Narrative Report | [Phase 3] Final Project

By: Kein Jake A. Culanggo

Phase 3 centers on the systematic evaluation of three convolutional neural network architectures to establish how well standard image-based deep learning models perform on the ASL fingerspelling task when trained from scratch on the Kaggle ASL dataset. The primary goal of this phase is to determine the baseline representational capacity of VGG16, ResNet18, and MobileNetV2 under controlled conditions, thereby creating a technical reference point that can later be compared against the practical demands of real-world recognition. This stage functions as an architectural benchmarking step, identifying how depth, residual connectivity, and lightweight design interact with the properties of a static ASL dataset. Although these experiments technically overlap with model experimentation, they are reported here because their outcomes directly inform the validity of the earlier data collection processes and help clarify whether the training dataset itself supports generalization to real testing conditions.

Although the quantitative results reported for VGG16, ResNet18, and MobileNetV2 provide clear evidence of architectural differences in representational capacity, these outcomes must be interpreted cautiously in relation to the earlier phases of the project. VGG16 and ResNet18 demonstrated strong generalization performance on the static Kaggle dataset, achieving test accuracies of 0.9722 and 0.9643, respectively, while MobileNetV2 exhibited a markedly lower accuracy of 0.7024 due to its constrained parameterization and lightweight design. These results suggest that high-capacity or residual architectures are more effective at capturing the fine-grained spatial structures required for ASL hand-sign recognition, particularly when trained from scratch on a dataset with limited intra-class variation. The stability of the loss curves for both VGG16 and ResNet18 and their rapid convergence toward low training and validation losses indicate that the dataset was sufficiently structured for these architectures to learn discriminative features in a static, controlled setting.

However, despite these strong numerical outcomes, the transition from dataset-based testing to real-world testing exposed a significant mismatch between model accuracy and practical usability. As stated in Phase 2, when the models were applied to our self-filmed ASL videos, performance degraded sharply: letters were confused with visually similar numbers, classification consistency deteriorated, and predictions became unreliable even under controlled filming conditions. This discrepancy necessitates a disclaimer. While this section reports model experimentation trends, the observations obtained during real-world testing were not part of Phase 3's architectural comparison but rather an extension of Phase 2's validation of the suitability and realism of the dataset. These performance issues arose not from model design choices but from our attempt to verify whether the dataset collected earlier truly supported generalization to authentic hand-sign scenarios.

The initial success of VGG16 and ResNet18 on the Kaggle dataset, contrasted with their failure to handle our own testing videos, suggests that the limitations of Phase 1 and Phase 2 are now clearer. It is possible that the dataset used for training lacked sufficient variability in lighting, background structure, arm visibility, skin tone variation, and finger articulation patterns. This is consistent with our observations: despite re-filming against plain white walls, controlling arm visibility, and standardizing hand position, the models treated the testing samples as out-of-distribution. These mismatches imply that the foundations established in Phases 1 and 2 were partially constrained by the initial dataset choice, and that the strong numerical performance recorded during Phase 3 should not be interpreted as evidence of real-world readiness.

Throughout this process, responsibilities remained consistent. Dela Cruz led the implementation, training, and troubleshooting of the CNN architectures. Abainza supported the computational requirements, including access to GPU resources essential for model iteration. Casino supervised the correctness and quality of the testing videos used in evaluation, ensuring that samples adhered to the expected input specifications. Culanggo refined and formalized the documentation, ensuring coherence across phases and correctly integrating insights that linked dataset limitations with model behavior.

Given the divergence between controlled dataset performance and real-world evaluation, the group is currently determining whether the problem arises from the limitations of the original training dataset or from the preprocessing pipeline applied to the testing videos. This analysis is ongoing. By next week, the group expects to establish whether a new dataset is required, whether the preprocessing workflow must be revised, or whether both components must be restructured to support models that generalize effectively beyond the static conditions of the Kaggle dataset.