# **Assignment 3: Implementing Convolutional Neural Networks (CNNs) for Butterfly Multiclass Image Classification**

## **Problem Statement**

The problem is to develop a **Convolutional Neural Network (CNN)** model that can classify butterfly images into their respective species. The dataset consists of butterfly images organized into training and testing sets, with labels specifying the species of each butterfly.

## **Objective**

* To preprocess and prepare image data for deep learning.
* To implement a **CNN model** using Keras and TensorFlow.
* To train the model for multiclass classification of butterfly species.
* To evaluate the model’s accuracy and ability to generalize on unseen test images.

## **Requirements**

* **Operating System**: Windows / Linux / macOS (Colab recommended for GPU support)
* **IDE / Platform**: Google Colab / Jupyter Notebook

### **Libraries and Packages Used**

* **TensorFlow / Keras** → Building and training CNN model
* **Pandas** → Reading CSV files containing image names and labels
* **NumPy** → Data manipulation
* **Matplotlib / Seaborn** → Visualization of training results and predictions
* **Scikit-learn** → Evaluation metrics
* **OpenCV / PIL** → Image processing (if needed)

## **Theory**

### **Definition**

A **Convolutional Neural Network (CNN)** is a deep learning model designed to automatically and adaptively learn spatial hierarchies of features from images. CNNs are highly effective for image classification tasks.

### **Structure**

1. **Input Layer** – Raw butterfly image pixels.
2. **Convolutional Layers** – Apply filters to extract local image features (edges, textures, patterns).
3. **Pooling Layers** – Downsample feature maps to reduce dimensionality and computation.
4. **Fully Connected Layers (Dense)** – Combine extracted features to form higher-level understanding.
5. **Output Layer** – Softmax activation to classify the image into one of the butterfly species.

## **Methodology**

1. **Dataset Preparation**
   * Import training and testing data using provided CSV files (Training\_set.csv and Testing\_set.csv).
   * Map image filenames to labels for supervised learning.
   * Normalize image pixel values and resize all images to a standard size (e.g., 150×150).
2. **Model Construction**
   * Define a Sequential CNN model in Keras.
   * Add convolutional + pooling layers to extract features.
   * Flatten the features and pass through dense layers.
   * Use **softmax activation** in output for multiclass classification.
3. **Training**
   * Compile the model with optimizer (Adam) and loss function (categorical\_crossentropy).
   * Train on training data with data augmentation (rotation, flip, zoom) to prevent overfitting.
4. **Evaluation**
   * Predict butterfly species on test data.
   * Evaluate performance using accuracy, confusion matrix, and classification report.

## **Advantages**

* Automatically learns important image features without manual feature extraction.
* Performs well on **multiclass image classification problems**.
* Can generalize well with proper data augmentation.
* CNNs are computationally efficient due to weight sharing and local connectivity.

## **Limitations**

* Requires a **large dataset** for high accuracy.
* Computationally intensive — training can be slow without GPU support.
* Susceptible to **overfitting** if dataset is small.
* Less interpretable compared to simpler machine learning models.

## **Working / Algorithm**

1. Input butterfly image is resized and normalized.
2. Pass through **convolutional layers** → extract low-level and mid-level features.
3. Apply **pooling layers** to reduce dimensions.
4. Flatten feature maps into a vector.
5. Pass through **dense layers** for classification learning.
6. Final **softmax output** assigns probability to each butterfly species.
7. Predicted label = class with the highest probability.

## **Conclusion**

This assignment successfully demonstrates the application of **Convolutional Neural Networks (CNNs)** for multiclass butterfly species classification. The CNN effectively extracted spatial features from butterfly images and classified them into species with high accuracy. This project highlights the importance of CNNs in computer vision tasks and their applicability in real-world biodiversity monitoring and conservation.