

In the previous code, Keras was used to load a pre-trained Convolutional Neural Network (CNN) model that was previously trained to classify brain MRI images into one of four classes: Mild Demented, Moderate Demented, Non Demented, and Very Mild Demented.

Here's a breakdown of what Keras did in the previous code:

1. `from keras.models import load_model`: This imports the `load_model` function from the Keras library, which is used to load a pre-trained model from a file.
2. `model = load_model('cnn_model.h5', compile=False)`: This loads the pre-trained CNN model from the file `cnn_model.h5` into a Keras model object called `model`.
3. `test_image = image.load_img(img_path, target_size=(176, 176))`: This loads an image from a file at `img_path` using the `load_img` function from Keras. The `target_size` argument resizes the image to a specified size of 176 x 176 pixels.
4. `test_image = image.img_to_array(test_image)/255.0`: This converts the image to a Numpy array and normalizes its pixel values to be between 0 and 1.
5. `test_image = test_image.reshape(1, 176, 176, 3)`: This reshapes the image array into the correct shape expected by the CNN model, which is a 4D tensor of shape `(batch_size, height, width, channels)`.
6. `predict_x = model.predict(test_image)`: This passes the image through the pre-trained CNN model to get the model's predicted class probabilities for each of the four classes.
7. `classes_x = np.argmax(predict_x, axis=1)`: This finds the class with the highest predicted probability using the `argmax` function from Numpy.
8. `return verbose_name[classes_x[0]]`: This returns the name of the predicted class using a dictionary called `verbose_name`, which maps each integer class label to its corresponding string name.

Overall, Keras was used to load a pre-trained CNN model, preprocess the input image to the correct format expected by the model, and use the model to make a prediction on the input image.