SMARTBRIDGE
Let's Bridge the Gap
a Veranda Enterprise

Smart Internz

# Model Development Phase Template

| | |
|---|---|
| Date | 03 July 2024 |
| Team ID | SWTID1720085445 |
| Project Name | Hydration Essentials: Classifying Water Bottle Images |
| Maximum Marks | 5 Marks |

## Model Selection Report

In this report, we will outline the criteria and process for selecting the most suitable model architectures for future deep learning and computer vision projects. This involves evaluating various models, considering factors such as performance, complexity, and computational requirements.

### Introduction

Selecting the right model architecture is crucial for the success of any deep learning and computer vision project. This report will guide the selection process, ensuring that the chosen model meets the project's specific needs and constraints.

### Evaluation Criteria

1. **Performance**
    - **Accuracy**: The primary metric for most classification tasks. Models will be evaluated based on their ability to correctly classify images.
    - **Precision and Recall**: Important for tasks where the cost of false positives and false negatives differs.
    - **F1-Score**: Provides a balance between precision and recall, especially useful in imbalanced datasets.
    - **ROC-AUC**: For binary classification tasks, this metric helps in understanding the trade-offs between true positive rate and false positive rate.
    - **Mean Squared Error (MSE) and Mean Absolute Error (MAE)**: For regression tasks, these metrics will be considered.

2. **Complexity**
   - **Model Architecture**: The depth and number of layers in the model. Simpler models are generally easier to understand and debug.
   - **Number of Parameters**: Models with fewer parameters are less likely to overfit and require less storage and memory.
   - **Training Time**: The time required to train the model to an acceptable performance level.
   - **Inference Time**: The time required for the model to make predictions on new data.

**Model Selection Report:**

| Model | Description |
|---|---|
| Convolutional Neural Networks (CNNs) | • **LeNet**: A simple and efficient architecture suitable for basic image classification tasks.<br>• **AlexNet**: A deeper network that brought significant improvements in performance for more complex tasks.<br>• **VGGNet**: Known for its uniform architecture and depth, providing a good balance between performance and complexity.<br>• **ResNet**: Introduces residual connections, enabling very deep networks without suffering from vanishing gradients.<br>• **Inception (GoogLeNet)**: Uses inception modules to balance computational efficiency and depth.<br>• **DenseNet**: Each layer receives input from all previous layers, promoting feature reuse and reducing the number of parameters. |
| RNN | • **Basic RNNs**: Useful for sequential data but suffer from vanishing gradient issues.<br>• **LSTM (Long Short-Term Memory)**: Designed to remember long-term dependencies, making them suitable for time-series analysis.<br>• **GRU (Gated Recurrent Unit)**: A simpler alternative to LSTM, often providing similar performance with fewer parameters. |