Project School Certificate

Title: Brain Haemorrhage Detection and Segmentation

Faculty in Charge: Dr. Devika Rubi

Session Duration: 12/06/2023 – 15/07/2023

Name: Arvind Agarwal

Roll Number: 21BD1A660B

Class: II-CSM-A

GitHub Link: Brain_Tumour_Segmentation

Index

Project Description

Introduction:

The aim of this project is to develop a system that can accurately segment brain tumours in CT scan images and present the segmented video to the user through a user-friendly Flutter frontend. The system utilizes the UNET model, a popular deep learning architecture for semantic segmentation tasks, to achieve high-quality and precise tumour segmentation.

Features:

CT Scan Image Processing: The system takes a CT scan image of a patient's brain as input. It preprocesses the image to enhance its quality, remove noise, and normalize intensity levels for better tumour segmentation results.

UNET Model for Segmentation: The pre-processed CT scan image is fed into a pre-trained UNET model. The UNET model is a convolutional neural network that has shown excellent performance in medical image segmentation tasks. It learns to segment the brain tumor from the input image by leveraging its hierarchical architecture and skip connections.

Brain Tumour Segmentation: The UNET model performs pixel-level segmentation, labelling each pixel as either tumour or non-tumour. The segmentation process involves analysing the spatial relationships between pixels to accurately identify tumour boundaries.

Video Generation: The system generates a segmented video of the brain tumour by combining multiple segmented frames obtained from consecutive CT scan images. This video provides a dynamic representation of the tumour's location, shape, and size over time.

Flutter Frontend: The segmented video is displayed to the user through a Flutter-based frontend. Flutter is a cross-platform framework for building mobile, web, and desktop applications. The frontend provides an intuitive user interface where the user can interact with the segmented video, view different slices, and navigate through the video timeline.

Interactive Visualization: The Flutter frontend allows users to manipulate the segmented video, such as adjusting playback speed, pausing, zooming, and scrolling through different frames. This interactivity enables medical professionals to analyse and understand the tumour's characteristics more effectively.

Performance Evaluation: The system incorporates performance evaluation metrics to assess the accuracy of the tumour segmentation results. Metrics such as dice coefficient and IoU can be computed to quantify the model's performance and provide feedback for further improvement.

Scalability and Deployment: The system is designed to be scalable, enabling it to handle large volumes of CT scan images and process them in a parallel and distributed manner. It can be deployed on cloud platforms to ensure availability and accessibility to medical professionals across various locations.

Intracranial haemorrhage - (HGE)

Intracranial haemorrhage, bleeding that occurs inside the cranium, is a serious health problem requiring rapid and often intensive medical treatment. For example, intracranial haemorrhages account for approximately 10% of strokes in the U.S., where stroke is the fifth-leading cause of death. Identifying the location and type of any haemorrhage present is a critical step in treating the patient.

Diagnosis requires an urgent procedure. When a patient shows acute neurological symptoms such as severe headache or loss of consciousness, highly trained specialists review medical images of the patient's cranium to look for the presence, location and type of haemorrhage. The process is complicated and often time consuming.

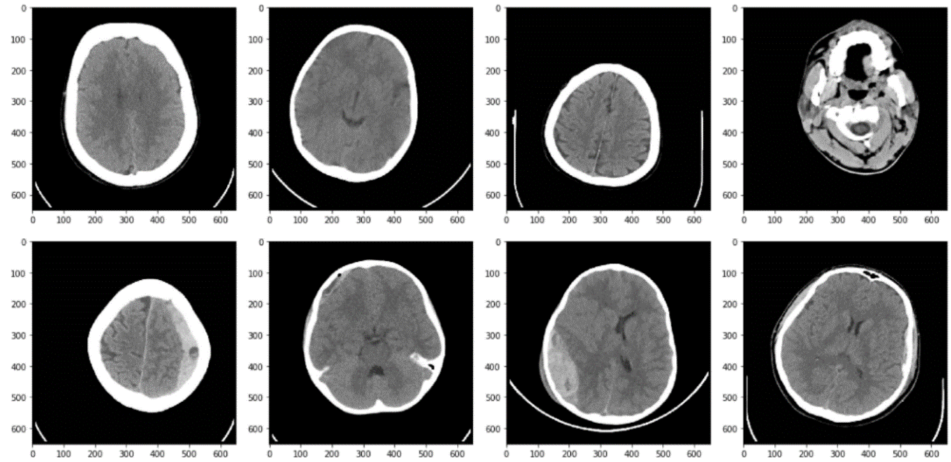
Dataset Description

This dataset consists of head CT (Computed Tomography) images in jpg format. There are 2500 brain window images for 82 patients. There are approximately 30 image slices per patient.

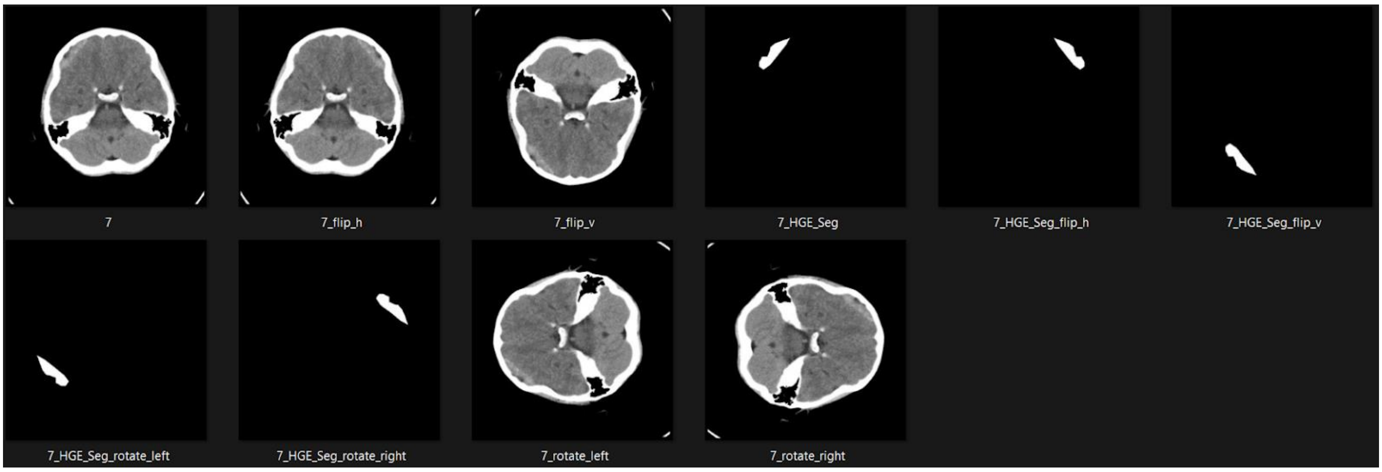
Original Images	Augmented Images	Flip Horizontal	Flip Vertical	Rotate Left	Rotate Right
318	1590	318	318	318	318

number of training examples = 1272
number of development examples = 159
number of test examples = 159
Images train shape: (1272, 256, 256, 3)
Masks train shape: (1272, 256, 256, 3)
Images Val (dev) shape: (159, 256, 256, 3)
Masks Val (dev) shape: (159, 256, 256, 3)
Images test shape: (159, 256, 256, 3)
Masks test shape: (159, 256, 256, 3)

Original Data:



Augmented Data:



Performance Metrics Used:

- **dice coefficient -** The Dice loss is a measure of the overlap between the prediction (y_pred) and the ground truth (y_true). It ranges from 0 to 1, where a Dice loss of 1 indicates perfect overlap (i.e., a perfect segmentation), while a Diceloss of 0 indicates no overlap.
- **Intersection Over Union (IoU)/Jaccard index**– IoU, also known as the Jaccard Index, is a metric used to quantify the percent overlap between the target mask and our prediction output. It's often used in segmentation problems to evaluate the quality of predictions. This function is generally used for evaluating segmentation tasks where the true and predicted outputs are binary masks of the same size.

Model Description

UNET is a convolutional neural network architecture specifically designed for semantic segmentation tasks. It was introduced by Ronneberger et al. in 2015 and has since become a widely adopted model for various medical image segmentation applications, including brain tumour segmentation.

The UNET architecture is characterized by its U-shaped structure, which consists of an encoder path and a corresponding decoder path. The encoder path resembles a traditional convolutional network that progressively reduces spatial dimensions while capturing higher-level features. On the other hand, the decoder path mirrors the encoder path but in a reverse manner, gradually upsampling the feature maps to obtain a final segmentation output.

Key components and operations of the UNET model include:

Contracting Path (Encoder):

Convolutional layers: These layers extract features from the input image using small receptive fields and learn increasingly complex representations.

Max pooling: After each convolutional block, max pooling is applied to reduce spatial dimensions, thus capturing contextual information at multiple scales.**Skip connections:** Before down sampling, the feature maps at each resolution level are preserved using skip connections. These connections serve as shortcuts that pass low-level and fine-grained information to the corresponding layers in the expanding path.

Expanding Path (Decoder):

Up-sampling: The feature maps from the encoder are upsampled to progressively recover spatial dimensions using either bilinear interpolation or transposed convolutions.

Concatenation: At each resolution level in the decoder, the up sampled feature maps are concatenated with the corresponding skip connections from the encoder. This helps to combine both low-level and high-level information for accurate segmentation.

Convolutional layers: These layers refine the concatenated feature maps and capture more precise spatial details.

Final Convolutional Layers:

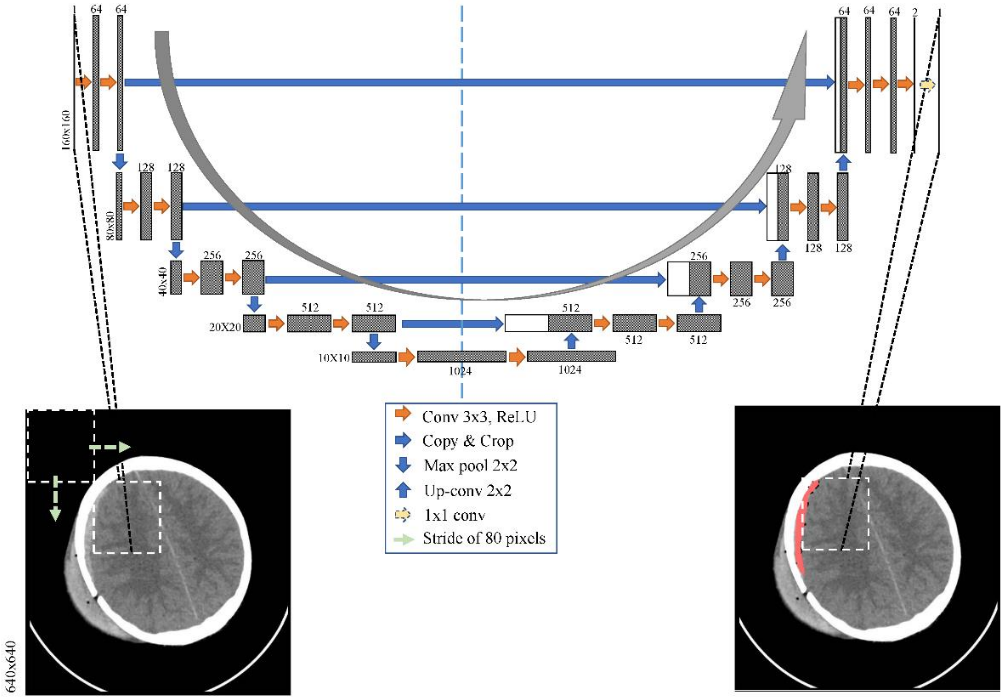
At the end of the expanding path, one or more convolutional layers are used to produce the final segmentation map.

Activation function: Often, a pixel-wise activation function such as the sigmoid or softmax activation is applied to the output to obtain probability maps, indicating the likelihood of each pixel belonging to a specific class.

The UNET model's architecture allows it to capture both local and global information effectively, making it suitable for segmenting objects with complex shapes and structures, such as brain tumours. Additionally, the skip connections enable the model to leverage both fine-grained and contextual information, improving the quality and accuracy of the segmentation results.

During training, the UNET model optimizes the network parameters by minimizing a suitable loss function, such as binary cross-entropy or Dice loss, comparing the predicted segmentation output with the ground truth labels. The model can be trained end-to-end using labelled data, where the input is the image, and the output is the corresponding segmentation mask.

In summary, UNET is a powerful architecture for semantic segmentation tasks that effectively combines the strengths of convolutional networks and skip connections. Its U-shaped design allows it to capture detailed information and context simultaneously, making it well-suited for brain tumour segmentation and other medical image analysis applications.

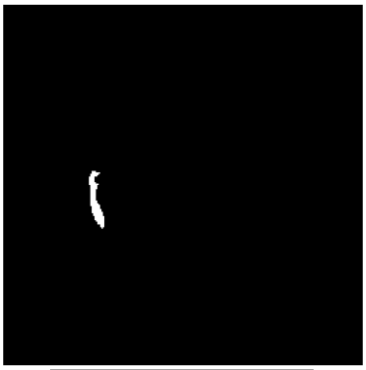


Conclusion:

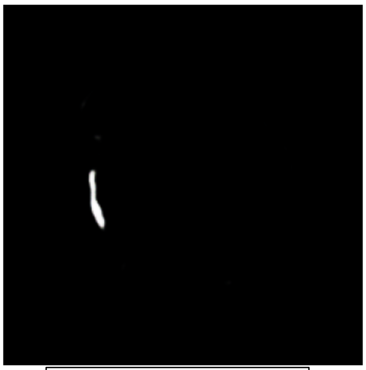
1. Image segmentation can be thought of as a combination of classification and localization tasks.
2. We wish to answer 2 questions in image segmentation – "what" and "where"?
3. The encoder path of the U-Net answers "what" is in the image and acts similarly to any CNN.
4. The decoder path of the U-Net answers "where" is the object in the image and produces a mask of the size of the original image.
5. Skip connections enable us to use the features learned in the encoder network to help generate our output mask.

Why UNET?

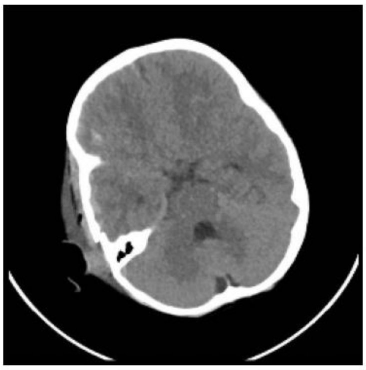
- Precise Localization
- Vastly employed in bio-medical Segmentation
- Versatile and Adaptable
- Better performance/compatible with limited data
- Spatial Resolution Recovery of data



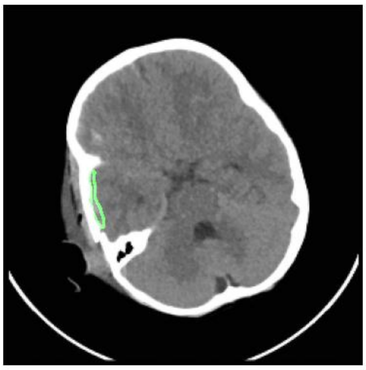
Ground Truth



Predicted Mask



CT-Scan



Overlapped-Image

```
print(model.evaluate(images_test,Seg_test))
✓ 51.6s

4/4 [==============================] - 51s 12s/step - loss: 0.0084 - iou: 0.5882 - dice_coefficients: 0.7462
[0.008428830653429031, 0.5882389545440674, 0.7462166547775269]
```

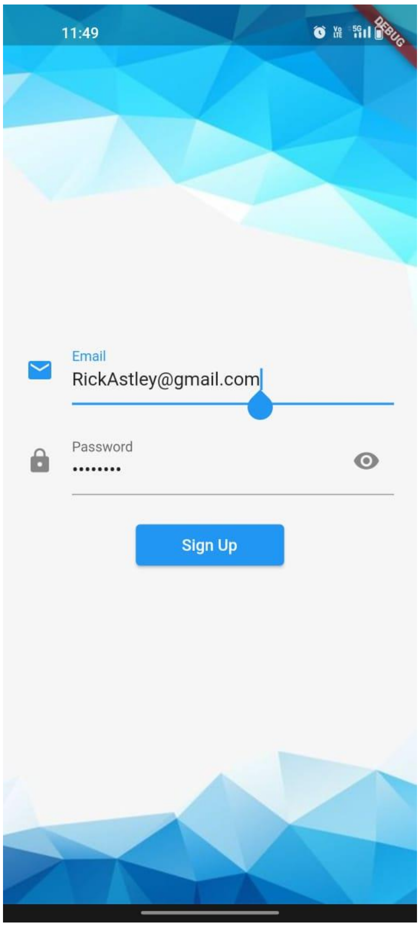
Dice Coefficient : 74.6

Implementation

Flutter:

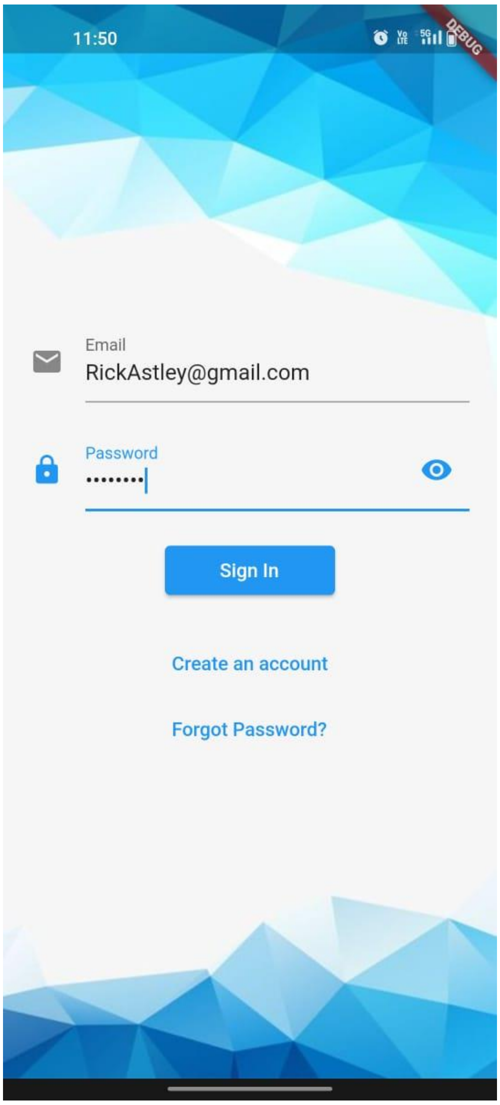
Flutter is an open-source UI software development kit (SDK) created by Google. It is designed to build beautiful, natively compiled applications for mobile, web, and desktop from a single codebase. Flutter allows developers to create high-performance, visually appealing, and fast applications with a rich set of customizable UI components.

Sign-Up Page using Email and Password:



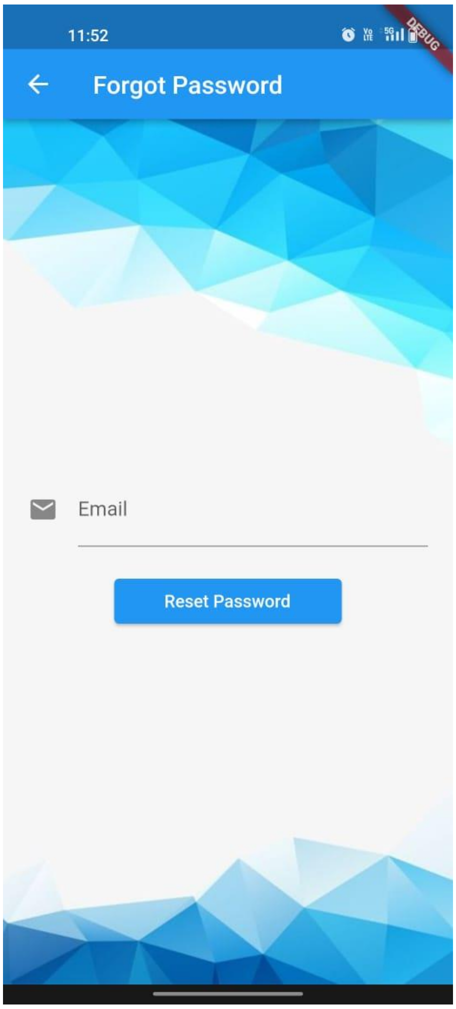
Login Page

To ensure a secure email-password verification process, Flutter has incorporated integrated authentication libraries and APIs, thereby establishing a robust authentication flow. The user interface has been meticulously crafted to deliver explicit feedback and error messages, capable of dynamically adapting to the authentication status at hand.



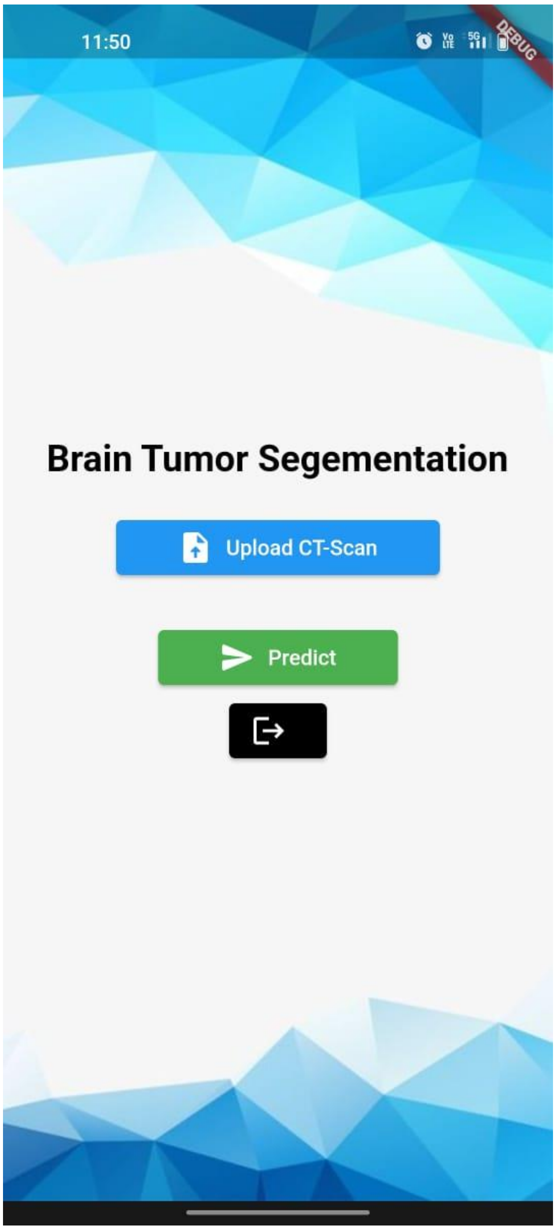
Forgot Password

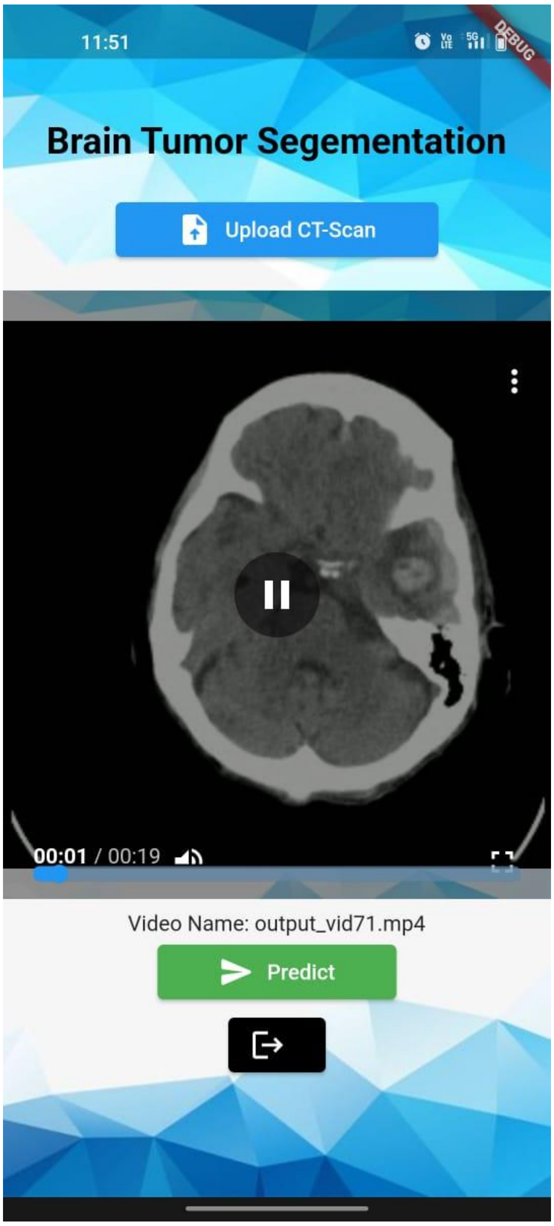
The "forgot password" functionality allows users to reset their passwords by receiving a password reset link via email. This feature enables users to regain access to their accounts in the event of forgotten passwords. Upon initiating the "forgot password" process, an email containing a unique link is sent to the user's registered email address. By clicking on this link, the user is redirected to a password reset page within the app. From there, they can enter a new password, confirming the change and subsequently gaining access to their account with the updated credentials.

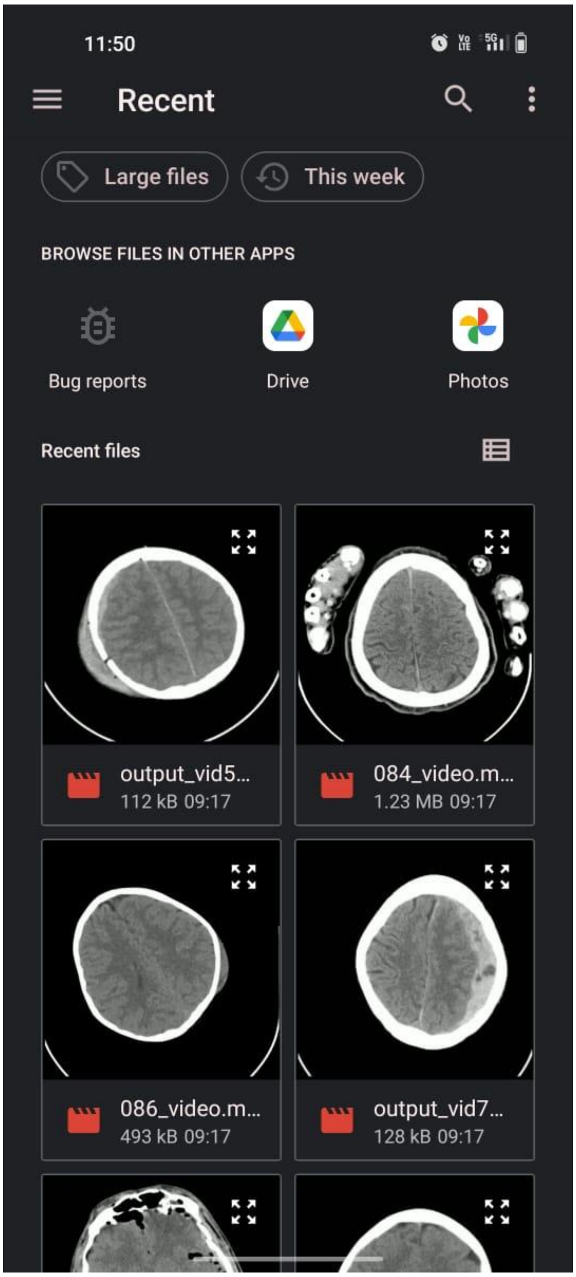


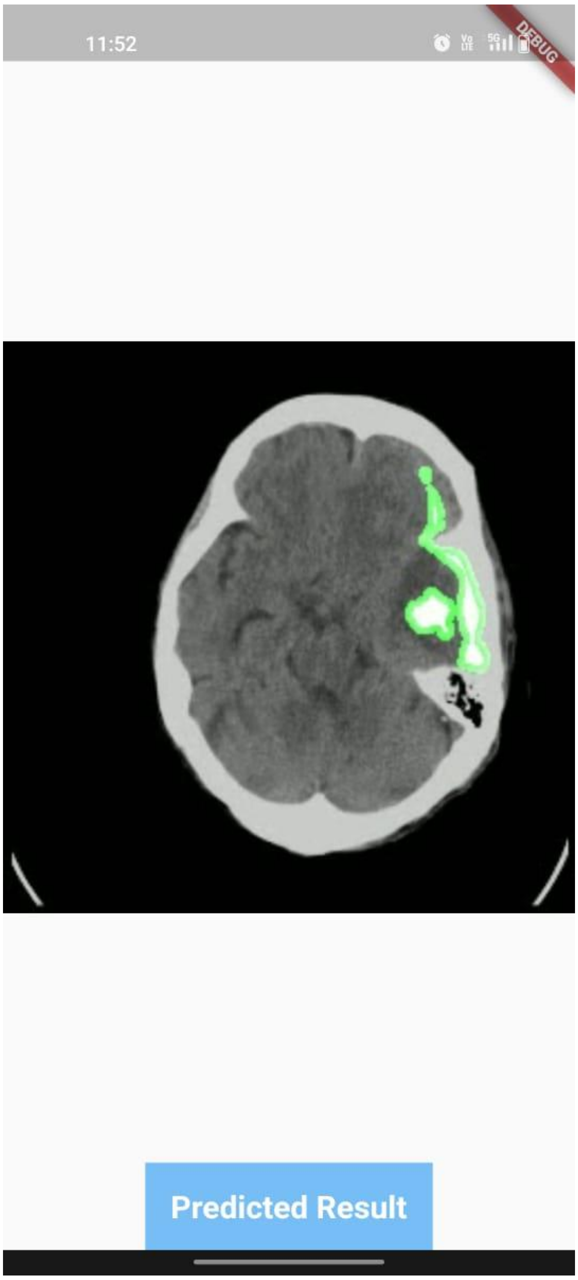
Home

The Home Page for video input of CT images leveraged the multimedia capabilities inherent in Flutter. Through the utilization of Flutter's widget suite, seamless video capture functionality from the device's camera or selection from the gallery was made possible. The user interface was meticulously designed to proficiently manage video processing operations and adeptly convey pertinent information to users, ensuring an optimal user experience.





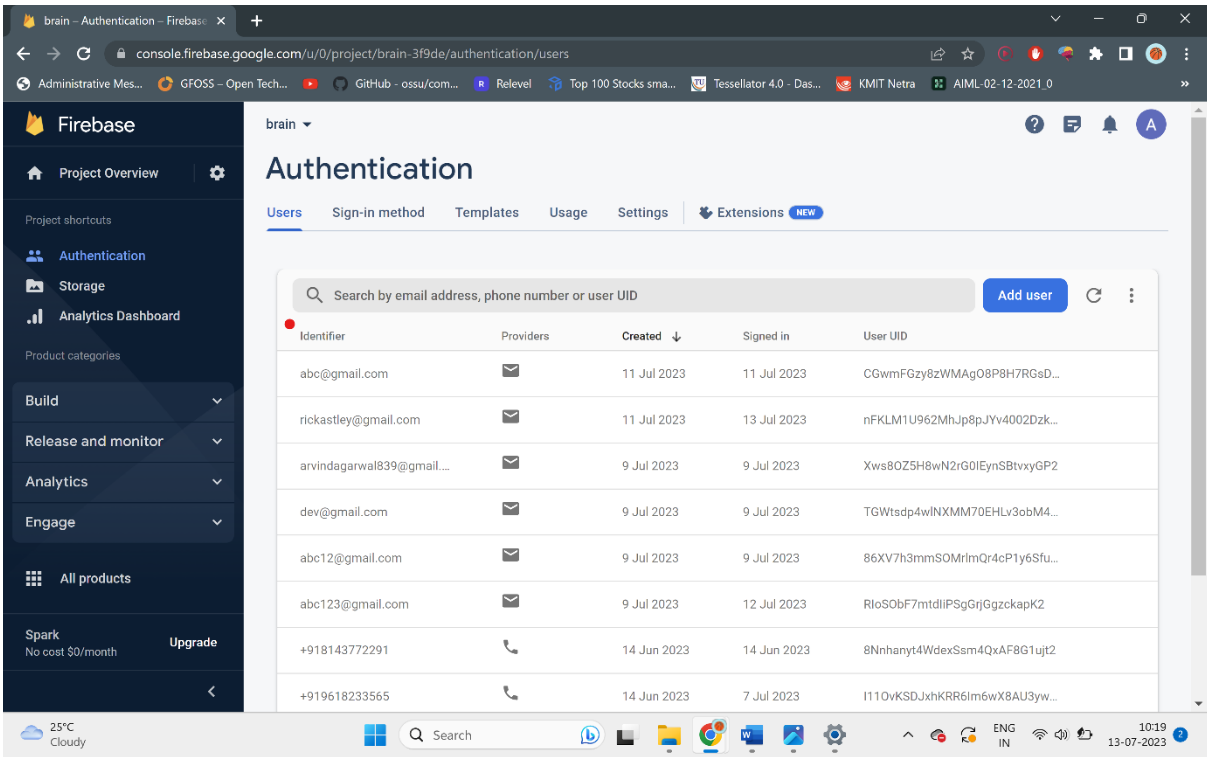

11:50
Recent
Large files
This week
BROWSE FILES IN OTHER APPS
Bug reports
Drive
Photos
Recent files
output_vid5...
112 kB 09:17
084_video.m...
1.23 MB 09:17
086_video.m...
493 kB 09:17
output_vid7...
128 kB 09:17


11:52
DEBUG
Predicted Result

Fire Base:

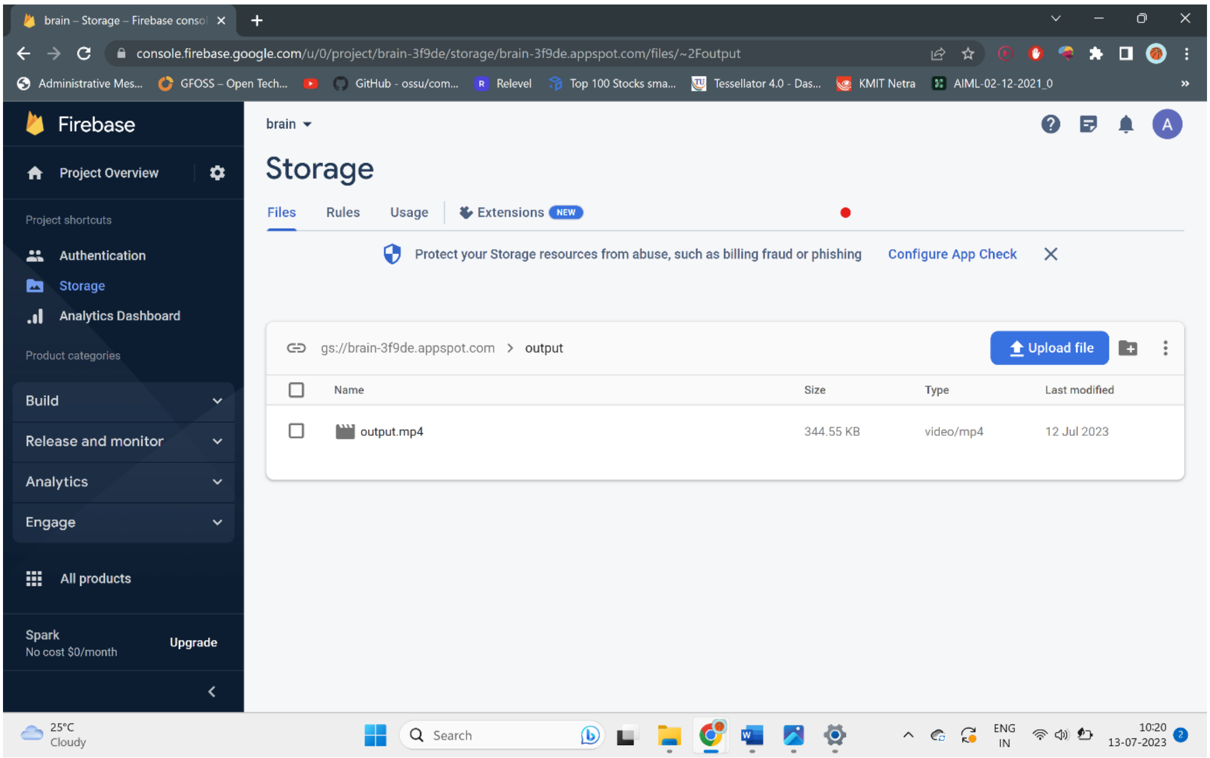
Using Firebase for authentication, email-password verification, phone OTP verification, and storing predicted videos in Firebase Storage is a great choice.

Email-Password Verification: Firebase Authentication offers built-in support for email and password authentication. It allows users to create accounts with email and password combinations, securely store user credentials, and handle password reset functionalities.



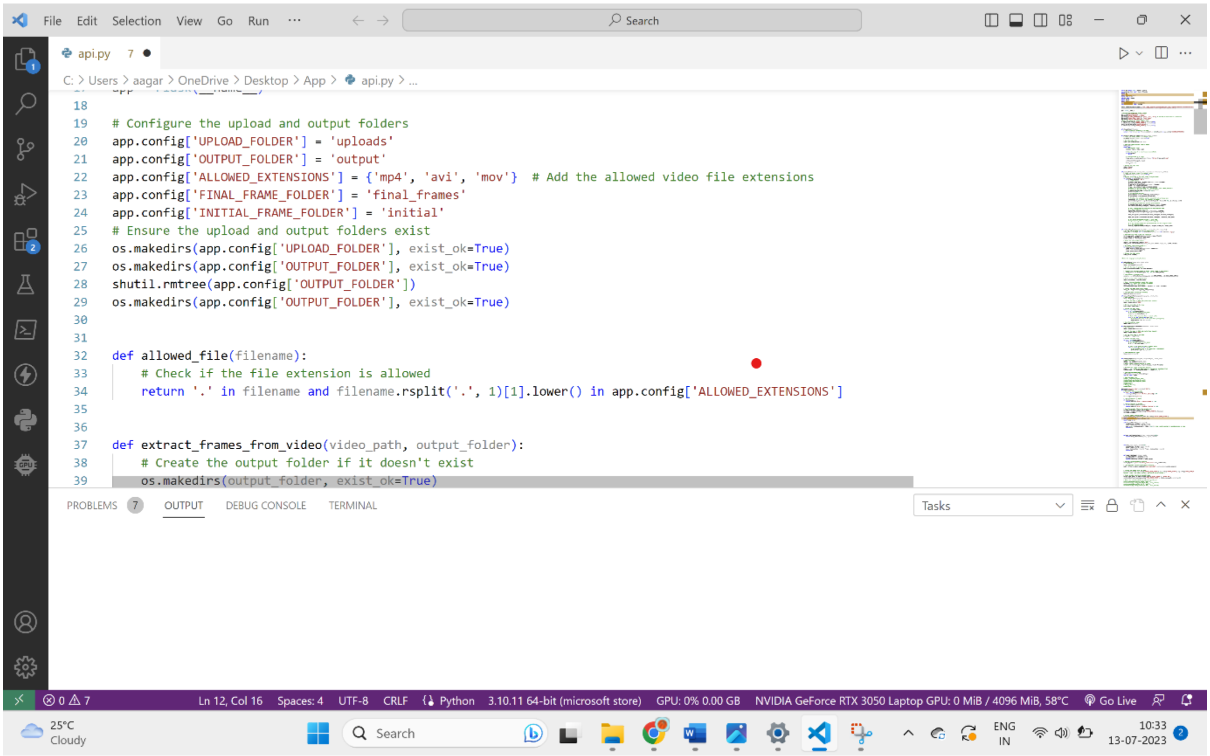
Firebase Storage:

Firebase Storage is a cloud-based storage service provided by Google's Firebase platform. Here the output/Segmented CT-Scan is stored and then presented to the user through flutter App.



Flask:

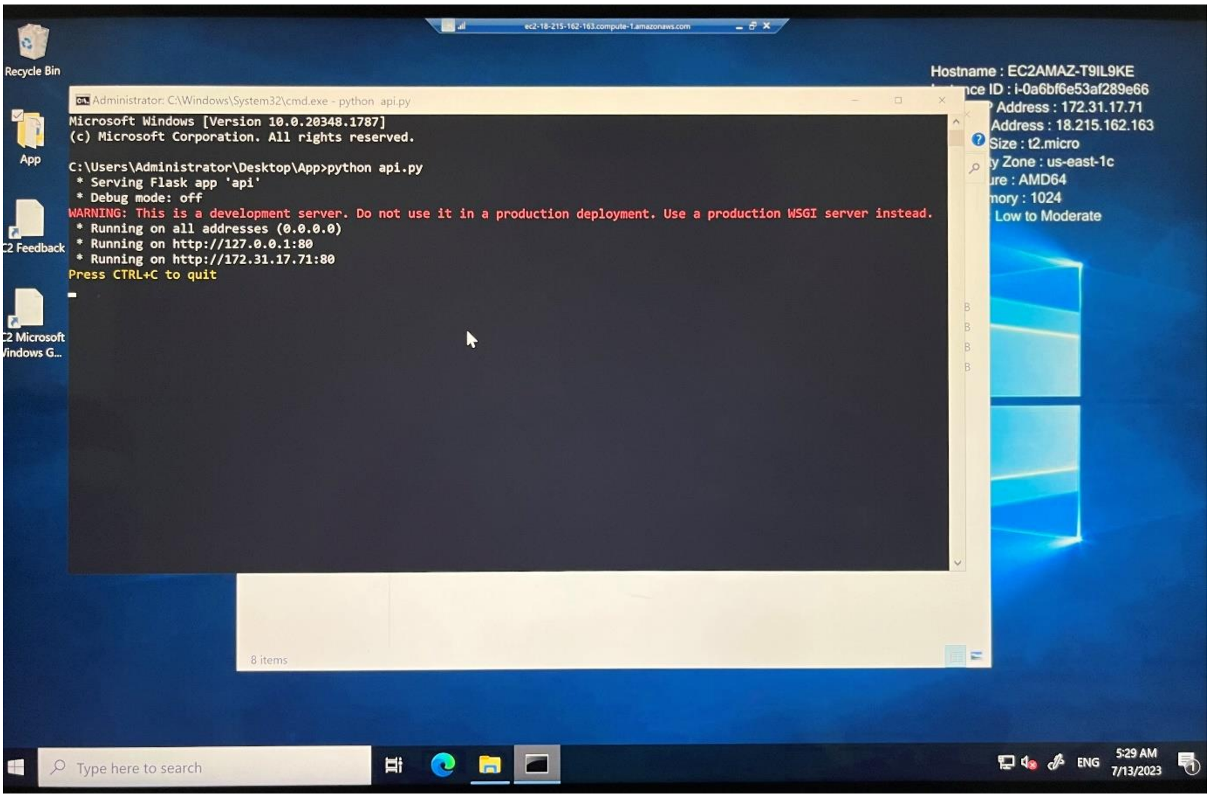
Flask API is a lightweight framework that facilitates the construction of Python-based APIs using Flask. It streamlines the management of HTTP requests, routing, and serialization, thereby affording developers the opportunity to concentrate on the fundamental features of their APIs. In the context of a deep learning (DL) project, Flask API proves particularly advantageous for seamlessly integrating DL frameworks like TensorFlow or PyTorch, thus enabling the provisioning of DL models through dedicated API endpoints. Noteworthy attributes of Flask API encompass efficient request handling, native support for serialization, and the capability to scale for deployment on cloud platforms. By virtue of Flask API's testing and debugging capabilities, one can ascertain the precision and dependability of their DL models, ensuring their readiness for real-world deployment.



AWS:

Amazon Web Services - essentially used for deployment purposes

An Amazon EC2 instance denotes a virtual server provisioned by Amazon Web Services (AWS) within its Elastic Compute Cloud (EC2) infrastructure. This offering enables enterprises to execute applications and services within a scalable and adaptable computing environment. EC2 instances furnish virtual machines with configurable attributes, encompassing CPU, memory, storage, and networking resources. AWS proffers an extensive array of instance types and sizes, catering to a diverse spectrum of workload prerequisites, thereby endowing users with virtually boundless virtual computing capabilities.



Horovod:

The main objective of Horovod is to overcome the challenges of training deep learning models on large-scale distributed systems by providing efficient and scalable communication patterns. It integrates with popular deep learning frameworks such as TensorFlow, PyTorch, and MXNet, allowing users to leverage their existing codebase and take advantage of distributed training capabilities.

```
print("Number of EPOCHS: ", EPOCHS)
print(f'GPU Training Duration: {total_duration:.2f} seconds')
```

```
Number of EPOCHS:  50
GPU Training Duration: 1012.00 seconds
```

CPU Training Duration : 15 hours

Model Accuracy:

Testing Accuracy:

```
print(model.evaluate(images_test,Seg_test))
✓ 51.6s

4/4 [==============================] - 51s 12s/step - loss: 0.0084 - iou: 0.5882 - dice_coefficients: 0.7462
[0.008428830653429031, 0.5882389545440674, 0.7462166547775269]
```

Training Accuracy:

```
4/4 [==============================] - 51s 13s/step - loss: 0.0078 - iou: 0.5407 - dice_coefficients: 0.7003
[0.00778543995693326, 0.5406542420387268, 0.7003418207168579]
```

References:

- flutter_auth
- Accuracy_metrics
- flask_deployment
- dataset
- chatGPT-debugTool
- flaskDocumentation
- AWS_EC2
- Youtube-Tutorials

Conclusion:

The Android Application for Segmenting Brain Haemorrhage is designed and the UNET model is deployed seamlessly on AWS EC2 instance and can be employed for Segmenting Brain Tumour just by uploading CT scan video.