# Abnormality detection based on musculoskeletal radiographs

Ritika Kumar, Noah Rae-Grant, Xin Wang

DS 5220, Fall 2024

[Github Repository]

## Abstract

Abnormality detection in musculoskeletal radiographs is crucial for diagnosing a wide range of conditions. This project evaluates two approaches: a unified model trained on all extremities and specialized models for individual extremities. Conducting experiments on building basic CNN architectures and pretrained models like ResNet-50 and DenseNet, we analyze their performance on the MURA dataset. The project aims to assess whether a unified model or specialized models for each extremity provide better performance in detecting abnormalities. The unified model processes images from all extremities, while the specialized models are trained for specific extremities, such as the hand, wrist, or elbow.

## Overview

### Introduction

Bone abnormality detection affects patient care, diagnostic speed, and treatment outcomes. Early and accurate detection helps in diagnosing conditions before they worsen, allowing for timely intervention. Radiographs are often the first line of imaging used to identify musculoskeletal issues, such as fractures, infections, or tumors. Our project aims to detect the presence of abnormality in 7 upper extremity classes based on radiographs of patients

### Motivation

We have a few different reasons we decided to tackle this project:

1. We wanted the content of our project to address a real-world problem that machine learning can aid.
2. We wanted to work more with images and convolutional neural networks.

To that end, we settled on bone abnormalities for the following reasons:

1. Enhanced Diagnostic Speed and Efficiency: ML-based image classification can analyze radiographs quickly and accurately, which is especially valuable in busy clinical settings where radiologists may have hundreds of images to review.
2. Improved Accuracy in Detection of Subtle or Rare Conditions: Advanced ML models trained on extensive, diverse datasets can detect patterns in radiographs that even experienced radiologists might overlook, particularly for uncommon or complex cases.
3. Standardized Diagnosis and Reduced Diagnostic Variability: By offering a standardized analysis, ML models can reduce variability and improve overall diagnostic reliability, ensuring that all patients receive the same high level of care.

# Experimental setup

## 1. Dataset and preprocessing

The MURA dataset contains 14,863 musculoskeletal studies of the arm, where each study contains one or more views and is manually labeled by radiologists as either normal or abnormal. The standard upper extremities include: Elbow, Finger, Forearm, Hand, Humerus, Shoulder, and Wrist.

The baseline model from the original paper[1] uses a 169-layer convolutional neural network to detect and localize abnormalities. The model takes as input one or more views for a study of an upper extremity. On each view, the model makes the binary prediction of abnormal if the probability of abnormality for the study is greater than 0.5.

The original model was evaluated on Cohen's kappa statistic, which expresses the agreement of the model with the gold standard. It was comparable to radiologist performance in detecting abnormalities on finger studies and equivalent on wrist studies, but performed lower than best radiologist performance in detecting abnormalities on elbow, forearm, hand, humerus, shoulder studies, and overall.

Transformations were applied to the training and validation images based on the model used for training.

**Data distributions** used for training, validation and testing: (Normal images are labeled 0 while abnormal images are labeled 1.)

| Upper extremity class | Training data | Validation data | Test data |
|---|---|---|---|
| Wrist | # Normal: 4626<br># Abnormal: 3183 | # Normal: 1139<br># Abnormal: 804 | # Normal: 364<br># Abnormal: 295 |

[1] Rajpurkar, Pranav, *et al*. “MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs,” May 2018.

| | | | |
|---|---|---|---|
| Shoulder | # Normal: 3341<br># Abnormal: 3316 | # Normal: 870<br># Abnormal: 852 | # Normal: 285<br># Abnormal: 278 |
| Hand | # Normal: 3224<br># Abnormal: 1200 | # Normal: 835<br># Abnormal: 284 | # Normal: 271<br># Abnormal: 189 |
| Finger | # Normal: 2482<br># Abnormal: 1612 | # Normal: 656<br># Abnormal: 356 | # Normal: 247<br># Abnormal: 214 |
| Elbow | # Normal: 2321<br># Abnormal: 1647 | # Normal: 604<br># Abnormal: 359 | # Normal: 235<br># Abnormal: 230 |
| Forearm | # Normal: 917<br># Abnormal: 548 | # Normal: 247<br># Abnormal: 113 | # Normal: 151<br># Abnormal: 150 |
| Humerus | # Normal: 546<br># Abnormal: 483 | # Normal: 127<br># Abnormal: 116 | # Normal: 148<br># Abnormal: 140 |

## 2. Model Training and evaluation

The problem of abnormality detection in musculoskeletal radiographs is inherently complex due to the diversity of anatomical regions and the subtlety of abnormalities. To address this, we adopted two distinct modeling approaches.

The first approach involves training a single model to process radiographs from all extremities collectively. This method ensures generalizability and simplicity, as the model is designed to handle diverse input data and learn shared patterns across different anatomical regions. A unified model is particularly useful in scenarios where computational resources are constrained or when deploying a single, all-encompassing diagnostic system is desired. However, this approach can struggle with capturing extremity-specific features, especially when certain abnormalities are rare or anatomically distinct.

In contrast, the second approach involves training individual models for each extremity, such as the hand, wrist, or elbow. By focusing on a single anatomical region, specialized models can learn region-specific patterns and intricacies, leading to improved accuracy and precision. This approach is particularly advantageous when there is sufficient labeled data for each extremity, allowing the models to optimize performance without interference from unrelated features of other extremities. Specialized models excel in tasks where domain-specific precision is critical, such as identifying rare or subtle abnormalities unique to a particular region.

We transfer these two approaches to 2 distinct sets of model architectures. One is building a convolution neural network from scratch and the other involves the use of pre-trained models to train a model on downstream tasks through transfer learning.

## Convolutional Neural Networks

Our goal with creating our own convolutional neural networks (CNNs) from scratch was to establish a baseline comparison between the smaller networks we've used in class, the 169-layer network from the original paper, and the transfer learning models we later created. Though we knew these would likely be too simple to produce any meaningful results, these CNNs were useful for testing different optimizations we later applied to the transfer learning models.

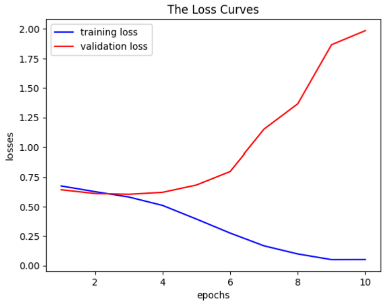
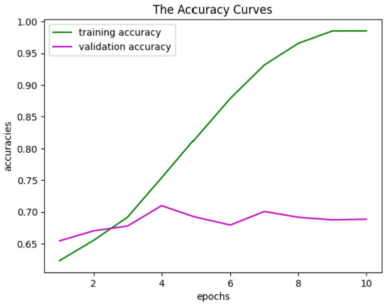
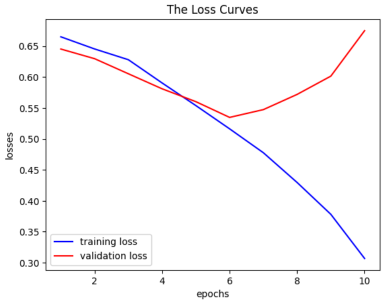
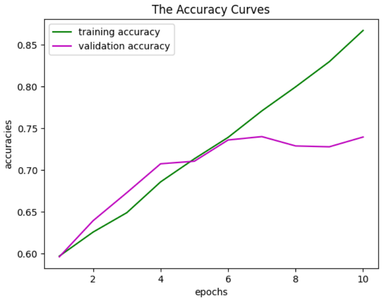
### a) Specialized Models

Our initial designs for the specialized model design, in which each part of the arm has its own model, were disheartening to say the least. The accuracy of the initial design attempts was little better than guessing randomly.

One such model, tested on the elbow data, plateaued at a .68 training loss after only two epochs and ended with a testing accuracy of 51%. Another, tested on the wrist and finger data, successfully reduced its training loss to .047, but still only had a testing accuracy of 55%, suggesting that the model overfit the data.

### Issues:

1. When calculating the training loss and validation loss for each epoch, it looks like validation loss cannot decrease. Test accuracy is 62.82%. Possible reason: The model (2(conv+pool)+2fc) on XR_WRIST is **too simple to learn the information** in training data. Using a deeper model (4(conv+pool)+fc+dropout+fc), the accuracy increased to 68.44%.

| | | |
|---|---|---|
| A simple model<br>2 (conv+pool)+2FC on wrist images<br><br>Test accuracy=62.82% |  |  |
| A deeper model (4(conv+pool)+fc+dropout+fc) on wrist images<br><br>Test accuracy=68.44% |  |  |

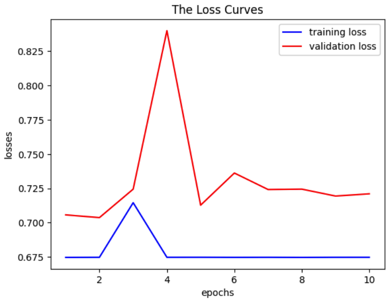
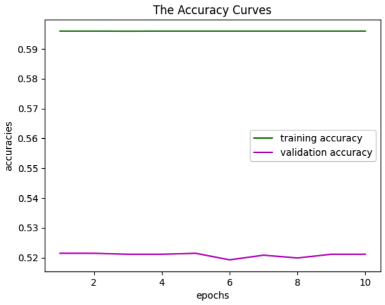
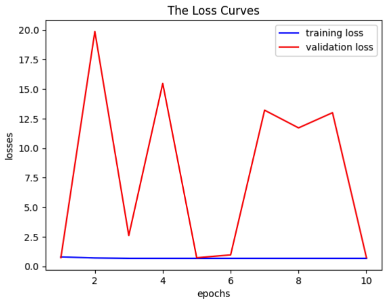
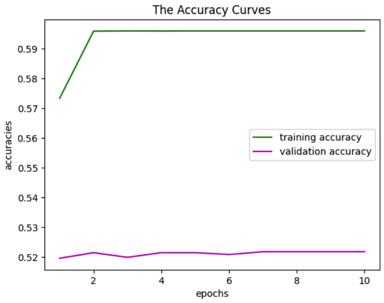
2. Initially, the outputs of the model were not passed through the sigmoid function correctly. Updating that later required a modification to the accuracy calculation function.

### Attempted Solutions and Optimizations

1. Batch normalization: After a convolution layer calculates its output, batch normalization will rescale the data and can (and often does) make the training process both faster and more stable.
2. Learning rate scheduling: This adjusts the learning rate between epochs depending on the test loss of the preceding epoch, allowing for a faster learning rate at the beginning, then a slower one so the model can fine-tune later.
3. Adaptive learning rate: Instead of using regular stochastic gradient descent, we used the Adam adaptive gradient descent algorithm. Adam updates the momentum of the algorithm based on running averages of the gradient and momentum.

### b) Unified Model

After our results with the specialized models, we did not expect much out of our from-scratch unified model. The attempts were, perhaps unsurprisingly, little better than random guessing.

| | | |
|---|---|---|
| First model: 2 * (conv + conv + pool + batch) + fc + dropout + fc<br>Testing accuracy = 52.11% |  |  |
| Second model: 2 * (conv + conv + pool) + batch + (conv + conv + conv + pool) + batch + fc + dropout + fc<br>Testing accuracy = 52.17% |  |  |

As shown by both "from-scratch" models, neither was really learning. This is likely due to having too few convolution layers for the kind of data we were training them on – meaning the models were too simple – and also the fact that we were trying to create a unified model from seven distinctive subsets of data.

Compared to the original paper's 169-layer convolutional neural network, our unified models capped out at 15 layers, the largest we were reasonably able to run[2] on the machines we had access to. Despite the failure of this model, it did provide a baseline comparison for our later transfer learning models.

## Transfer learning

As highlighted by the previous experiments, convolutional neural networks (CNNs) have certain limitations when applied to complex tasks such as abnormality detection in musculoskeletal radiographs. Hence, we moved on to leveraging more advanced architectures such as ResNet and DenseNet, which utilize transfer learning to significantly enhance model performance by building on pretrained weights from large-scale datasets.

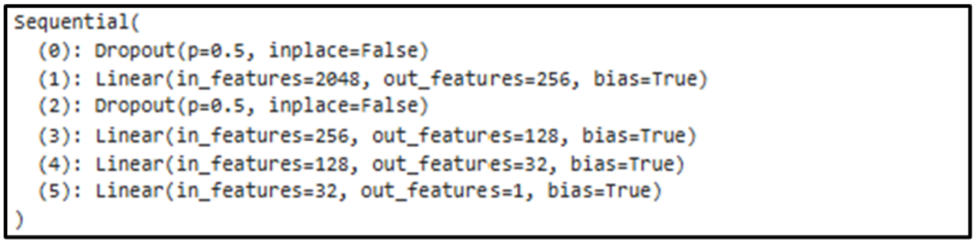
### 1. ResNet-18 or ResNet-50 on separate classes

ResNet is ideal for XR image binary classification due to its ability to train deep networks effectively using residual connections, capturing intricate patterns critical for diagnosis. Its pre-trained models enable transfer learning, reducing data requirements and improving accuracy. ResNet-18 and ResNet-50 were chosen for this task. The number of features from ResNet-18 is 512 while the number of features from ResNet-50 is 2048.

**Hyperparameter settings**:

| Batch Size | Optimizer | | | Epochs |
|---|---|---|---|---|
| | Name | Learning rate | Other params | |
| 32 | SGD with Momentum | 0.001 | Momentum:0.8<br>Weight decay: 0.001 | 20 with Early Stopping patience: 4 |

**Classifier architecture:**



```
Sequential(
  (0): Dropout(p=0.5, inplace=False)
  (1): Linear(in_features=2048, out_features=256, bias=True)
  (2): Dropout(p=0.5, inplace=False)
  (3): Linear(in_features=256, out_features=128, bias=True)
  (4): Linear(in_features=128, out_features=32, bias=True)
  (5): Linear(in_features=32, out_features=1, bias=True)
)
```

Using the same hyperparameter settings and binary classifier, the test accuracy varies significantly across the seven extremity classes, reflecting differences in data complexity and model generalization. The loss curves (Appendix A) provide insights into these dynamics, showcasing variations in overfitting behavior and validation performance. Training loss decreases smoothly for all classes, confirming consistent learning, but the validation loss trends diverge. Early overfitting is prominent in "Wrist", "Shoulder",

[2] This took over 5 hours anyway.

"Hand", and "Elbow" with validation loss rising around epochs 4-6. In contrast, "Forearm" and "Humerus" exhibit delayed overfitting, with validation loss increasing only after epoch 12, suggesting better overall stability. "Hand" and "Finger" show pronounced fluctuations in validation loss, likely due to inherent dataset complexity, imbalanced labels, or noisier data distributions.

The "Humerus" class demonstrates the most stable loss trends and generalization, achieving the highest test accuracy among all classes, indicating the dataset for this class is relatively clean and well-separated. On the other hand, the "Hand" and "Finger" classes may require further regularization techniques, such as increased dropout or data augmentation, to mitigate fluctuations and improve performance. Early stopping effectively prevents excessive overfitting across all classes, halting training when validation performance begins to degrade, but the timing of overfitting varies depending on the dataset's characteristics.

Overall, the variation in test accuracy reflects the influence of dataset separability and class-specific complexity on model performance. While most classes achieve respectable accuracy, the results indicate that classes with noisier or imbalanced data ("Hand", "Finger") might benefit from additional preprocessing or tailored training strategies. The "Humerus" class stands out with robust generalization, demonstrating how dataset quality and separability directly impact model performance.

**Interesting phenomenon**: While the test accuracy on wrist images is only 78.3% using ResNet-50 + the classifier, it increases to 83% when using ResNet-18 + the same classifier. One possible explanation for this phenomenon is that the number of features from ResNet-50 (2048) is much higher than that from ResNet-18 (512). As a result, in the classifier, directly reducing the number of features from 2048 to 256 in the first fully connected (FC) layer might be less effective than reducing the features from 512 to 256. The same phenomenon occurred when training ResNet-18 and ResNet-50 on the whole dataset where ResNet-18+classifier results in test accuracy 78.54% while ResNet-50+classifier results in 77.20%.

2. DenseNet

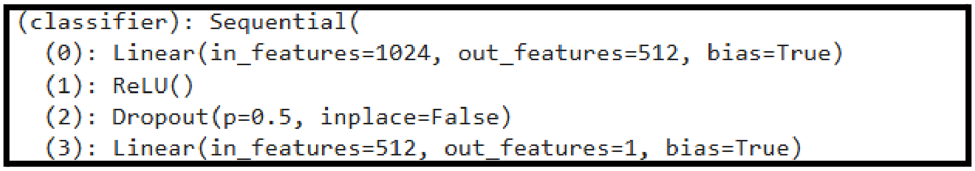
DenseNet was chosen for this task due to its unique architecture that promotes feature reuse and efficient gradient flow. Unlike traditional CNNs, DenseNet connects each layer to every other layer in a feed-forward manner, ensuring that features learned in earlier layers are directly accessible to later layers. This connectivity reduces the risk of vanishing gradients and helps the model learn richer feature representations, which is crucial for detecting subtle abnormalities in musculoskeletal radiographs.

The DenseNet version chosen for the task was densenet-121. The computational power used for training was 2xT4 GPU and training took around 2.5 hours for the unified model.

**Hyperparameter settings:**

| Batch Size | Optimizer | | | Epochs |
|---|---|---|---|---|
| | Name | Learning rate | Other params | |
| 32 | SGD with Momentum | 1e-4<br>With scheduler<br>Step size = 3<br>Gamma: 0.3 | Momentum:0.9<br>Weight decay: 1e-4 | 20 with Early Stopping patience: 2 |

**Classifier Architecture:**



```
(classifier): Sequential(
  (0): Linear(in_features=1024, out_features=512, bias=True)
  (1): ReLU()
  (2): Dropout(p=0.5, inplace=False)
  (3): Linear(in_features=512, out_features=1, bias=True)
```

We utilized Stochastic Gradient Descent (SGD) with momentum as the optimization strategy to enhance model convergence and stability during training. Other optimizers, such as Adam and AdamW, were very unstable during training and resulted in less test accuracy, despite reaching similar training accuracies. To prevent overfitting, especially on a limited medical dataset like MURA, we employed early stopping. This technique monitors the model's performance on a validation set and halts training when the validation performance stops improving for a specified number of epochs.

As this is a binary classification of abnormality, we used binary cross-entropy with logits as the loss function. This choice was made because binary cross-entropy effectively quantifies the difference between the predicted probability of a class and the actual class label. Using the logits version ensures computational efficiency and avoids potential instability when calculating probabilities directly.

a. Unified Model
   The unified model performed well using the above configuration, reaching a validation accuracy of 81.03% and test accuracy of 78.32%. While hyperparameter tuning the main issue were overfitting after around 9 epochs as well as unstable training using optimizers like Adam and AdamW which usually are known to achieve good results.

   SGD with momentum as the optimizer solved the issue of unstable training and using Early stopping helped tackle overfitting.

b. Specialized models
   The specialized model took on an average 1 hour of runtime. While there was considerable tuning applied for the specialized wrist model, not all models could be optimally tuned given the time as well as computational constraints.

While most of the models with light tuning were able to beat the benchmark set by the original paper, the specialized model for hand and wrist are still not at par with the set benchmark.

In general, DenseNet was able to perform well on abnormality detection. As further scope of this approach, hyperparameter tuning of specialized models has potential to improve model performance. Additionally, using other versions of DenseNet (e.g 201) can also help in improving performance.

## Results

| Model Category | Total Images in Dataset (Augmented Total) | Original Paper's Model | "From Scratch" Model | Transfer Learning Model (ResNet) | Transfer Learning Model (DenseNet) |
|---|---|---|---|---|---|
| | | **Test Accuracy** | | | |
| Specialized (Elbow) | 1,912 (5,396) | 71% | 50.54% | 77.85% | 80.00% |
| Specialized (Finger) | 2,110 (5,567) | 38.9% | 46.42% | 73.54% | 75.92% |
| Specialized (Forearm) | 1,010 (2,126) | 73.7% | 49.83% | 78.41% | 77.41% |
| Specialized (Hand) | 2,185 (6,003) | 85.1% | 58.91% | 67.17% | 70.87% |
| Specialized (Humerus) | 727 (1,560) | 60% | 51.39% | 82.64% | 84.72% |
| Specialized (Shoulder) | 3,015 (8,942) | 72.9% | 49.38% | 74.42% | 76.37% |
| Specialized (Wrist) | 3,697 (10,411) | 93.1% | 68.44% | 83.00% | 83.00% |
| Unified (Overall) | 14,656 (40,005) | 70.5% (average of all 7 categories) | 52.17% | 78.54% | 78.32% |

## Conclusion

While we don't need to reinvent the wheel with convolutional neural network design, understanding what goes into a successful one is key for knowing how to effectively fine-tune pre-trained models. Our difficulties with designing an effective model for the different categories of musculoskeletal radiographs, or at least one that could perform significantly better than random guessing, demonstrated the sheer amount of work that goes into moving beyond simple convolutional networks into more complex designs.

Our "from scratch" 15-layer model design only performed better in the finger category, and even then it was less accurate than random guessing should be. Though they weren't able to compete with the pre-trained models, we did still learn what worked and what didn't when creating our "from scratch" models, and we were able to take those optimizations into our work with the transfer learning models.

For most extremity categories, we were able to create a transfer learning model that performed better than the original paper's model. For example, both the ResNet-50 and the DenseNet models out-performed the other two models in the finger category.

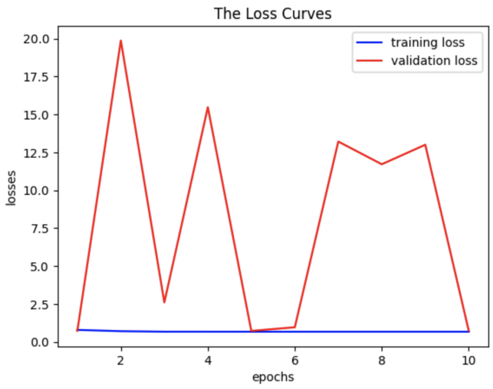
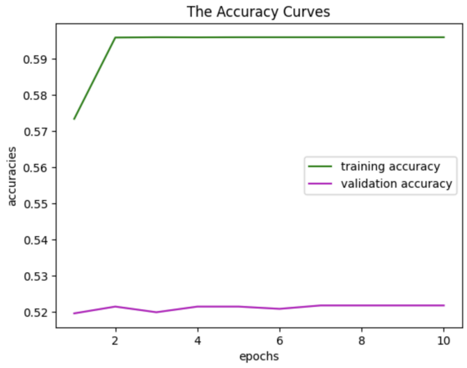
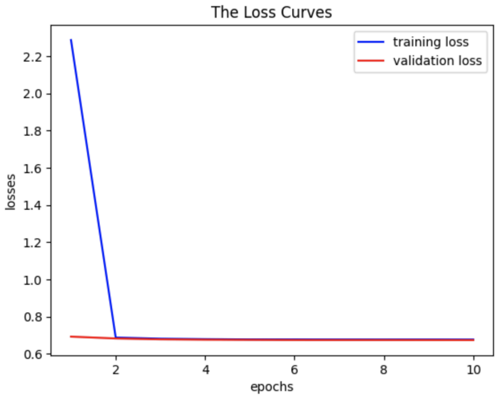
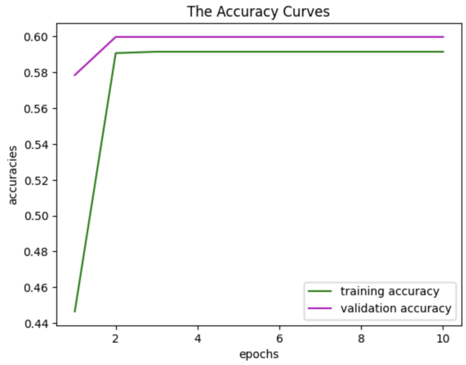
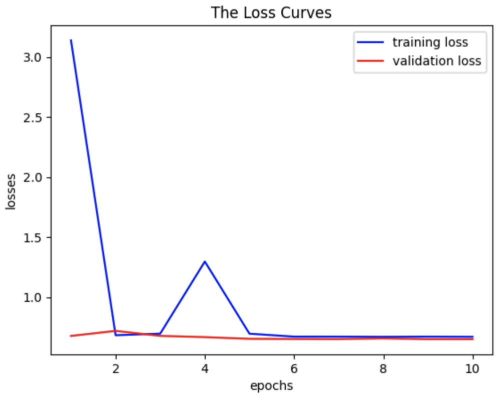
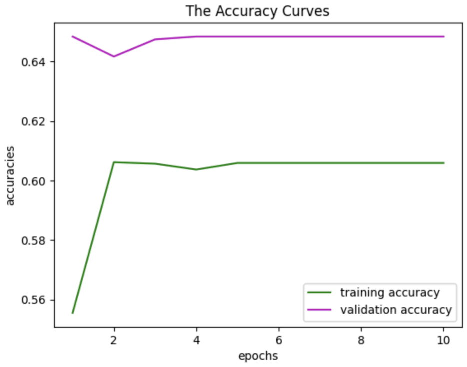
However, unlike the other specialized categories, we were not able to train models for the hand or wrist that performed at or better than the original paper's accuracy. Our wrist models had reasonably high accuracy at 83%, and we likely could achieve better results with sufficient tweaking of hyperparameters in the future.

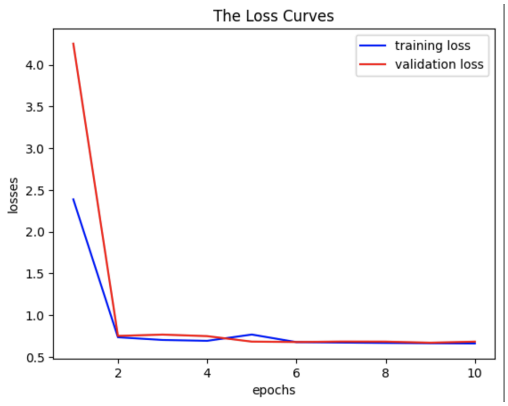
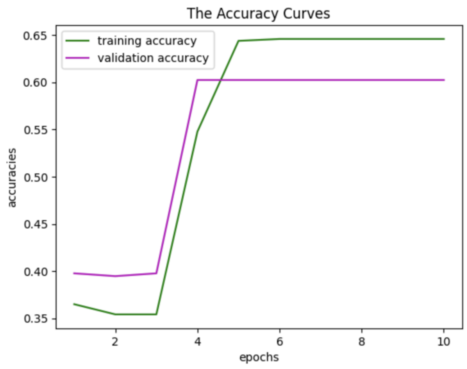
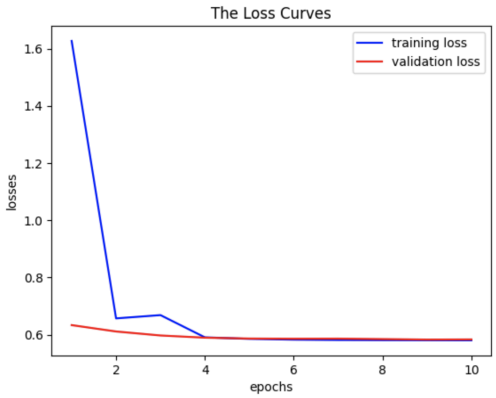
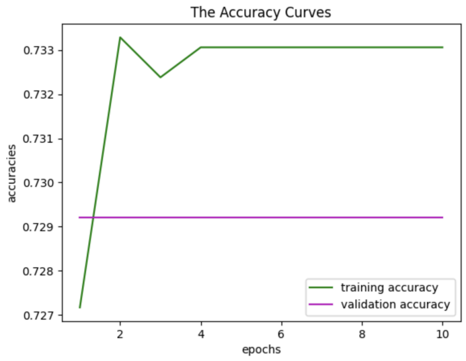
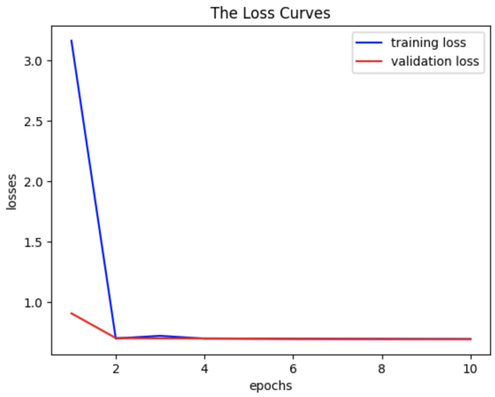
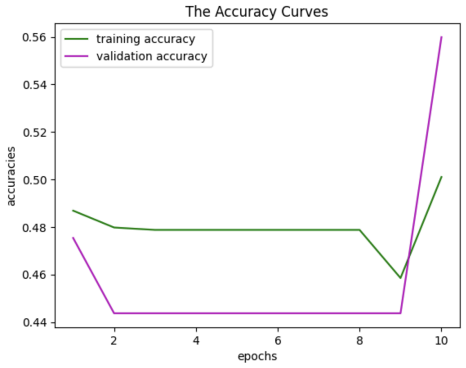
Despite this, both the DenseNet and ResNet unified models had a higher accuracy than the average of the original model, suggesting that we were able to achieve our goal of a more accurate general-purpose model.

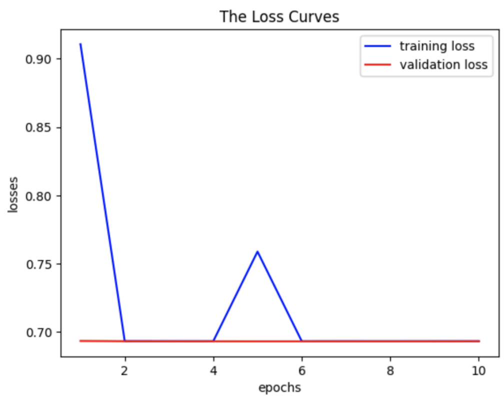
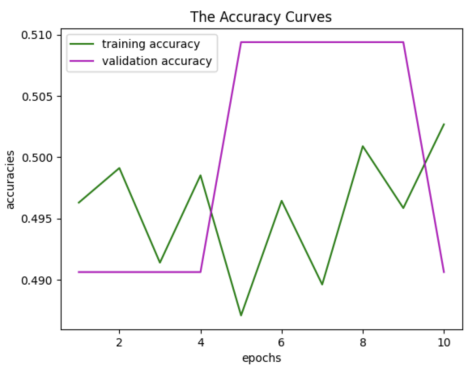
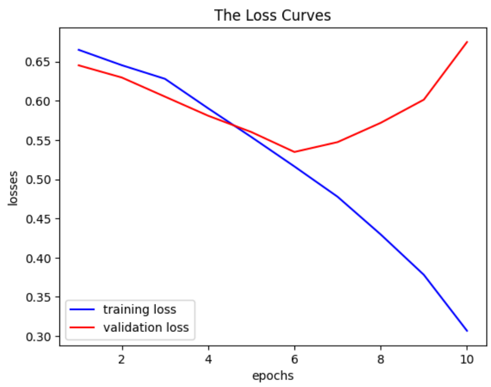
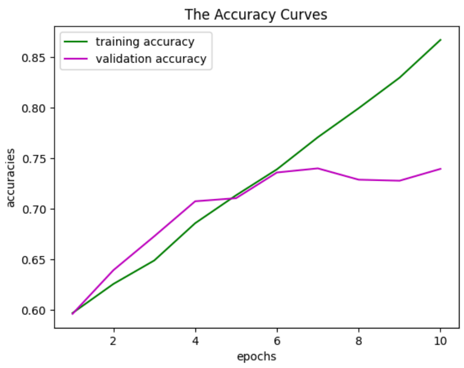
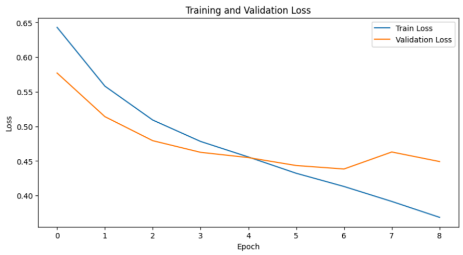
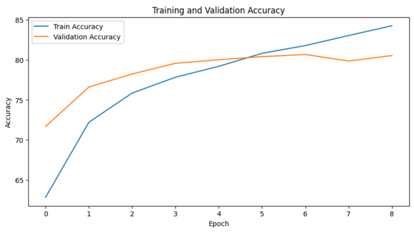
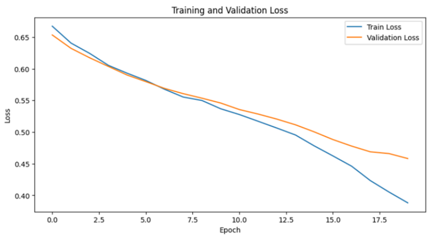
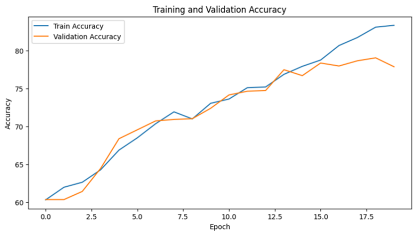
## References

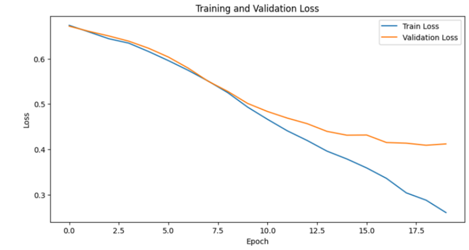
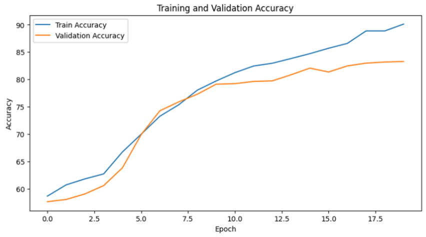
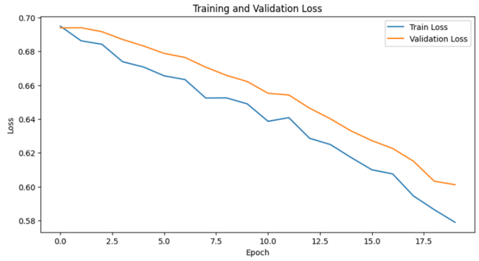
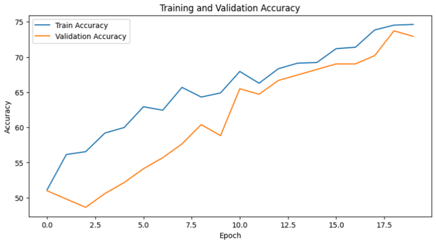
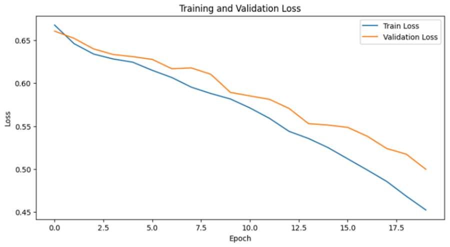
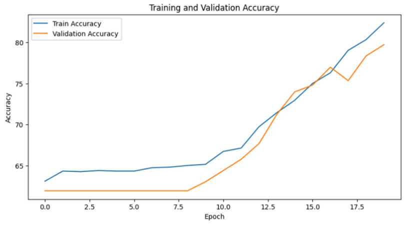
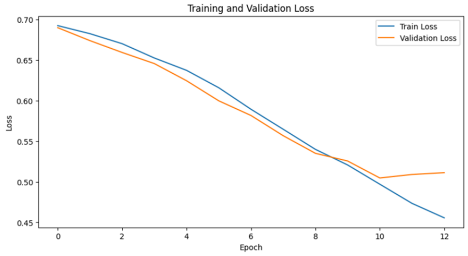
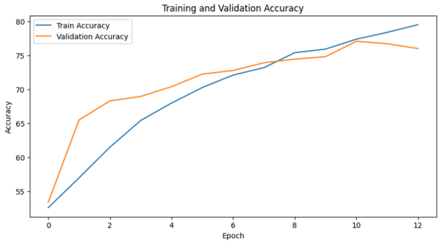
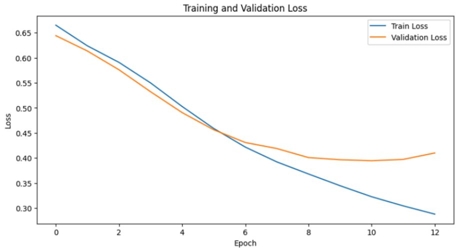
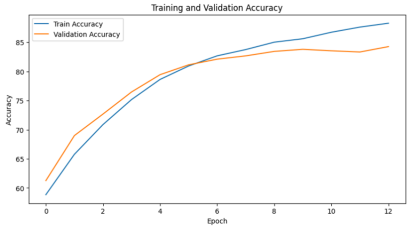
- Rajpurkar, Pranav, *et al*. "MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs," May 2018.

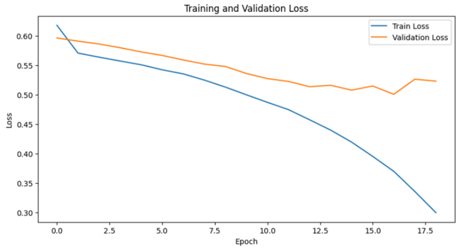
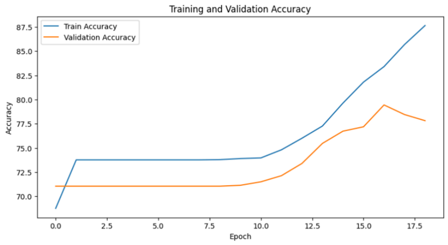
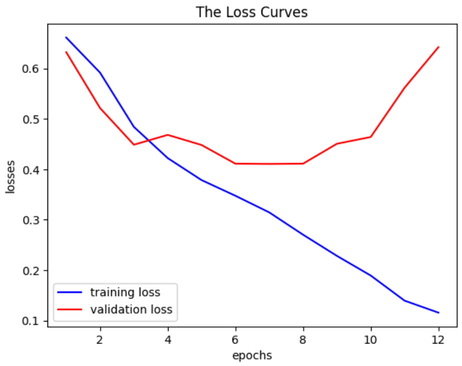
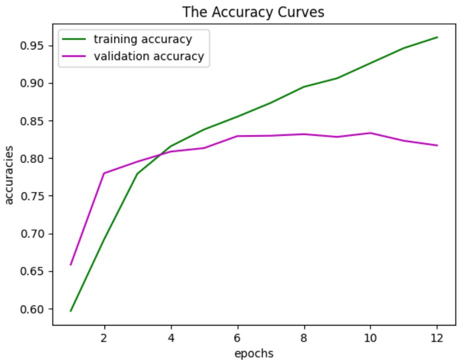
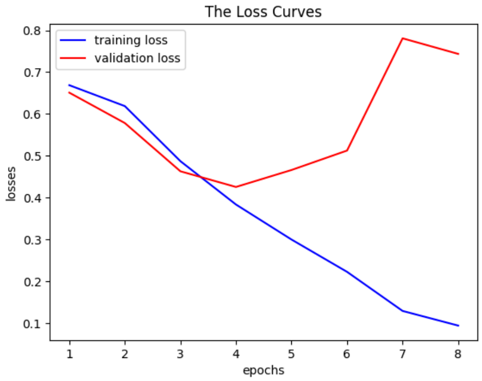
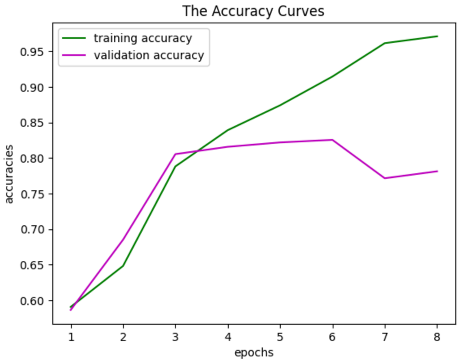
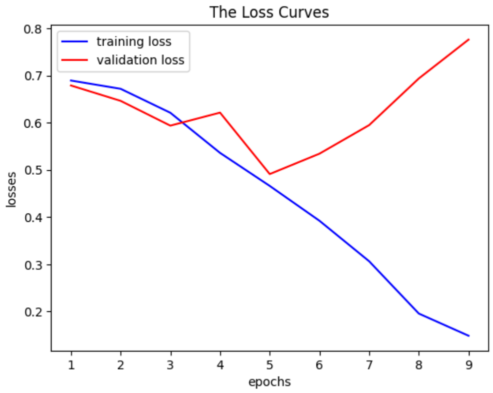
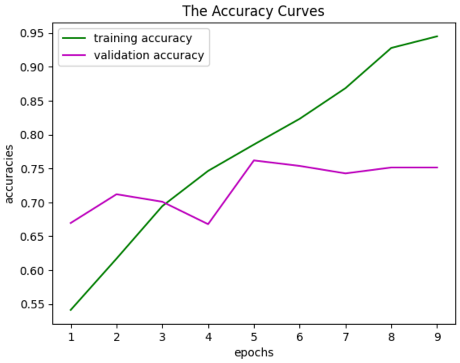
# Appendix A: Loss and Accuracy Curve Graphs

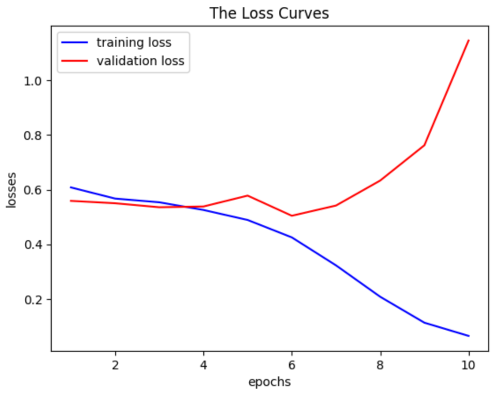
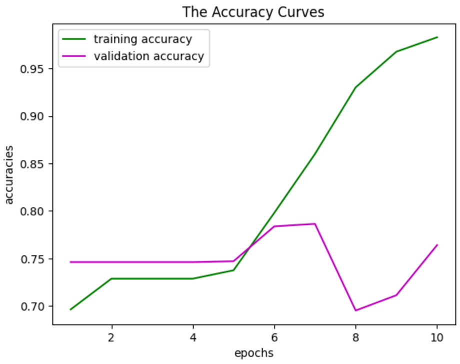
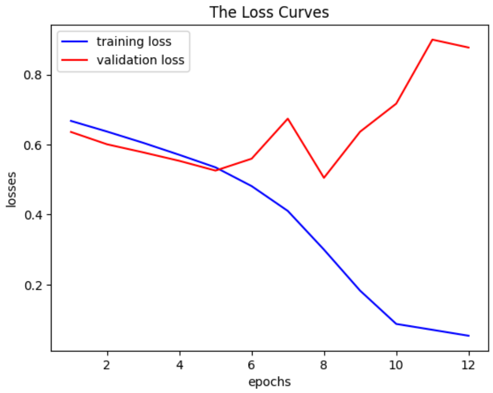
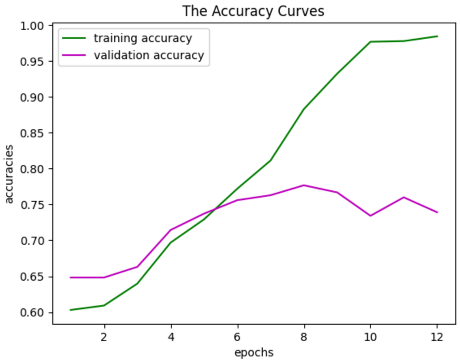
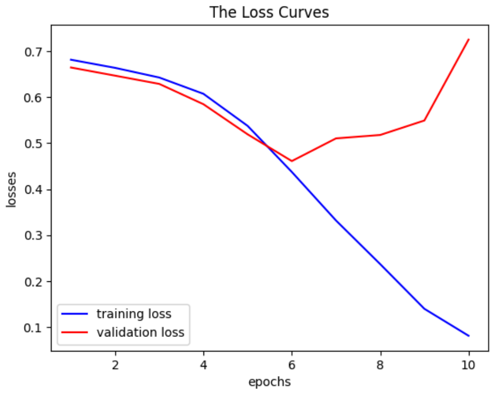
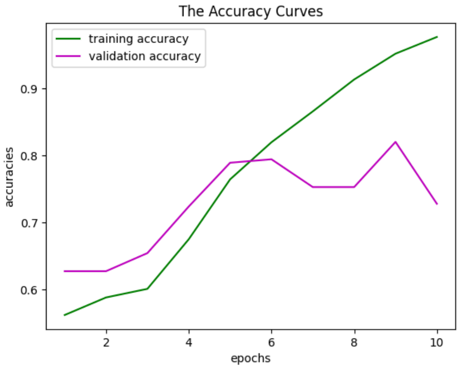
| **Model** | **Loss Curves** | **Accuracy Curves** |
|---|---|---|
| Unified CNN |  |  |
| Specialized CNN (Elbow) |  |  |
| Specialized CNN (Finger) |  |  |

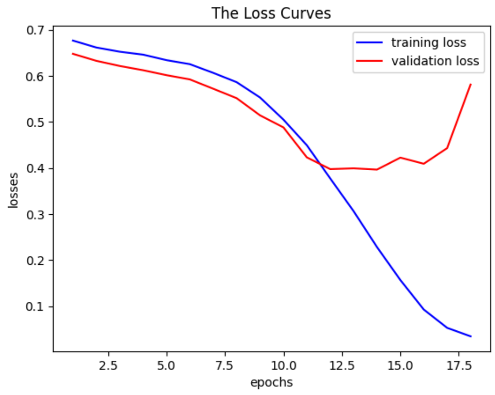
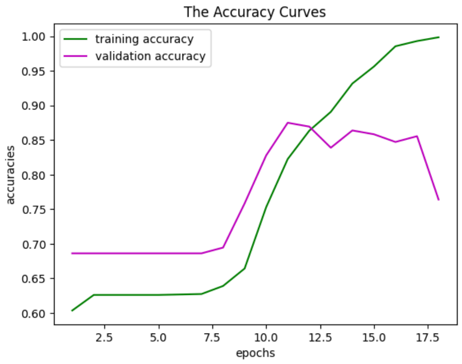
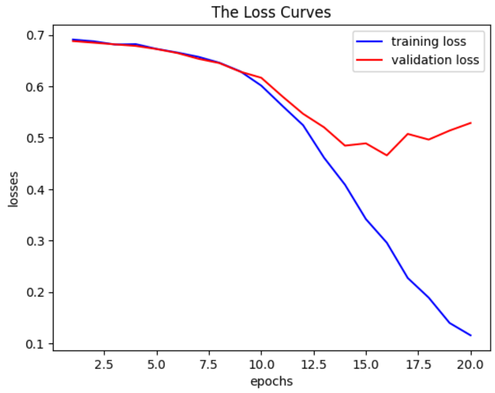
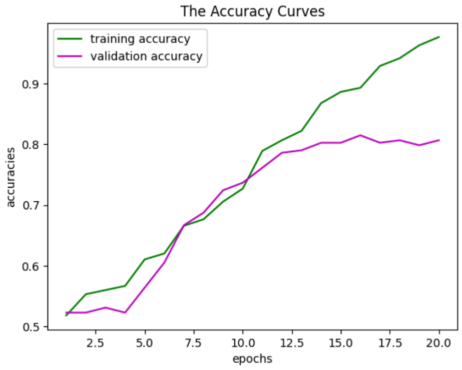
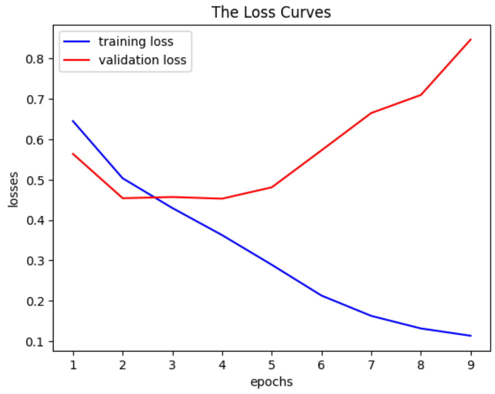
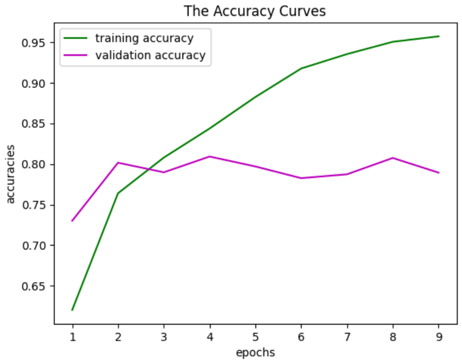
<table>
<tr>
<td>Specialized CNN (Forearm)</td>
<td> </td>
<td> </td>
</tr>
<tr>
<td>Specialized CNN (Hand)</td>
<td> </td>
<td> </td>
</tr>
<tr>
<td>Specialized CNN (Humerus)</td>
<td> </td>
<td> </td>
</tr>
</table>

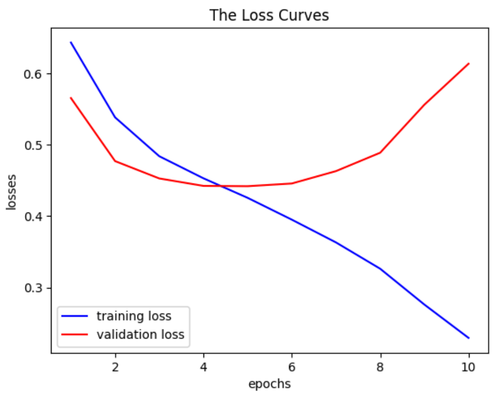
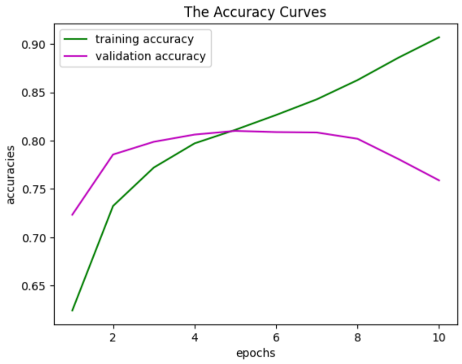
| | | |
|---|---|---|
| Specialized CNN (Shoulder) |  |  |
| Specialized CNN (Wrist) |  |  |
| Unified DenseNet |  |  |
| Specialised DenseNet (Finger) |  |  |

| | | |
|---|---|---|
| Specialised DenseNet (Elbow) |  |  |
| Specialised DenseNet (Humerus) |  |  |
| Specialised DenseNet (Forearm) |  |  |
| Specialised DenseNet (Shoulder) |  |  |
| Specialised DenseNet (Wrist) |  |  |

| | | |
|---|---|---|
| Specialised DenseNet (Hand) |  |  |
| Specialised ResNet-18 (Wrist): best model was at epoch 7, test accuracy=83.00% |  |  |
| Specialised ResNet-50 (Wrist): best model was at epoch 4, test accuracy=78.30% |  |  |
| Specialised ResNet-50 (Shoulder): best model was at epoch 5, test accuracy=74.42% |  |  |

| | | |
|---|---|---|
| Specialised ResNet-50 (Hand): best model was at epoch 6, test accuracy=67.17% |  |  |
| Specialised ResNet-50 (Finger): best model was at epoch 8, test accuracy=73.54% |  |  |
| Specialised ResNet-50 (Elbow): best model was at epoch 6, test accuracy=77.85% |  |  |

| | | |
|---|---|---|
| Specialised ResNet-50 (Forearm): best model was at epoch 14, test accuracy=78.41% |  |  |
| Specialised ResNet-50 (Humerus): best model was at epoch 16, test accuracy=82.64% |  |  |
| Unified ResNet-50, best model was at epoch 4, test accuracy=77.20% |  |  |

| | | |
|---|---|---|
| Unified ResNet-18, best model was at epoch 5, test accuracy=78.54% |  |  |