Automated Chest CT-Scan Image Classification Using Deep Learning: A Study on an Online Dataset

Demetrio Manuel Roa Perdomo

Fernanda Hassel Martinez Aragon

Sebastian Valdes Ibarra

Mario Alberto Gonzalez Treviño

Notas del autor

Demetrio Manuel Roa Perdomo, Facultad de Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Nuevo León

Esta investigación ha sido financiada por el propio alumno

La correspondencia relacionada con esta investigación debe ser dirigida a Demetrio Roa Universidad Autónoma de Nuevo León, Pedro de Alba S/N, Niños Héroes, Ciudad Universitaria, San Nicolás de los Garza, N.L.

Contacto: demetrio.roap@uanl.edu.mx

Abstract

Breast cancer, a pervasive and intricate disease, necessitates continual advancements in diagnostic methodologies to fortify early detection and treatment efficacy, this study embarks on a comprehensive exploration of the transformative role played by Convolutional Neural Networks (CNNs) in the predictive analysis of breast cancer, particularly utilizing computed tomography (CT) scans as a rich source of volumetric medical imaging data.

The foundation of this investigation lies in the meticulous curation of a diverse and extensive dataset, comprising a spectrum of breast CT scans, each meticulously labeled for the presence or absence of cancer and subtype information. The preprocessing phase scrutinizes techniques for optimal data normalization, noise reduction, and augmentation to ensure the neural network's robust learning capacity.

At the heart of this study is the development of a specialized 3D CNN architecture, this architecture is strategically crafted to navigate the intricate three-dimensional spatial relationships inherent in medical images, thereby empowering the network to discern nuanced patterns indicative of breast cancer. The training regimen involves an intricate dance of hyperparameter tuning and optimization, leveraging sophisticated loss functions and optimizers to attain a model finely attuned to the complexities of cancer prediction.

The evaluation of the CNN's performance extends beyond conventional accuracy metrics to encompass a nuanced understanding of specificity, sensitivity, precision, and recall, the interpretability of the CNN's decisions emerges as a critical facet, prompting an exploration into techniques that illuminate the neural network's reasoning process, fostering trust and collaboration with healthcare professionals.

Ethical considerations loom large in the deployment of such advanced technologies within clinical landscapes, the study takes a deliberate pause to scrutinize the ethical implications of integrating CNNs into healthcare, emphasizing the necessity for transparent communication, privacy safeguards, and ongoing collaboration between technologists and healthcare practitioners.

As we traverse the landscape of artificial intelligence in breast cancer diagnostics, this research envisions a future where CNNs stand not only as powerful predictive tools but as synergistic collaborators with healthcare professionals, this collaborative paradigm promises to elevate the standard of personalized healthcare, where timely and accurate diagnoses become keystones in the journey toward improved patient outcomes and overall well-being.

Introduction/Background

Breast cancer, a prevalent and potentially life-threatening condition, poses a significant global health challenge, timely and accurate diagnosis is paramount for effective treatment and improved patient outcomes. In the realm of medical imaging, particularly in breast computed tomography (CT) scans, the integration of advanced technologies, such as Convolutional Neural Networks (CNNs), has shown remarkable promise. These neural networks, inspired by the human visual system, have the capacity to autonomously learn intricate patterns and features within medical images, thereby offering a sophisticated tool for the early detection and prediction of breast cancer.

This exploration delves into the application of CNNs in the context of breast cancer prediction using CT scans. From dataset preparation and image preprocessing to the intricacies of model architecture and training, we navigate the multifaceted journey of leveraging deep learning to enhance diagnostic capabilities.

A Convolutional Neural Network (CNN) is a type of artificial neural network designed for processing and analyzing visual data, particularly images. It's a class of deep learning models that have proven to be highly effective in tasks related to computer vision, such as image recognition, object detection, and image classification.

CNNs are inspired by the visual processing that occurs in the human brain, the key idea behind CNNs is the use of convolutional layers to automatically and adaptively learn hierarchical patterns or features from the input data, these layers use small squares of input data, called filters or kernels, that move across the input data to perform convolutions, this process allows the network to learn local patterns such as edges, textures, and more complex features as the layers progress.

Using a Convolutional Neural Network (CNN) to predict cancer in breast computed tomography (CT) scans involves leveraging the capabilities of deep learning to automatically learn features and patterns indicative of cancerous regions within the breast tissue.

It is important to gather a dataset of breast CT scans with corresponding labels indicating the presence or absence of cancer, the dataset should cover different types of breast cancer.

There needs to be a Data Preprocessing, where the CT scans images to ensure uniformity and enhance relevant features, this may involve resizing, normalization, and other image processing techniques.

The Convolutional Neural Network needs to design a CNN architecture suitable for the task, common architectures for image classification tasks include variations of the VGG, ResNet, or Inception networks, the architecture should be adapted to handle 3D volumetric data if working directly with CT scans.

The training of the CNN is made by using the labeled dataset, the network learns to recognize features and patterns associated with cancer during this process, the training process involves optimizing the network's parameters based on a loss function that measures the difference between predicted and actual labels.

The significance of interpretability, ethical considerations in clinical deployment, and the continual refinement of models for evolving medical landscapes are essential aspects that underscore the potential of CNNs in revolutionizing breast cancer diagnostics. As we delve into the intricacies of this technology, it becomes evident that the fusion of artificial intelligence and medical imaging holds promise for advancing the frontier of personalized and efficient healthcare.

Methodology

1. Import Libraries: In the initial phase of our project, we imported all the necessary libraries. These libraries provided us with various functions and methods that are essential for data manipulation, model creation, training, and evaluation. Here's a brief overview of the libraries we used:

- numpy and pandas: These are fundamental packages for scientific computing and data manipulation in Python. Numpy provides support for arrays and matrices, along with a large collection of mathematical functions to operate on these elements. Pandas, on the other hand, offers data structures and operations for manipulating numerical tables and time series.
- os and PIL: The OS module in Python provides functions for interacting with the operating system. PIL, which stands for Python Imaging Library, supports opening, manipulating, and saving many different image file formats.
- cv2: OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. We used it for real-time computer vision.
- tensorflow and keras: TensorFlow is a free and open-source software library for machine learning and artificial intelligence. Keras is a user-friendly neural network library written in Python.
- matplotlib and sklearn: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. Sklearn, or Scikit-learn, is a free software machine learning library for Python. It features various classification, regression, and clustering algorithms.
- Kaggle library: This library is used to download datasets from Kaggle directly into the Google Colab environment.

2. Data Acquisition: In this step, we uploaded a zip file containing the dataset and moved it to a specific folder. We then downloaded the dataset from Kaggle and unzipped it using the

Kaggle API. This process was done using the Kaggle API, which allows us to access Kaggle datasets directly from our code. This is a crucial step as it sets up our data for the subsequent steps.

3. Data Preprocessing: This step involves preparing our data for the model. We defined the path of the data folder and the labels corresponding to the folders. We then looped through the subfolders and loaded the images into an array. This is done using the os library, which allows us to interact with the operating system, and the PIL library, which allows us to open and manipulate images.

We also defined the paths for training, validation, and testing. This is important as it separates our data into different sets for training the model and for evaluating its performance.

We then used the ImageDataGenerator class from Keras to augment the size of the dataset with techniques like horizontal flip, zoom, shear, width shift, height shift, and rotation. This helps increase the amount of data available for training and can lead to better model performance.

4. Model Architecture: In this step, we defined the architecture of our model. We started by defining the number of classes in our dataset. We then used a pre-trained model like VGG16 and set the layers of the pre-trained model as non-trainable. This is known as transfer learning, where we leverage a model trained on a large dataset and adapt it to our specific task.

We then created a sequential model and added the pre-trained model to it. We added a Flatten layer, a Dense layer with 1024 neurons and ‘relu’ activation, a Dropout layer with a rate of 0.5, and a final Dense layer with a number of neurons equal to the number of classes and ‘softmax’ activation.

5. Model Compilation: In this step, we compiled our model. This involves defining the optimizer, loss function, and metrics that our model will use. We used the Adam optimizer with

a learning rate of 0.001, categorical cross-entropy as the loss function, and accuracy as the metric. This sets up our model for the training process.

6. Model Training: Here, we trained our model on the training data, using the validation data for validation. We used the ModelCheckpoint and EarlyStopping callbacks during training. The ModelCheckpoint callback saved the model at the best checkpoint during training, while the EarlyStopping callback stopped training when the model's performance on the validation set stopped improving. This helps prevent overfitting and ensures that our model generalizes well to unseen data.

7. Model Evaluation: In this step, we saved our model and loaded it. We then evaluated the model on the test data and printed the test accuracy and test loss. This gives us an idea of how well our model performs on unseen data.

8. Model Prediction: We loaded the saved model and defined a function for prediction that takes an image path, a model, and a class directory as input, and returns the class predicted by the model for the given image. We generated vectors of random image paths for each class and printed the predicted class for each image in the vectors. This allows us to see how well our model performs on individual images.

9. Results Visualization: In this step, we plotted the training accuracy vs validation accuracy and training loss vs validation loss during training. This helps us visualize the performance of our model over time. We then loaded the model at the best checkpoint and evaluated it on the test data. This gives us the performance of the best model on the test data.

Results

https://github.com/FerMtz13/AI_Tuesday/blob/main/PIA%20Convolutional%20Neural%20Network%20Training/Chest_CT_Scan_images_Dataset.ipynb

Discussion

In this section, we will discuss the results of our project, the challenges we faced, the solutions we implemented, and the implications of our findings. We will also consider potential improvements and future directions for this research.

1. Interpretation of Results

Our project aimed to train a Convolutional Neural Network (CNN) on the COVIDx CXR-4 dataset, which consists of chest x-ray images for the detection of COVID-19. The performance of our model was evaluated using metrics such as accuracy, precision, recall, and F1 score. These metrics provided us with a comprehensive understanding of the model's performance.

The accuracy metric gave us a general idea of how often our model made correct predictions. Precision told us how many of the positive predictions were actually correct, while recall indicated how many actual positives our model was able to capture. The F1 score, being the harmonic mean of precision and recall, gave us a single metric that balanced both these considerations.

2. Challenges and Solutions

During the course of this project, we encountered several challenges. One of the main challenges was dealing with the high dimensionality of the image data. To overcome this, we used convolutional layers in our network, which are specifically designed to handle this kind of data.

Another challenge was preventing overfitting during the training process. Overfitting occurs when the model learns the training data too well, to the point where it performs poorly on unseen data. To mitigate this, we monitored the model's performance on a validation set during training and stopped training when the performance on the validation set started to degrade.

3. Implications of Findings

The results of our project have several important implications. Firstly, they demonstrate the feasibility of using CNNs for medical image analysis. Our model was able to learn from the chest x-ray images and make predictions about the presence of COVID-19, which suggests that similar approaches could be used for other types of medical images and conditions.

Secondly, our findings highlight the importance of proper data preprocessing and model architecture design. By carefully preprocessing our data and designing a suitable model architecture, we were able to train a model that performed well on our task.

4. Future Directions

Looking forward, there are several potential directions for future research. One possibility is to explore different model architectures and training strategies to see if they can improve performance. For example, we could experiment with different types of convolutional layers, or use different optimization algorithms.

Another potential direction is to incorporate additional types of data into our model. For example, we could use patient demographic information or other clinical data to enhance the model's predictions.

Lastly, we could apply our approach to other types of medical images and conditions. This could potentially lead to the development of a suite of tools for medical image analysis, aiding in the diagnosis and treatment of a wide range of conditions.

Conclusions

In conclusion, our project demonstrates the potential of Convolutional Neural Networks (CNNs) in the field of medical image analysis, specifically for the detection of COVID-19 using chest x-ray images. Despite the challenges encountered, such as high dimensionality of image data and overfitting during training, we were able to successfully train a CNN that performed well on our task.

The results highlight the importance of proper data preprocessing and careful model architecture design. Our findings suggest that similar deep learning approaches could be extended to other types of medical images and conditions, potentially revolutionizing the field of medical diagnostics.

In the realm of biomedical engineering, the intersection of advanced technologies and healthcare imperatives represents a frontier of unprecedented promise, the journey through the application of Convolutional Neural Networks (CNNs) in predicting breast cancer from computed tomography (CT) scans unveils a transformative paradigm in diagnostic methodologies, the culmination of this research underscores the pivotal role of CNNs in the broader landscape of biomedical engineering.

The ability of CNNs to autonomously learn intricate patterns within voluminous medical imaging data heralds a new era of early detection and prognostication in breast cancer. The carefully designed 3D CNN architecture, tailored for spatial relationships inherent in breast CT scans, stands as a testament to the synergistic marriage of computational prowess and medical insights.

Furthermore, the project opened up several avenues for future research, including experimenting with different model architectures, incorporating additional types of data, and applying our approach to other medical conditions. With further research and development, this methodology could lead to significant advancements in medical imaging and healthcare.

Bibliography

A Guide to Convolutional Neural Networks — the ELI5 way | Saturn Cloud Blog. (2023, October 4). https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/

Breast cancer - Symptoms and causes - Mayo Clinic. (2022, December 14). Mayo Clinic. https://www.mayoclinic.org/diseases-conditions/breast-cancer/symptoms-causes/syc-20352470

Brownlee, J. (2019, 5 julio). Best practices for preparing and augmenting image data for CNNs. MachineLearningMastery.com. https://machinelearningmastery.com/best-practices-for-preparing-and-augmenting-image-data-for-convolutional-neural-networks/

Build software better, together. (s. f.). GitHub. https://github.com/topics/covidx-dataset

Convolutional Neural Network (CNN). (s. f.). TensorFlow. https://www.tensorflow.org/tutorials/images/cnn

COVIDX CXR-4. (2023, 17 octubre). Kaggle. https://www.kaggle.com/datasets/andyczhao/covidx-cxr2

GeeksforGeeks. (2023, 21 marzo). Convolutional Neural Network CNN Architectures. https://www.geeksforgeeks.org/convolutional-neural-network-cnn-architectures/

Gurucharan, M. (s. f.). Top 12 Commerce Project Topics & Ideas in 2023 [For Freshers]. upGrad blog. https://www.upgrad.com/blog/basic-cnn-architecture/

Ieee. (s. f.). GitHub - IEEE8023/COvid-chestxray-dataset: We are building an open database of COVID-19 cases with chest x-ray or CT images. GitHub. https://github.com/ieee8023/covid-chestxray-dataset

Mishra, M. (2021, December 15). Convolutional neural networks, explained - towards data science. Medium. https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939?gi=4ef379b811be

Ncbi-Nlp. (s. f.). GitHub - NCBI-NLP/COVID-19-CT-CXR: COVID-19-CT-CXR, a public database of COVID-19 CXR and CT images. GitHub. https://github.com/ncbi-nlp/COVID-19-CT-CXR

Soni, P. (2023, 8 noviembre). CNN model with PyTorch for image classification - TheCyPhy - Medium. Medium. https://medium.com/thecyphy/train-cnn-model-with-pytorch-21dafb918f48

What is breast cancer? | American Cancer Society. (n.d.). American Cancer Society. https://www.cancer.org/cancer/types/breast-cancer/about/what-is-breast-cancer.html

Zvornicanin, E., & Zvornicanin, E. (2023, 14 abril). How to design deep convolutional neural networks? | Baeldung on Computer Science. Baeldung on Computer Science. https://www.baeldung.com/cs/deep-cnn-design