Classifying **dyslexia using Convolutional Neural Networks (CNNs)** involves using deep learning techniques to analyze and classify features from input data, such as brain images or text. Below is a general methodology and flow chart for training and testing a CNN for dyslexia classification.

**Methodology:**

**Data Collection:**

Gather a dataset containing relevant information for dyslexia classification. This could include brain images, text data, or a combination of both.

Ensure the dataset is labeled with binary classes (e.g., dyslexic vs. non-dyslexic).

**Data Preprocessing:**

Clean and preprocess the data. This may involve resizing images, normalizing pixel values, or tokenizing and encoding text data.

Split the dataset into training and testing sets.

**Architecture Design:**

Design the CNN architecture. This involves defining the number of convolutional layers, pooling layers, fully connected layers, and the activation functions to be used.

Consider using pre-trained models or architectures proven effective in similar tasks.

**Model Compilation:**

Compile the model by specifying the loss function, optimizer, and evaluation metric.

Common loss functions for binary classification include binary cross-entropy, and popular optimizers include Adam or SGD.

**Training:**

Train the CNN using the training dataset.

Feed the input data through the network, calculate the loss, and update the model parameters through backpropagation.

Monitor training progress and adjust hyperparameters if needed.

**Validation:**

Evaluate the model on a validation set to ensure it generalizes well to unseen data.

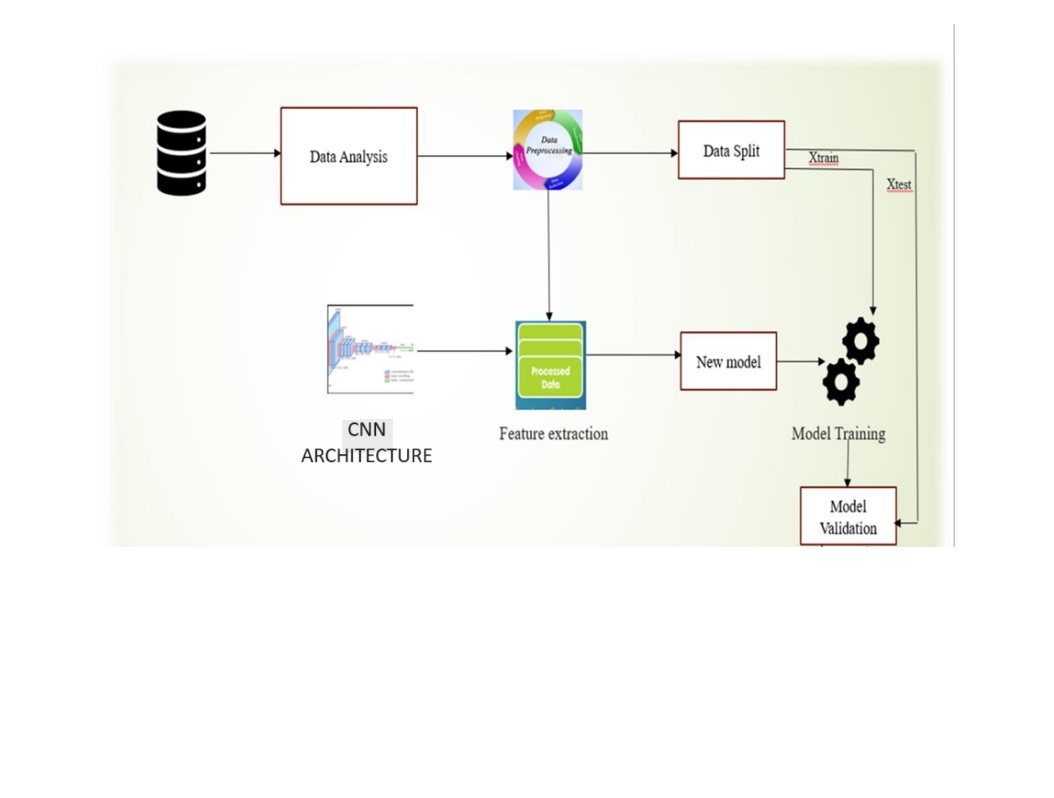
Adjust the model or training parameters based on the validation performance.

**Testing:**

Assess the trained model on the testing dataset to evaluate its real-world performance.

Calculate metrics such as accuracy, precision, recall, and F1-score.

**CLASSIFICATION USING CNN**

GENERAL SYSTEM ARCHITECTURE OF CNN

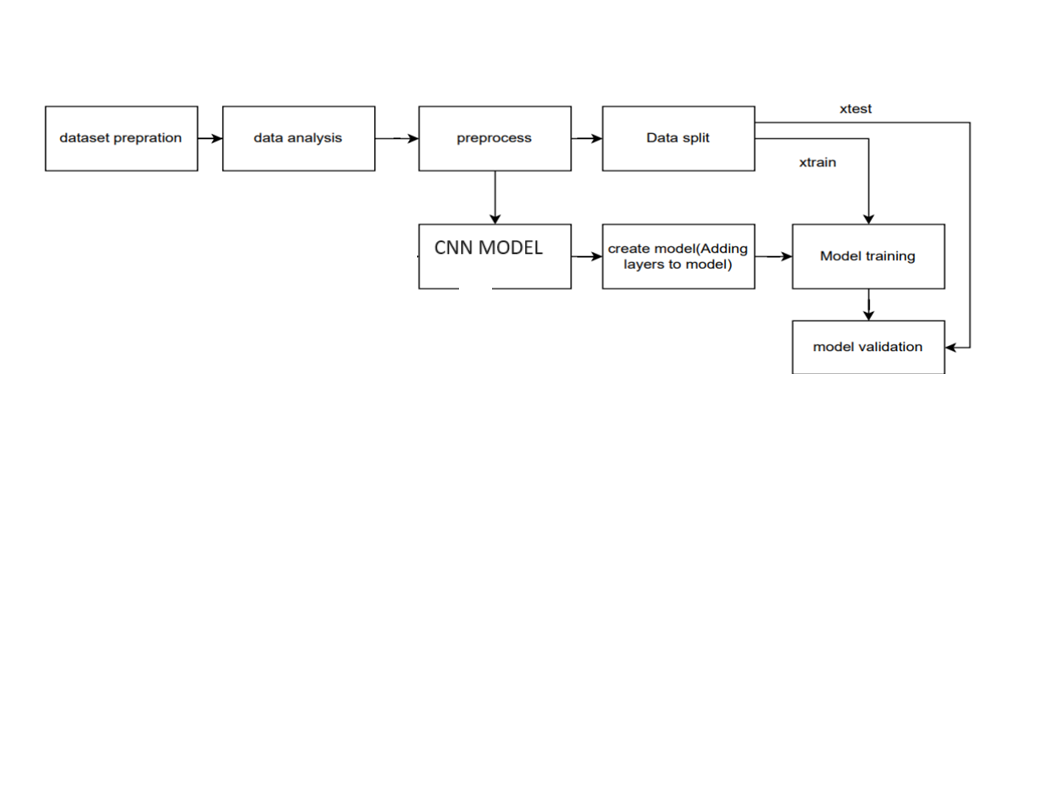
Class3

Class2

Class1

Class 1

Class5

.WORK FLOW

CLASS1

CLASS3

CLASS2