**Project Idea: Classroom Occupancy Detection Using Image Classification**

**Concept:**

Develop an image classification system to determine whether university classrooms or study spaces are occupied or vacant. This tool can help students and staff quickly find available spaces for study or meetings.

**Why This Project?**

**Direct Relevance**: Specifically tailored to a common need in university settings.

Simplicity: Focuses on a binary classification task (occupied vs. vacant), which is simpler than multi-class classification.

**Practical Utility:** Assists students and staff in efficiently utilizing campus spaces.

**Educational Value:** Provides a hands-on introduction to image classification and binary classification problems.

**Dataset Creation:**

**Source:** Capture images of various classrooms and study spaces within your university. Ensure to take pictures at different times to represent both occupied and vacant states.

**Classes:** Two categories – ‘Full class’ and ‘Empty class’.

Image Requirements: Color images (RGB), covering diverse lighting conditions and angles.

**Dataset Split:**

Training Set: ~70% of images.

Validation Set: ~20% for model tuning.

Test Set: ~10% for final evaluation.

**Model Development:**

Approach: Use a simple Convolutional Neural Network (CNN) model. For beginners, models like ResNet or VGG with pre-trained weights on a dataset like ImageNet can be a good starting point, adapting them to your specific task.

**Tools:** Leverage frameworks like TensorFlow or PyTorch, which are beginner-friendly and have strong community support.

Prototype Application:

Design: A basic application (web or mobile) where users can see real-time information about the occupancy of rooms.

Functionality: Incorporate a feature to display the nearest vacant room or study space.

Documentation:

Include the process of collecting and labeling data, model training, and application development in your creative brief.

Discuss the challenges encountered, especially in terms of dataset creation and model accuracy, and how you overcame them.