CSEN 240 - Machine Learning
Team 2
Knee Osteo Classification CNN Project

Ulises Chavarria
Maximilian Khan

Objective:

Construct a convolutional neural network with the highest rate of accuracy against a locked testing test for the classification of osteoarthritis images between normal, osteopenia, and osteoporosis.

Approach:

Ulises focused on the fine-tuning of various models through a custom written testing suite, while Max experimented while the overall construction of models with added custom features.

Models Used (imagenet base):

Xception, ResNet50, ResNet50V2, ResNet101, ResNet101V2, ResNet152, ResNet152V2, Vision Transformer (ViT), EfficientNetV2S, EfficientNetV2M, EfficientNetV2L, EfficientNetV2B0

Feature Experimentation & Layering:

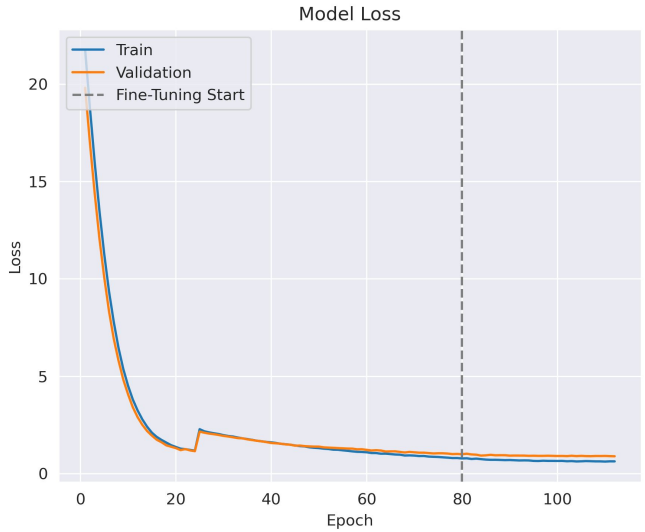
Augmentation, Batch Normalization, Dense, Dropout, Global Average/Max Pooling, Sobel Filters

Test Suite Features:

Logging, JSON Config, Traceability, Remote Access, Cross-Platform Development, Makefile, Utility Script (Model Creator, Plotter, &c), Mixed Precision, Asserting Deterministic Outcomes

SANTA CLARA UNIVERSITY

SCHOOL OF ENGINEERING




```
Users > ulises > github > CSEN-240-Project > out > 2025-03-11T01:16:20_Darwin > ≡ knee_osteo_2025-03-11T01:16:20_Darwin.log
'025-03-11 01:16:20,520 - INFO - config.json copied to out/2025-03-11T01:16:20_Darwin/config_2025-03-11T01:16:20_Darwin.json
'025-03-11 01:16:20,521 - INFO - /Users/ulises/github/CSEN-240-Project/knee-osteo.py copied to out/2025-03-11T01:16:20_Darwin/scripts
'025-03-11 01:16:20,521 - INFO - create_model_util.py copied to out/2025-03-11T01:16:20_Darwin/scripts/utils
'025-03-11 01:16:20,521 - INFO - system_specs_util.py copied to out/2025-03-11T01:16:20_Darwin/scripts/utils
'025-03-11 01:16:20,522 - INFO - plot_utils.py copied to out/2025-03-11T01:16:20_Darwin/scripts/utils
'025-03-11 01:16:20,522 - INFO - knee-osteo project copied to out/2025-03-11T01:16:20_Darwin/scripts
'025-03-11 01:16:20,522 - INFO - --- START ---
'025-03-11 01:16:20,522 - INFO - Start Time: 2025-03-11T01:16:20
'025-03-11 01:16:20,526 - INFO - --- SYSTEM SPECIFICATIONS ---
'025-03-11 01:16:20,526 - INFO - System Platform: Darwin | 24.3.0 | Darwin Kernel Version 24.3.0: Thu Jan  2 20:22:58 PST 2025; root:x
'025-03-11 01:16:20,526 - INFO - CPU: arm | Cores: 10 | Threads: 10 | Max Frequency: 4 MHz
'025-03-11 01:16:20,526 - INFO - RAM: 16.00 GB (Total) | Available RAM: 10.83 GB | Used RAM: 4.11 GB
'025-03-11 01:16:20,526 - INFO - GPU (TF): [PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
'025-03-11 01:16:20,526 - INFO - GPU Driver: N/A
'025-03-11 01:16:20,526 - INFO - --- SYSTEM SPECIFICATIONS ---
'025-03-11 01:16:20,526 - INFO - --- config.json ---
'025-03-11 01:16:20,526 - INFO - config: {'enable_plots': True, 'save_plots': True, 'show_plots': False, 'enable_tf_determism': Fals
'025-03-11 01:16:20,526 - INFO - hyperparameters: {'unfreeze_last_n_layers': 150, 'batch_size': 64, 'img_size': [224, 224], 'channels'
'025-03-11 01:16:20,526 - INFO - augmentation: {'rotation_range': 180, 'width_shift_range': 0.1, 'height_shift_range': 0.1, 'shear_ran
'025-03-11 01:16:20,526 - INFO - --- config.json ---
'025-03-11 01:16:20,527 - INFO - Using Metal backend for acceleration on Apple Silicon
'025-03-11 01:16:20,652 - INFO - Plot saved at out/2025-03-11T01:16:20_Darwin/label_distribution_2025-03-11T01:16:20_Darwin.png
'025-03-11 01:16:20,880 - INFO - Plot saved at out/2025-03-11T01:16:20_Darwin/sample_images_2025-03-11T01:16:20_Darwin.png
'025-03-11 01:16:20,933 - INFO - Creating Learning Rate Scheduler with ReduceLROnPlateau
'025-03-11 01:16:20,933 - INFO - Creating model with base model: Xception
'025-03-11 01:16:21,653 - INFO - Model summary (BEFORE): None
'025-03-11 01:18:44,282 - INFO - Epoch   1 | Accuracy: 0.3499 | Loss: 9.7260 | Val Accuracy: 0.2778 | Val Loss: 6.7449
'025-03-11 01:20:59,059 - INFO - Epoch   2 | Accuracy: 0.4130 | Loss: 5.4878 | Val Accuracy: 0.3439 | Val Loss: 4.1144
'025-03-11 01:23:14,733 - INFO - Epoch   3 | Accuracy: 0.5155 | Loss: 3.0221 | Val Accuracy: 0.2646 | Val Loss: 4.1222
```

Operating Systems Used:

MacOS, Ubuntu

GPU/Processing Used:

Apple M1 (16GB), Apple M4 (16GB), Nvidia RTX 3070 (8GB), Nvidia RTX 3090 TI (24GB)

Total Hours Testing & Training:

85+ Combined

Longest Test Run:

~12 Hours

Winning Model:

EfficientNetV2S w/ Batch Normalization, Dense, and Dropout @ 1.5 hours

Head: [Dense 1024 + Dense 512 + Dense 256 + Dense 3] * Batch N + Dropout

3024 Training Samples Augmented

F1 Scores:

Class 1: 89%

Class 2: 89%

Class 3: 85%

Total Test Accuracy: 87.57%

What Worked Well:

Frozen Epochs, Unfreezing Trainable Layers Slowly, High Epoch Number, Lower Learning Rate, Collaboration, 32 Batch Size, Noise, Imagenet works by default

Takeaway + Potential Improvements:

Lower the added dense layers, Sobel is a mixed bag and typically unnecessary, choose a lower dropout than 0.5 and bring back Monte Carlo dropout, and employ the use of an ensemble model with attention (global vs. local feature extraction, which is why we first thought to use Sobel). Batch training.