**MSDS692 – Data Science Practicum 1**

**Progress Report for Week4**

**Project Details:**

The project aims to build a bone fracture detection system using deep learning and computer vision. The system will classify X-ray images into fractured vs. non-fractured categories, supporting clinicians in early and accurate diagnosis.

* *Type of Task: Image classification using CNNs and transfer learning (ResNet, EfficientNet).*
* *Data: Bone Fracture Detection CV dataset + MURA dataset (~2–3 GB combined).*
* *Methods: CNNs, preprocessing (augmentation, normalization, denoising), Grad-CAM for interpretability.*
* *Evaluation Metrics: Accuracy, Precision, Recall, F1-score, AUC.*

**Project Timeline:**

* Week 1 – Literature review, finalize problem scope, dataset exploration (DONE)
* Week 2 – Data preprocessing and cleaning (normalization, augmentation) (DONE)
* Week 3 – Baseline model development (simple CNN) (DONE)
* Week 4 – Implement transfer learning models (ResNet, EfficientNet) (DONE)
* Week 5 – Model training and hyperparameter tuning (In Progress)
* Week 6 – Model evaluation and visualization (ROC curves, Grad-CAM)
* Week 7 – Compare models, optimize performance, interpretability testing
* Week 8 – Final report preparation, results presentation, and documentation

**Planned Work for the Week:**

* *Implement transfer learning models (ResNet, EfficientNet).*
* *Adapt preprocessing pipeline for transfer learning input requirements.*
* *Fine-tune models using pre-trained weights.*

**Progress for the Week:**

* Completed integration of ResNet and EfficientNet architectures for fracture classification.
* Preprocessing pipeline finalized for transfer learning (224×224 inputs, ImageNet normalization).
* Successfully fine-tuned ResNet on the combined dataset, achieving promising preliminary results.
* EfficientNet successfully set up and trained for initial runs.
* Class imbalance addressed through oversampling and class-weighted loss functions.
* Verified training stability with loss/accuracy curves and validated balanced performance across classes.

**Roadblocks/Issues:**

* Computational limitations: Full training runs on EfficientNet are still slow on local hardware; exploring cloud GPU solutions.
* Some label inconsistencies in merged datasets required additional cleaning.
* Dataset imbalance (fractured vs. non-fractured images) requires class weighting and oversampling (partially addressed with custom loaders).
* Computational resources: Training ResNet on full dataset is slow on local machine; GPU/cloud resources may be needed.

**Plan for next Week:**

* Conduct hyperparameter tuning for ResNet and EfficientNet.
* Run extended training experiments on cloud GPU for efficiency.
* Begin evaluation with Grad-CAM visualizations for interpretability.
* Start comparing model performances (baseline CNN vs. ResNet vs. EfficientNet).