IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Review of Computer Engineering Studies
Vol. 10, No. 4, December, 2023, pp. 48-53
Journal homepage: http://iieta.org/journals/rces

Advanced AI-Driven MRI Tumor Diagnosis System on AWS with Enhanced HTTP/2 Data Transfer

Nassr N. Khamis*, Rahma S. Mustaf

Department of Systems Engineer, Al-Nahrain University, Baghdad 10072, Iraq

Corresponding Author Email: nassrnafea@gmail.com

https://doi.org/10.18280/rces.100401

Received: 14 October 2023
Revised: 20 November 2023
Accepted: 3 December 2023
Available online: 28 December 2023

Keywords:
Amazon Web Services, deep learning, HTTP/2, Magnetic Resonance Imaging, Picture Archiving and Communication System, VGG-16, Y-Net

ABSTRACT

Magnetic Resonance Imaging (MRI) is a critical tool in the detection and characterization of brain tumors, offering precise information on tumor location, size, and properties. This information is vital for healthcare professionals to devise personalized treatment plans and select suitable therapeutic approaches while preserving neurological functions. Presented here is an innovative AI-based MRI tumor diagnosis system, accessible to a wide range of medical professionals including radiologists, specialists, and registrars. The system integrates advanced web service applications, providing an improved user experience. The diagnosis system is built utilizindg a Y-Net architecture with transfer learning, demonstrating a classification loss of 0.2152 and an accuracy rate of 0.9075. Remarkably, for segmentation tasks, the system achieves an accuracy of 0.9983, a dice coefficient of 0.9036, and an Intersection over Union (IoU) of 0.8225. Hosted on Amazon's cloud servers, the system is readily available to medical center personnel, ensuring widespread accessibility. For deployment on the Amazon cloud, a virtual machine running the Ubuntu Linux distribution was established, and the training model was successfully deployed online. To bolster the reliability of interactions among various users, the deployment of HTTP/2 for internet connections is recommended. This protocol significantly enhances the handling of substantial MRI data, fostering improved transfer speeds and effective collaboration in the diagnosis and treatment of brain tumors. This implementation culminates in a sophisticated Picture Archiving and Communication System (PACS) with cloud-based functionalities for the storage, retrieval, management, and distribution of medical images and associated patient data. The MRI PACS system guarantees precise and error-minimized interactions among users, excelling in data interchange speed and accuracy. Simultaneously, the system's core AI functionalities, including segmentation, classification, localization, and tumor area calculation, are executed flawlessly, ensuring integrity and no data loss or corruption.

1. INTRODUCTION

The field of deep learning continues to thrive, with its applications extending across image recognition, identification, segmentation, and disease diagnosis, presenting more efficient solutions than traditional machine learning techniques [1]. Its proficiency is particularly pronounced in medical image diagnosis using MRI, where deep learning has significantly reduced the time and effort involved in the manual classification and segmentation of brain tumors, thus revolutionizing the analysis of extensive medical image databases. This technological advancement provides medical professionals with enhanced insights into patients' internal organs, thereby facilitating more accurate assessments and therapy planning [2, 3].

In the domain of brain tumor segmentation, detection, and classification, machine learning and deep learning techniques have been extensively adopted. These technologies aim to automate the aforementioned processes, augmenting the efficiency of medical assessments and therapy planning [3]. The decision to implement deep learning models as standalone entities or within cloud-based solutions is pivotal. While standalone models offer rapidity and simplicity in business processes, including training, prediction, and decision-making, their limitations become evident when handling large datasets, requiring constant updates and rapid response for an extensive user base [4]. Conversely, hosting deep learning models on AWS EC2 provides a robust and adaptable environment, facilitating efficient training and deployment at scale. The integration of HTTP/2 as a communication protocol addresses the constraints of its predecessor, HTTP/1.x. This advanced protocol introduces features such as header compression, multiplexing, prioritization, and complete request and response multiplexing. These enhancements are crucial in improving web performance and user experience [5]. It is essential that these capabilities be incorporated into the proposed system to optimize its performance.

The primary objective of this article is to present a novel methodological advancement in the field of intelligent MRI PACS systems. This has been achieved through the development of an innovative system for MRI-based identification and segmentation, utilizing the Y-Net architecture. This system is seamlessly integrated into the web environment and leverages the robust capabilities of HTTP/2. It is hosted on Amazon Web Services (AWS), ensuring a powerful and flexible platform for its deployment. This system exemplifies the integration of deep learning technologies in medical imaging, enhancing accuracy and efficiency in MRI image analysis. The deployment on a web-based platform, coupled with cloud hosting, ensures high accessibility, scalability, and performance. These features are indispensable for healthcare professionals and researchers seeking advanced medical image interpretation and diagnostic solutions.

Designed with user-friendliness in mind, the system's architecture addresses the limitations and capitalizes on the strengths of existing solutions. The emphasis is on ease of implementation, usage, and further development, ensuring a comprehensive approach to medical image analysis. The CoviExpert platform presents an innovative approach to COVID-19 diagnosis using chest X-rays. This user-centric platform allows for effortless uploading of X-ray images, providing rapid predictions and enabling informed decision-making. Its design incorporates an intuitive user interface, complete with sections for general information, FAQs, and contact details. While CoviExpert enhances the diagnostic process through its accessible and streamlined design, it must be noted that accuracy is contingent upon the quality of the images provided. The system's potential for misinterpretation necessitates user support, especially for those less experienced with medical imaging [6].

The focus shifts to pneumonia, a global health concern and a leading cause of mortality, particularly among children and the elderly. Chest radiographs, known for their affordability and rapid results, are extensively used for pneumonia diagnosis. The application of a convolutional neural network (CNN) to classify chest X-rays into categories, namely, normal, bacterial pneumonia, and viral pneumonia, has shown promising results. Enhanced by data augmentation, the CNN's accuracy in this classification task has been improved from 70% to 78.37%. This advancement underscores the potential of deep learning in early pneumonia detection. The integration of this model into a user-friendly web application hosted on AWS simplifies the process of image uploads, thus increasing accessibility for medical professionals and the broader community [7]. Nevertheless, challenges have been observed in the initial model, notably in terms of lower accuracy and elevated loss, thereby underscoring the necessity for enhanced training data. The variability in validation loss has been identified as a factor potentially impacting the consistency of predictions. Additionally, the web application's provision of disease-related information for healthcare professionals has been found wanting.

A groundbreaking method for disease detection using machine learning algorithms is introduced. These algorithms analyze symptoms, thereby obviating the need for laboratory tests. The research applied six supervised machine learning algorithms, namely, J48 decision tree, random forest, support vector machine, k-nearest neighbors, naïve Bayes, and artificial neural networks, to the "COVID-19 Symptoms and Presence Dataset" obtained from Kaggle. Techniques such as hyperparameter optimization and 10-fold cross-validation were employed to enhance the performance of these algorithms. A comparative analysis, focusing on accuracy, sensitivity, specificity, and the area under the Receiver Operating Characteristic (ROC) curve, indicated that the random forest, support vector machine, k-nearest neighbors, and artificial neural networks algorithms exhibited superior performance. They achieved an accuracy of 98.84%, sensitivity of 100%, specificity of 98.79%, and an area of 98.84% under the ROC curve. Moreover, a user-friendly web application was developed, enabling the prediction of COVID-19 based on current symptoms. This tool offers real-time disease detection without relying on laboratory tests, kits, or devices. However, it is crucial to note that the predictions provided by this application are not official and may lead to false positives or negatives. Therefore, laboratory testing remains imperative for confirmation. Patients are advised to seek telemedicine or medical consultation for an accurate diagnosis and appropriate guidance. It is recommended that this tool be used in tandem with standard testing methods [8].

2. RESEARCH METHODOLOGY

To implement an intelligent MRI PACS website based on Y-Net for both segmenter and classifier training deep learning model, we divide the process into two distinct stages, as illustrated in Figure 1.

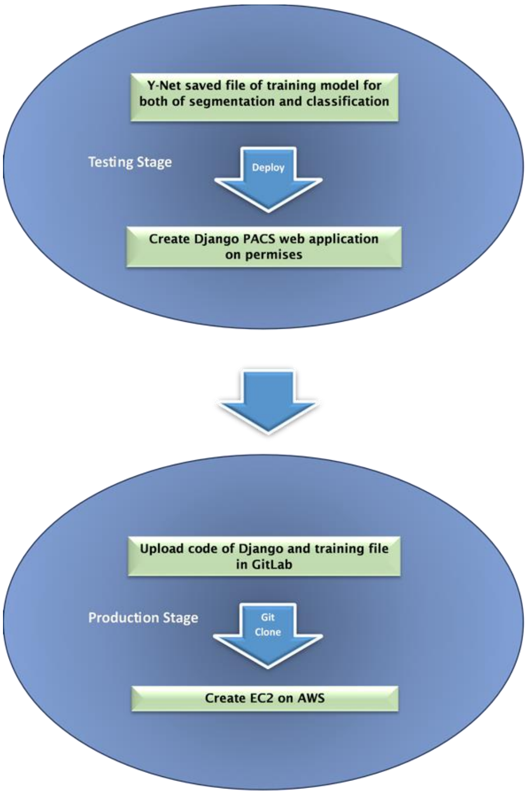

Y-Net saved file of training model for both of segmentation and classification
Testing Stage
Deploy
Create Django PACS web application on permises
Upload code of Django and training file in GitLab
Production Stage
Git Clone
Create EC2 on AWS

Figure 1. Processes flowdiagram of proposed PACS system

2.1 Stage one: Design, training, deployment, and testing

In the first stage, our focus is on designing, training, deploying, and testing an MRI-based AI using Y-Net [9]. This involves the integration of the AI model with a web application on a local machine. The main components of this stage are as follows:

2.1.1 MRI diagnosis deep learning

For the diagnosis of brain tumors through magnetic resonance imaging (MRI), we utilize deep learning (DL) techniques [3]. These techniques facilitate brain tumor identification, detection, and classification, and we also employ transfer learning (TL) [10]. Y-Net DL architecture is applied for both classification and segmentation of brain images, as demonstrated in Figure 2. The trained model is saved locally as an. hdf5 file.

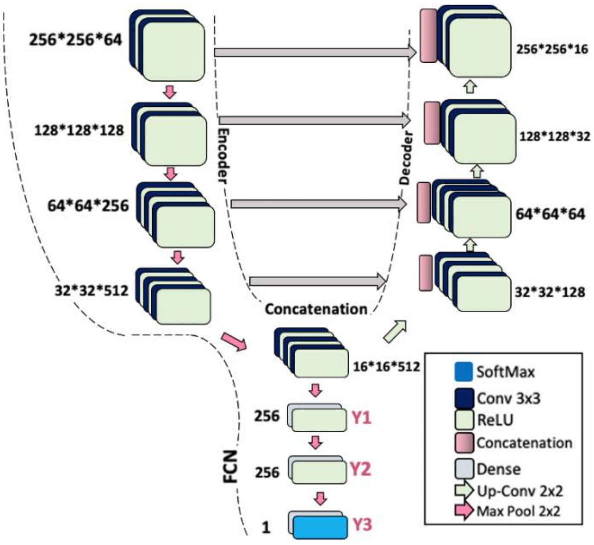


Figure 2. Y-Net architecture for both of classification and segmentation MRI brain [10]

2.1.2 Y-Net model-based web application deployment

Incorporating the Digital Imaging and Communication in Medicine (DICOM) standard for medical screenings, we consider industry standards and developments such as those by the American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA). These standards help preserve historical data and support integration with PACS and other communication systems, including waveforms, reports, and IDs.

To make the PACS system accessible to radiologists, healthcare professionals, and users, we utilize cloud services to create dedicated interfaces and tools within the PACS user interface. Our approach begins with implementing a high-level Python web framework called Django, which follows the model-view-controller (MVC) architectural pattern, often referred to as MVT (Model-View-Template).

Django offers a robust set of tools and features that streamline development, enhance security, and ensure scalability [11]. The main components of Django's MVT architecture are depicted in Figure 3. When a user interacts with our PACS system by accessing a URL or performing any action, the user's web browser sends an HTTP request to the Django server.

To facilitate this interaction, a URL dispatcher maps the incoming URL to a specific view based on URL patterns defined in our application file "urls. py." This file contains a list of URL patterns and their corresponding view functions. The view function processes the user's request and returns an HTTP response, which may involve interaction with the model (database) for data retrieval and processing.

The view function interacts with the Model to retrieve the necessary data, in this case, a list of all patients from the database using the Patient. objects. all () query. Finally, the view function renders the retrieved data into an HTML response using a template that defines the structure of the HTML page and inserts dynamic content from the data retrieved in the view [12].

The rendered HTML content is returned as an HTTP response to the user's browser, displaying the page, which, in this example, is a list of patients.

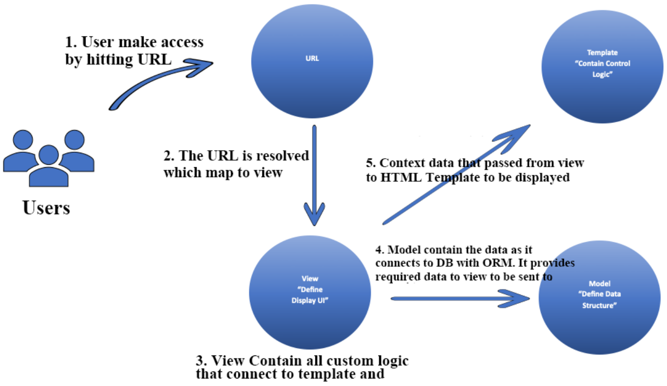

1. User make access
by hitting URL
URL
Template
"Contain Control
Logic"
Users
2. The URL is resolved
which map to view
5. Context data that passed from view
to HTML Template to be displayed
4. Model contain the data as it
connects to DB with ORM. It provides
required data to view to be sent to
View
"Define
Display UI"
Model
"Define Data
Structure"
3. View Contain all custom logic
that connect to template and

Figure 3. Model, view and template process

2.2 Stage two: Global access network with AWS

In the second stage, we make the intelligent MRI PACS system accessible on the global network, particularly the Amazon Cloud network. This step is crucial for reaching a broader audience, including medical professionals and users worldwide. To achieve global accessibility, we leverage the Amazon Web Services (AWS) cloud platform. Here's how we do it:

2.2.1 Gitlab: A centeralized development platform

Before deploying our system on AWS, we upload our code to GitLab, a powerful version control platform. This step ensures efficient project organization and collaborative work among team members. GitLab facilitates change management, version tracking, and maintains a centralized code repository. GitLab serves as a comprehensive platform for streamlining development processes, fostering collaboration, and automating software delivery. It is a crucial tool for development teams, covering various aspects, from code creation to deployment [13].

2.2.2 AWS hosting

Amazon Web Services offers various hosting solutions, and one of the most prominent is Elastic Compute Cloud (EC2). This web service is central to cloud computing needs, allowing users to lease virtual machines known as instances for running applications on a scalable and reliable platform [14]. EC2 is the preferred choice for hosting deep learning models with web applications due to its scalability, reliability, high performance, security, ease of use, cost-efficiency, and robust community and support. This combination of features makes EC2 a top choice for meeting the computational demands of deep learning tasks while ensuring flexibility and cost-effectiveness.

2.2.3 Configuring EC2 instance

To host our PACS system, we need to configure an EC2 instance. This involves launching an Amazon Machine Image (AMI), which utilizes the Ubuntu operating system. The chosen instance type is t2.2xlarge, equipped with 32GB of RAM and 8 virtual CPUs (vCPUs) based on Intel Xeon. Security group configuration is essential, permitting specific port access for HTTP and SSH.

2.2.4 Key pair creation

An integral step in this process is creating a key pair. AWS generates a public-private key pair. The private key is securely stored on our local machine, while the public key is associated with our EC2 instance. This key pair allows secure access to our EC2 instance through SSH for activities such as cloning the code repository from GitLab.

3. RESULT

The intelligent PACS (Picture Archiving and Communication System) designed for the segmentation and classification of brain tumors and accessible through a web application consists of several key components:

- Administrator: At the core of the system is the administrator, responsible for managing user accounts and their roles. This individual ensures the smooth operation of the system, overseeing its various functions.
- Registrators: Registrators play a critical role in the initial phase of the patient journey. They are tasked with gathering essential information from incoming patients, including their unique ID, first name, sex, and age. This data collection is the first step in the process.
- Radiologists: Highly specialized medical professionals, radiologists, are responsible for capturing and uploading MRI (Magnetic Resonance Imaging) images of the patients. Their expertise in acquiring high-quality images is essential for the accurate analysis that follows.
- Specialists: The specialists, often medical experts or diagnosticians, are the end-users of the system. They receive the MRI images and the results of the deep learning analysis. Their responsibilities encompass interpreting these outcomes, including the localization, segmentation, classification, and tumor area calculation. They also provide comprehensive case notes and save diagnostic reports as PDF files.

The workflow within this system is well-defined and as shown in Figure 4:

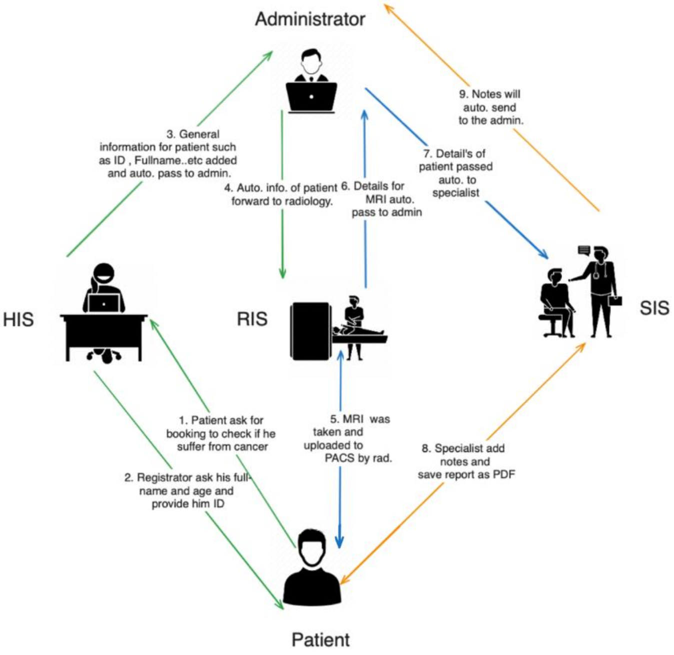
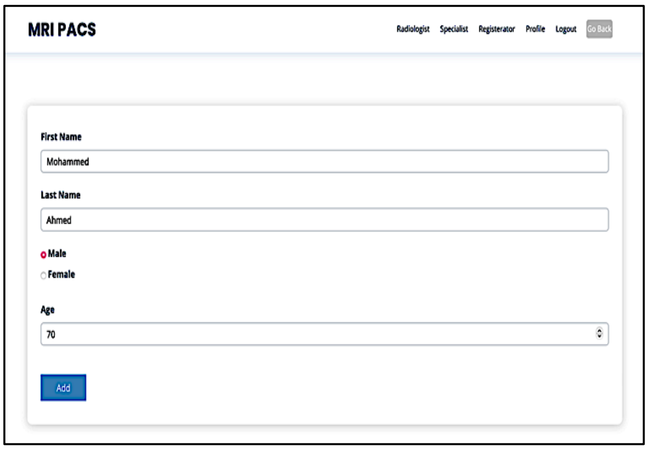

Administrator
3. General information for patient such as ID , Fullname..etc added and auto. pass to admin.
4. Auto. info. of patient forward to radiology.
6. Details for MRI auto. pass to admin
7. Detail's of patient passed auto. to specialist
9. Notes will auto. send to the admin.
HIS
RIS
SIS
1. Patient ask for booking to check if he suffer from cancer
2. Registrator ask his full-name and age and provide him ID
5. MRI was taken and uploaded to PACS by rad.
8. Specialist add notes and save report as PDF
Patient

Figure 4. Process of MRI PACS system

(1) Data Entry by Registrators: The process begins when registrators input vital patient information into the system, which acts as the foundation for subsequent analyses, information of patient including First Name, Last Name, Age and Geneder as shown in Figure 5.



First Name
Mohammed

Last Name
Ahmed

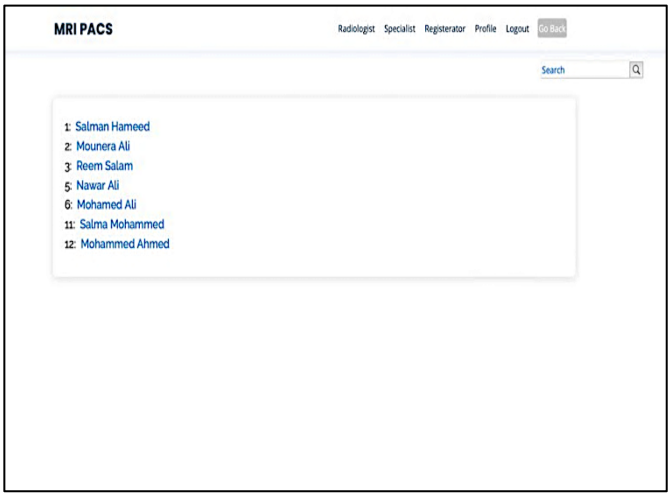
● Male
○ Female

Age
70

Add

Figure 5. Registerator add patient in PACS

(2) Information Transfer to Administrator: The collected data is then passed on to the administrator, who manages the administrative aspects of the system as shown in Figure 6.



1: Salman Hameed
2: Mounera Ali
3: Reem Salam
5: Nawar Ali
6: Mohamed Ali
11: Salma Mohammed
12: Mohammed Ahmed

Figure 6. List of patients in radiology page

(3) Assignment to Radiologists: The administrator assigns the patient case to radiologists, who are tasked with capturing and uploading the MRI images, ensuring that the image quality meets diagnostic standard sas shown in Figure 7.

(4) Deep Learning Analysis: The core of the system's intelligence lies in the deep learning model. This model performs tasks such as localization, segmentation, classification, and tumor area calculation, and presents the results to the radiologist and specialists.

(5) Specialist's Role: Specialists analyze the deep learning results and apply their clinical expertise to interpret and refine the findings. They also contribute vital case notes to provide context and insights as shown in Figure 8.

(6) Report Generation: Specialists compile diagnostic reports that encapsulate the patient's condition, including the tumor's location, characteristics, and any pertinent clinical notes. These reports are saved in PDF format for ease of accessibility and sharing as shown in Figure 9.

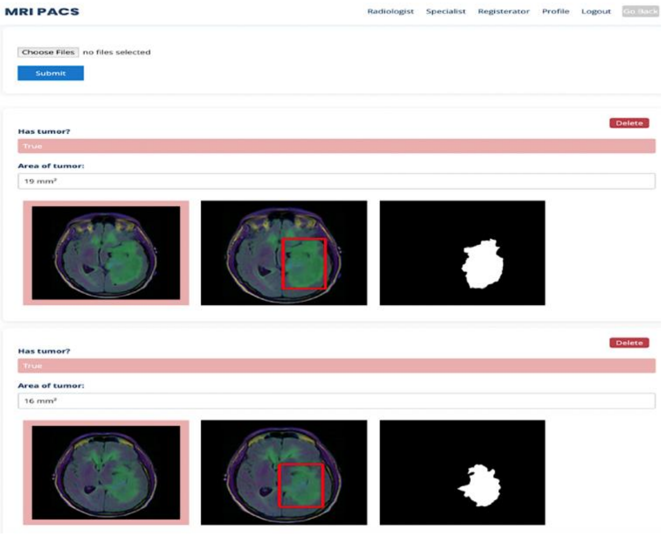
By orchestrating these components and implementing deep learning technology, the PACS system enhances the accuracy and efficiency of diagnosing brain tumors, ultimately improving patient care and medical decision-making.

In the realm of healthcare database modeling which is used SQLite, the initial phase involves creating five distinct tables with specific roles as shwon in Figure 10. User Profile, User, Doctor, Tumor, and Person Tables, these tables form the foundation of the database structure.

The User Table establishes a one-to-one relationship with the User_Profile Table, differentiating user accounts based on type. Additionally, it facilitates a one-to-many relationship with the Doctor Table, enabling doctors to contribute notes for patient case reports.

The Tumor Table is central, with a one-to-many relationship with the Doctor Table, allowing the association of multiple tumors with a single doctor. It also establishes a many-to-one relationship with the Person Table, signifying each tumor's connection to a single patient while allowing each patient to be associated with multiple tumors.

As for the implementation of HTTP/2, this technology can significantly enhance the web application's performance and responsiveness. HTTP/2 is particularly well-suited for applications that rely on multiple data transfers, such as medical imaging and deep learning analysis.



no files selected

Has tumor?

True

Area of tumor:

19 mm²

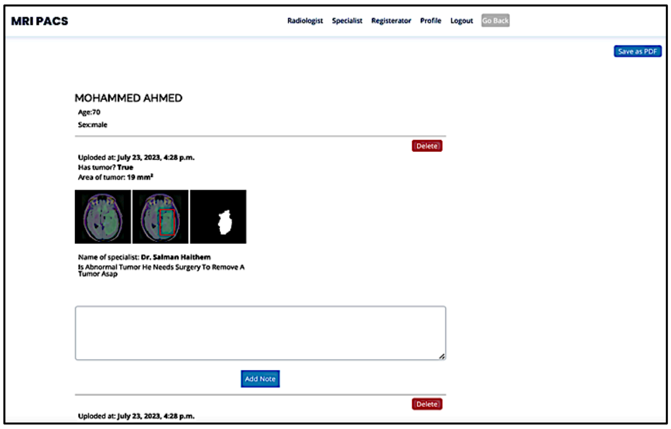
Has tumor?

True

Area of tumor:

16 mm²

Figure 7. Radiologist upload MRI for patient and result will be display immediately



MOHAMMED AHMED

Age:70

Sex:male

Uploded at: July 23, 2023, 4:28 p.m.

Has tumor? True

Area of tumor: 19 mm²

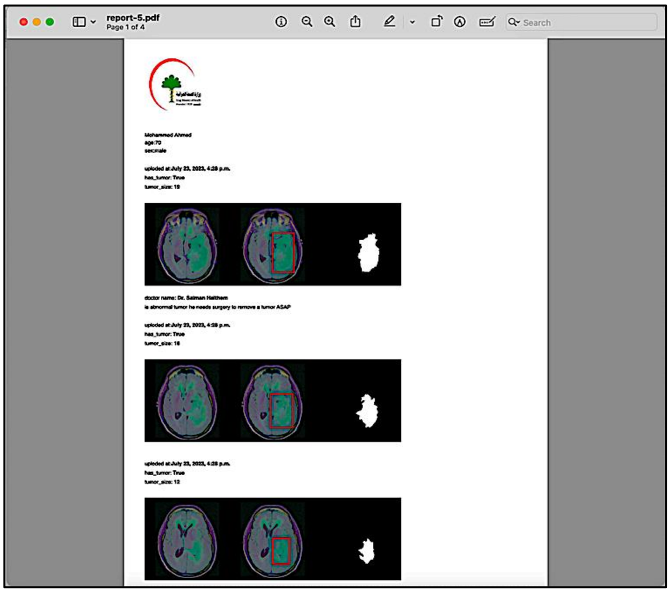
Name of specialist: Dr. Salman Haithem

Is Abnormal Tumor He Needs Surgery To Remove A Tumor Asap

Uploded at: July 23, 2023, 4:28 p.m.

Figure 8. Specialist will add his note in each image

To enable HTTP/2, it can be integrated into the web server that hosts the PACS system. Modern web servers like Apache Server to improve the speed and efficiency of web applications and websites, as faster loading times and better resource management are allowed by it. This is achieved with the following command, as shown in Figure 11.



Mohammed Ahmed

age:70

sex:male

uploded at:July 23, 2023, 4:28 p.m.

has_tumor: True

tumor_size: 19

doctor name: Dr. Salman Haithem

is abnormal tumor he needs surgery to remove a tumor ASAP

uploded at:July 23, 2023, 4:28 p.m.

has_tumor: True

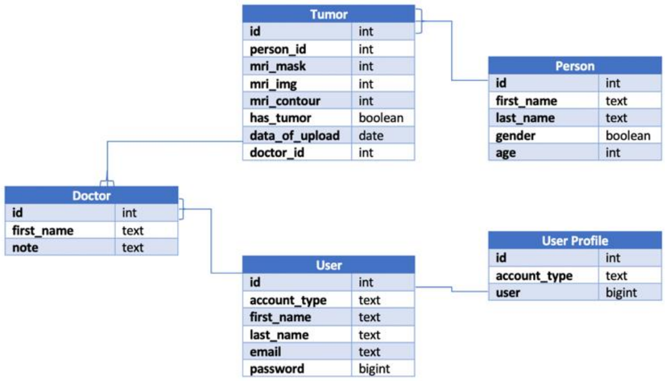
tumor_size: 16

uploded at:July 23, 2023, 4:28 p.m.

has_tumor: True

tumor_size: 12

Figure 9. Saved report of patient by specialist as PDF



Tumor	
id	int
person_id	int
mri_mask	int
mri_img	int
mri_contour	int
has_tumor	boolean
data_of_upload	date
doctor_id	int

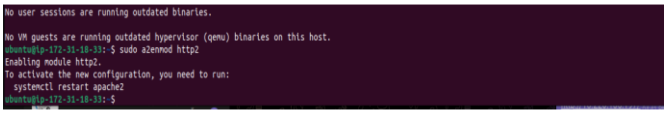
Person	
id	int
first_name	text
last_name	text
gender	boolean
age	int

Doctor	
id	int
first_name	text
note	text

User	
id	int
account_type	text
first_name	text
last_name	text
email	text
password	bigint

User Profile	
id	int
account_type	text
user	bigint

Figure 10. Relationship database PACS system



```
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-18-33:~$ sudo a2enmod http2
Enabling module http2.
To activate the new configuration, you need to run:
  systemctl restart apache2
ubuntu@ip-172-31-18-33:~$
```

Figure 11. Activate HTTP/2

4. CONCLUSION

By utilizing AWS to host the PACS system, healthcare providers can take advantage of the cloud's scalability, reliability, and security features. AWS provides a robust infrastructure that ensures the PACS system's high availability, seamless data backup, and disaster recovery capabilities. Additionally, the cloud-based setup reduces the need for on-premises hardware maintenance, resulting in cost savings and greater flexibility in resource allocation.

Enabling HTTP/2 for the PACS system brings significant benefits to data transfer efficiency and user experience. With HTTP/2's multiplexing, the system can process concurrent requests over a single connection, reducing latency and accelerating image retrieval. Header compression minimizes overhead, optimizing bandwidth usage and further enhancing system performance. Furthermore, the adoption of HTTP/2, together with SSL/TLS encryption requirements, ensures the secure transmission of sensitive patient information.

The combination of AWS hosting and HTTP/2 support guarantees swift access to medical images, quicker retrieval times for clinicians, and ultimately improves patient care. Moreover, the PACS system's dynamic scalability in AWS enables healthcare institutions to adapt seamlessly to varying workloads, efficiently handling a growing number of medical imaging studies.

In conclusion, implementing an MRI PACS system hosted in AWS and enabling HTTP/2 signifies a forward-looking approach that harnesses cutting-edge technology to deliver a secure, high-performance, and dependable medical imaging platform for healthcare providers and their patients.

REFERENCE

[1] Razzak, M.I., Naz, S., Zaib, A. (2018). Deep learning for medical image processing: Overview, challenges and the future. Classification in BioApps: Automation of Decision Making, 323-350. https://doi.org/10.1007/978-3-319-65981-7_12

[2] Zhang, Z., Li, G., Xu, Y., Tang, X. (2021). Application of artificial intelligence in the MRI classification task of human brain neurological and psychiatric diseases: A scoping review. Diagnostics, 11(8): 1402. https://doi.org/10.3390/diagnostics11081402

[3] Das, S., Aranya, O.R.R., Labiba, N.N. (2019). Brain tumor classification using convolutional neural network. In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, pp. 1-5. https://doi.org/10.1109/ICASERT.2019.8934603

[4] Singh, P., Singh, P. (2021). Model deployment and challenges. Deploy Machine Learning Models to Production: With Flask, Streamlit, Docker, and Kubernetes on Google Cloud Platform, 55-66. https://doi.org/10.1007/978-1-4842-6546-8_2

[5] Tuyl, R., Northcutt, S. (2016). Practical approach to detecting and preventing web application attacks over HTTP/2. SANS Institute Reading Room site.

[6] Arivoli, A., Golwala, D., Reddy, R. (2022). CoviExpert: COVID-19 detection from chest X-ray using CNN. Measurement: Sensors, 23: 100392. https://doi.org/10.1016/j.measen.2022.100392

[7] Alshamrani, K., Alshamrani, H.A., Asiri, A.A., Alqahtani, F.F., Mohammad, W.T., Alshehri, A.H. (2022). The use of chest radiographs and machine learning model for the rapid detection of pneumonitis in Pediatric. BioMed Research International, 2022. https://doi.org/10.1155/2022/5260231

[8] Villavicencio, C.N., Macrohon, J.J., Inbaraj, X.A., Jeng, J.H., Hsieh, J.G. (2022). Development of a machine learning based web application for early diagnosis of COVID-19 based on symptoms. Diagnostics, 12(4): 821. https://doi.org/10.3390/diagnostics12040821

[9] Khamis, N.N., Mustaf, R.S. (2023). Colored MRI biomedical image tumor classification and segmentation based on transfer learning of modified Y-Net. In ITM Web of Conferences. EDP Sciences, Vol. 56.

[10] Aurna, N.F., Yousuf, M.A., Taher, K.A., Azad, A.K.M., Moni, M.A. (2022). A classification of MRI brain tumor based on two stage feature level ensemble of deep CNN models. Computers in Biology and Medicine, 146: 105539. https://doi.org/10.1016/j.compbiomed.2022.105539

[11] Uzayr, S.B. (2022). Mastering Django: A Beginner's Guide, 1st ed. Boca Raton: CRC Press. https://doi.org/10.1201/9781003310495

[12] Shaw, B., Badhwar, S., Guest, C., KS, B.C. (2023). Web Development with Django: A definitive guide to building modern Python web applications using Django 4.

[13] Cannon, S. (2023). Succeed in software: A Comprehensive guide to software career excellence. N.p.: Alien Creations.

[14] Murty, J. (2008). Programming amazon web services: S3, EC2, SQS, FPS, and SimpleDB. O'Reilly Media, Inc.