Project Design Phase-I
Solution Architecture

Team ID	Team - 592300
Project Name	Mushroom classification using deep learning
Maximum Marks	5 Marks

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

1. Problem Definition:

- **Business Problem:** Accurate identification of edible and poisonous mushrooms is crucial for health and safety. Misidentification can lead to serious health issues, including death. Current manual identification methods are prone to human errors.
- **Solution:** Use deep learning to automate the classification process. This would reduce human error, and potentially offer a faster and more reliable identification method.

2. Structure of the Software

- **Input Layer:** Accepts image data of mushrooms. This might be raw pixel values of certain dimensions (e.g., 128x128x3 for color images).
- **Hidden Layers:** Several convolutional layers to detect features (like cap shape, stem, color), followed by pooling layers, and fully connected layers.
- **Output Layer:** Binary classification (e.g., edible or poisonous) or multi-class classification if identifying specific species or categories.

3. Characteristics & Behaviour:

- **Performance:** The model should have high accuracy (above 95% would be desirable) and low false positives, especially for poisonous classification.
- **Reliability:** It should offer consistent results across varying lighting conditions and mushroom orientations.
- **Scalability:** The system should handle a large number of images and be able to learn from new data.
- **User Interface:** A simple interface where users can upload images and receive classification results.

4. Features:

- **Image Upload:** Users can upload mushroom images.
- Real-time Classification: Once an image is uploaded, the software provides a classification result in real-time.
- **Database Integration:** Store images and corresponding classifications for future reference and to improve the model.
- **Feedback Mechanism:** Users can confirm if the classification was correct, helping the model learn.
- **Alert System:** If a mushroom is identified as poisonous, the software sends a clear warning.

5. Development Phases:

- **Data Collection:** Gather a substantial number of mushroom images and label them.
- **Data Preprocessing:** Normalize, resize, and augment the images to make them suitable for training.
- **Model Training:** Use a deep learning framework like TensorFlow or PyTorch to train a model.
- **Validation & Testing:** Split the dataset into training, validation, and testing sets to ensure the model performs well.
- **Deployment:** Once the model is robust, deploy the software for end-users.
- **Feedback Loop:** Implement a system for continuous learning from user feedback.

6. Solution Requirements:

- **Hardware:** GPU for faster training and prediction.
- **Software:** Deep learning frameworks (like TensorFlow or PyTorch), a web framework for the interface (like Flask or Django), and a database system.
- **Dataset**: A large, labeled dataset of mushroom images.
- **Skills**: Expertise in deep learning, web development, and database management.
- Specifications:
- **Image Resolution:** Determine a standard resolution, such as 128x128 or 256x256 pixels.
- Data Augmentation: Use techniques like rotation, zooming, and horizontal flipping to increase dataset size.
- **Model Architecture**: Start with popular architectures like ResNet or VGG, then fine-tune.
- Loss Function: Binary cross-entropy for binary classification or categorical cross-entropy for multi-class.
- **Optimizer:** Algorithms like Adam or SGD to minimize the loss function.
- With these specifications and a well-structured plan, the project will have a clear direction and better chances of success.

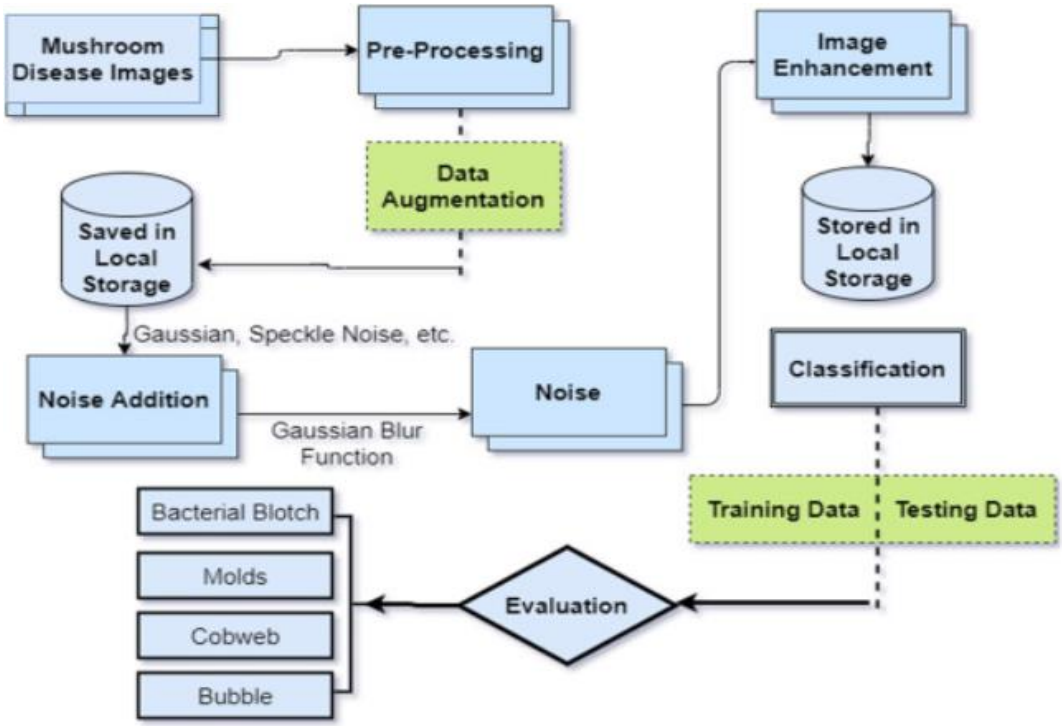

Figure 1: Architecture and data flow of mushroom classification using deep learning