**Model Performance Report**

**Overview**

This report summarizes the development and evaluation of a custom Convolutional Neural Network (CNN) model designed for image classification of handwritten prescriptions. The model was developed to outperform previous architectures, including VGG16, by optimizing input image size and architecture complexity.

**Model Development**

**Initial Attempts**

Initially, I employed a pre-trained VGG16 model, which is known for its robustness in image classification tasks. However, the model required input images of size (224 \times 224), leading to increased computational costs and longer training times.

To address this, I experimented with average pooling layers in the architecture, aiming to reduce overfitting and improve generalization. However, these attempts did not yield satisfactory results, as the model struggled to capture essential features from the smaller images, leading to lower accuracy on both validation and test datasets. The accuracy that I got with average pooling and max pooling was around 60 percent.

**Custom CNN Architecture**

After analysing the performance of the VGG16 model and the limitations of average pooling, we designed a custom CNN architecture optimized for (64 \times 64) pixel input images. The architecture included:

* **Convolutional Layers**: Three convolutional layers with increasing filter sizes (32, 64, and 128) to capture hierarchical features.
* **Max Pooling Layers**: Used instead of average pooling to retain important spatial information.
* **Dense Layers**: A fully connected layer followed by a dropout layer to mitigate overfitting, culminating in a SoftMax output layer for classification.

**Performance Metrics**

The model was evaluated using three datasets: training, cross-validation (CV), and test datasets. The following metrics were recorded:

| **Dataset** | **Accuracy (%)** | **Loss** |
| --- | --- | --- |
| Training | 91.19 | 0.12 |
| Cross-Validation | 73.46 | 1.009 |
| Test | 73.46 | 1.009 |

The custom CNN outperformed the VGG16 model, which achieved a maximum accuracy of 73.46% on the test dataset. The reduction in image size and the tailored architecture contributed to faster training times and improved performance.

**Resources Utilized**

**Hardware Specifications**

* **GPU**: NVIDIA GeForce RTX 3060
* **CPU**: 8 CPUs
* **RAM**: 16 GB

**Software Stack**

* **Framework**: TensorFlow 2.x for model development and training.
* **Libraries**: Keras for building the CNN architecture, NumPy and Pandas for data manipulation, and scikit-learn for evaluation metrics.
* **Environment**: Jupyter Notebook for interactive development and experimentation.

**Conclusion**

The custom CNN model demonstrated significant improvements over previous models, particularly in terms of accuracy and training efficiency. The decision to optimize the architecture for smaller input sizes and to avoid average pooling layers proved beneficial. Future work will focus on further fine-tuning the model and exploring advanced techniques such as data augmentation and transfer learning to enhance performance on more complex datasets.