**Experiments**

Block-wise Tuning

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Fine Tuning |  |  |  |  |  |  |
| 1 | No fine-tuning (scratch model) | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | Fully-connected Layer (FCL) only | 0 | 0 | 0 | 0 | 0 | 0.1 |
| 3 | Block 4 | 0 | 0 | 0 | 0 | 0.1 | 0.1 |
| 4 | Block 3-4 | 0 | 0 | 0 | 0.1 | 0.1 | 0.1 |
| 5 | Block 2-4 | 0 | 0 | 0.1 | 0.1 | 0.1 | 0.1 |
| 6 | Block 1-4 | 0 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| 7 | Training all layers | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |

3 hyperparameter tuning quick => just learning rate (0.1, 0.01, 0.001, 0.0001) + learning decay + report epoch convg

1. VGG19 – hyper parameter tuning <= according to William not needed to replicate.
2. ResNet Scratch + Hyperparameter tuning = report best ResNet and best hyperparamter
3. (std + mean) Transfer Learning (Only)
4. (minmax) Transfer learning
5. Hyperparameter tuning
6. Use same hyperparameters for block-wise training report
7. On best model – do final hyperparameter report
8. Data augmentation (Transformation)
9. Balanced Data
10. 5-fold validation

The table above, shows the concept of “freezing layer”, which is simply setting the learning rate of that block to zero. When conducting block-by-block tuning, the block is allowed to learn by setting a learning rate of 0.1.

Hyperparameter Setting

The optimal hyperparameters were chosen from preliminary experiments. A grid-search approach was conducted on Model 3 from table /ref{1}, in which all the layers prior to block 4 is frozen, this means that block 4, average pooling layer and FC layer is allowed to learn, during the training phase. Model 3 was chosen for hyperparameter tuning, since this model only allows one block to learn, and this would allow us to analyse the behaviour of the general block-wise tuning, as well as, obtaining a reasonable hyperparameter for the block learning rate ().

A smaller batch size is used in our experiments, compared to Swati et al.’s experiment, this was due to our GPU capacity restrictions, therefore, a batch size of 16 was used rather than the 64 used in the paper. The results from this preliminary experiment shows that the optimal block learning rate is = [blah], and the optimal initial learning rate for the remaining layers (outside of all the blocks), = [blah].

Our preliminary experiment also showed that data augmentation of [blah] improves the accuracy by [blah], the transformations include [blah] because [evidence][R]

Additionally, the best optimiser for our experiments was [blah SGD with momentum?] **ADD EVIDENCE FROM PAPERS**, with a momentum value of [blah], and a learning rate scheduler of [blah], as shown in the table below:

Hyperparameter tuning on model 3 (table 1)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Initial Learning Rate () | Data Augmentation  (Y/N) | Block Learning Rate () | Learning Rate Decay | Epoch  Convg. | Train  Acc (%) | Val  Acc (%) |
| 3.1 |  |  |  |  |  |  |  |
| 3.2 |  |  |  |  |  |  |  |
| 3.3 |  |  |  |  |  |  |  |
| 3.4 |  |  |  |  |  |  |  |
| 3.5 |  |  |  |  |  |  |  |
| 3.6 |  |  |  |  |  |  |  |
| 3.7 |  |  |  |  |  |  |  |

**Results**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Fine Tuning | Recall | Specificity | Precision | F1-score | Train Acc (%) | Val Acc (%) |
| 1 | No fine-tuning (scratch model) |  |  |  |  |  |  |
| 2 | Fully-connected Layer (FCL) only |  |  |  |  |  |  |
| 3 | Block 4 |  |  |  |  |  |  |
| 4 | Block 3-4 |  |  |  |  |  |  |
| 5 | Block 2-4 |  |  |  |  |  |  |
| 6 | Block 1-4 |  |  |  |  |  |  |
| 7 | Training all layers |  |  |  |  |  |  |

What is the test accuracy on the best model?

**VGG19**

Scheduling rate (gamma) = 0.9

Nesterov’s momentum (mu) = 0.9

Base-learning rate of each layer = 2 \* alpha\_b

Initial learning rate alpha = 0.01

Alpha\_b (controls the block learning) = 0.1 (unfreeze), 0 (freeze- no learning)

Minibatch = 64 (max for GPU)

Epochs = 50 (stop automatically if 15 epochs does not run)

Comparison with State-of-the-art

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Model** | **Transfer Learning (TF) or Scratch (S)** | **Fine-tuned Blocks** | **Recall** | **Specificity** | **Precision** | **F1-score** | **Accuracy** |
| 1 | Baseline  VGG19 | TF | B1 – B6 |  |  |  |  |  |
| 2 | ResNet18 | S | - |  |  |  |  |  |
| 5 | ResNet18 | TF | B1 – B6 |  |  |  |  |  |
| 4 | ResNet18 + Balanced Data | TF | B1 – B6 |  |  |  |  |  |

Comparison with other papers on the same CE-MRI dataset classification

Proposed = best model so far

|  |  |
| --- | --- |
| **No** | **Accuracy** |
| Cheng et al. |  |
| Paul et al. |  |
| Abiwananda et al. |  |
| Swati et al. |  |
| Proposed |  |

Confusion matrix of VGG19, ResNet18 and ResNet18 + Balanced Data

5-fold comparison with State

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |