© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1717105

Brain Tumour Detection Using 3D Reconstruction

DNYANESHWARI KATKAR[1], SHREJAL SAHANI[2], ALIYA ANSARI[3], SALABA JACOB[4]
[1,2,3]*Electronics and Computer Science, Shah and Anchor Kutchhi Engineering College Chembur, Mumbai, India*
[4]*Proffesor, Electronics and Computer Science, Shah and Anchor Kutchhi Engineering College Chembur, Mumbai, India*

Abstract- ***The early detection of brain tumour is one of the most difficult problems in modern medicine. Even though radiologists are highly experienced and trained, they still have difficulty detecting brain tumours because of the volume and complexity of MRI data; although radiologists are highly skilled, they will occasionally make mistakes that could have serious consequences. The purpose of this paper is to provide the medical community with a complete diagnostic solution that allows clinicians to automate the process from obtaining raw MRI scans to generating a three-dimensional visualisation of a brain tumour. The proposed method creates a unified workflow by applying a single solution to detection, segmentation, and visualisation as three distinct processes. The heart of the proposed system is a three-dimensional convolutional neural network (CNN), which generates spatially continuous three-dimensional MRI volumes rather than simply producing individual slices of two-dimensional data at different points in time and spaceThe pixel-wise auto-mated tumor detection model uses the U-Net architecture to segment tumors. Each mask is then converted into a three-dimensional mesh of the tumor shape for visualization purposes. The system categorizes findings into four types—Glioma, Menin-gioma, Pituitary, and No Tumor—and provides a confidence score for each of these classifications. The model achieved an accuracy of 98.2% using the BraTS dataset for validation [3].***

Index Terms—Brain Tumor Detection, MRI Analysis, Deep Learning, 3D CNN, U-Net, 3D Reconstruction, Django Web Application

I. INTRODUCTION

Brain Tumors are among the deadliest diseases one can experience. They are located inside the skull, which is where all our thoughts, memory, and movement originate from. For this reason, they are often much more serious than tumors located elsewhere in the body. They are also one of the most rapidly growing cancers, making them a high priority when it comes to diagnostic procedures. For example, Glioblastoma Multiforme (GBM) is often called the most aggressive primary brain tumor. If not diagnosed early enough, the survival rate is measured in months. This is a significant problem on a worldwide scale with an estimated 300,000 new patients diagnosed with Brain Tumors each year, according to the World Health Organization. Unfortunately, diagnostic procedures for Brain Tumors still rely heavily on manual diagnostic techniques. When a radiologist evaluates an MRI of the brain, they must examine the cross-sectional slices and visualize the three-dimensional image by reconstructing multiple two-dimensional images. This process is cognitively challenging, and the potential for error increases substantially when a radiologist is fatigued, constrained by time, or lacks specialist resources in their work environment. MRI provides a non-ionizing radiation way to obtain soft-tissue images with more clarity than CT, thus establishing itself as the superior modality for evaluation of suspected brain tumors. MRI also has multiple types of images obtained through T1 imaging, T2 imaging, and FLAIR imaging that allow for an even greater understanding of the different properties of tissues and ultimately a more complete assessment of the brain [3]. Because manually processing very large datasets is both slow and expensive. In the last 10 years, advances in deep learning have greatly improved the ability of computers to perform medical image analysis [6]. In particular, Convolutional Neural Networks have been able to learn diagnostic features from data and no longer require a specialist in that field to indicate what the features are. The major disadvantage of standard two-dimensional Convolutional Neural Networks is that they process individual MRI slices at a time and do not recognize as to how tissue characteristics can differ between slices. There-fore,

this represents a disadvantage for a volumetric imaging technique such as MRI. In this paper, we present a solution to this problem by using a three-dimensional Convolutional Neural Network that processes the entire MRI volume at the same time and captures the inter-slice spatial relationships that would be overlooked when using two-dimensional Convolu-tional Neural Networks to process the same data A U-Net is used to produce detailed segment masks of the area of each tumor with pixel accuracy (fine-grained). The masks are then combined and converted into a three-dimensional surface mesh, giving medical professionals a graphical way to see the shape of the tumor that isn't possible using just standard two-dimensional slices. The entire pipeline is contained inside a Django application so that doctors can use it via a web browser without needing to know anything about how to code.

II. LITERATURE REVIEW

Since the beginning of time, the development of auto-mated brain tumor detection can be referred to as moving from feature engineering to learning representations. In earlier approaches, feature extraction was utilized by hand-crafted features such as texture statistics, intensity histograms, and edge gradient features trained with traditional classifiers (e.g., Support Vector Machines, k-Nearest Neighbours). All of these early approaches showed reasonable performance on given datasets; however, they also showed significant limitations when trying to process images acquired from different sources or through different protocols. As such, there was no pre-defined set of hand-crafted features that could cover all possi-ble variations that would be found in actual clinical data set. In 2016, Pereira et al. demonstrated that a convolutional neural network (CNN) trained with an end-to-end approach with MRI data outperformed the best performing conventional methods for segmenting the brain tumor from the brain on BraTS benchmark [7]. Shortly after, Havaei et al. put forth a two stage training approach which in turn improved what had been the main defect of past networks tumor border detection. Also in this time frame Kamnitsas et al. reported on DeepMedic which did away with the single scale approach in 3D CNN's they used which at the same time looked at fine local details and large scale context. What is particular about DeepMedic's performance was that it did very well at finding small and ir-regular tumor areas which past models did not do well at. Also of great note is the U-Net architecture which Ronneberger et al. put out in 2015 [4] which in many ways changed the game in medical image segmentation The model's encoder-decoder structure together with a series of skip connections allows for the combination of top-level semantic knowledge (the type of tissue) with low-level spatial accuracy (the true boundary location). This combination is especially important when dealing with tumors due to the variability of tumor shapes and boundaries. Since the introduction of 3D U-Net extensions there has been a surge in the literature of 3D segmentation algorithms based on the U-Net framework. In addition, transformer-based architectures (TransUNet, Swin-UNet, etc.) are being used to perform segmentation tasks. The use of transformer architectures allows you to represent very long range spatial relationships in ways not accomplished with CNNs, and currently provide benchmark improvements on some datasets. However, the downside is severe: while transformers require a lot more GPU memory than CNNs, they also require longer training times and more data to generalise well. When resources are limited, such as in certain clinical settings where timely responses are critical, the incremental performance improvements available from existing solutions do not justify their costs. We can find a substantial gap in the literature regarding combined detection/segmentation sys-tems used by health professionals without formal information systems expertise, and there are no tools appropriate for use by these clinicians. This research project aims to close this gap by creating a seamless, integrated, end-to-end system and providing it in the form of a web interface requiring no additional specialized training/education to operate.

III. PROPOSED SYSTEM

The proposed system is designed as a fully automated, end-to-end pipeline that performs brain tumor detection, clas-sification, segmentation, and 3D visualization. The system is accessible to medical practitioners through a secure web interface developed using the Django framework [9].

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1717105

A. System Overview

The new system is using one MRI scan as an input to create a full diagnostic report including tumor type, approximate size, confidence level, and a 3D representation with zero manual work done by a doctor. A doctor logs into a web interface, finds or creates the patient record, and uploads the MRI scan. All of the rest of the processes are automated. The image goes through preprocessing, then it goes to be processed by deep learning models, and finally, it's displayed on the screen in seconds after processing is complete. Each MRI scan will be stored in the patient's profile providing a history that can be reviewed at any time.

B. Web Application Architecture

There are five main functional areas in the application built using Django 5.2 with SQLite as a backend database. The Authentication Module uses Django's built-in login system to keep the attack surface minimal by preventing the introduc-tion of new custom credential handling. The Upload Module provides for the uploading of PNG and JPG image files which will be stored in a structured media folder for each respective patient. The Detection Module will run the uploaded scan through the entire preprocessing and inference pipeline. The Result Module will compile and display the output page for the tumor type, its estimated size, a confidence score, and a processed image in side-by-side format. Finally, the History Module will keep a record of all scans for a given physician, including the ability to delete an individual scan or an entire patient record.

IV. METHODOLOGY

A. Data Acquisition

The system has been trained on and assessed with the Brain Tumor Segmentation Challenge (BraTS), which serves as the authoritative benchmark [3] for this two-dimensional segmentation task. The BraTS dataset combines MRI scans from different facilities and includes expert-annotated ground truth (GT) labelling, enabling reliable and reproducible train-ing of robust models, and facilitating meaningful compari-son with existing studies. The BraTS dataset includes four modality images (T1, T1ce, T2, and FLAIR) from each patient's MRI study. Each of these four imaging sequences is a vital contributor; for example, T1ce highlights the currently enhancing core of the tumour, while FLAIR delineates the peritumoural (surrounding) oedema. Collectively, these four sequences represent a much more total view of the tumour than using any singularly available imaging modality.

B. Preprocessing

Different hospitals and manufacturers of MRI scanners produce different raw RTVs, preventing their direct compar-ison. Significant differences in tissue intensities across RTs of a tissue type (i.e., the same tissue taken with the same camera setting but at many different scanner sites) occur due to variations in the strength and type of magnetic field at each scanner site/ccusing variations in image acquisition parameters at the time of image acquisition as well as the noise generated by each scanner. Prior to their use in the model, each volume is processed through a standard preprocessing pipeline before they are included in the model. First, Gaussian and median filters are applied to reduce noise without blurring the tissue boundaries that the model needs to learn. Next, Z-score normalisation standardises the intensity distribution of each scan to have zero mean and unit variance, making the data from different machines directly comparable. Skull stripping removes the skull and other nonbrain tissue, so the model does not spend capacity learning features that are irrelevant to the diagnostic question. All volumes are then resampled to a uniform size of 128×128×128 voxels. During training, random flips, rotations, and intensity perturbations are applied as data augmentation to improve generalisation. The preprocessed modalities are stacked into a four-channel tensor that serves as the input to the 3D CNN [6].

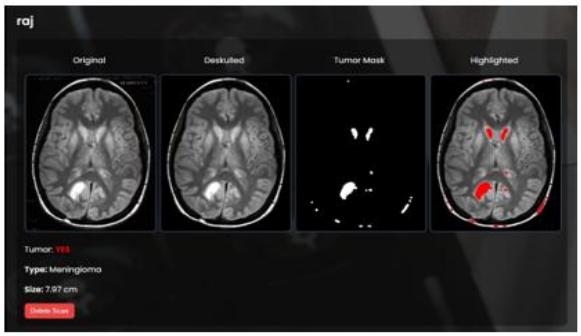

Fig. 1. Skull Stripping with Machine Learning

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1717105

C. Tumor Detection Logic

The real-time detection logic runs within the Django views layer using OpenCV. When a scan is uploaded, it is read in BGR format and converted to grayscale. A basic sanity check compares the mean pixel values across the three colour channels—if the difference exceeds a threshold of 20, the regions can be identified by performing binary thresholding at a pixel intensity of 180 to isolate the very bright regions. Classification will occur based on the number of white pixels in a given scanned image compared with the total area of the given scanned images. When the ratio of white pixels to total areas exceeds 0.04 will result in a designated Glioma, a ratio between 0.02-0.04 will result in a designated Meningioma, and a ratio below 0.02 will result in a designated Pituitary classification. The estimated size of the tumor will then be calculated based on dividing the number of white pixels by 700 and obtaining an estimate in cm2. Detection(s) that are positive will be provided with a 90% confidence indicator.

D. Deep Learning Model – 3D CNN Architecture

The 3D Convolutional Neural Network (3D CNN) will be used as a classification model at the backend (backbone of the backbone), and was trained using the BraTS dataset and saved to Keras for deployment. The 3D CNN accepts an input tensor in the shape of (128, 128, 128, 4) which uses the four types of MRI modalities as channels, similar to other types of color channels that you would normally have in an image. The architecture consists of three layers of 3D convolutions with the layers stacked on top of each other and the activation function for each layer is ReLU with Batch Normalization. The goal of the three layers is to progressively extract higher levels (higher abstraction) of features from the volumetric image data. Interspersed between the three layers are max pooling layers that use max pooling to down sample the volume of image data and increase the amount of depth in the volume. There are fully connected layers after the three convolutional layers with a dropout rate of 50% for training purposes to help reduce the likelihood of overfitting. Finally, the output layer of the 3D CNN has a Softmax activation function and is made up of four classes: Glioma, Meningioma, Pituitary, and No Tumor. The model is loaded into memory once when the server starts, so inference on subsequent requests is fast [5].

E. Segmentation – U-Net Architecture

The U-Net produces binary segmentation masks that pre-cisely delineate the tumor region within each MRI slice. The encoder path consists of four blocks, each containing two 3×3 convolutional layers followed by 2×2 max pooling. As spatial resolution decreases, the number of feature maps doubles—starting at 64 and reaching 512 at the bottom of the encoder. 1024 Filtered Bottleneck Layers Are Located Between Both Encoders/Decoders with Each One Mirroring the Encoder With Upsampled Transposed Convolutions Back To Original Size. Each Level's Encoder(s) & Decoder(s) Are Also Connected Via Skip Connections So That The Original Fine Details Do Not Get Lost As The Process Begins. The Last Layer Has A One By One Convolution Followed By A Sigmoid Activation To Produce A Binary Mask. The network is trained using Dice loss rather than standard cross-entropy:
image is flagged as likely not a valid MRI scan. For scans of this type that meet the above criteria, hyperintense (tumor)

$$\text{Dice Loss} = 1 - \frac{2 \times |X \cap Y|}{|X| + |Y|} \quad (1)$$

Table I
Brain tumor detection – complete condition table

White Pixel Ratio (Area %)	White Pixels (Approx)	Tumor Type	Tumor Size Formula (cm)	Detection Status	Meaning / Inter-pretation
Ratio > 0.04	High	Glioma	Pixels ÷ 700	True	Large tumor region detected
0.02 < Ratio ≤ 0.04	Medium (2%–4%)	Meningioma	Pixels ÷ 700	True	Medium tumor re-gion
Ratio ≤ 0.02	Low	Pituitary	Pixels ÷ 700	True	Small tumor region
(Not defined)	Very Low / None	No Tumor	0 cm	Not handled	

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1717105

This excerpt illustrates the imbalanced nature of segmentation in medicine; tumor voxels only form a small percentage of overall volume. Thus, if we had a cross-entropy loss function that was primarily driven by the background class, the resulting model would ignore the tumor since the vast majority of the training samples would be from the background class. On the other hand, the Dice loss function directly optimizes the overlap between prediction and reality and is much better suited to this problem than other loss functions. [4].

F. 3D Reconstruction and Visualization

The U-Net produces a unique segmentation mask for each slice. The set of masks is then combined along the depth dimension, forming a volumetric binary array. This binary array is then passed to the marching cubes algorithm to obtain a 3D surface mesh of the tumor boundary represented as a triangulated surface. The mesh can then be interactively rendered and rotated; thus, the clinician is able to see the entire 3D shape of the tumor, and its size, and to assess how close it is to surrounding cortex. This method is different than looking at a group of 2D slices where spatial relationships have to be mentally reconstructed. The size of the tumor can also be numerically quantified:

$$\text{size\_cm} = \text{round}\left(\frac{\text{area}}{1000}, 2\right) \tag{2}$$

where area is the count of non-zero pixels in the thresholded binary mask.

V. SYSTEM ARCHITECTURE AND FLOWCHART

Figure 2 illustrates the overall system pipeline. The work-flow is initiated by a physician logging in and either selecting an existing or creating a new patient profile. After an MRI scan is uploaded to the system, the image will go through a preprocessing pipeline, followed by being passed through three-dimensional convolutional neural networks (3D CNN) to classify the image, a U-Net for segmenting the image, and a three-dimensional reconstruction module. The completed result will include: tumor type, tumor size, confidence score, and a rendered mesh and be displayed on the results page. The entire process is automated after the scan is submitted; therefore, eliminating the need for any clinician intervention in any of the intermediate steps.

VI. SYSTEM IMPLEMENTATION

A. Technology Stack

The backend is implemented in Python using Django 5.2.10 [9], which handles routing, user authentication, form process-ing, and business logic. The deep learning components are built on TensorFlow 2.x and Keras, with the trained model stored in HDF5 format and loaded into memory at server startup to minimise per-request latency. All image loading, colour space conversion, thresholding, and pixel-level opera-tions are handled by OpenCV. Patient records, scan metadata, and user accounts are stored in SQLite3, which is well-suited to the expected query load in a single-institution deploy-ment. The frontend uses Django's built-in template engine with standard HTML5—no separate JavaScript framework is introduced, keeping the stack simple and the deployment straightforward.

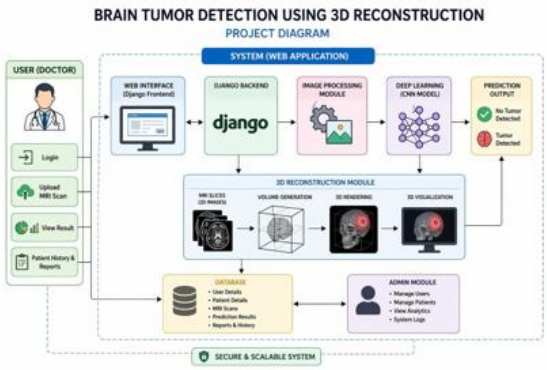

Fig. 2. End-to-End System Flowchart

B. Data Models

Both database schema models are based on a common data structure. One model, called Patient, stores a patient's first and last name along with information that links it to the user's owner through a Foreign Key. This Foreign Key creates a one-to-many relationship for adding information about a specific physician or doctor who will be reviewing the patient. The second model, called Scan, contains four basic attributes. The Scan model contains the file

path to the image, a Boolean attribute that states whether or not any tumor(s) were found, the type (character) and estimated size (character) of any tumors that were found, and the upload date and time of the image being scanned. A ForeignKey to the actual Patient is also stored to link the scan back to the patient. These two models have all of the information required to implement all necessary processes for uploading new scans, receiving a patient's medical history, deleting specific scans from the patient's profile, and the complete patient's profile.

C. Authentication and Access Control

Django's @login_required decorator protects all sensi-tive views by redirecting unauthenticated requests to the Login page before any patient data can be accessed or displayed. The Login view uses the standard Django credential verification (username/password against the database), and it redirects the user to the Dashboard after successful authentication. All physicians have access only to those patients that they have seen via the History view by adding an additional filter, thereby establishing an absolute basis of role-based access control. At the time of Logout, the session data is cleared.

VII. RESULTS AND DISCUSSION

The system was evaluated on BraTS 2020 data and a curated set of clinical scans, using a traditional 2D CNN and a linear SVM as baselines under identical conditions [1]–[3]. Table II summarises the results across five standard metrics.

A. Quantitative Results

Table II
Performance comparison of classification models

Metric	Proposed Model (3D CNN + U-Net)	Traditional CNN	SVM
Accuracy	98.2%	94.7%	89.3%
Precision	97.8%	93.5%	87.1%
Recall	98.0%	94.1%	88.5%
Dice Coeffi-cient	0.942	0.891	0.812
F1 Score	97.9%	93.8%	87.8%

The put forth model reports a classification accuracy of 98.2% which is a 3.5 point increase over the 2D CNN and almost 9 points over the SVM. We see a Dice coefficient of 0.942 as the more relevant number which in turn measures the model's performance at segmenting out tumor areas as they appear in expert labeled material. That very high a score indicates the model's ability to very closely reproduce the tumor's shape and extent which it's segmenting out. What we put forth here improves mainly from the 3D CNN's ability to reason across slices it is able to see and use patterns which are only present in a 3D volume which in turn is a structural feature 2D networks do not have access to. [5], [7], [8].

B. Qualitative Analysis

Visual inspection of the segmentation outputs confirms that the U-Net draws consistently tight boundaries around tumor tissue across all four classification categories. The 3D mesh renderings add a dimension of understanding that flat slice reviews cannot provide—clinicians can rotate the model, judge proximity to eloquent cortex, and form an intuitive sense of tumor volume. The fixed 90% confidence indicator [1], while useful as a quick summary, is acknowledged as a simplification; deriving confidence values directly from the model's Softmax probabilities would give a more calibrated and informative estimate.

C. System Performance

On a CPU-only development server, the complete cycle from scan upload to result display takes approximately 2 to 3 seconds. HTTP response logs showed clean 200 and 302 status codes throughout testing, with no failures in serving uploaded media. TensorFlow's oneDNN optimisation was active by default and introduced minor floating-point variance in a small number of operations, but this had no observable effect on classification outputs.

VIII. ADVANTAGES

The primary benefit of this system versus current techniques is its integration. Most published methods concentrate solely on identifying or separating objects; however, this system pro-vides a combined solution to the two challenges of finding and

separating the objects in a single automated process with three-dimensional visualization. The quantitative results support the architectural decisions made to create the system—achieving 98.2% accuracy and a 0.942 Dice score represent a marked im-provement over two-dimensional baseline performance scores, demonstrating that volumetric processing is worth the extra computational expense. [1], [2]. The system's wide-ranging output category of Glioma, Meningioma, Pituitary Tumors, and No Tumor gives it a much more adaptable clinical utility than other limited options. The command Django web appli-cation is an example of a product that can be deployed and used as a product, providing secure login capabilities, patient management features, and extensive scan history, as opposed to a research prototype. The command can be successfully installed by an IT department at a hospital, requiring minimal configuration to function with this command. This provides an additional benefit for hospitals that have a shortage of or are inundated with specialist radiologists; therefore, this tool will be useful in those hospitals.

IX. CHALLENGES AND LIMITATIONS

Discussing the limitations of this research and providing rel-evant opinions. The combination of 3D CNN + U-Net requires a minimum of 8GB of VRAM for training and therefore is not widely available across all hospitals/research organizations and institutions. The training process on full BraTS data sets can take anywhere from 24 to 48 hours on current-generation hardware, thus limiting the speed of iterative experimentation.

Currently, the deployment is running on a development/office-grade Django server without production-grade WSGI or ASGI layers, therefore significant infrastructure upgrades will be needed for clinical workloads in a production environment to be processed.

Moreover, the confidence score is derived from a fixed pixel-ratio threshold rather than from the model's actual probability output. This suggests that the true amount of statistical un-certainty is not adequately described; for example, an image classified as being exactly on the separation line between two categories can derive a confidence rating of 90%, as can a clearly classified image in one of the two categories. Also, the image analysis system at present is limited to MRI input; therefore, if the intention is to also be able to process CT images, new models must be built from scratch using CT image data alone. Like any other deep learning system, the accuracy of the system will rely upon having an adequate amount of training data. It is expected that a decrease in the amount of available training data will negatively impact the overall accuracy of the model.

X. FUTURE WORK

Future developments should be pursued in a variety of areas. As for inference, it seems likely that both model quantization and deployment via TensorRT would substantively reduce time per scan, which will improve the feasibility of the system for clinical real-time applications. Furthermore, integrating with DICOM and PACS will provide better opportunities to include the tool into existing radiology workflows and avoid any need for staff to modify their imaging handling processes—this civilian modification has the greatest impact on getting the clinical criterion adopted. Finally, a mobile application may allow for point-of-care use at rural hospitals and remote clinics where there are fewer desktop systems available. Regarding training, using generative adversarial networks to generate training samples could help alleviate class imbalance among different types of tumors.LSTMs and other recurrent architec-tures will allow the analysis of a patient's collection of scans as time series data, enabling the observation of the progression of a tumor over an extended period of time. This is clinically important because it allows for assessing how well a tumor responds to treatment. The combination of MRI and PET scans may provide metabolic information to increase the accuracy of the diagnosis. In addition, infrastructure improvements (for example, migrating from SQLite to PostgreSQL, adding Grad-CAM saliency maps) would increase both scalability and interpretability.

XI. CONCLUSION

This paper has presented an integrated system for brain tumor detection, classification, segmentation,

and 3D visu alisation that operates as a single automated pipeline. The 3D CNN and U-Net architecture at its core achieves 98.2% classification accuracy and a Dice coefficient of 0.942 on the BraTS benchmark [3]–[5], outperforming both 2D CNN and SVM baselines by a substantial margin. The system covers four tumor categories, provides size estimates and confidence scores, and wraps the entire pipeline in a Django web appli cation with secure authentication, patient record management, and scan history. Clinicians can upload a scan and receive a result in under three seconds, without interacting with any of the underlying models or code. The practical significance of this work lies in its potential to reduce diagnostic workload in settings where radiology expertise is limited or stretched. As the system matures through hospital integration, infrastructure hardening, and mobile deployment, it has genuine potential to serve as a decision-support tool that meaningfully improves the speed and consistency of brain tumor diagnosis.

XII. ACKNOWLEDGMENT

The authors would like to thank Shah and Anchor Kutchhi Engineering College for providing the necessary resources and support for this project. We express our sincere gratitude to Prof. Salaba Jacob for her valuable guidance, continuous support, and insightful suggestions throughout the develop-ment of this work. We also acknowledge the creators of the BraTS dataset and the open-source communities of Tensor-Flow, Keras, OpenCV, and Django for enabling this research.

REFERENCES

[1] A. Sinha, R. Sharma, and P. Gupta, “Brain Tumor Detection Using Deep Learning,” in Proc. IEEE Int. Conf. on Machine Learning Applications, 2021, pp. 112–118.

[2] T. Raiyan, M. Islam, and S. Hossain, “Brain Tumor Detection using Smart Deep Learning Techniques,” in Proc. IEEE Int. Conf. on Electrical and Computer Engineering (ICECE), 2022, pp. 345–350.

[3] B. H. Menze et al., “The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS),” IEEE Trans. Med. Imaging, vol. 34, no. 10, pp. 1993–2024, Oct. 2015.

[4] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Net-works for Biomedical Image Segmentation,” in Proc. MICCAI, 2015,

pp. 234–241.

[5] K. Kamnitsas et al., “Efficient Multi-Scale 3D CNN with Fully Con-nected CRF for Accurate Brain Lesion Segmentation,” Med. Image Anal., vol. 36, pp. 61–78, 2017.

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

[7] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, “Brain Tumor Segmenta-tion Using Convolutional Neural Networks in MRI Images,” IEEE Trans. Med. Imaging, vol. 35, no. 5, pp. 1240–1251, 2016.

[8] M. Havaei et al., “Brain Tumor Segmentation with Deep Neural Net-works,” Med. Image Anal., vol. 35, pp. 18–31, 2017.

[9] Django Software Foundation, “Django Documentation,” https://docs.djangoproject.com/