1. Code Execution Flow:

Libraries Used:

TensorFlow: For building and training the Convolutional Neural Network (CNN).

Keras: For building and training the CNN.

Pillow: For image processing and manipulation.

Tkinter: For creating a GUI to select an image from the file explorer.

Key Steps:

Model Creation: A CNN model is created using Keras Sequential API. The model consists of convolutional layers, max-pooling layers, and fully connected layers.

Data Preprocessing: ImageDataGenerator from TensorFlow is used to preprocess the training and test images. This includes rescaling, shearing, zooming, and horizontal flipping.

Model Training: The model is compiled using the Adam optimizer and binary cross-entropy loss function. It is then trained on the training set and evaluated on the test set.

Model Saving: After training, the model is saved to a file named "Gmodel.h5".

Image Selection: The user is prompted to select an image from the file explorer using a Tkinter GUI.

Image Processing: The selected image is loaded, resized to the required dimensions, and converted to an array.

Prediction: The model predicts whether the image contains glaucoma or not.

Annotation: The predicted result (glaucoma or not glaucoma) is annotated on the image.

Saving Annotated Image: The annotated image is saved to the output directory.

2. Algorithms Used:

Convolutional Neural Network (CNN): CNNs are a class of deep neural networks, most commonly applied to analyzing visual imagery. They have proven to be highly effective in image classification tasks.

ImageDataGenerator: Used for real-time data augmentation during training. It generates batches of tensor image data with real-time data augmentation.

3. Accuracy Percentage:

The accuracy percentage of the model is calculated after training and evaluation on the test set using the evaluate method. It represents the proportion of correctly classified images out of the total number of images in the test set.