VAAL UNIVERSITY OF TECHNOLOGY

Inspiring thought. Shaping talent.

Faculty	:	Applied and Computer Sciences
Department	:	Computer Sciences
Subject	:	Business Analysis 3.2
Subject Code	:	AIBUY3A
Lecturer	:	Mr M Matsela
Moderator	:	Mr XP Piyose
Year	:	2025

Assessment Name:	Project Documentation – QUANTUM.AI
Assessment Date:	13 October 2025

STUDENT NUMBER	SURNAME AND INITIALS	SIGNATURE
223515760	Chigbu CCI	
223232327	Chauke VL	
223412325	Bopape AA	
223307211	Moeletsi E	
224119052	Manana KOM	
223389706	Shangule NT	
223082252	Segale T	
223354147	Molise L	

DECLARATION

We,

Chidi Chibuikem Iheanacho Chigbu

Vushaka Lyborn Chauke

Angel Adequate Bopape

Emily Moeletsi

Katleho Oreratile Masego Manana

Nseketelo Tishenia Shangule

Tumelo Segale

Lesego Molise

declare that:

- The contents of this project represent our own unaided work, and the project has not previously been submitted for academic examination towards any qualification.
- We have not engaged in plagiarism or the use of unauthorized materials within our work.
- All resources used in this work have been appropriately cited and acknowledged.
- This work represents our own opinions and not necessarily those of the Vaal University of Technology.

Signed	Date
Signed	Date
Signed	Date
Signed	Date
Signed	Date
Signed	Date
Signed	Date
Signed	Date

Table of Contents

Table of Figures

DOCUMENTATION ASPECTS

1. Brainiac’s Relevance To The Theme

The diagnosis of brain tumors is one of the most critical tasks in healthcare. Despite its significance, it remains a time-consuming activity that requires a lot of resources and expertise to transition from differentials to diagnosis.

Medical practitioners often rely on manual inspection of MRI scans, a process that is prone to human errors even in developed countries. Our proposed solution leverages the BRISC2025 dataset – a high-quality, vetted dataset containing multiple MRI scans already reviewed by experts in radiology and neurology - to build a solution that assists and supports healthcare at Chris Hani Baragwanath Hospital in tumour detection and classification.

The solution adopts deep learning for image classification to identify the type of brain tumor (Glioma, Meningioma, Pituitary Adenomas, or No Tumor). This AI solution also aligns with South Africa’s goal of leveraging the power of AI technologies to create innovative ideas and solutions in the healthcare industry. This will reduce diagnostic workload and provide for faster response from diagnosis to corrective actions for such a life-threatening condition.

2. Business Objectives

- To reduce the time required for diagnosis.
- To improve the accuracy of brain tumor diagnosis by leveraging the BRISC 2025 dataset.
- To improve patients’ prognosis by early detection and allowing for efficient clinical decision-making.
- To reduce healthcare costs for both patients and hospitals without sacrificing quality decision-making.
- To demonstrate how AI solutions like brAiniac can automate medical processes.
- To promote the use and adoption of AI solutions like brAiniac in more remote and low-resource locations in South Africa.

3. Business Success Criteria

- To achieve more than 80% overall test accuracy when classifying brain MRI scans into the different categories: Glioma, Meningioma, No tumor, and Pituitary.
- To process 1000 test images and generate a confusion matrix and a classification report in less than 5 minutes on limited CPU power (<= 16 GB ram).
- To be able to predict on a single brain MRI scan and classify it while showing a confidence score for each diagnosis in less than 10 seconds.
- To achieve more than 80% user adoption from participating radiologists and neurologists from Chris Hani Hospital.

4. Business Background

With increasing demand for a better quality of life, healthcare has been a paramount topic. Due to this necessitated demand, South Africa faces a scarcity of key specialists in (neurology and radiology) as seen in this article from UCT news of 2021, stating that there are 150 neurologists and only about 35 work in public hospitals like Chris Hani Baragwanath Hospital. These shortages, together with equipment constraints, contribute to MRI backlogs that delay time-critical diagnosis, as cited in an article from HEALTH-E News stating, "About 3500 patients are currently awaiting MRI scans, underscoring a severe diagnostic backlog".

We at QUANTUM.AI, having recognised these constraints the public health sector faces in South Africa, have developed a practical AI decision–support tool that analyses brain MRI scans to detect and classify tumors while reducing workloads for clinicians. By automating parts of the diagnostic process with deep learning, brAIniac provides a fast, consistent, and diagnostic assessment with confidence scores, thus helping medical experts prioritize cases and accelerate the time taken to medical interventions.

5. Requirements

Functional requirements

- The system will accept brain MRI scans in digital image formats such as JPG.

- The system must normalize and pre-process images before analyzing.
- The system must predict each MRI image into one of the following classes of [glioma, meningioma, no_tumor, pituitary] as well as return the confidence score.
- The system will generate summary reports on test data.

Non-Functional requirements

- The system will handle at least 30 MRI scans per hour concurrently.
- The system will maintain consistent performance across different MRI scan images.
- The system will support expansion to other hospitals in the future.
- The system must be highly available about 99.99% of the time.
- The system will have a user-friendly dashboard for non-technical users to navigate easily.

6. Constraints

- The system analyzes brain MRI images only. Other medical imaging, such as PET, CT, and X-rays, as well as other body regions, are out of scope.
- Classification is currently restricted to four classes [glioma, meningioma, no_tumor, pituitary]
- Model performance is validated only on data similar to BRISC2025.
- Testing of the system will be done on a limited compute workstation (less than 16 GB RAM).

7. Risks

RISK	**LIKELIHOOD**	**IMAPCT**	**MITIGATION**
Misclassification of tumors	Medium	High	Ensure all images are normalized and sized appropriately. If the confidence score is below 80% then probe for further review.
Over-reliance	Medium	High	Disclaimer for decision- support only.

			Transparent reporting system along with confidence score prediction to prevent blind trust.
Privacy data breach	Low	High	Comply with South Africa's POPIA Act and principles of least privilege to collect information about patient data. Clearly define what data is retained. Implementing RBAC when being deployed.
Poor User Adoption	Medium	High	2-week pilot operation to teach about the system. Appoint a champion from Chris Hani Hospital. Provide user documentation.
Technical failure	Medium	High	Host solution on cloud services with high availability incorporated in their SLA, as well as other services, such as the fault domains and availability zones.

Table 1: Risk table

8. Tools And Techniques

A. Programming Language

- Python – the primary language due to its wide range of AI libraries, large community support for various features, and ease of deployment

B. Machine Learning / Deep Learning techniques

- Supervised Image classification with Convolutional Neural Networks using Keras Sequential API
- Data Augmentation (zooms, rotations, shifts) on appropriate data
- Normalization of all image data

C. Python Libraries Used

- **os** - for file system operations and path checks on the local machine

- **random** – for reproducibility
- **numpy** – for numerical array math computations
- **tensorflow** – backbone for keras and input pipelines
- **matplotlib** – To show images and evaluation on a UI
- **Keras**- for model definitions, layers, training, prediction, and metrics tracking
- **sklearn** - for evaluation reports such as confusion matrix and classification report

D. Development and collaboration tools

- Meetings / WhatsApp – to discuss the project and share ideas and resources
- Git/ GitHub – for version control, code collaboration and project management, and task tracking
- Visual Studio Code – IDE for developing the Brainiac code base and pushing code fragments to GitHub

E. Support tools

- Grammarly – for formatting documentation
- Microsoft Word – for documenting the project
- Canva – for creation of a poster

9. Problem Definition

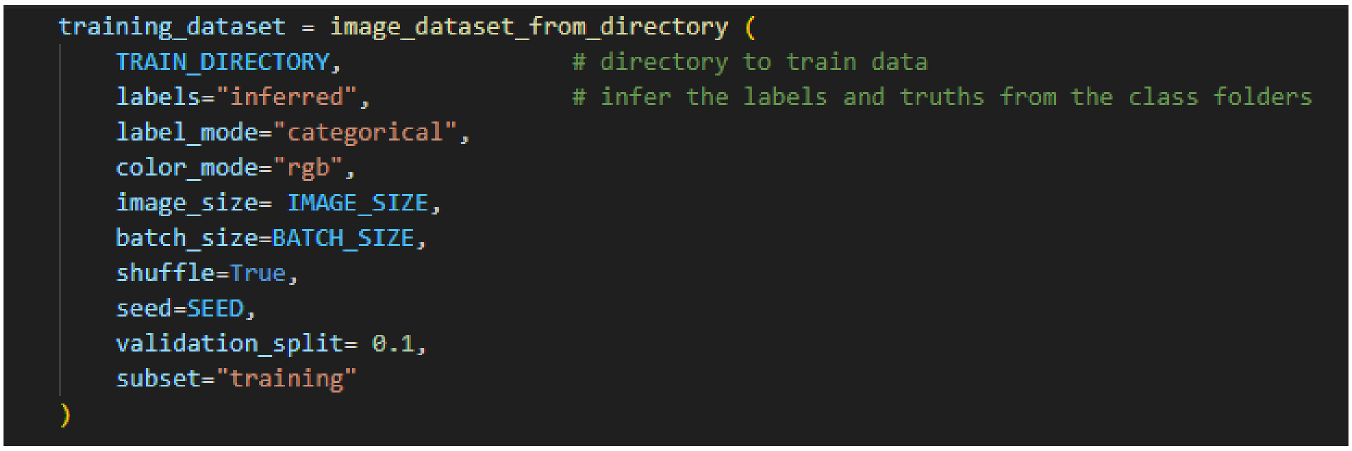
Diagnosing brain tumors with MRI scans is a very important, yet time and resource-intensive task. In South Africa and other developing areas, there are not enough medical experts to fulfil this task on time. In public hospitals such as Chris Hani Hospital, the problem is even more emphasized with the few experts they are able to spare being overtasked, which can lead to mistakes, delays in treating patients, and even missed findings.

This problem is very critical and should be adequately addressed, as brain tumors can result in death or poor prognosis for untimely interventions. Early detection of brain tumors and improved accuracy of categorization of tumor type can greatly improve a patient's prognosis by bringing timely clinical decision-making to them. Our proposed solution, brAIniac, uses the BRISC2025 dataset and AI ML/DL techniques to analyse and classify brain MRI images. This solution will help specialists reduce their workload and improve decision-making.

THEORETICAL ASPECTS

1. Machine Learning Approach

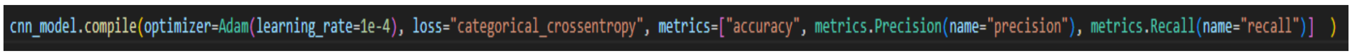
We have adopted a **supervised machine learning** approach to create brAiniac. By using *Keras.utils* function, **image_dataset_from_directory**, we implemented a ***multi-class image classification task,*** as our model was designed to train on our labelled data inferred from their respective folders, acting as our truth labels.



```
training_dataset = image_dataset_from_directory (
    TRAIN_DIRECTORY,             # directory to train data
    labels="inferred",           # infer the labels and truths from the class folders
    label_mode="categorical",
    color_mode="rgb",
    image_size= IMAGE_SIZE,
    batch_size=BATCH_SIZE,
    shuffle=True,
    seed=SEED,
    validation_split= 0.1,
    subset="training"
)
```

Figure 1: Multi-task image classification task

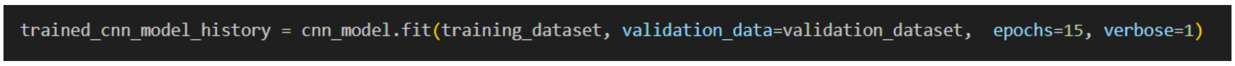
The model was implemented using a **Convolutional Neural Network (CNN)** built with a **Keras Sequential** classifier and optimized with the **Adam** algorithm and **categorical cross-entropy loss**.



```
cnn_model.compile(optimizer=Adam(learning_rate=1e-4), loss="categorical_crossentropy", metrics=["accuracy", metrics.Precision(name="precision"), metrics.Recall(name="recall")]  )
```

Figure 2: The model was implemented using a Convolutional Neural Network (CNN)

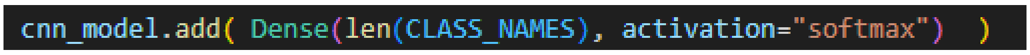
Our model uses the packaged version of the dataset created to train via supervised learning and assess itself from the inferred truths.



```
trained_cnn_model_history = cnn_model.fit(training_dataset, validation_data=validation_dataset,  epochs=15, verbose=1)
```

Figure 3: The model uses the packaged version of the dataset created

After which it outputs for each data a class via one of the four classes identified from our dataset



```
cnn_model.add( Dense(len(CLASS_NAMES), activation="softmax")   )
```

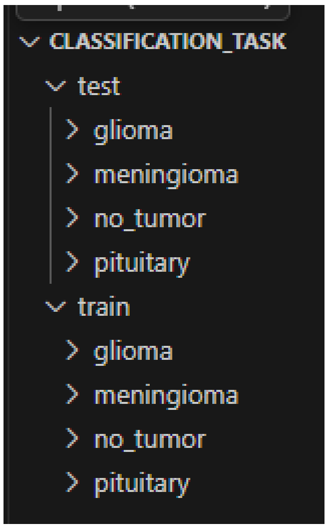
Figure 4: Outputs for each data class via one of the four classes

This approach was selected for its proven efficiency in learning spatial features from image data, making it a strong choice for our brain tumour classification task.

2. Data

Brainiac employed the use of the **BRISC2025** dataset sourced from the KAGGLE dataset library and focused on the brain tumour classification task.

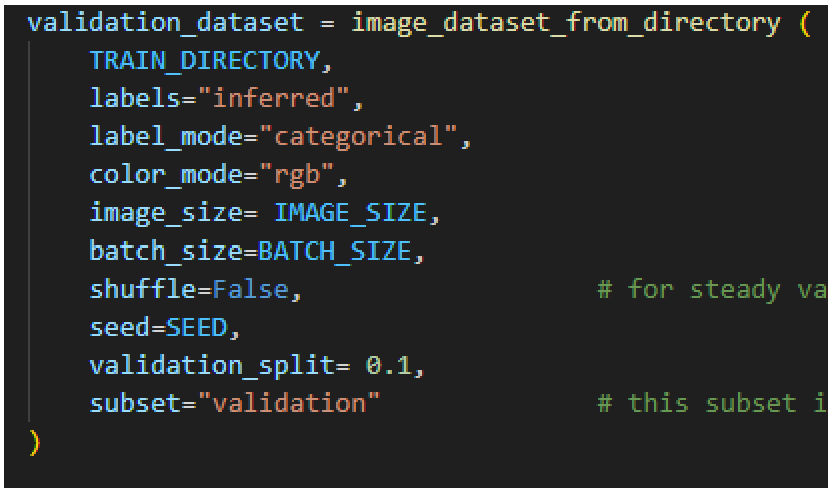
The folder structure includes train and test folders, each with four class folders: glioma, meningioma, no_tumor, and pituitary. In total, there are 6000 JPG images (5000 train images and 1000 test images), each sized at 512 X 512 pixels.



```
CLASSIFICATION_TASK
  test
    glioma
    meningioma
    no_tumor
    pituitary
  train
    glioma
    meningioma
    no_tumor
    pituitary
```

Figure 5: The BRISC2025 dataset was sourced from the KAGGLE dataset library and focused on the brain tumour classification task

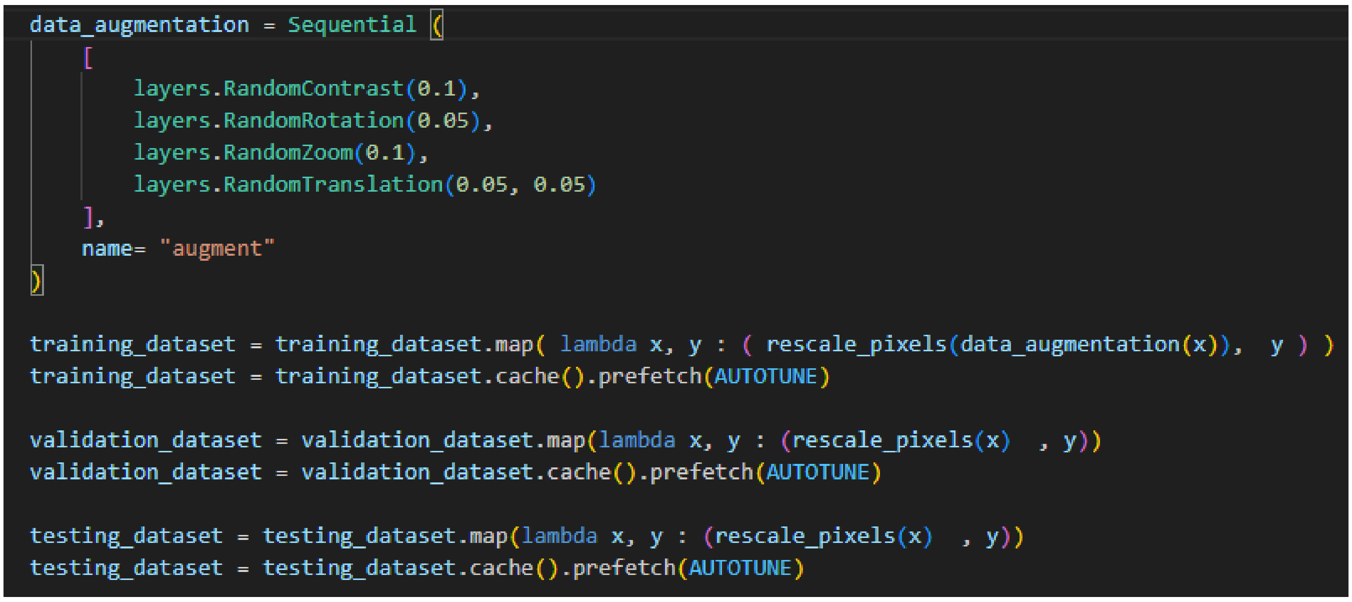
For training purposes, 10% of the training data was set aside for validation and to track metrics per epoch.



```
validation_dataset = image_dataset_from_directory (
    TRAIN_DIRECTORY,
    labels="inferred",
    label_mode="categorical",
    color_mode="rgb",
    image_size= IMAGE_SIZE,
    batch_size=BATCH_SIZE,
    shuffle=False,                # for steady va
    seed=SEED,
    validation_split= 0.1,
    subset="validation"           # this subset i
)
```

Figure 6: 10% of the training data was set aside

In the codebase, all images were resized to 224 X 224 pixels to improve efficiency and normalized to [0, 1]. To improve the robustness of training, we applied data augmentation on the training set only, such as small contrasts, rotations, zooms, and translations. The validation set and testing set were only normalized.

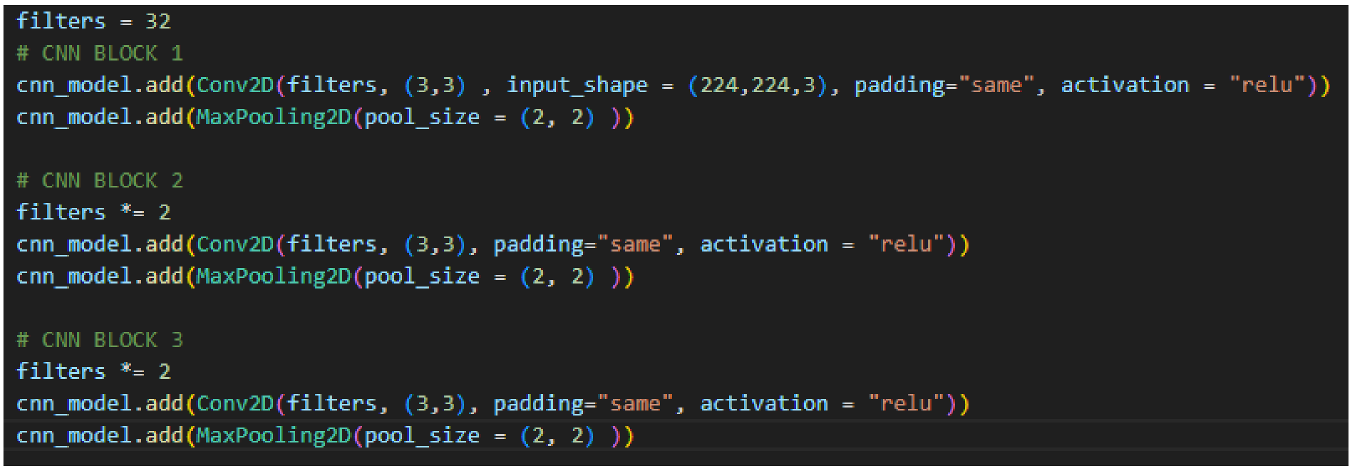


```
data_augmentation = Sequential (
    [
        layers.RandomContrast(0.1),
        layers.RandomRotation(0.05),
        layers.RandomZoom(0.1),
        layers.RandomTranslation(0.05, 0.05)
    ],
    name= "augment"
)

training_dataset = training_dataset.map( lambda x, y : ( rescale_pixels(data_augmentation(x)),  y ) )
training_dataset = training_dataset.cache().prefetch(AUTOTUNE)

validation_dataset = validation_dataset.map(lambda x, y : (rescale_pixels(x)  , y))
validation_dataset = validation_dataset.cache().prefetch(AUTOTUNE)

testing_dataset = testing_dataset.map(lambda x, y : (rescale_pixels(x)  , y))
testing_dataset = testing_dataset.cache().prefetch(AUTOTUNE)
```

Figure 7In the codebase, all images were resized to 224 X 224 pixels to improve efficiency and normalized to [0, 1].

3. Model

Brainiac uses a **Convolutional Neural Network** with a **Keras Sequential** classifier. We implemented a 3-block system of Convolutional and Pooling layers using **Keras.layers' Conv2D** and **MaxPooling2D** classifiers, by reducing the spatial data by half and increasing feature detectors by 2, we ensured that each block learns from

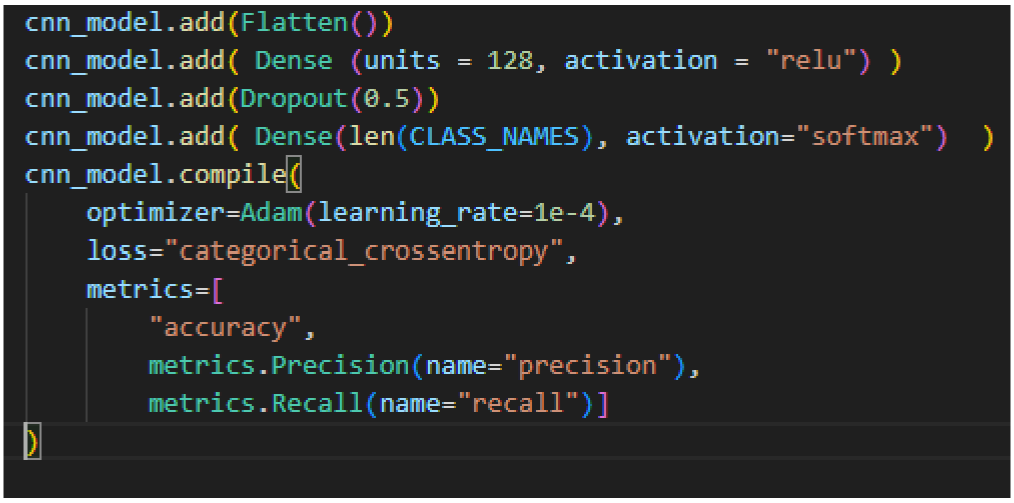


```
filters = 32
# CNN BLOCK 1
cnn_model.add(Conv2D(filters, (3,3) , input_shape = (224,224,3), padding="same", activation = "relu"))
cnn_model.add(MaxPooling2D(pool_size = (2, 2) ))

# CNN BLOCK 2
filters *= 2
cnn_model.add(Conv2D(filters, (3,3), padding="same", activation = "relu"))
cnn_model.add(MaxPooling2D(pool_size = (2, 2) ))

# CNN BLOCK 3
filters *= 2
cnn_model.add(Conv2D(filters, (3,3), padding="same", activation = "relu"))
cnn_model.add(MaxPooling2D(pool_size = (2, 2) ))
```

the preceding block.

Figure 8: Implementation of a 3-block system of Convolutional and Pooling layers using Keras.layers' Conv2D and MaxPooling2D classifiers

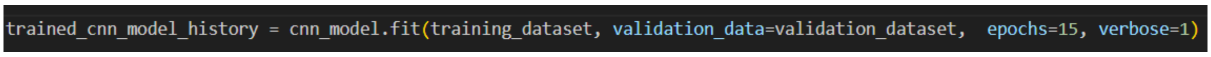
A Flatten layer was used to condense the outputs from the blocks into 1D feature vectors for our fully connected dense layer. The dense layer adopts the “relu” activation algorithm to evaluate our model’s decision. A dropout layer of 50% was employed to promote regularization and avoid overfitting before the final dense layer for outputting results for each image. The model was then compiled using the Adam optimizer and metrics to be tracked and assigned within.



```
cnn_model.add(Flatten())
cnn_model.add( Dense (units = 128, activation = "relu") )
cnn_model.add(Dropout(0.5))
cnn_model.add( Dense(len(CLASS_NAMES), activation="softmax")  )
cnn_model.compile(
    optimizer=Adam(learning_rate=1e-4),
    loss="categorical_crossentropy",
    metrics=[
        "accuracy",
        metrics.Precision(name="precision"),
        metrics.Recall(name="recall")]
)
```

Figure 9: A Flatten layer was used to condense the outputs

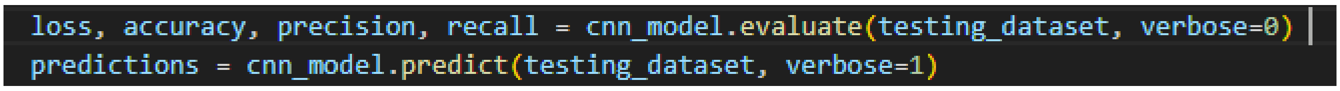
The model was then set to train for 45 epochs using the training dataset and validation dataset to track progress per epoch.



```
trained_cnn_model_history = cnn_model.fit(training_dataset, validation_data=validation_dataset,  epochs=15, verbose=1)
```

Figure 10: The model was then set to train for 45 epochs

After which, the model was then used to evaluate and predict on the unseen test data



```
loss, accuracy, precision, recall = cnn_model.evaluate(testing_dataset, verbose=0)
predictions = cnn_model.predict(testing_dataset, verbose=1)
```

Figure 11: The model was then used to evaluate and predict on the unseen test data

4. Solution Techniques

- The team used a weighted-scale decision method to evaluate several project ideas based on feasibility and social impact. The highest-scoring idea, **Brainiac,** was selected for implementation.
- Each team member contributed modular Python functions based on their assigned task. Various code versions were evaluated and merged via GitHub branches.

- All images from the BRISC2025 dataset were resized from 512 x 512 to 224 x 224 pixels to reduce computational cost and training time per epoch without sacrificing key spatial features.
- Every image was normalized from an RGB range of [0-255] to [0-1] using **Keras' layers.Rescaling** function to prevent model bias towards large pixel values.
- We applied a fixed global seed on all random generators used (Python, Numpy, and TensorFlow) to ensure reproducible results per run and allow for easier evaluation of the model per run.
- The training dataset was further augmented using the Sequential model, with random zooms, contrast, rotations, and shifts applied to random images, to improve the robustness of the training process of our CNN model and enhance generalization.
- Datasets were cached in RAM memory and prefetched using **tensorflow.data.AUTOTUNE** efficiency algorithm to enable parallel data loading and maximize CPU utilization, thus improving training efficiency.
- We used TensorFlow data AUTOTUNE to evaluate local machine CPU features and load in parallel to maximize compute power efficiency
- The convolution layer used the "same" padding to include edge pixels during feature extractions and ensure a uniform spatial coverage.
- The number of filters was doubled after each Convolution-Pooling block to compensate for spatial reduction caused by the Pooling layer, and allow for learning of complex abstractions
- Our model adopted the **ReLU** activation to introduce a non-linear learning pattern and improve the efficiency of the learning process.
- A dropout of 50% was applied to randomly deactivate half of the neurons during training to prevent feature memorization, causing overfitting and thus promoting generalization
- The model was compiled using the Adam optimizer algorithm to ensure our model gains adaptive learning capabilities.
- The training data was split, and 10% set aside for validation to be used for tracking the learning process per epoch cycle and help detect underfitting or overfitting in training.

- We ensured to implement a confidence score when predicting every single image to help users measure how certain the model is about its diagnosis – essential for real-world application of Brainiac.
- Python's built-in error handler was integrated into all functions to identify, isolate, and handle runtime errors efficiently, improving debugging and maintainability.

5. Deep Learning

Brainiac applies Deep Learning techniques through a Convolutional Neural Network using the Keras Sequential classifier. CNNs are well-suited for image-based tasks because they can process two-dimensional spatial data and automatically extract meaningful visual features, making them an ideal choice for medical image classification tasks.

The CNN architecture consists of 3-Convolution-Pooling blocks, allowing the model to progressively learn and represent more complex spatial features at each block level.

ReLU activation functions were applied to introduce non-linear learning behaviour, enabling the network to capture irregular tumour patterns.

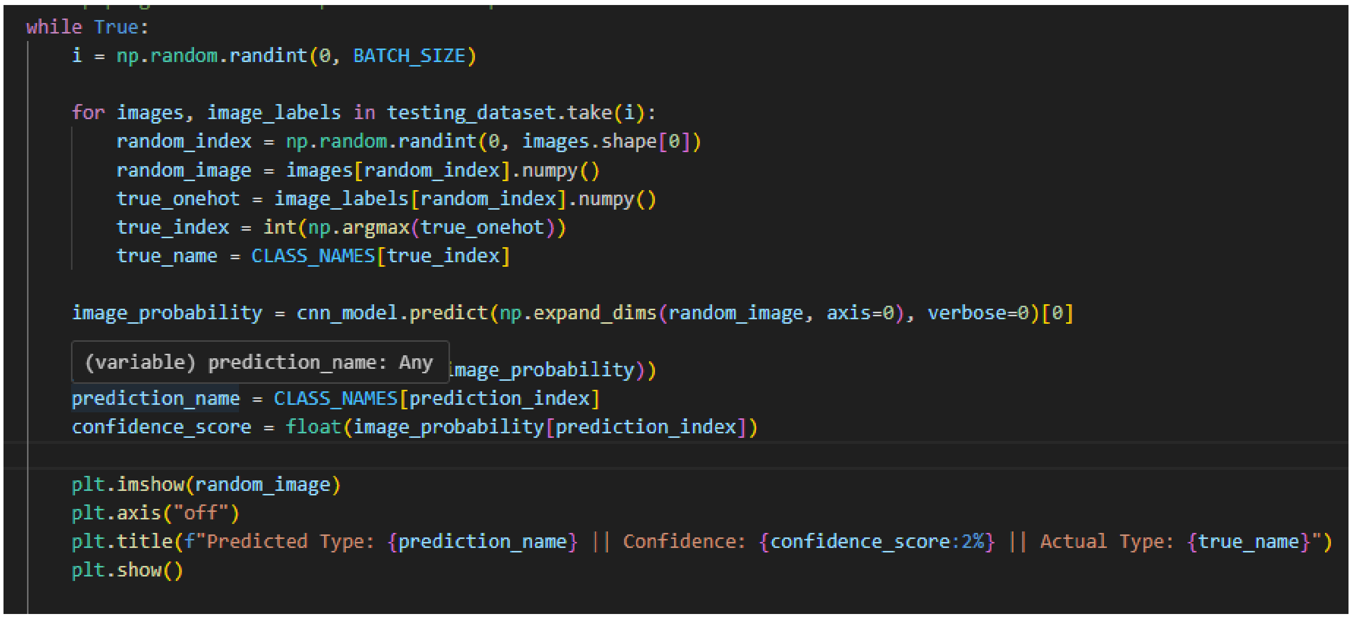
A Flatten layer was used to summarize 2D spatial features to ID feature vectors, which were then passed into a Dense layer to perform high-level reasoning and classification.

Finally, a Dropout layer was integrated to promote regularization, reduce overfitting, and enhance the model's ability to generalize to unseen brain MRI scans.

6. Other Features

Brainiac integrated another feature into its solution - the ability to predict on a single MRI scan. This feature is particularly useful for real-world applications and promotes ease of use for non-technical users. The function randomly selects an image from any of the testing batches and displays it through a graphical interface using **Matplotlib's pylot** module.

For each displayed image, the system presents the model's predicted tumour class, the actual class. This provides an interactive and visual way to interpret the model's decision-making process and assess its certainty in real-world use, thereby improving practical usability in a clinical context.



```
while True:
    i = np.random.randint(0, BATCH_SIZE)

    for images, image_labels in testing_dataset.take(i):
        random_index = np.random.randint(0, images.shape[0])
        random_image = images[random_index].numpy()
        true_onehot = image_labels[random_index].numpy()
        true_index = int(np.argmax(true_onehot))
        true_name = CLASS_NAMES[true_index]

    image_probability = cnn_model.predict(np.expand_dims(random_image, axis=0), verbose=0)[0]

    (variable) prediction_name: Any image_probability))
    prediction_name = CLASS_NAMES[prediction_index]
    confidence_score = float(image_probability[prediction_index])

    plt.imshow(random_image)
    plt.axis("off")
    plt.title(f"Predicted Type: {prediction_name} || Confidence: {confidence_score:2%} || Actual Type: {true_name}")
    plt.show()
```

Figure 12: The function randomly selects an image from any of the testing batches and displays it through a graphical interface using Matplotlib's pylot module

Reference

Anon., n.d. *Artificial Intelligence with Python.* s.l.:s.n.

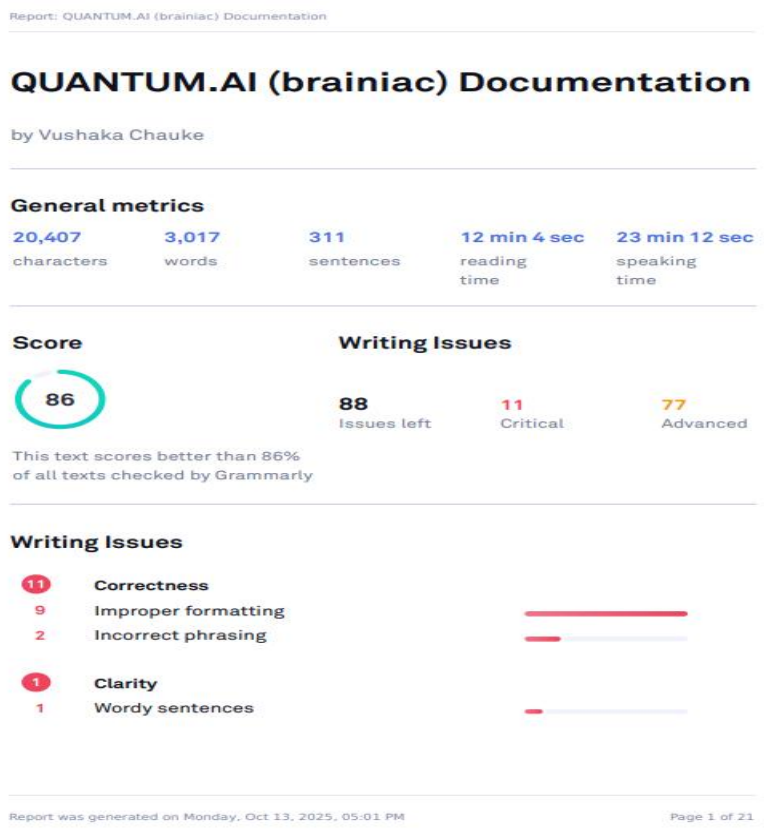
South African National Department of Health. (2019). National Digital Health Strategy for South Africa 2019-2024. Available at: https://www.health.gov.za/wp-content/uploads/2020/11/national-digital-strategy-for-south-africa-2019-2024-b.pdf

Health-e News. (2025). Patients left in limbo as MRI services collapse in Gauteng. [online] Available at: https://health-e.org.za/2025/06/30/patients-left-in-limbo-as-mri-services-collapse-in-gauteng-mri/

BRISCDataset. (2025). BRISC2025. [dataset] Available at: https://www.kaggle.com/datasets/briscdataset/brisc2025

University of Cape Town. (2021). Neurologists. [online] Available at: https://www.news.uct.ac.za/images/userfiles/downloads/media/2021_11_19_Neurologists.pdf

GRAMMARLY REPORT



Report: QUANTUM.AI (brainiac) Documentation

QUANTUM.AI (brainiac) Documentation

by Vushaka Chauke

General metrics

20,407	3,017	311	12 min 4 sec	23 min 12 sec
characters	words	sentences	reading time	speaking time

Score

86

This text scores better than 86% of all texts checked by Grammarly

Writing Issues

88	11	77
Issues left	Critical	Advanced

Writing Issues

11	**Correctness**
9	Improper formatting
2	Incorrect phrasing
1	**Clarity**
1	Wordy sentences

Report was generated on Monday, Oct 13, 2025, 05:01 PM

Page 1 of 21