**Project Proposal: Food101 Object Classification**

**Objective:**

Develop a machine learning model to accurately classify food images from the Food-101 dataset.

**Scope:**

1. **Data Preparation:** Collection of data from the source and preparation for the pre-processing of data.
2. **Model Selection**: Evaluate and choose deep learning models (e.g., CNNs) to identify which architecture may be best suited for the food classification task.
3. **Transfer learning**: Consider utilizing pre-trained models to reduce computational time and cost.
4. **Model Training and Evaluation**: Train models and evaluate performance.
5. **Optimization**: Fine-tune hyperparameters and architectures.
6. **Outcome**: Predict the class of the food object.
7. **Documentation**: Comprehensive documentation of the process and findings.

**Data Description:**

The Food101 dataset, originally used in the paper "Food-101 – Mining Discriminative Components with Random Forests" by Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. It consists of 16,256 images across 17 food classes. Data can be split into training and testing images, where each class can be roughly split into 70-30. Classes are typically the names of the food, e.g. apple-pie. Food-101 dataset can be obtained by visiting this original link given below: <https://www.kaggle.com/datasets/dansbecker/food-101>

**Plan:**

1. **Data Exploration:** Download the Food-101 dataset.
2. **Sampling:** Use a portion of dataset to reduce computational load and meet the system requirements.
3. **Data visualization:** Load the dataset, visualize data distribution, and identify any inconsistencies.
4. **Preprocessing**: Resize, normalize, and labeling.
5. **Model Selection**: Experiment with CNN architectures like VGG16, ResNet50.
6. **Model Transfer Learning:** Freeze the pre-trained model weights and add new layers on top for classification specific to the 17 food classes.
7. **Model Training:** Train the model on the training set, monitor validation performance to prevent overfitting.
8. **Evaluation**: Use metrics like accuracy, precision, recall, and F1-score.
9. **Hyperparameter Tuning:** Optimize hyperparameters (learning rate, epochs) for best performance**.**
10. **Outcome Prediction:** Apply the trained models to new data and validate their predictions.
11. **Documentation**: Detailed reporting of all steps and results with the help of synopsis, introduction, abstract, methodology etc.