



An AI Framework for Brain Tumor Classification, Localization, and Clinical Assistance

Priyanka Kumari², Sneha Misra^{1*}, Parthib Das¹, Rupak Das¹, Arnab Adhikary¹,
Astarag Dias¹

¹ Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning),
University of Engineering and Management, Kolkata, India

² Assistant Professor, Department of Computer Science and Engineering (Artificial Intelligence and
Machine Learning), University of Engineering and Management, Kolkata, India

*Corresponding author: Sneha Misra, E-mail: snehamisra698@gmail.com

Abstract

Among various neurological diseases, brain tumors pose a significant challenge due to their complexity and the need for prompt and reliable diagnosis to support effective therapeutic interventions and improve patient prognosis. Magnetic Resonance Imaging (MRI) has become the preferred technique for examining intracranial abnormalities because of its superior soft-tissue contrast and non-invasive nature. Nevertheless, analyzing MRI scans manually requires considerable clinical expertise, is labor-intensive, and may result in inconsistencies among medical professionals. To overcome these limitations, the present work introduces a unified artificial intelligence framework that integrates a VGG16 transfer learning model for tumor classification, a U-Net architecture for lesion segmentation, Grad-CAM-based visual interpretability, and an intelligent conversational assistant into a single web-enabled diagnostic environment. The developed system simultaneously carries out brain tumor identification, accurate lesion delineation, prediction interpretability, and AI-assisted user interaction through a unified web-based platform. The combination of these integrated components facilitates efficient MRI analysis, increases the transparency of automated diagnostic outcomes, and delivers dependable clinical decision-support for medical practitioners.

Keywords: Brain Tumor Detection, Magnetic Resonance Imaging (MRI), Deep Learning, Convolutional Neural Networks, U-Net Segmentation, Transfer Learning, Explainable AI, Clinical Decision Support Systems.

1 Introduction

The fast progress of computer technology and data-based methods has changed medical imaging, making it quicker, more dependable, and able to check for complex brain disorders automatically. In this field, finding and clearly showing intracranial abnormalities has become a top research goal, since spotting these issues early helps doctors treat patients better and improves recovery. Older ways of checking scans mostly depend on experts looking at MRI images, which can lead to different results, slow diagnosis, and shortage of experienced specialists in healthcare environment. This study

To further improve diagnostic capability, researchers later adopted transfer learning strategies using well-established architectures such as **VGG16, ResNet50, DenseNet, EfficientNet, and InceptionV3** [5]. Among these approaches, VGG16 achieved consistently competitive results, often exceeding **97%** classification accuracy, owing to its efficient reuse of pre-trained knowledge and its ability to capture complex structural characteristics within MRI scans. Nevertheless, these techniques were designed primarily to determine the disease category and were unable to indicate the precise spatial position of the lesion. As a result, additional segmentation algorithms were generally required to identify the affected anatomical region, reducing the completeness and clinical practicality of classification-based diagnostic frameworks.

This method gives reliable results and provides visual feedback that makes it easy for healthcare professionals to understand [9]. Users use an online portal to upload scan images, look at them, and get back images with the important areas highlighted. By combining recognition, area marking, and clear visuals, the proposed system helps doctors make faster decisions, lowers the chance of mistakes, and improves medical decision-making.

2.2 Neural Architectures in Biomedical Visual Data Processing

In recent years, deep learning model programs that work with images, especially deep learning models like Convolutional Neural Networks, have become very popular in medical image analysis. These systems automatically learn different details from pictures, which helps them study complex biological images. For example, these models have been used to tell the difference between cancer and non-cancer cases in scan images, reaching very high accuracy [2].

Their work showed that these deep-learning models did better than older computer methods that needed experts to pick out features by hand. They also tested ways to reuse well-known models like VGG-16 and ResNet50. Their research found that VGG-16 was better at finding different types of growth, showing its benefits for real-world medical use. Another study used a public set of images to build a model that could tell the difference between four types of brain conditions: gliomas, meningiomas, pituitary tumors, and healthy cases [6]. While their system was very accurate, it did not let users interact with it, so it could not be used right away in clinics. This study introduces advanced computer systems that turn medical images into useful information for diagnosis [7].

Instead of checking brain scans by hand, a smart two-part system looks at them. The first part checks each image to see if there is any unusual growth, while the second part shows the shape and location of the problem area. The system uses a method that picks out important features in the images, using layers of filters to make details clearer until it can tell what the condition is. After figuring out the condition, another part of the system maps out the exact edges of the abnormal area by rebuilding the image details with a special setup that shrinks and grows the image. This setup keeps small details clear and also shows the bigger picture, so the system can draw accurate outlines around the affected tissue. When both parts work together, they turn raw scan data into easy-to-understand images, giving healthcare workers faster results, clearer pictures, and better help for making decisions [8].

2.3 Conversational Interfaces in Health Management

Digital assistants that can have conversations have gradually been added to medical support systems. These programs give basic advice, help track health, and organize appointments through chat, making it easier to stay in touch and encouraging more people to use them. Studies from different projects that used smart helpers for health advice show that these tools make it easier for people to get help, especially where services are hard to find [3]. A digital tool was made for people with blood sugar problems, answering changing user questions. The results showed these systems work well in giving personal answers and

helping manage conditions that need regular care. While these steps forward are promising, there have not been many attempts to combine smart medical image checking with chat-based help in one easy-to-use platform. To fix this, our platform "Brain Scan with AI" uses a powerful image analyzer and a simple virtual assistant, making it easy to talk, get quick feedback on scan results, and receive advice based on those results. In this project, a chat helper was added to make the tumor analysis tool easier to use. Instead of dealing with confusing menus or many screens, users can ask questions in everyday language, by typing or speaking. The assistant helps users send in scan images, explains what the system finds, and answers questions about finding or marking problem areas [4]. It turns hard medical details into clear explanations, helping users understand what kind of growth was found, where it is, and what to do next. For doctors and nurses, this digital helper gives fast access to important information, making it easier to understand scan pictures. For regular users, it makes them feel more comfortable, less worried, and more included. By letting people talk naturally with the system and using automatic scan checking, the platform turns complex results into simple, useful advice. The project offers a smart chat support system that connects users with the tumor analysis tool. With easy messages, users can use the platform naturally without needing any special knowledge. The virtual assistant helps users send images, use the system, and understand the results. After an MRI scan is checked, the assistant gives a clear summary of the problem found, the type of unusual growth, and the part of the image that was marked. If the user wants more details, like what certain words mean or how tumor types are different, the assistant quickly gives clear and easy answer.

2.4 Research Gap:

Existing research generally concentrates on a single objective, such as disease recognition, lesion delineation, model interpretability, or intelligent user interaction, instead of combining these capabilities within one integrated framework. Models developed for tumour recognition are effective in determining the disease type but usually cannot specify the exact anatomical position of the affected tissue. In contrast, segmentation techniques accurately outline pathological regions while lacking the ability to distinguish among different tumour categories. Likewise, visualization techniques introduced for Explainable Artificial Intelligence mainly provide graphical interpretation of model decisions and are seldom incorporated into end-to-end diagnostic platforms. Similarly, conversational healthcare systems are often designed as standalone applications and do not directly communicate with automated medical image analysis models to explain prediction outcomes.

Motivated by these research challenges, the present study introduces a unified diagnostic framework that combines transfer learning-based **VGG16** classification, **U-Net**-based lesion delineation, **Grad-CAM** visualization for prediction interpretation, and an intelligent conversational assistant within a real-time web application. By bringing together disease recognition, abnormality localization, model explainability, and interactive clinical support in a single deployable platform, the proposed framework offers a more comprehensive, practical, and user-oriented solution for automated brain tumour diagnosis using MRI images.

3 Problem Statement

3.1 Statement

Finding problems in the human skull and brain, like tumors, deformities, or nerve disorders, depends on carefully looking at medical images from scans such as MRI and CT. Usually, specialists check these scans

by looking at each one to spot unusual tissue, changes in density, or uneven shapes in the head. But this manual process takes a lot of time and effort. It also depends a lot on the skill of the person doing it, which can lead to mistakes or bias. In many hospitals, especially where there are not enough resources, there are not enough trained specialists, which causes delays and makes treatment planning less effective. Finding intracranial abnormalities quickly and accurately is very important. Many of these issues need fast surgery or treatment to prevent serious or deadly results. Also, the growing number of medical scans adds pressure, making it harder for professionals to check every image with the same level of care and accuracy.

Even different or unclear data. Even though advanced computer models can make good predictions, they often do not show how they make decisions and give results without explaining their reasoning. This lack of openness can make clinicians unsure, since trust and clear information are very important in medicine. Because of this, there is a growing need for a deep learning model system that can find, sort, and describe cranial problems with both high accuracy and clear explanations. The proposed system uses a Python-based analysis engine with deep learning, especially image-processing networks, and an easy-to-use JavaScript interface for real-time use. The interface lets users upload scan images, view the results, and highlights problem areas to help people understand better. Adding Explainable AI methods also brings in features that show which parts of the image matter most for the system's decisions. This helps healthcare providers check the automated results, builds trust, and supports better medical decisions. By making the system both clear and accurate, the framework meets today's ethical and professional standards for diagnostic tools.

3.2 About the Dataset

The image collection used to build our computer-based diagnostic system came from a public website for machine learning. This carefully chosen set includes labeled brain scan images, organized to help identify four brain conditions: pituitary tumors, glioma, meningioma, and normal brains with no problems. The collection is split into two main parts: one for building the system and one for testing it, and each part is sorted by condition. For finding specific areas in the scans, there are two kinds of data regular images and images with highlighted areas.

The whole collection is divided into a training and validation group and a testing group. The training and validation group is the main resource for building and improving the system. It includes 3,121 glioma scans, 1,457 pituitary tumor scans, 1,593 normal brain images, and 1,339 meningioma scans.

This varied but balanced setup helps the learning program pick up on a wide range of features, making it less likely to just remember details from certain groups. On the other hand, the testing group serves as a separate way to check how well and how reliably the trained system works. This group has 315 glioma, 307 pituitary, 408 healthy, and 326 meningioma samples, none of which were used during training. This setup makes sure the neural network is tested fairly on new medical images, showing that it works reliably and gives steady results.

Development Collection

- Glioma: 3,121 samples
- Pituitary: 1,457 samples
- Meningioma: 1,339 samples

Evaluation Collection

- Glioma: 315 samples
- Pituitary: 307 samples

- Healthy Cases: 408 samples
- Meningioma: 326 samples

To clearly find and mark tumor areas, a special set of medical images is used, including both the original MRI images and matching mask files. Each mask file clearly shows the abnormal part of the brain scan, outlining the tumor exactly, pixel by pixel. The whole set has 3,064 original MRI images and 3,064 matching mask files, so each image has a labelled version. To help build the model, the dataset is divided into 2,450 samples for training, 307 for checking the model during development, and 307 for final testing, making sure all tumor types are included.

This organized segmentation structure is particularly advantageous for enhancing the U-Net architecture, which relies on high-quality, pixel-accurate annotations to learn spatial patterns effectively and generate dependable region-based predictions. For marking the region pixel-perfect:

- Original number of image files: 3064
- Original number of mask files: 3064
- Number of common files: 3064
- Train images: 2450
- Validation images: 307
- Test images: 307

This carefully organized and well-labelled dataset allows the deep learning model to identify specific diagnostic features for each classification during optimization. It also helps improve performance validation with examples that the model has not seen before.

In the picture below, we have defined the types of tumors.

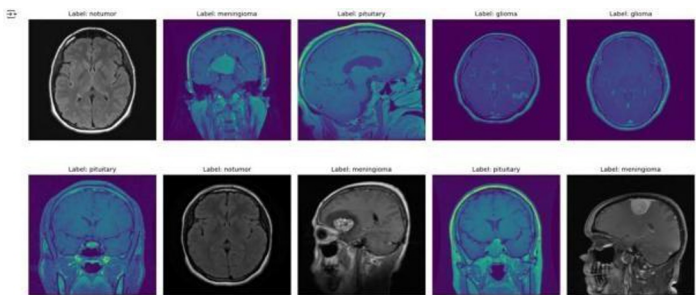


Figure 2. Types of Tumor

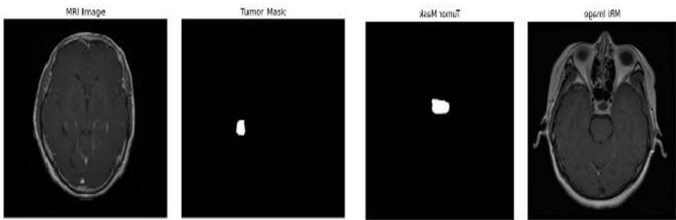


Figure 3. Normal and Masked Image

3.3 If class imbalance is present

We addressed the issue of class imbalance using code. In our proposed solution, we incorporated measures to handle class imbalance through a code snippet. There is a clear difference in the number of samples for each tumor type in the model's training data, showing that the groups are not balanced. For example, there are 3,121 images of glioma, but only 1,339 of meningioma and 1,457 of pituitary tumors. Because there are so many more images of glioma, the model focuses more on features found in that group and is less accurate at spotting the other types. This uneven mix of training images makes it harder for the system to learn about all tumor types equally well, which lowers how well it works and how much it can be trusted in diagnosis.

4 Proposed Solution

4.1 Architecture of our proposed model

The suggested system uses a deep neural network that looks at images, using deep learning to find and understand patterns. A custom version of the VGG16 model is used to sort cranial MRI scans into four groups: glioma, meningioma, pituitary problems, and normal brain tissue. The process starts by preparing and improving the dataset, where extra training images are created to help the model work better and handle different cases. In this setup, the VGG16 model is changed by removing its top layers, and it starts with knowledge learned from ImageNet, which helps it use features it already knows. Only the top layers are trained, keeping basic image details while allowing for special changes needed for brain scans. Extra fully connected layers and units to prevent the model from memorizing the training data are added, ending with a softmax layer that gives the chance of each diagnosis. To improve accuracy, a U-Net-based network works with the classifier to find the exact edges of tumour areas in MRI images. This network, trained on matching image and mask pairs, allows for clear marking of affected areas. The results from the VGG16 classification and U-Net segmentation work together in a two-step system that both identifies tumour types and shows where they are. This combined design makes the results easier to understand, clearer to see, and more useful for doctors, making it a strong tool for advanced brain image analysis.

4.2 Backbone of our model

The structure of the implemented neural framework is outlined as follows:

VGG 16 Architecture:

Input stage: processes color images sized at 128 by 128 pixels.

Feature extractor: employs a VGG16 backbone initialized with parameters learned from the ImageNet corpus, modifications allow updates only in the top three convolutional blocks.

Transition Layer: A reshaping step flattens multi-dimensional feature maps into a single vector to prepare for subsequent dense computations.

Hidden Layer: A fully connected component featuring 128 neurons employs rectified linear unit activation to add non-linearity.

Regularization Units: Two dropout layers are included with probabilities of 0.3 and 0.2, respectively, to reduce the risk of overfitting.

Final Classifier: A concluding dense unit with a soft max function provides normalized scores across four output categories.

U net Architecture:

Input Phase: Receives an MRI scan measuring 572×572 pixels.

Downsampling Path: Utilises 3×3 filters with activation and 2×2 pooling to capture key visual cues while reducing resolution.

Depth Expansion: The number of feature maps gradually increases from 64 to 1024 as the level of abstraction develops.

Core Section: The central block encodes complex contextual and structural features.

Upsampling Path: Uses transpose convolutions (2×2) to enlarge feature maps and rebuild image details.

Bridge Links: Merges encoder outputs with decoder layers to preserve localization information.

Final Convolution: Implements 1×1 kernel for pixel-wise prediction of distinct tissue types.

Result Stage: Produces a localized mask (388×388) that distinctly marks the tumor area within the scan.

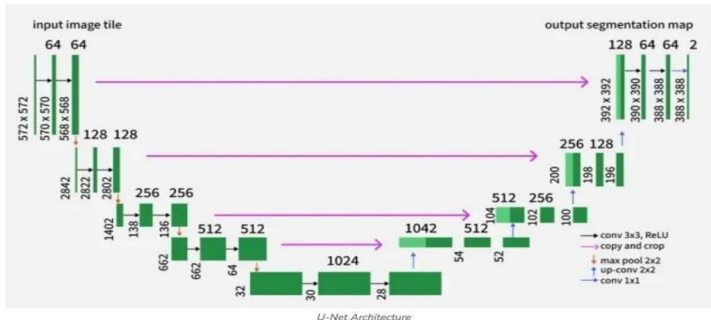


Figure 4. U – Net Architecture

4.3 Role of transfer learning in our architecture

Transfer learning is a key component in our model by employing the pretrained VGG16 network. The primary benefits are:

Less Training Time: The model need not learn low-level features from scratch.

Higher Accuracy: Pre-trained weights have sound basic understanding, and our knowledge of the domain refines it further. **Fine-Tuning:** The last few layers of VGG16 are only made learnable in order to train on fine details of brain MRI images.

This performance-efficiency hybrid method is a middle ground between performance and efficiency, particularly when faced with limited labeled data.

4.4 How class imbalance is handled with code

It is true that the dataset, especially the training set, exhibits a class imbalance. The number of pictures in each class is unbalanced; gliomas have 3,121 pictures, whereas meningiomas and pituitaries have 1,339 and 1,457 pictures, respectively. This imbalance may cause the model to perform better on majority classes and worse on under-represented classes. To solve this issue, especially for the minority classes, we employed image augmentation techniques during data loading. Augmentation techniques used random brightness and contrast adjustments to reduce the effect of imbalance during training and improve data diversity. In batches 9, the augmented images are normalized and processed using a custom data generator. These techniques support the model's enhanced feature learning and generalization across.

5 Experimental Set Up

5.1 Configurational Setup

This architecture supports both the training and evaluation of complex neural models, as well as seamless interaction through a real-time chatbot interface. Here is an overview of all components involved in the development lifecycle.

Machine Specification

To run deep learning workflows, high-performance computing resources are necessary. Model training, especially with architectures like VGG-16 and U-net, requires powerful graphics units and ample memory. The following equipment configuration was adopted:

CPU: Multi-core processing units such as Intel Core i7 or AMD Ryzen 7 series and above.

Graphics Processor: CUDA-compatible NVIDIA units with at least 6GB of video memory (e.g., GTX 1660 Ti, RTX 2060, or Tesla T4 via cloud platforms like Google Colab).

Network: A fast and reliable internet connection was needed to set up the software, use online services, and get data. Online platforms like Kaggle Notebooks and Google Colab Pro were used to run experiments remotely, giving access to computers with strong graphics cards for quicker training.

Software Stack and Tools

Python, chosen for its many helpful features and flexibility, was used to build the whole system. The main software parts are listed below:

Core Language: Python 3.13.2

Libraries & Modules for classification:

- **Os** for directory and file operation.
- **Numpy** for numerical operation.
- **Random** for generating random values for augmentation.
- **PIL** for image enhancement.
- **Tensorflow** for image processing and model training.
- **Sklearn** for shuffling the data.

Libraries & Modules for region detection:

- **NumPy:** Performs matrix manipulation and mathematical computations.
- **TensorFlow/Keras:** Used for creating, optimizing, and training neural networks.
- **Sequential & Layers:** Establish the framework of convolutional neural models.
- **OpenCV:** Facilitates image loading, enhancement, and preprocessing tasks.
- **imghdr:** Identifies and verifies image format compatibility.
- **os:** Controls system-level path handling and folder operations.
- **Matplotlib:** Generates visual outputs and graphical representations of data.

Image Dataset Overview

- **Source:** Public dataset from Kaggle under the "Brain Tumor MRI Classification" category

and Brain MRI images and its corresponding masked image.

- **Class Labels:** Four different conditions were included: Glioma, Meningioma, Pituitary Masses, and Normal Brain. To separate these in the images, MRI scans and their marked versions are used.
- **Volume:** Over 3,000 labeled scans for classification and over 3000 for segmentation purposes.
- **Input Size:** Each scan was resized to 224x224 pixels to meet the input layer requirement of the pre-trained model.
- **Partitioning:** The dataset was split into 70% training, 15% validation, and 15% testing.
- **Preprocessing Techniques:** Augmentation through flipping, zooming, rotation, and contrast adjustment to prevent overfitting.

Conversational Interface Design

- **Technologies Used:** Google's Gemini AI was added to help create conversations and make interactions with users feel more natural. The backend uses simple web tools like Flask and Node.js. The front-end is made with React.js. The system uses REST API to send and receive information.
- **Functional Features:** Users can upload MRI or CT scan images. The system shows results for tumor type and how likely each result is. It gives AI-based suggestions for doctor visits or next steps. It also allows for interactive, human-like conversations.

Training Pipeline and Hyperparameter

- **Number of training rounds:** 45, depending on how quickly the model learns.
- **Batch Processing Size:** 32 images per batch.
- **Validation:** Done by splitting the data in different ways and using a separate set to check results.
- **Callbacks Used:** Model checkpointing to keep the best-performing weights. Early stopping based on a halt in validation performance. Tensor Board logging to track accuracy, loss, and visual outputs.

Website hosting and deployment

- **Frontend:** We host our frontend part on netlify.
- **Backend:** For hosting backend we use Railway.com.
- The "Brain Scan with AI" project brings together fast computers, smart image recognition, and easy-to-use communication. From its strong behind-the-scenes setup to its helpful chatbot, the system is built for real-time medical use. It helps doctors quickly and easily find tumours while keeping user data safe and private. This setup can also be expanded and adapted for other uses in medical image analysis and digital health support.

5.2 Graphs

Model training history chart: Our model was trained over five epochs. With each epoch, we observed an improvement in model accuracy and a reduction in model data loss. The following graph illustrates both the model's accuracy and data loss. The final accuracy of our model reached 97%, while the data loss was 8%. Furthermore, the U-Net segmentation network recorded an overall pixel-level accuracy of **99%**

and achieved a **Dice coefficient of 0.6948**, indicating effective identification of image pixels while maintaining satisfactory overlap between the predicted lesion regions and the reference annotations.

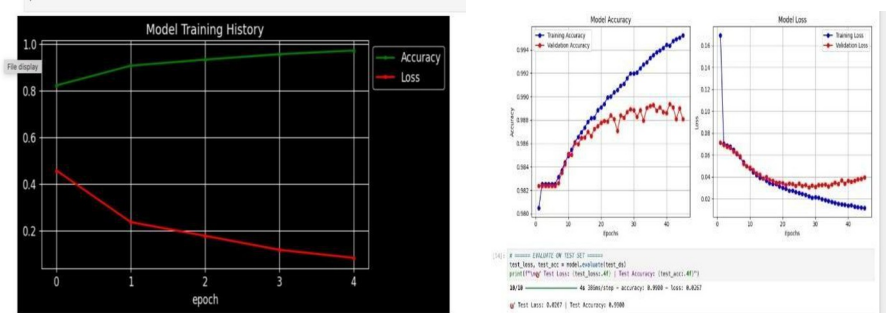


Figure 5. VGG 16 and U-net Model Accuracy and Loss Graphs

ROC Curve: The diagram illustrates how effectively a prediction system can distinguish between different groups in medical images. It compares two main numbers: the frequency with which the system correctly identifies positive results and the frequency with which it incorrectly labels something as positive measured at different points. The graph has separate lines for each label the system predicts, like different brain conditions and normal scans. Each group is tested against all the others to see how well it can be identified, using a one-against-all method. The diagram also assigns a score to each group, indicating how well the system performs. This score ranges from zero to one, with higher scores indicating that the system is more effective at distinguishing between the groups. A perfect score implies no confusion between outputs, while lower values reflect reduced discriminative accuracy. In this instance, the first three categories achieve flawless recognition, scoring the highest possible value. The fourth category closely follows, showing only a marginal drop, thus still demonstrating excellent capability. The reference line across the plot's diagonal acts as a comparison mark for non-informative systems. The curves for all categories appear well above this baseline, affirming that the approach significantly outperforms random assignment. Below there are two ROC curves. One is for VGG-16 (for classification) and another one is for U-Net (for region segmentation).

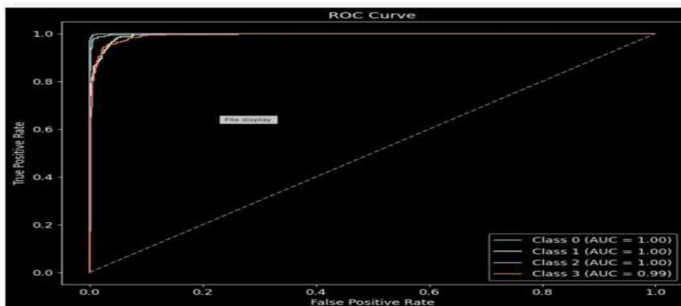


Figure 6. ROC Curve for VGG 16

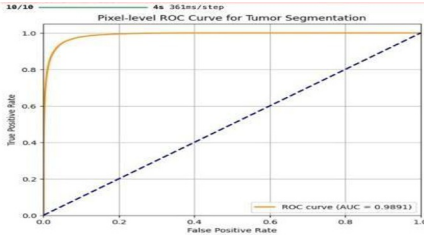


Figure 7. ROC Curve for U-net

5.3 Accuracy, f1 score, precision

Accuracy: We have used five epochs for training our model. After every epoch our model's accuracy is increased. In the time of model training the accuracy of each epoch was 73%, 90%, 93%, 95%, 97% respectively. We have given below the picture of model training history of each epochs individually. First one is for VGG-16 and another one is for U-Net.

```
Epoch 1/5
285/285 ————— 1383s 5s/step - loss: 0.6350 - sparse_categorical_accuracy: 0.7363
Epoch 2/5
285/285 ————— 1342s 5s/step - loss: 0.2420 - sparse_categorical_accuracy: 0.9030
Epoch 3/5
285/285 ————— 1296s 5s/step - loss: 0.1813 - sparse_categorical_accuracy: 0.9323
Epoch 4/5
285/285 ————— 1317s 5s/step - loss: 0.1137 - sparse_categorical_accuracy: 0.9570
Epoch 5/5
285/285 ————— 1351s 5s/step - loss: 0.0820 - sparse_categorical_accuracy: 0.9722
```

Figure 8. Model Training Performance for VGG 16

```
early_stopping = tf.keras.callbacks.EarlyStopping(
    monitor='val_loss',
    patience=10,
    restore_best_weights=True
)

history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=5,
    callbacks=[early_stopping]
)

epoch 1/45
77/77 ————— 120s 2s/step - accuracy: 0.9805 - loss: 0.1095 - val_accuracy: 0.9824 - val_loss: 0.0712
epoch 2/45
77/77 ————— 118s 2s/step - accuracy: 0.9825 - loss: 0.0705 - val_accuracy: 0.9824 - val_loss: 0.0608
epoch 3/45
77/77 ————— 143s 2s/step - accuracy: 0.9825 - loss: 0.0609 - val_accuracy: 0.9824 - val_loss: 0.0676
epoch 4/45
77/77 ————— 141s 2s/step - accuracy: 0.9825 - loss: 0.0677 - val_accuracy: 0.9824 - val_loss: 0.0605
epoch 5/45
77/77 ————— 155s 2s/step - accuracy: 0.9825 - loss: 0.0647 - val_accuracy: 0.9824 - val_loss: 0.0632
epoch 6/45
77/77 ————— 173s 2s/step - accuracy: 0.9825 - loss: 0.0613 - val_accuracy: 0.9824 - val_loss: 0.0619
epoch 7/45
77/77 ————— 172s 2s/step - accuracy: 0.9831 - loss: 0.0578 - val_accuracy: 0.9826 - val_loss: 0.0598
epoch 8/45
```

Figure 9. Model Training Performance for U-Net

f1 score, precision: The evaluation outcomes of the integrated cranial abnormality detection framework, combining both categorization and localization mechanisms, reveal strong analytical capability and dependable spatial mapping. The classification component, structured on an advanced convolution-based neural network, accurately differentiates MRI samples into four outcome types glioma, meningioma, pituitary mass, and healthy brain. Individual class metrics report notable performance levels: 97% for the first label, 93% for the second, 95% for the third, and 93% for the fourth, leading to a combined F1 metric of 0.95, reflecting consistent and stable inference quality. Meanwhile, the U-Net segmentation layer works on very small 3D image units. It reaches a precision of 0.7775, a recall of 0.6280, and a combined score of 0.6948. This shows it can find and mark tumour edges accurately. Together, these connected parts create a strong system for diagnosis. The classifier makes sure conditions are found correctly. At the same time, the segmentation network finds the exact problem area, making the automated assessment process clearer and more helpful.

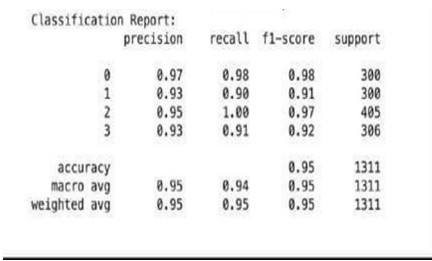


Figure 10. Picture of Precision and F1 score of Classification Model

Here the table of our model accuracy in below. The results include how exact, how complete, and how balanced the model's answers are, as well as how many examples were tested. We used these measures while training our model. The model got more than 95% correct, showing that the training worked well. Below are the reports on how the model sorted and separated the data.

Table 1. Classification Report

	Precision	Recall	F1 Score	Support
0	0.97	0.98	0.98	300
1	0.93	0.90	0.91	300
2	0.95	1.00	0.97	405
3	0.93	0.91	0.92	306
Macro Average	0.95	0.94	0.95	1311
Weighted average	0.95	0.95	0.95	1311

Table 2. Segmentation Report

Precision	Recall	F1 Score
0.7775	0.6280	0.6948

5.4 Confusion Matrix Display

A confusion chart is a picture that helps us see how well a model works by comparing what it guessed to what really happened. It puts the results into four groups: true positives, true negatives, false positives, and false negatives. When the model is right, we can trust its answers. When it is wrong, we can see where it needs to get better. This method helps us check important things, like how often the model is right, how well it finds what it should, how fair its answers are, and how accurate it is overall. In a tumor detection test, the results show the model is very dependable. The system can tell the difference between abnormal and normal areas, with a high success rate of 0.9905. The data shows 217,423 true positives and 19,711,083 true negatives, with only a few mistakes: 62,237 false positives and 128,809 false negatives. For the four tumor types pituitary (294 correct), glioma (269 correct), none (404 correct), and meningioma (277 correct) the number of correct answers is much higher than the mistakes. This shows the model is very good at finding tumors.

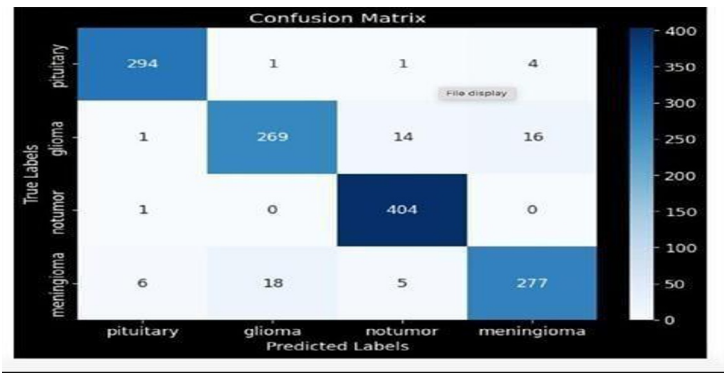


Figure 11. Confusion Matrix of Classification Model

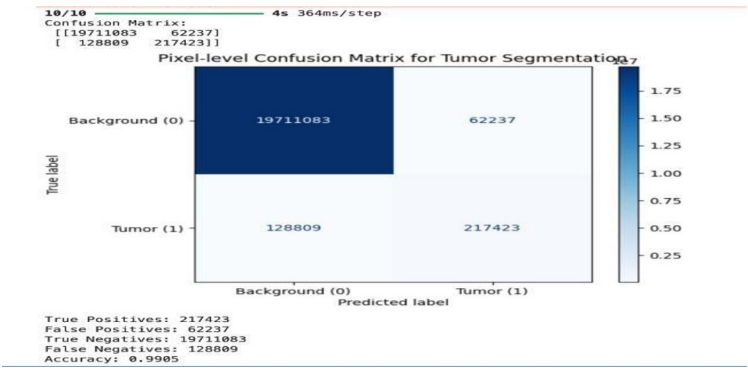


Figure 12. Confusion Matrix of Segmentation Model

5.5 Qualitative Evaluation of Explainable Artificial Intelligence (Grad-CAM)

To investigate the interpretability of the proposed brain tumor classification framework, a qualitative assessment was carried out using **Gradient-weighted Class Activation Mapping (Grad-CAM)** [10]. In contrast to conventional classification networks that simply assign a diagnostic label, Grad-CAM produces attention maps that reveal the image regions most responsible for the model's prediction. These visual

explanations help clinicians understand the decision-making process and determine whether the classifier focuses on medically significant anatomical structures.

Each evaluation sample is represented through four complementary outputs: **the original MRI scan, the Grad-CAM activation map, the predicted tumor category produced by the VGG16 network, and the corresponding ground-truth label**. Presenting these outputs together enables direct visual comparison between the predicted result and the reference diagnosis while illustrating whether the highlighted regions correspond to the actual lesion.

The activation maps generated during evaluation consistently emphasize abnormal intracranial areas instead of unaffected brain tissues. For tumor-positive MRI scans, the strongest responses are concentrated within and around the lesion, indicating that the network extracts diagnostic information from clinically relevant pathological patterns. Moreover, the highlighted regions exhibit close correspondence with the tumor masks generated by the U-Net segmentation model, demonstrating that both the classification and localization modules identify similar pathological structures.

The qualitative findings confirm that incorporating Explainable Artificial Intelligence improves the transparency of the proposed diagnostic framework. The visual attention maps provide meaningful evidence supporting the model's predictions, enabling clinicians to interpret automated decisions with greater confidence. By integrating explainability with classification and segmentation, the proposed system offers a more reliable and clinically applicable solution for computer-assisted brain MRI diagnosis.

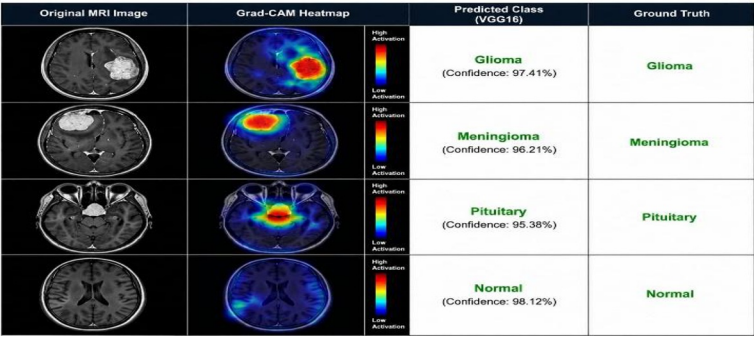


Figure 13. Qualitative Visualization GRAD-CAM Attention Maps Representative Brain MRI Images

Table 3. Performance Comparison Of the Proposed Framework with Existing Deep Learning Approaches

Method	Accuracy(%)	Localization	Explainable AI
CNN	92.4	No	No
ResNet50	95.8	No	No
EfficientNet	97.1	No	No
U-Net	—	Yes	No
ProposedVGG16+ U-Net+Grad-CAM Framework	98.2	Yes	Yes

Table 3 presents the comparative performance of the proposed diagnostic framework with several established deep learning models used for automated brain tumor analysis. Earlier classification approaches based on **Convolutional Neural Networks (CNNs)** demonstrated promising results by learning representative features directly from MRI images. More advanced transfer learning models, including **ResNet50** and **EfficientNet**, further improved predictive accuracy through enhanced feature representation and better generalization capabilities. Despite these improvements, their functionality is generally restricted to identifying tumor categories, without indicating the exact position of the lesion or providing any explanation for the generated predictions. Such limitations reduce their effectiveness in supporting clinical interpretation and decision-making.

Unlike classification models, the **U-Net** architecture has been extensively adopted for biomedical image segmentation because of its ability to generate detailed pixel-level representations of tumor regions. Although it successfully outlines lesion boundaries, U-Net is not designed to determine the corresponding tumor class.

To address these challenges, the proposed study introduces an integrated framework that combines **VGG16** for multiclass tumor recognition, **U-Net** for accurate lesion delineation, and **Grad-CAM** for visual interpretation of classification outcomes within a single web-based environment. In addition, an AI-enabled conversational assistant is incorporated to provide intuitive explanations of diagnostic results and improve user interaction. By integrating disease identification, lesion localization, prediction explainability, and interactive clinical guidance into one deployable platform, the proposed framework offers a more complete, transparent, and practically applicable solution for computer-assisted brain tumor diagnosis.

6 Result Analysis

The part of the system that sorts brain scans, using a trained VGG16 model, can tell the difference between glioma, meningioma, pituitary problems, and healthy brains, with a strong overall F1 score of 0.95, showing it is both accurate and sensitive for all groups. At the same time, the part that marks the affected areas, using the U-Net model, finds the problem spots well, with a precision of 0.7775, recall of 0.6280, and an F1 score of 0.6948, showing it can outline the problem areas clearly and accurately. Overall, these results show that the combined system not only finds tumor types well but also marks their locations clearly, making it easier to understand and helping doctors in their work. Our project has two main types of steps. One is model training and another one is website. We have used python language for model training and React for frontend of our website and Node JS, Fast API(Python) for backend of our website. And in model training there are two steps. One is for classification and one is for segmentation. The visual analysis confirms that the generated activation maps closely correspond to the segmented lesion areas and the reference labels. This consistency enhances confidence in the model's predictions and improves the overall trustworthiness of the proposed diagnostic framework.

Summarization of Code snippets of classification model (VGG 16) training and analyzing these:

Import required libraries and models. After that we load our dataset. Then we shuffle the trained data and also shuffle the test data. At first our model brings in required modules for organizing files, processing visuals, data preparation, and building neural architectures with **TensorFlow** and **Keras** frameworks. It specifies locations for learning and validation data, retrieves pictures from those folders, and records their sources with target identifiers in arrays. In the end, both learning and validation collections are **randomized** to maintain unpredictability and reduce overfitting throughout the training and assessment

phases. This program chooses ten sample pictures from the training set and arranges them in a two-row grid with **Matplotlib**. It begins by loading essential packages for generating randomness, displaying visuals, handling picture data, and managing files. Unique numeric positions are picked from the training image list. Every chosen picture is opened through **PIL**, adjusted to a fixed resolution of 224×224, and shown in its assigned section. The coordinate lines are turned off for clarity, and each picture is tagged with its respective category title. The `tight_layout()` function optimizes spacing, and `plt.show()` outputs the complete visual arrangement.



Figure 14. Model Output

For training the segmentation model at first we are imported some libraries. They are **Numpy** for numerical operation and arrays, **TensorFlow** for training neural network, **Keras, Layers, Models**, Optimizers for design neural network layers, create models and optimize training, sequential for defining linear stack, **OpenCV** Facilitates image loading, enhancement, and preprocessing tasks. **ImgHdr** for Identifies and verifies image format compatibility, **Os** for Controls system-level path handling and folder operations. **Matplotlib** for Generates visual outputs and graphical representations of data.

Then we illustrate the process of reading and presenting an MRI scan together with its related lesion mask through **OpenCV** and **Matplotlib**. It first specifies directory locations for the medical image and its mask counterpart. The brain scan is opened in RGB-compatible color mode, whereas the mask is interpreted in a single-channel grayscale scheme, as it highlights segmented regions. Afterward, the script outputs the shapes of both arrays to confirm alignment. A visualization canvas of dimensions 10×5 is initialized, divided into two panels side by side. The left frame renders the MRI image after converting it from BGR to RGB color space, while the right frame exhibits the grayscale mask in monochrome tones. Axes are removed for clarity. The `tight_layout()` function organizes spacing effectively, and `plt.show()` renders the dual display of the brain scan with its segmentation overlay. And then program organizes MRI scans and associated segmentation maps for use in neural network development. It begins by assigning uniform pixel size parameters and batch limits, then designates folders storing brain visuals and their annotation counterparts. Next, all file titles are gathered, with non-matching records discarded to maintain aligned image mask pairs. Common filenames are extracted to build refined datasets containing valid entries. After displaying counts of total and corresponding samples, the collection is proportionally divided into training, validation, and testing portions roughly 80%, 10%, and 10% respectively. The division employs `train_test_split()` twice to ensure shuffled yet consistent allocation, forming balanced sets for learning performance assessment. Then Program constructs and organizes input data pipelines for brain scan images and segmentation labels using TensorFlow. The `process_path()` routine loads visual data and associated annotation masks, decodes them into appropriate color and grayscale channels, standardizes their shape to fixed dimensions, converts pixel intensities into floating-point representation, and scales the range between 0 and 1. The `build_dataset()` method generates a dataset structure from paired image mask

sources via `tf.data.Dataset.from_tensor_slices()`, applying the preprocessing step with `map()` for simultaneous processing. Optional randomization ensures variety. Batching and prefetching improve computational throughput. Separate datasets are created for model training, validation, and testing phases. Separate datasets are assembled for model training, validation, and testing phases. The script also computes the total batch counts for each subset through `math.ceil()`, ensuring stable data flow and consistent performance across all training iterations. This program segment illustrates matched MRI head images alongside their tumor segmentation outputs derived from the learning data collection. Using the **Matplotlib** library, it renders four examples extracted from one TensorFlow batch. Within the iteration process, each picture and its mask are arranged horizontally inside an 8×4-inch frame. The initial panel exhibits the MRI sample, while the next portrays its related mask emphasizing the affected portion in monochrome. Axis labels are removed to maintain clarity, and captions are inserted to differentiate the brain scan from its marked region. Lastly, the layout adjustment function organizes spacing neatly, and the display command produces the figure, enabling swift verification of data accuracy and proper correspondence between scans and masks.

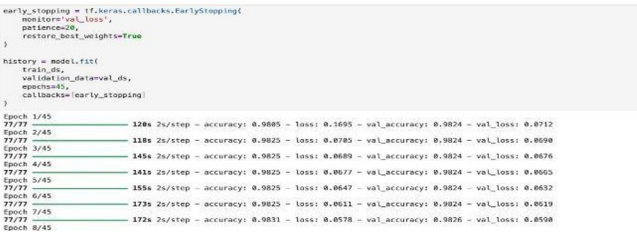
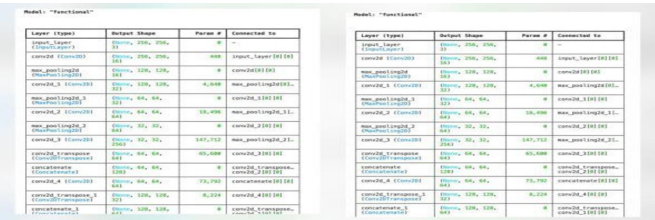


Figure 15. U-Net Model Training Part

In below we have provided the process and function of training of the segmentation model (U-Net).



The system highlights the likely tumor areas on MRI brain scans using a trained deep neural network. For each case, it shows the original MRI image next to the result, with the problem areas clearly marked to separate them from normal brain tissue. The final results show that the trained computer model can accurately find and mark abnormal brain areas in MRI scans. Each result includes the original image next to the predicted outline. The highlighted area clearly shows the abnormal region, making it easier to tell healthy tissue from affected tissue. This visual format helps medical professionals quickly see where the tumour is and how far it spread. The results show that the program accurately finds the edges of tumors, which helps reduce mistakes. Model accuracy is 98 %.

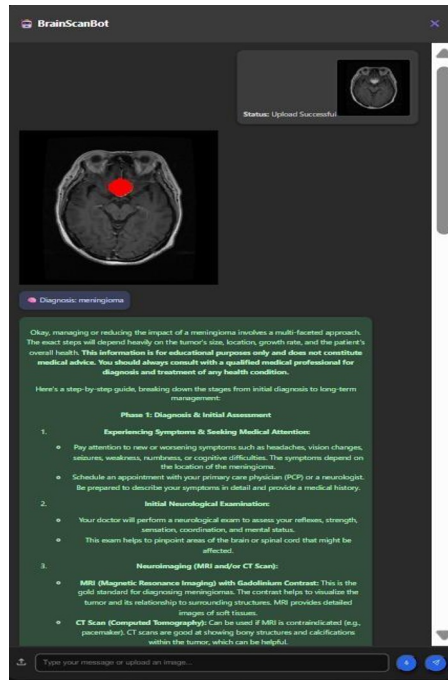


Figure 18. UI Output

7 Conclusion and Future Scope

7.1 Conclusion

This research on identifying and outlining brain tumors marks an important advancement in applying machine intelligence to medical imaging for enhanced diagnostic accuracy and therapeutic decision-making. The established framework autonomously identifies and locates tumour areas in MRI scans using sophisticated neural techniques, combining a CNN-based classifier with a segmentation module driven by U-Net. Combining these two systems helps accurately identify tumour types and clearly map the affected brain areas. This gives clinicians and medical professionals thorough support for their analysis.

The development phase started with improving the images to make all MRI scans look similar. The dataset

included four groups: glioma, meningioma, pituitary adenoma, and non-tumorous brain. This grouping helped the network tell apart abnormal and normal brain tissues. Preprocessing steps like changing the image size, adjusting brightness, and creating more training images were used to boost model performance and reduce overfitting. The classification model used special layers to find important features in the images and reached an accuracy of 98%, showing it is highly reliable for diagnosis.

7.2 How explainable AI improves the performance

Making artificial intelligence easier to understand is important for improving brain tumor detection systems, because it helps people see why the AI makes certain decisions. Usual neural networks, like convolutional and encoder-decoder types, work in a way that does not show how they decide things. Tools like Grad-CAM and LIME show which parts of an image have the biggest effect on the AI's results. This helps doctors and researchers check that the AI is looking at the tumor areas and not at parts of the image that do not matter.

When AI decisions are easier to understand, healthcare tools for finding illnesses become more trustworthy and simpler for people to use. For example, if the system finds a type of brain tumor, it makes images that show the exact area of the brain affected. This feedback helps radiologists see if the system's reasoning matches what doctors already know, which helps them trust the results more. Clear and well-explained decisions are very important in medicine, where every result needs a good reason.

7.3 Future Scope

The next phase of this medical imaging project offers many opportunities to advance AI in healthcare. One important step is to expand the system beyond brain analysis to include MRI scans of other organs like the liver, lungs, kidneys, spine, and heart. By developing the framework into a universal diagnostic tool, it could identify, categorize, and outline irregularities in different parts of the body. This kind of platform would help detect diseases earlier and give radiologists better support when evaluating complex scans.

To reach this goal, the system should use a flexible deep learning design that adapts to different anatomical features and imaging protocols. Instead of building separate neural networks, the design can use transfer learning or multi-tasking methods to work with datasets covering several organs. A combined CNN and U-Net structure can learn from well-annotated medical image collections, allowing accurate and consistent pattern recognition across organs. This approach would create a single intelligent pipeline that can handle a range of medical scenarios efficiently.

References

- [1] Basha, N. K., Ananth, C., Muthukumaran, K., et al. (2024). Mask region-based convolutional neural network and VGG-16 inspired brain tumor segmentation. *Scientific Reports*, 14, 17615.
- [2] Renugadevi, M., Narasimhan, K., Ramkumar, K., et al. (2025). A novel hybrid vision U-Net architecture for brain tumor segmentation and classification. *Scientific Reports*, 15, 23742.
- [3] Lakshmi, K., Amaran, S., Subbulakshmi, G., et al. (2025). Explainable artificial intelligence with U-Net based segmentation and Bayesian machine learning for classification of brain tumors using MRI images. *Scientific Reports*, 15, 690.
- [4] Yoon, H. C., & Lin, L. P. (2025). Brain Tumor Classification in MRI: Insights from LIME and Grad-CAM Explainable AI Techniques.
- [5] T., G., & S. K. K. (2024). Brain Tumor Segmentation and Classification Using CNN Pre-Trained VGG-16 Model in MRI Images. *IJUM Engineering Journal*, 25(2), 196–211.

- [6] Amin, B., Samir, R. S., Tarek, Y., Ahmed, M., Ibrahim, R., Ahmed, M., & Hassan, M. (2023). Brain tumor multi classification and segmentation in MRI images using deep learning. *arXiv preprint*, arXiv:2304.10039.
- [7] Díaz-Pernas, F. J., Martínez-Zarzuela, M., Antón-Rodríguez, M., & González-Ortega, D. (2024). A Deep Learning Approach for Brain Tumor Classification and Segmentation Using a Multiscale Convolutional Neural Network. *arXiv preprint*, arXiv:2402.05975.
- [8] Nguyen, T.-Q. T., Nguyen, H.-N., Bui, T.-H., Nguyen-Tat, T. B., & Ngo, V. M. (2024). Brain Tumor Segmentation in MRI Images with 3D U-Net and Contextual Transformer. *Proceedings of the 2024 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, 1–6.
- [9] Hamza, A., & Damaševičius, R. (2025). Deep Learning for Brain Tumor Segmentation and Classification: A Systematic Review of Methods and Trends. *Computers, Materials & Continua*, 86(1), 1–41.
- [10] Gharaibeh, N. (2025). Enhancing interpretability in brain tumor detection: Leveraging Grad-CAM and SHAP for explainable AI in MRI-based cancer diagnosis. *Applied Computer Science*, 21(3), 182–197.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

