Antonopoulos Ilias p3352004
Ndoja Silva p3352017

X-Ray Abnormality Detection

03/04/2022

An approach based on deep neural networks for the Bone X-Ray Deep Learning Competition (Stanford ML Group) using the MURA dataset.

The **live** code for this project can be found at:
https://github.com/ilias-ant/x-ray-abnormality-detection (main branch).

The **freezed** code for this project (aka the assignment deliverable) can be found at:
https://github.com/ilias-ant/x-ray-abnormality-detection/tree/submission (submission branch).

The two branches are now in-sync, but we reserve the right to update the main branch in the future finding new approaches, better results etc. (see 5.Future Work).

Contents

1 The Problem

Musculoskeletal conditions are extensively present in the population, affecting over 1.7 billion people worldwide. Musculoskeletal disorders are the major cause of disability worldwide, affecting the locomotor system. Chronic pain, as well as impairments in mobility, dexterity, and functional capacity, plague patients. The use of a musculoskeletal (bone) X-ray is critical in the diagnosis of abnormalities. In recent years, deep learning algorithms have increasingly been applied in musculoskeletal radiology and have produced remarkable results.

Our objective is to create a classifier that can assert - given an X-ray - whether it depicts an abnormality or not.

2 The Dataset

MURA is a dataset of musculoskeletal radiographs consisting of 14,863 studies from 12,173 patients, with a total of 40,561 multi-view radiographic images. Each belongs to one of seven standard upper extremity radiographic study types: elbow, finger, forearm, hand, humerus, shoulder, and wrist. Each study was manually labeled as normal or abnormal by board-certified radiologists from the Stanford Hospital at the time of clinical radiographic interpretation in the diagnostic radiology environment between 2001 and 2012 [4].

This is the folder structure of the dataset:

```
└─train {data subset}
|   └──XR_ELBOW {study type}
|       |  └──patient00011 {patient}
|       |         └──study1_negative {study with label}
|       |               └──image1.png {view}
|       |               └──image2.png
|       |               └──image3.png
|       |               └──...
   ...


└─valid {data subset}
|   └──XR_HUMERUS {study type}
|       |  └──patient11216 {patient}
|       |         └──study1_negative {study with label}
|       |               └──image1.png {view}
|       |               └──image2.png
|       |               └──...
```

The task is the binary classification of the studies. The overall probability of abnormality for the study is computed by taking the arithmetic mean of the abnormality probabilities output by the network for each view (image). The model makes the binary prediction of abnormal if the probability of abnormality for the study is greater than 0.5.

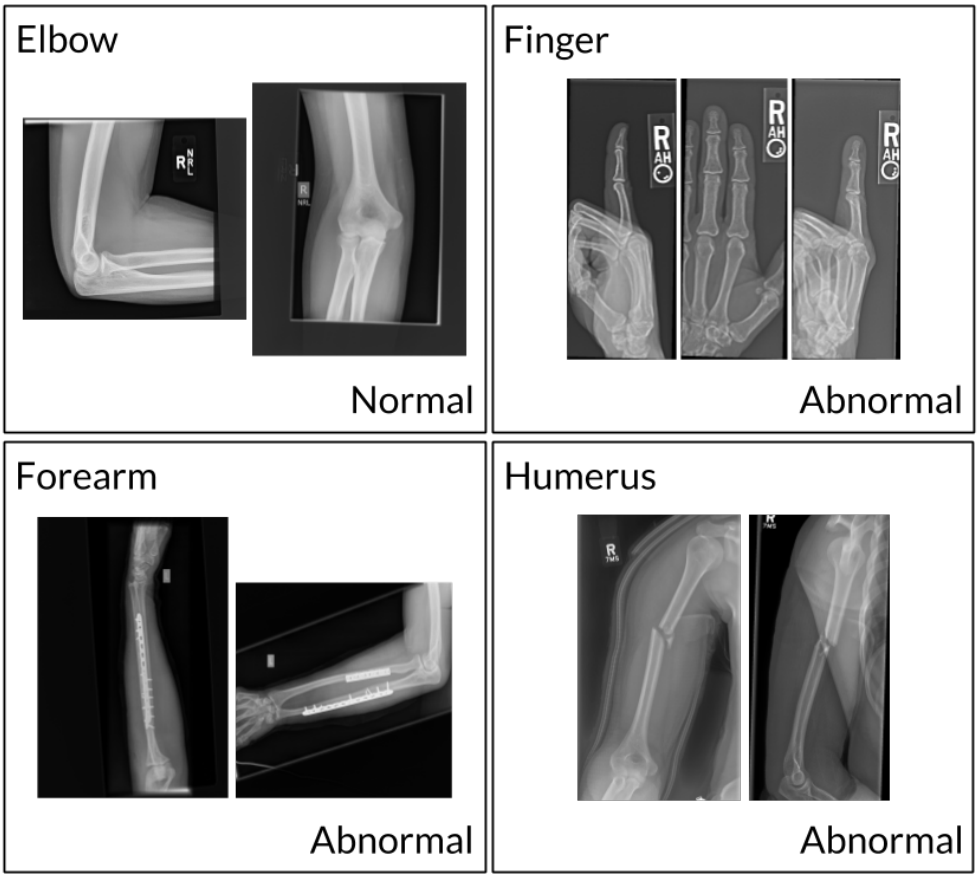


Figure 1: Sample of both normal & abnormal x-rays present in the dataset. These examples show a normal elbow study (left), an abnormal finger study with degenerative changes (middle left), an abnormal forearm study (middle right) demonstrating operative plate and screw fixation of radial and ulnar fractures, and an abnormal humerus study with a fracture (right). Taken from [4]

We can also observe that the dataset is imbalanced, having significantly less abnormal studies than normal. The overall imbalance ratio is: 0.625.

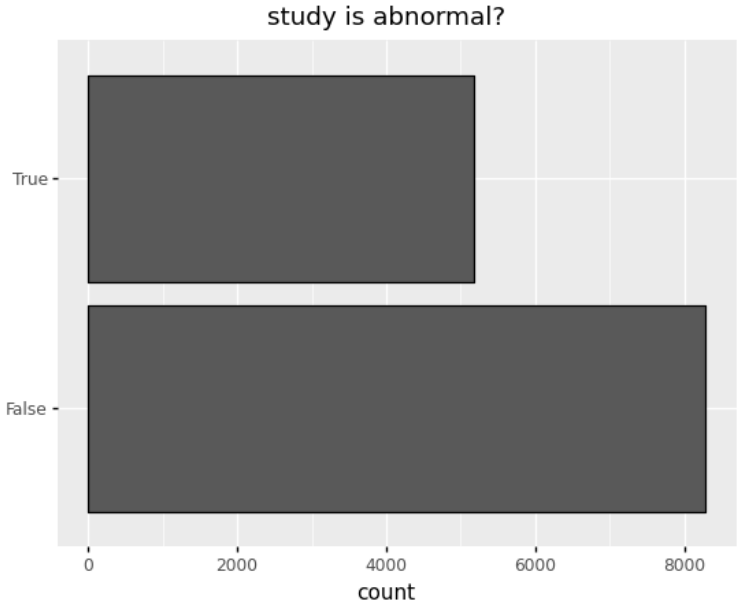


Figure 2: Number of normal and abnormal studies

Also, the majority of the studies typically contain two or three views (images). It is highly unlikely (i.e. an outlier) for a study to contain more than five views.

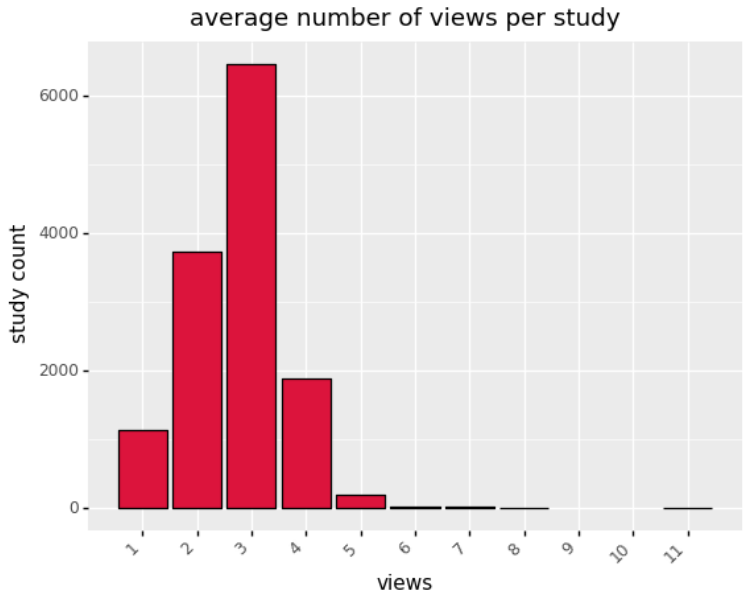


Figure 3: histogram of views per study

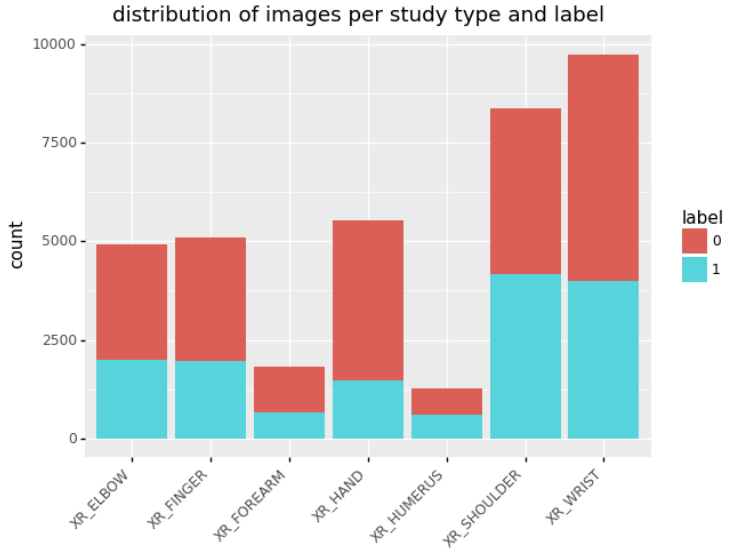


Table 1: imbalance ratios for the 7 study types

Study type	Imbalance ratio
Shoulder	0.99
Humerus	0.89
Wrist	0.692
Elbow	0.686
Finger	0.627
Forearm	0.568
Hand	0.366

The imbalance ratios above are computed as the ratio of the minority class (abnormal) size over the majority class (normal) size. The Hand study type is the most imbalanced one, while the Shoulder study type is almost perfectly balanced.

3 Approaches

We have experimented with 3 different approaches:

- training from scratch a classic Convolutional Neural Network.
- transfer-learning: via fine-tuning a DenseNet-169, with weights trained on ImageNet.
- transfer-learning: via fine-tuning a VGG-19, with weights trained on ImageNet.

A large part of the preprocessing phase is kept the same across all 3 models. The image size is fixed at $(224x224)$. We augmented the data during training by applying random lateral inversions and rotations of up to 30 degrees. Pixel points outside the boundaries of the view are filled with 0.

Importantly, we settle on a 80%-20% training-validation split for all our experiments. The training dataset consists of 29447 images, while the validation dataset of 7461 images.

A very important thing to keep in mind for the rest of this reading is that we try to classify images (aka views) as best as possible by assuming that good view classification performance will naturally translate to good study (contains one or more views) classification performance. So, every reported metric from now on will be accompanied by the labels:

- view-level: if the metric refers to the classification of views (images).
- study-level: if the metric refers to the classification of studies.

We are ultimately interested for the **study-level** classification performance, so you keep that in mind.

3.1 Convolutional Neural Network

The first approach to the problem training from scratch a Convolutional Neural Network architecture. In the data augmentation process we explicitly rescaled the pixels' values in the range of $[0, 1]$. The model takes as input 3-channeled images i.e. $(224x224x3)$

Our model consists of 4 convolutional blocks and each one consists of:

- the convolutional layer - each with a progressively larger no. of filters, an l_2 reguralizer and a ReLU activation function. Kernel size was selected as $(3x3)$ with a stride and dilation of 1.
- a batch normalization operation
- a max pooling operation
- a dropout scheme, with a rate of 0.25

After this convolutional base, the output is flattened and passed to a dense layer of 128 units. Lastly, the output layer consists of a single unit layer with a sigmoid nonlinearity. We have optimized the binary cross-entropy loss function, via the Adam optimizer with learning rate $1e-4$.

We have trained 8 such models in total: one for the entire training dataset and 7 individual models on the respective standard upper extremity radiographic study types.

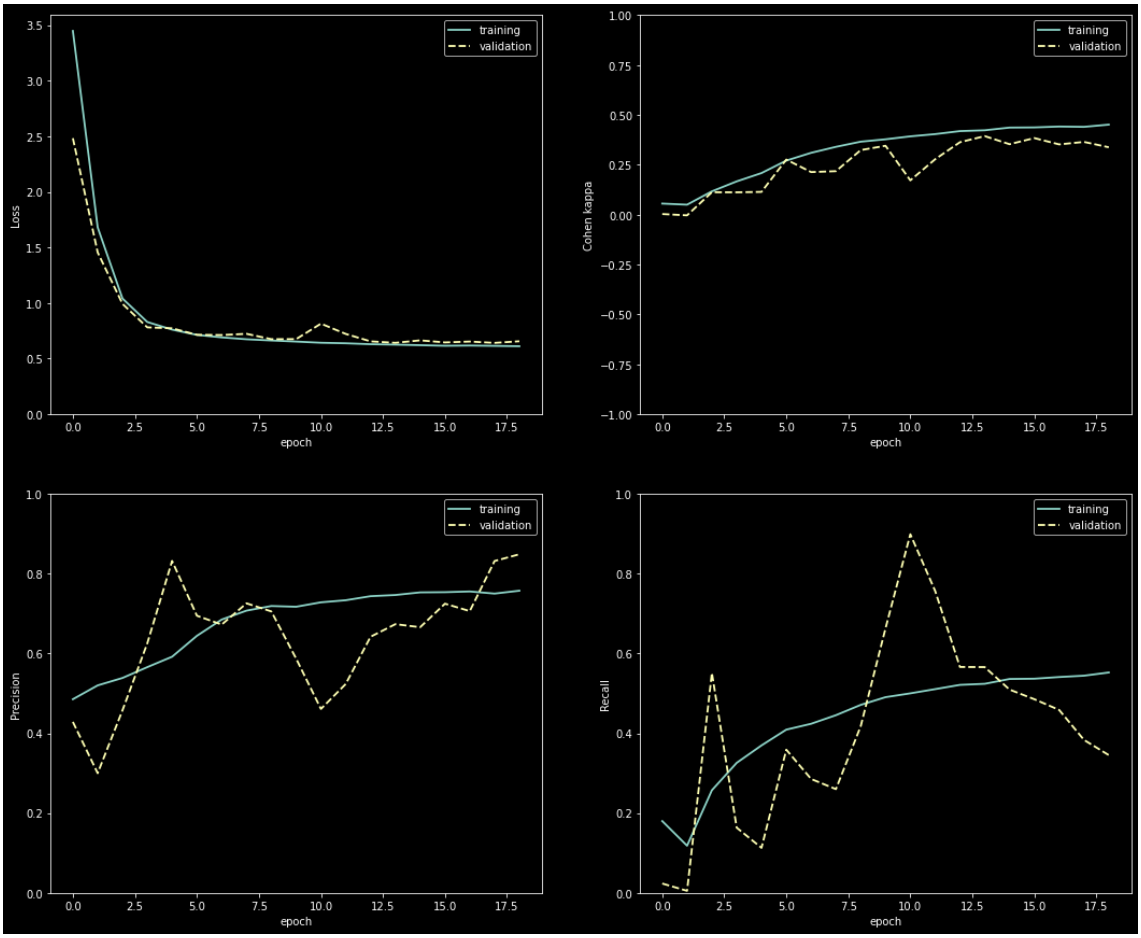


Figure 4: the history plots for the all-study-types CNN model training

3.2 DenseNet196, pretrained on ImageNet

Apart from the previous classical Convolutional Neural Network architecture - trained from scratch - we explored other more advanced techniques, like the Dense Convolutional Network (DenseNet), which connects each layer to every other layer in a feed-forward fashion [3]. We have opted for a feature extraction and fine-tuning scheme in which our original base model is a DenseNet-169 with weights pre-trained on ImageNet [2].

Apart from the aforementioned data augmentation, we have used the typical DenseNet image preprocessing.

For the feature extraction part, we have not included the fully-connected layer at the top of the pretrained network and we have applied global max pooling as a feature extractor. The idea of the max pooling stems from the fact that we want to 'emphasize' any sharp features on the image (e.g. hardware placed on human body that most definitely suggests abnormality). We have then freezed the entire convolutional base - to prevent the weights from being updated during training - and added a simple classification head that consists of a dropout layer and a dense unit, with sigmoid activation function. The final model consists of $12.644.545$ params, with only 1665 of them being actually trainable.

We have then trained this model (i.e. the classification head of it) over 20 epochs, using binary crossentropy as a loss function and Adam as the loss optimizer with a learning rate of $1e-4$. Early stopping has been employed over the validation cohen's kappa (κ) to provide further regularization. You can consult the appendix 7.2 for the model architecture.

The training concluded at epoch 11, with a restoration of the model weights from the best epoch, 6. This model yields a validation cohen's kappa (κ) of: 0.42 (view-level).

To increase performance even further, we have then unfreezed the entire convolutional base and retrained the entire model end-to-end over 10 epochs, with a very low learning rate ($1e-5$) to avoid catastrophic forgeting. To avoid potential overfitting, we have used early stopping and learning rate reduce-on-plateau schemes.

The fine-tuning process improved the classification now yielding a validation cohen's kappa (κ) of: 0.64 (view-level).

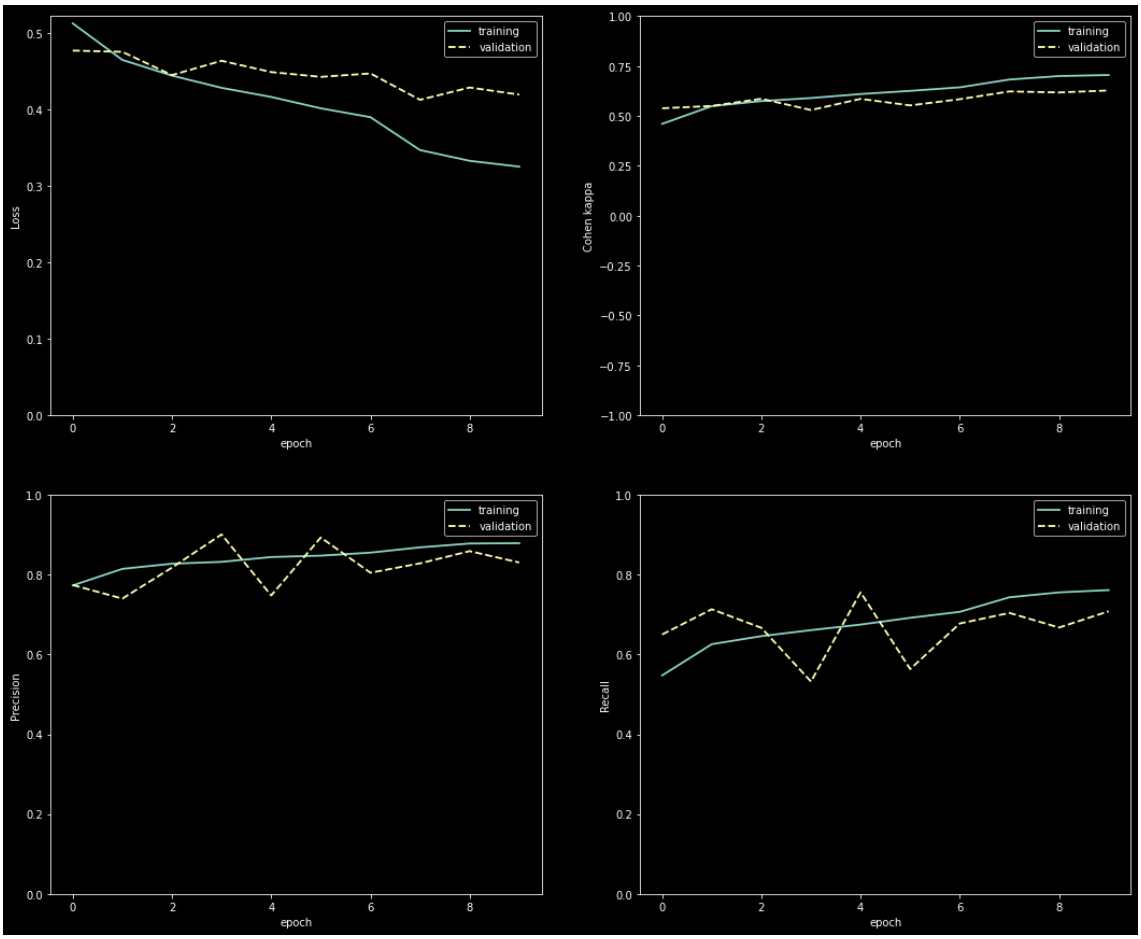


Figure 5: the history plots for the fine-tuning phase of DenseNet-169

3.3 VGG19, pretrained on ImageNet

We have also experimented with Very Deep Convolutional Networks with 19 weight layers (VGG19)
We again opted for a feature extraction and fine-tuning scheme in which our original base model is now a VGG19 with weights pre-trained on ImageNet [2].

Apart from the aforementioned data augmentation, we have used the typical VGG-19 image preprocessing.

For the feature extraction part, we have not included the fully-connected layer at the top of the pretrained network and we have applied global max pooling as a feature extractor. We have then freezed the entire convolutional base - to prevent the weights from being updated during training - and added a simple classification head that consists of a dropout layer and a dense unit, with sigmoid activation function. The final model consists of $20.024.897$ params, with only 513 of them being actually trainable.

We have then trained this model (i.e. the classification head of it) over 20 epochs, using binary crossentropy as a loss function and Adam as the loss optimizer with a learning rate of $1e-4$. Early stopping has been employed over the validation cohen's kappa (κ) to provide further regularization. You can consult the appendix 7.3 for the model architecture.

The training concluded at epoch 16, with a restoration of the model weights from the best epoch, 11. This model yields a validation cohen's kappa (κ) of: 0.36 (view-level).

To increase performance even further, we have then unfreezed the entire convolutional base and retrained the entire model end-to-end over 10 epochs, with a very low learning rate ($1e-5$) to avoid catastrophic forgeting. To avoid potential overfitting, we have used early stopping and learning rate reduce-on-plateau schemes.

The fine-tuning process improved the classification now yielding a validation cohen's kappa (κ) of: 0.6 (view-level).

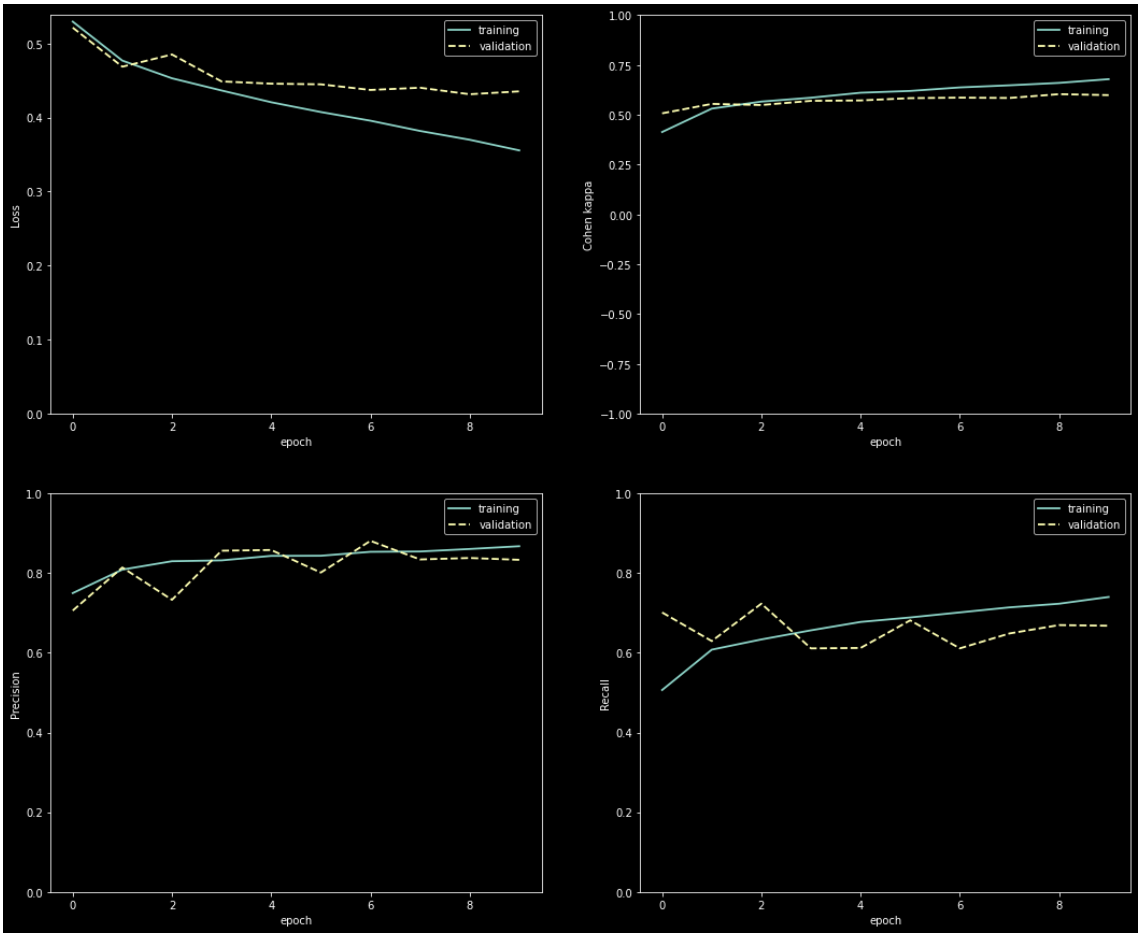


Figure 6: the history plots for the fine-tuning phase of VGG-19

3.4 Ensemble model: DenseNet-169 + VGG-19

Ensemble learning has been proven to produce improved and more robust performance than a single model [1]. Here, we combine the previous two pretrained architectures to create an ensemble classifier.

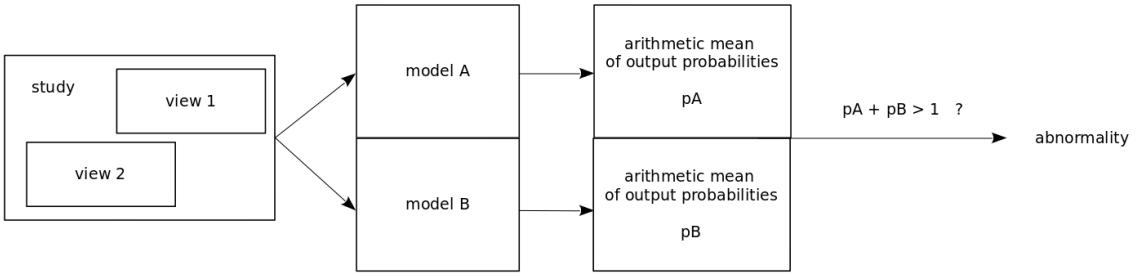


Figure 7: we classify as abnormal when the sum of the respective mean output probabilities exceeds 1.

3.5 Ensemble model: CNN with a designated CNN classifier for wrists

This is an experiment to create a model where the inference for a particular upper extremity region (study type) is being delegated to a model that has been trained exclusively for this region (in our case here Wrist, but several other combination can be tried - we tried Wrist because we have observed promising results on the validation dataset).

For this ensemble model to practically be meaningful, we make the assumption that we have (or we can acquire) the knowledge of the upper extremity region in which a given unseen image belongs to. If, during inference, we only have unseen images without any additional metadata then this approach does not stand.

4 Evaluation

4.1 Metrics

The metrics we have chosen are the following :

- Cohen's kappa (κ) - lies in the range $[-1, 1]$. A score of -1 represents complete disagreement between two raters whereas a score of 1 represents complete agreement between the two raters. A score of 0 means agreement by chance.
- ROC-AUC - lies in the range $[0, 1]$. A score of 0 represents that model's predictions are all worng, while a score of 1.0 represents model's predictions are correct.
- F_1 - lies in the range $[0, 1]$ which is the harmonic mean of precision and recall, both calculated as percentages.

4.2 Results

We have held the testing dataset **private** until the end, in order to assess and compare the resulting models. The training dataset consists of 3197 images.

Table 2: final evaluation on **test** dataset - format loosely adopted from [1]

metrics	models	Overall	Shoulder	Elbow	Humerus	Hand	Wrist	Forearm	Finger
Cohen's kappa (κ)	CNN	0.386	0.17	0.50	0.44	0.18	0.51	0.44	0.37
	CNN + wrist-CNN	0.400	0.17	0.50	0.44	0.18	0.58	0.44	0.37
	DenseNet-169	**0.629**	0.54	**0.72**	0.72	0.43	**0.71**	**0.65**	**0.59**
	VGG-19	0.598	0.51	0.67	0.7	0.42	0.67	0.62	0.56
	DenseNet + VGG	**0.629**	**0.57**	0.7	**0.73**	**0.45**	0.7	0.63	0.58
ROC-AUC	CNN	0.69	0.59	0.75	0.72	0.58	0.74	0.72	0.68
	CNN + wrist-CNN	0.7	0.59	0.75	0.72	0.58	0.79	0.72	0.68
	DenseNet-169	0.79	0.75	0.82	0.85	0.69	0.82	0.8	0.77
	VGG-19	0.79	0.76	0.82	0.85	0.69	0.82	0.8	0.77
	DenseNet + VGG	**0.81**	0.78	0.84	0.87	0.7	0.84	0.81	0.79
F_1	CNN	0.64	0.67	0.7	0.72	0.3	0.68	0.65	0.6
	CNN + wrist-CNN	0.65	0.67	0.7	0.72	0.3	0.74	0.65	0.6
	DenseNet-169	0.75	0.73	0.79	0.84	0.57	0.79	0.77	0.73
	VGG-19	0.75	0.73	0.79	0.84	0.57	0.79	0.77	0.73
	DenseNet + VGG	**0.77**	0.77	0.81	0.86	0.6	0.8	0.78	0.74

5 Future Work

Restrictions on both time and computing capabilities forced us to leave some potential areas for further improvement unexplored. Namely:

- utilize the Class Activation Map (CAM) technique, indicating the discriminative region(s) used by the model to make predictions. This can help us improve the efficacy of our classifiers.
- more complex classification heads for the feature extraction scheme employed to both DenseNet-169, VGG-19 pretrained models - we have added a very simple and minimal classification head, but we could have opted for a typical MLP or CNN.
- resample the minority class (abnormal studies) naturally through image augmentation, in order to create a more balanced training dataset. We suspect that this idea can be crucial to specific sub-datasets (e.g. Hand) where the classes are heavily imbalanced.
- more rich data augmentation techniques, to enhance the training dataset.
- experiment with pretained residual neural networks (ResNet) as well, and use them as part of an ensembling scheme.

6 References

[1] He Minliang et al. *A calibrated deep learning ensemble for abnormality detection in musculoskeletal radiographs*. 2021. DOI: 10.1038/s41598-021-88578-w. URL: https://doi.org/10.1038/s41598-021-88578-w.

[2] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

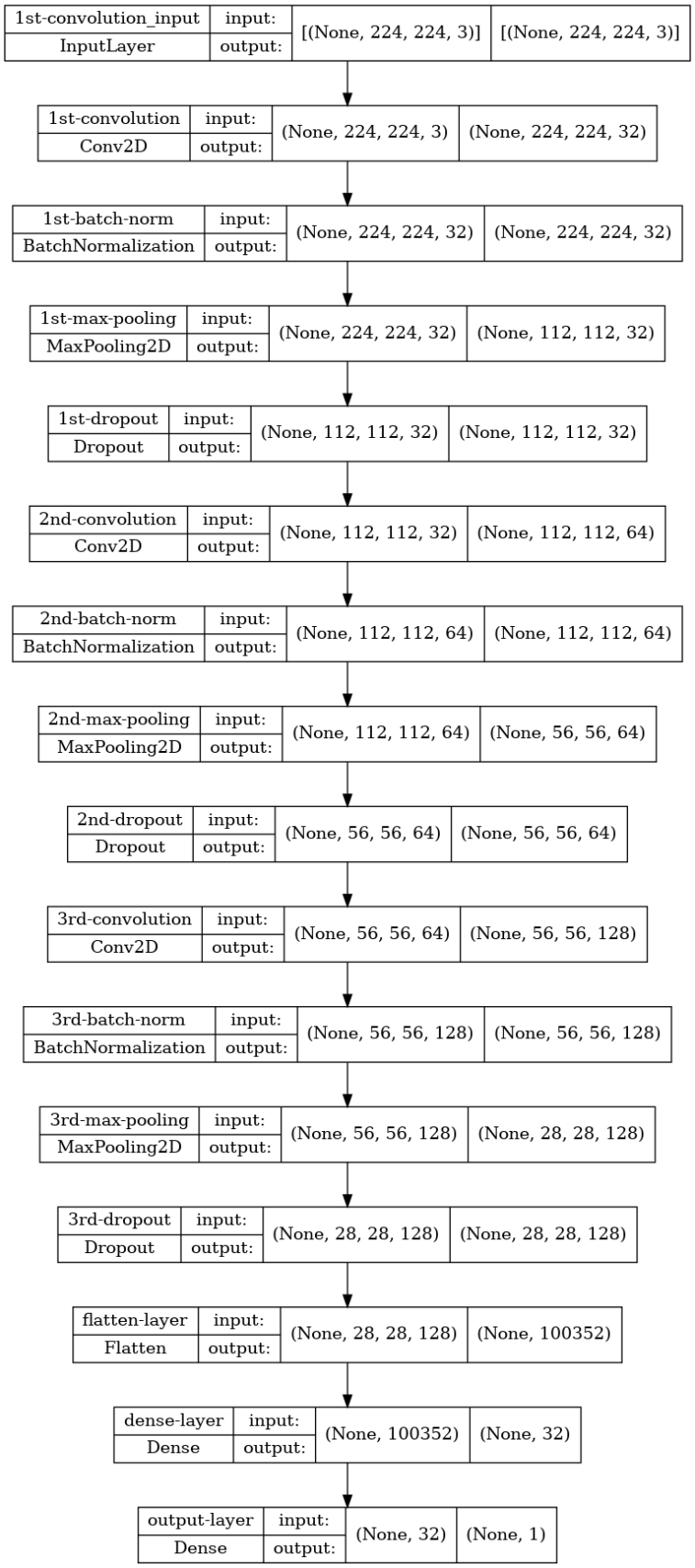
[3] Gao Huang et al. *Densely Connected Convolutional Networks*. 2016. DOI: 10.48550/ARXIV.1608.06993. URL: https://arxiv.org/abs/1608.06993.

[4] Pranav Rajpurkar et al. *MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs*. 2017. DOI: 10.48550/ARXIV.1712.06957. URL: https://arxiv.org/abs/1712.06957.

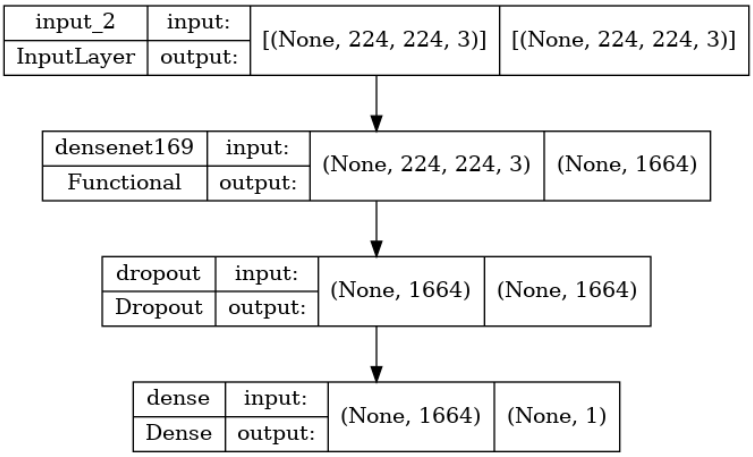
[5] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. DOI: 10.48550/ARXIV.1409.1556. URL: https://arxiv.org/abs/1409.1556.

7 Appendix

7.1 CNN architecture



7.2 DenseNet-169 architecture



7.3 VGG-19 architecture