Script for Project\_Brain\_Tumour-v4 Applied AI Presentation

Load Dataset

* "Welcome to our presentation! We'll discuss how we load and preprocess your dataset. We'll be using a custom function to handle the images stored in our dataset directory."
* "First, we define the path to our dataset, which is stored in Google Drive. Then, we create a function called load\_and\_preprocess\_data. This function takes in the directory of the data and the phase (whether it's training or testing data)."
* "The function iterates through each label in the phase directory, loads each image in grayscale, resizes it to 128 by 128 pixels, applies histogram equalisation to enhance contrast, and normalises the image to a range of 0 to 1. The images and labels are stored in lists and converted to numpy arrays."
* "This method is particularly useful because it standardises all images to the same size and scale, ensuring consistent input for our machine learning models. By normalising the images, we also improve the performance and convergence of neural networks."

Cell 3: CNN with Attention Mechanisms

* "In this segment, we explore how to build a Convolutional Neural Network (CNN) with attention mechanisms. Attention mechanisms help the network focus on important parts of the input image."
* "We start by defining the attention module. This module includes both channel attention and spatial attention. Channel attention focuses on 'what' is important in the image, while spatial attention focuses on 'where' the important features are."
* "The attention\_module function first applies global average pooling, then dense layers with ReLU and sigmoid activations to get channel weights. It multiplies these weights with the input to get channel attention. Next, it applies a 1x1 convolution with sigmoid activation to get spatial weights and multiplies these with the channel-attended output."
* "By integrating this module into our CNN, we can enhance the network's ability to learn important features, potentially improving accuracy and robustness."

Cell 4: Data Augmentation with GANs and Preprocessing

* "Now, let's delve into data augmentation using Generative Adversarial Networks (GANs) and preprocessing techniques."
* "We start by performing histogram equalisation on our images to enhance contrast. This is followed by defining an augmentation sequence using the imgaug library, which includes horizontal flips, random rotations, and Gaussian blur."
* "Applying this augmentation sequence to our dataset generates a variety of transformed images. This helps the model generalise better by exposing it to different versions of the training images."
* "Ultimately, data augmentation is a powerful technique to increase the diversity of our training data without actually collecting more data, leading to better-performing models."

Cell 5: Super-Resolution using CNN

* "Up next, we explore the world of super-resolution using Convolutional Neural Networks (CNN). Super-resolution is the process of enhancing the resolution of an image."
* "We define a model using convolutional layers, max pooling, and upsampling layers. The model is designed to take a low-resolution image as input and produce a high-resolution image as output."
* "We then compile the model with Adam optimiser and mean squared error loss and train it using low-resolution and high-resolution image pairs. The trained model can then be used to enhance the resolution of new images."
* "Super-resolution can be incredibly beneficial for applications requiring detailed images, such as medical imaging and satellite imagery."

Cell 6: Noise Reduction with Deep Learning

* "Noise in images can significantly degrade their quality. In this segment, we show how to reduce noise using a deep learning approach with an autoencoder."
* "We build a denoising autoencoder model with convolutional and upsampling layers. The model is trained to reconstruct clean images from noisy inputs."
* "To create noisy images, we add Gaussian noise to our equalised images. The model is then trained to minimise the difference between the noisy inputs and the clean outputs using binary cross-entropy loss."
* "This method is effective for removing noise while preserving important details in the images, making it useful for applications in photography, medical imaging, and more."

Cell 7: Feature Extraction with Wavelet Transformations

* "Feature extraction is a crucial step in image processing. In this segment, we demonstrate how to use wavelet transformations for feature extraction."
* "Wavelet transformations decompose an image into different frequency components. We use the PyWavelets library to perform a 2D discrete wavelet transform on our images."
* "The transformation yields four sets of coefficients: LL, LH, HL, and HH, representing different frequency components of the image. These features can be used for various image processing tasks."
* "Wavelet transformations provide a multi-resolution analysis of images, which is beneficial for tasks like image compression and texture analysis."

Cell 8: Combine Pre-processed Data

* "In this segment, we combine various pre-processed images to create a comprehensive dataset for training our model."
* "We stack equalised, augmented, super-resolution, and denoised images along a new dimension. This combined dataset is then flattened for model training."
* "Combining different versions of the images can provide a more robust training set, helping the model generalise better across different types of pre-processing."
* "By using a diverse dataset, we improve the model's ability to handle various real-world scenarios, leading to better performance and accuracy."

Cell 9: Data Augmentation with Keras

* "Data augmentation is a key technique for enhancing the robustness of machine learning models. Let's see how to implement it using Keras."
* "We start by encoding our labels using LabelEncoder from scikit-learn. Next, we define an ImageDataGenerator for data augmentation, specifying transformations like rotation, width and height shifts, shear, zoom, and horizontal flips."
* "We ensure our image data is in the correct format by expanding dimensions if necessary. Then, we create a data generator for the training images and train our model using this augmented data."
* "Data augmentation helps in generating a larger variety of training samples, which can significantly improve the model's generalisation and prevent overfitting."

Cell 10: Explainability with Grad-CAM

* "Understanding how a model makes decisions is crucial for model interpretability. In this segment, we explore Grad-CAM, a technique for visualising model attention."
* "Grad-CAM uses the gradients of the target concept to produce a heatmap highlighting important regions in the input image. We define a function that computes Grad-CAM for a given image and model."
* "The function extracts the activations of a specified convolutional layer and computes the gradient of the predicted class with respect to these activations. The resulting heatmap shows where the model focuses when making its prediction."
* "Visualising model attention with Grad-CAM helps in understanding model behaviour, identifying potential biases, and ensuring the model makes decisions for the right reasons."
* "And that wraps up our detailed exploration of various advanced image processing and machine learning techniques. We hope you found these explanations insightful and are inspired to apply these methods in your own projects. Thank you for joining us!"