 10.1007/s13244-018-0639-9

13. Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

14. Naeem, A., Farooq, M. S., Khelifi, A., & Abid, A. (2020). Malignant Melanoma Classification Using Deep Learning: Datasets, Performance Measurements, Challenges and Opportunities. IEEE Access, 8 , 110575-110597. DOI: 10.1109/ACCESS.2020.3001507

15. Bassel, A., Abdulkareem, A.B., Alyasseri, Z.A.A., Sani, N.S., & Mohammed, H.J. (2022). Automatic Malignant and Benign Skin Cancer Classification Using a Hybrid Deep Learning Approach. Diagnostics, 12, 2472. DOI: 10.3390/diagnostics12102472

16. Alo, M., Baloglu, U. B., Yıldırım, Ö., & Acharya, U. R. (2019). Application of deep transfer learning for automated brain abnormality classification using MR images. Cognitive Systems Research, 54 , 176-188. DOI: 10.1016/j.cogsys.2018.12.007

17. Schirrmacher, R., Xue, Y., Chen, S., Qin, J., Liu, Y., Huang, B., & Chen, H. (2017). Application of Deep Learning in Automated Analysis of Molecular Images in Cancer: A Survey. Contrast Media & Molecular Imaging, 2017, 9512370. DOI: 10.1155/2017/9512370

18. Ibrahim, D. M., Elshennawy, N. M., Sarhan, A. M. (2021). Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases. Computers in Biology and Medicine, 132, 104348. DOI: 10.1016/j.compbiomed.2021.104348.

19. Zhou, L., Zhang, Z., Chen, Y.-C., Zhao, Z.-Y., Yin, X.-D., & Jiang, H.-B. (2019). A Deep Learning-Based Radiomics Model for Differentiating Benign and Malignant Renal Tumors. Translational Oncology, 12(2), 292-300. DOI: 10.1016/j.tranon.2018.1

20. Korobkin, M., Brodeur, F. J., Francis, I. R., Quint, L. E., Dunnick, N. R., Londy, F. J., ... & Podrabsky, P. A. (1996). CT time-attenuation washout curves of adrenal adenomas and nonadenomas. AJR. American Journal of Roentgenology, 166(2), 531-536. DOI: 10.2214/ajr.166.2.8553963

21.Boland, G. W. L., Dwamena, B. A., Sangwaiya, M. J., Goehler, A. G., & Francis, I. R. (2008). Characterization of adrenal masses using unenhanced CT: Initial work in 165 consecutive patients. Radiology, 249(3), 756-764. DOI: 10.1148/radiol.2493080372

22. Szolar, D. H., Korobkin, M., Reittner, P., Berghold, A., Hausegger, K. A., & Ballmer, P. E. (2005). Adrenal masses: Characterization with multiphasic CT protocol with 10-minute delay. Radiology, 234(2), 479-485. DOI: 10.1148/radiol.2342032017

23. Libé, R., & Huillard, O. (2023). Adrenocortical carcinoma: Diagnosis, prognostic classification and treatment of localized and advanced disease. Cancer Treatment and Research Communications, 37. DOI:

10.1016/j.ctarc.2023.100759