Automated Scoring of Rheumatoid Arthritis using Deep Neural Networks

Authors : Khanh-Tung Nguyen-Ba[#], Jay Ji-Hyung Ryu[1], Rui Bai[2], Yilin Wu[1], Yingnan Wu[#], Xiaofu He[#]

Abstract

We investigated a solution to the problem of detecting joints and measuring narrowness and erosion from Rheumatoid Arthritis (RA) from medical images. Our solution used FasterRCNN networks for joint localization and classification and EfficientNet for scoring. To our knowledge, this is the first time a full solution for detection, classification and scoring of RA joints using Neural Network has been proposed. We achieved IOU mAP scores of 99% on both narrowing and erosion joint localization, but there remains to be challenges regarding scoring the damage on the joints. Our annotated dataset of 367 patients will be a good resource for future RA research.

1. Introduction

The inflammation of the synovium by Rheumatoid Arthritis (RA) creates pain and stiffness of the joints for patients[1]. Radiographers usually assess the severity of the disease and monitor the patient response to treatment by manually inspecting erosion and joint space narrowing on hand and feet radiographs. This manual process is subjective and costs a lot of time. Our goal is to deliver an automated solution that is fast and objective[2].

The grading of radiographs in RA is often done using the Sharp/van der Heijde (SvH) scoring method[3]. This includes: 16 areas for joint erosion and 15 areas for joint space narrowing for each hand, and 6 areas for erosion and 6 areas for joint space narrowing for each foot. The scores for joint narrowing and joint erosion are then summed up to give an overall SvH. The score for joint narrowing is on the scale of 0 to 4, 0 is normal and 4 is the most severe. The score for erosion is on the scale of 0 to 5 for hands and 0 to 10 for feet. Visualization of the details can be found in Appendix 8.1.

The RA2 Dream Challenge [2] provides the first time a significant dataset of radiographs which are available for proposals of automated solutions. As a participant in this challenge, we proposed a neural network solution for joint localization, classification and scoring. We also created a cleanly annotated data set for future RA research.

2. Previous Work

There are very few papers based on the literature on the application of neural networks on RA joint localization and scoring. Perhaps, this is due to the lack of publicly available RA medical images and annotations, and also the lack of shared and reproducible work from the medical imaging community.

There are some preliminary works that investigated the application of deep learning on RA. In [4], the authors trained a CNN to output binary classification on a whole image on a small dataset of 135 radiographs. In [5], regions of interest are proposed using a handcrafted multiscale gradient vector flow method before feeding into a CNN, which was done on a very small dataset of 30 patients.

Released recently in December 2019, [6] is the only paper we can find that proposed to localize the joints first before scoring. The joints were first localized using a cascade classifier with Haar-like features, and then feeded to a CNN for scoring. Their solution performed well on PIP, IP and MCP joints, but failed to capture many wrist and intercarpal joints, which are often more indicative of the RA damage.

We believe that the application of deep learning on RA research is still growing. More structured data needs to be shared and more collaboration needs to be formed to move field forward.

3. Challenges from the dataset

The first and foremost challenge is the size of the dataset. There are 367 patients, each with 4 images of Left Hand, Right Hand, Left Foot and Right Foot, respectively. This is a small dataset in the context of deep learning. Also, the image quality is often inconsistent, e.g., some have either low resolution, or missing EXIF data, or are under/overexposed. We need to preprocess the images to address this issue.

Secondly, the RA2 Dream Challenge only provides the score of each joint but not the location. Challengers need to either annotate the bounding boxes manually, or come up with an unsupervised solution. We chose to manually label part of the images, then trained a preliminary FasterRCNN model to create synthetic labels for the rest. Finally, we manually assessed and corrected all synthetic labels to get a clean set of annotations. In total, we produced a set of annotations for 15414 joints for narrowing (367 patients x (12 foot joints + 32 hand joints)) and an additional 4404 joints for erosion (367 patients x 12 additional joints).

Lastly, the distribution of scores is heavily right-skewed. The majority of the joint scores are 0 (0 = normal, 4 = most severe).

4. System Architecture

Our system architecture is two-fold: a neural network for joint detection and classification, and a neural network for joint scoring, respectively. Ideally, a single-shot architecture that combines localization, classification and scoring all at once would be more efficient. Due to the nature of this challenge that we did not have bounding box annotations in the first place, we started out solving

the localization problem first, hence the two-fold architecture. See Appendix 8.2

Following the two-fold system architecture, we chose FasterRCNN [7] for joint localization and classification, and EfficientNet[8] for scoring, respectively.

4.1 Localization

For joint localization, FasterRCNN was chosen for its known superior performance in proposing regions of interest using a neural network compared to RCNN and FastCNN that uses a selective search algorithm. Compared to YOLO, another high performing network, it fares better in its ability to capture smaller and overlapping objects, which is especially important to detect wrist joints. Please refer to the original FasterRCNN paper [7] for more details.

We used a FasterRCNN model pre-trained on the COCO dataset, and replaced the head layer to output 42 classes for narrowing joints and 44 for erosion joints, respectively.

4.2 Scoring for Joint Narrowing and Joint Erosion

We used a classification model to predict joint narrowing and erosion damage scores. EfficientNet[8] was chosen as our model considering its state-of-art record in image classification problems and rather parsimonious number of parameters compared to competing models as described in the paper. Specifically, we used EfficientNet B4 pretrained on ImageNet data among all model family members and all layers were retrained with our dataset.

5. Dataset and Training Protocol

5.1 Dataset

Our dataset included 4 images of Left Hand, Right Hand, Left Feet and Right Feet for each of 367 patients. There were two sets of labels, one for narrowing joints of hands and feets, and one for erosion joints of hands (because the narrowing joints and erosion joints for feet are the same).

For narrowing, we cropped out 15414 narrowing hand and feet joints for training. For erosion, we cropped out 12478 erosion hand joints and merged them with 4404 narrowing feet joints for training.

For both localization and scoring, we splitted the dataset into 90% for training and 10% for validation. The cropped images of the joints were converted to grayscale, normalized, padded, and resized to 224x224. We applied a series of data augmentation techniques, including random crop, horizontal flip, rotation, and distortion. Only train images were augmented while normalization, padding and resizing were applied to the validation set, as well.

5.2 Training and Validation Protocol

We trained two FasterRCNN detection networks separately, one for localizing narrowing joints and one for erosion joints of the hand.

After training, the validation detection results were first filtered so that for each joint we only selected the maximum objectness score. Because we knew the exact joints to label for each image, we could find the false negatives right away.

We applied IOU threshold = 0.5 to find the true/false positives, and calculated precision, recall and AP on the validation set.

After the two FasterRCNN were trained, we applied them to the images to get bounding boxes for the joints, and cropped out narrowing joints and erosion joints separately. The narrowing feet joints were merged with the erosion joints to get a full set of erosion joints.

Because of highly imbalanced score distribution with most of the images scored as 0, we oversampled minor classes using ImbalancedDatasetSampler[9] implemented using Pytorch as we fed data into the network. We expected that data augmentation could mitigate part of the overfitting problem introduced by oversampling. For joint narrowing scores, Stochastic Gradient Descent (SGD) was used with an initial learning rate = 0.001 and momentum = 0.9. L2 regularization was set at 0.001.

For erosion scores, the score was even more skewed since the scales for hands and feet were not the same. Only a few of the foot images were scored as 10. Oversampling and data augmentation were also introduced to improve the imbalanced dataset. Similar to the joint narrowing model, EfficientNet B4 is used for training. SGD was used with an initial learning rate = 0.001 and momentum = 0.9. L2 regularization was set at 0.005 since overfitting for erosion scores was more evident.

6. Results

Both narrowing and erosion FasterRCNN converged quickly after a few epochs and achieved 99% IOU mAP on the validation set.

The results were encouraging as we found no false negatives in the validation set for both narrowing and erosion joints. For narrowing there were 8 joints that were false positives. For erosion there was only 1 joint.

Our joint narrowing scoring model achieved 57.62% accuracy on the validation set. When tested on a validation set that used the PyTorch WeightedRandomSampler to balance the distribution of the classes, the accuracy was 60.09%. The erosion scoring model achieved 17% of validation accuracy and 18.6% of balanced accuracy. By checking the original image, erosion was less obvious than narrowing. If severe narrowing happened together with erosion, erosion would

be invisible. Validation accuracy was low because we mainly focused on improving accuracy on resampled balanced validation sets as we were targeting to increase the average of the weighted absolute error.

7. Discussion and Conclusion

7.1. Discussion

We learned that the quality of the RA images are not consistent and the size of the cropped joints can be completely different. One direction for future improvement is to upscale the low-resolution images and provide more details using super resolution techniques.

Although FasterRCNN performed well (capturing 99% of the joints), it is the undetected joints that are usually the ones most severely damaged. Our solution architecture, therefore, performed less effectively for the most severe cases. An interesting direction would be to build a single-shot detector that could look at the overall morphology of the hand, and, be based on training images and learn cross-correlations involving other-than-joint-level features inferred in the hidden layers, impute a joint score for each without actually examining each joint specifically.

Our scoring model didn't overcome extreme imbalance in the dataset. Imbalance in class distribution is a typical problem of RA data as most of the cases are classified as normal. Dealing with imbalanced data composes the fundamental challenge of this problem and requires further research.

Moreover, for erosion score, the accuracy could be improved if separate models were built on hands and feet due to the different scoring scale. Our scoring model was trained on combined datasets for hands and feet due to the small size of the dataset. Hence, foot images scoring 5 were mixed with hand images scoring 5 but the severity for erosion was different. This could be improved by separating the data or relabeling foot images.

7.2. Conclusion

We proposed a full neural-network-based solution for the automated scoring of RA. We also produced a fully annotated set of RA images for 367 patients, the largest to date so far, which would be helpful for future RA research.

8. Acknowledgement

We would like to thank the organizers of the RA2 Dream Challenge to pose this challenging but meaningful problem for the Machine Learning community to solve.

9. Authors Statement

X.H. conceived and supervised the study, K.N., J.R., Yilin W., Yingnan W. contributed to manually annotating, K.N., Yilin W. established joint detection and labeling model, R.B., Yingnan W. contributed to image preprocessing, K.N., J.R., Yilin W. built joint score prediction model, J.R., R.B., X.H. refined the prediction models, K.N., J.R., R.B. wrote the manuscript, J.R., R.B., X.H. proofread the manuscript.

10. References

[1] Smolen, J., Aletaha, D., Barton, A. et al. Rheumatoid arthritis. Nat Rev Dis Primers 4, 18001 (2018). https://doi.org/10.1038/nrdp.2018.1

[2] RA2 Dream Challenge https://www.synapse.org/#!Synapse:syn20545111

[3] van der Heijde D.M, van Leeuwen M.A, van Riel P.L, van de Putte L.B. Radiographic progression on radiographs of hands and feet during the first 3 years of rheumatoid arthritis measured according to Sharp's method (van der Heijde modification). J Rheumatol. 22(9), 1792-1796 (1995)

[4] Üreten, K., Erbay, H. & Maraş, H.H. Detection of rheumatoid arthritis from hand radiographs using a convolutional neural network. Clin Rheumatol 39, 969–974(2020). https://doi.org/10.1007/s10067-019-04487-4

[5] Murakami, S., Hatano, K., Tan, J. et al. Automatic identification of bone erosions in rheumatoid arthritis from hand radiographs based on deep convolutional neural network. Multimed Tools Appl 77, 10921–10937 (2018). https://doi.org/10.1007/s11042-017-5449-4

[6] Hirano, T., Nishide M., Nonaka, N., Seita, J., Ebina, K., Sakurada. K, Kumanogoh, A., Development and validation of a deep-learning model for scoring of radiographic finger joint destruction in rheumatoid arthritis, Rheumatology Advances in Practice, 3, rkz047 (2019) https://doi.org/10.1093/rap/rkz047

[7] Ren, S. He, K., Girshick, R. and Sun, J.. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 10.1109/tpami.2016.2577031 (2016) http://dx.doi.org/10.1109/TPAMI.2016.2577031

[8] Tan, M., Le, Q.V., EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, arXiv:1905.11946 (2019) https://arxiv.org/abs/1905.11946

Source code:

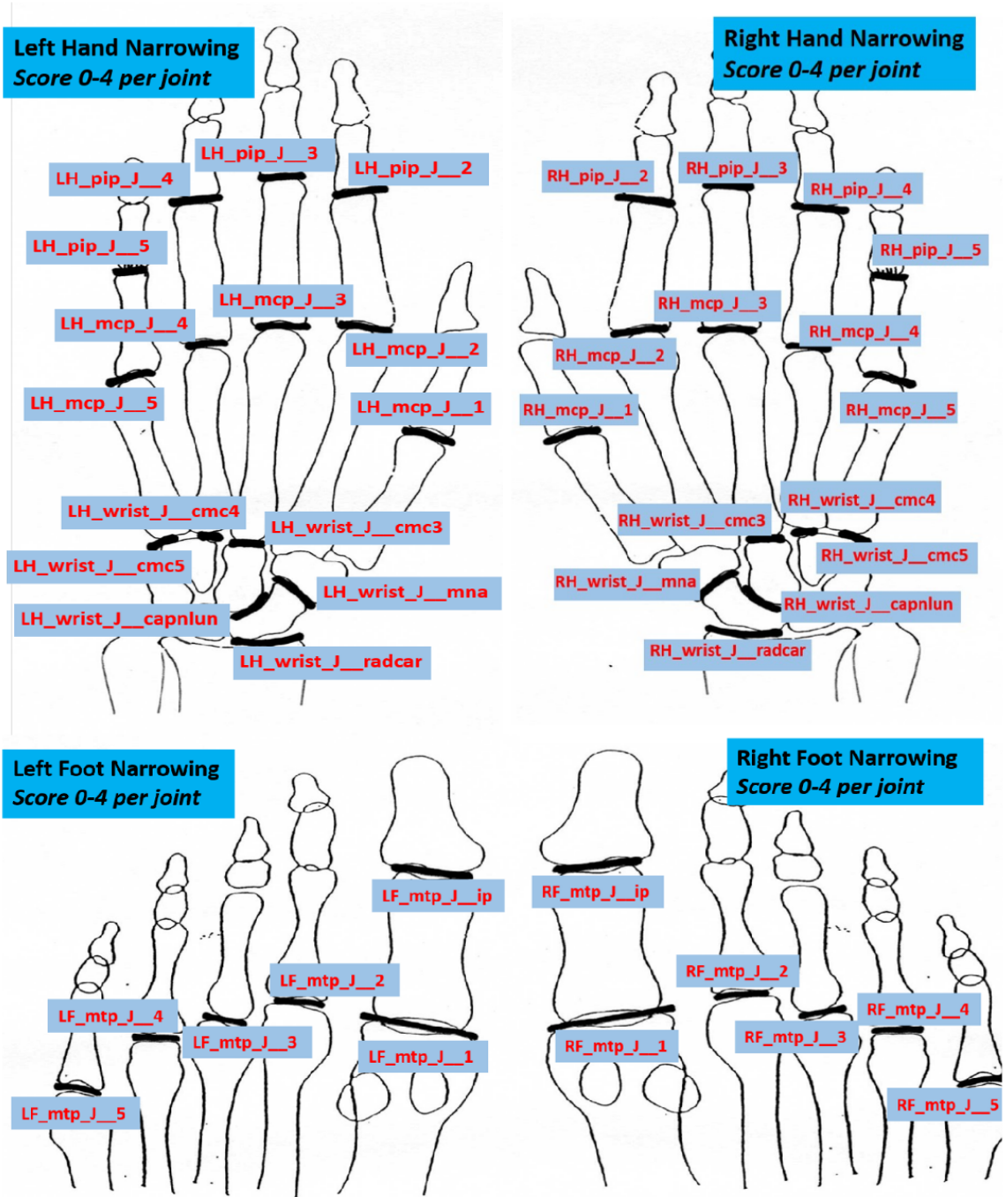
[9] EfficientNet https://github.com/lukemelas/EfficientNet-PyTorch

[10] ImbalancedDatasetSampler https://github.com/ufoym/imbalanced-dataset-sampler

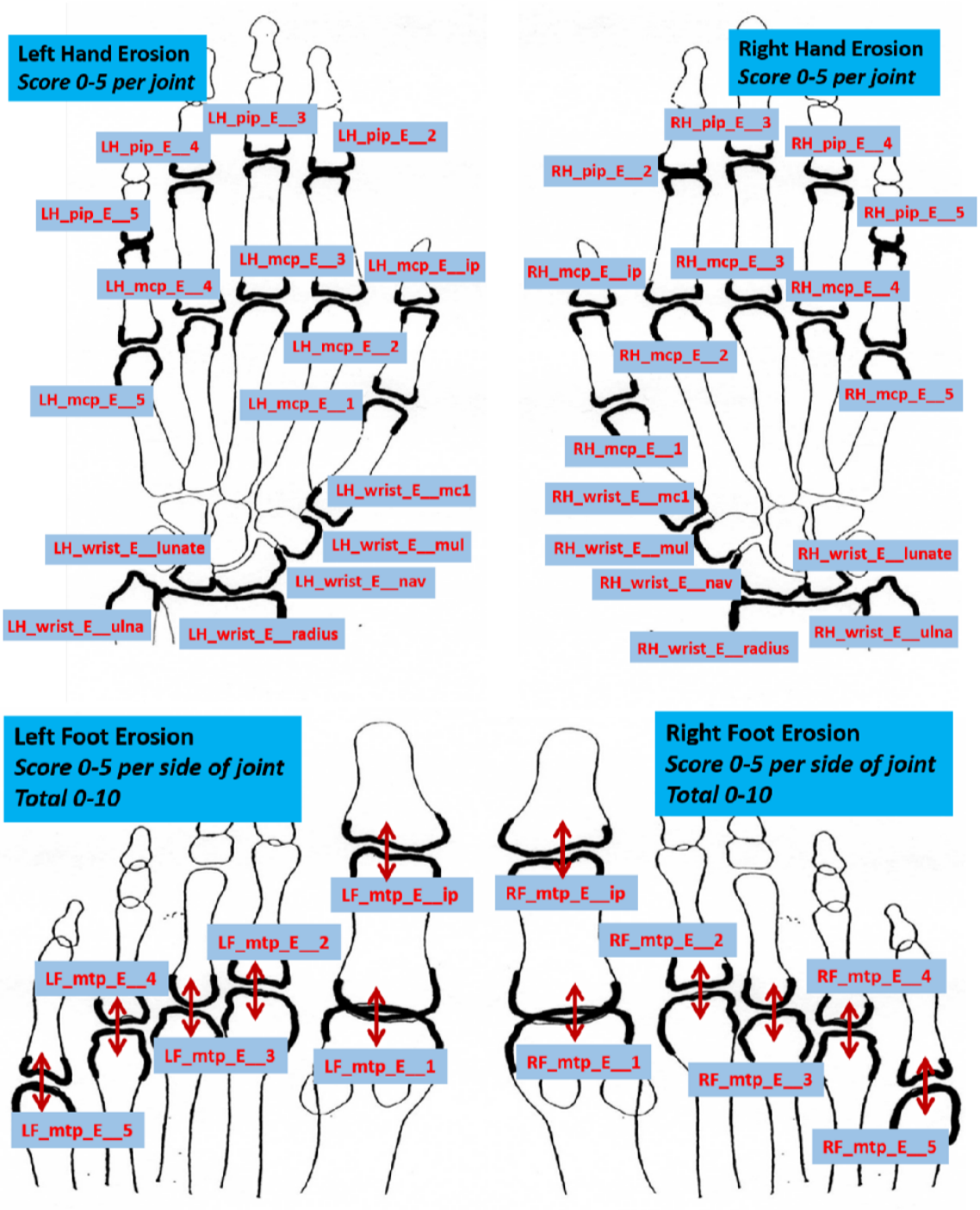
8. Appendix

8.1 Sharp/van der Heijde scoring method

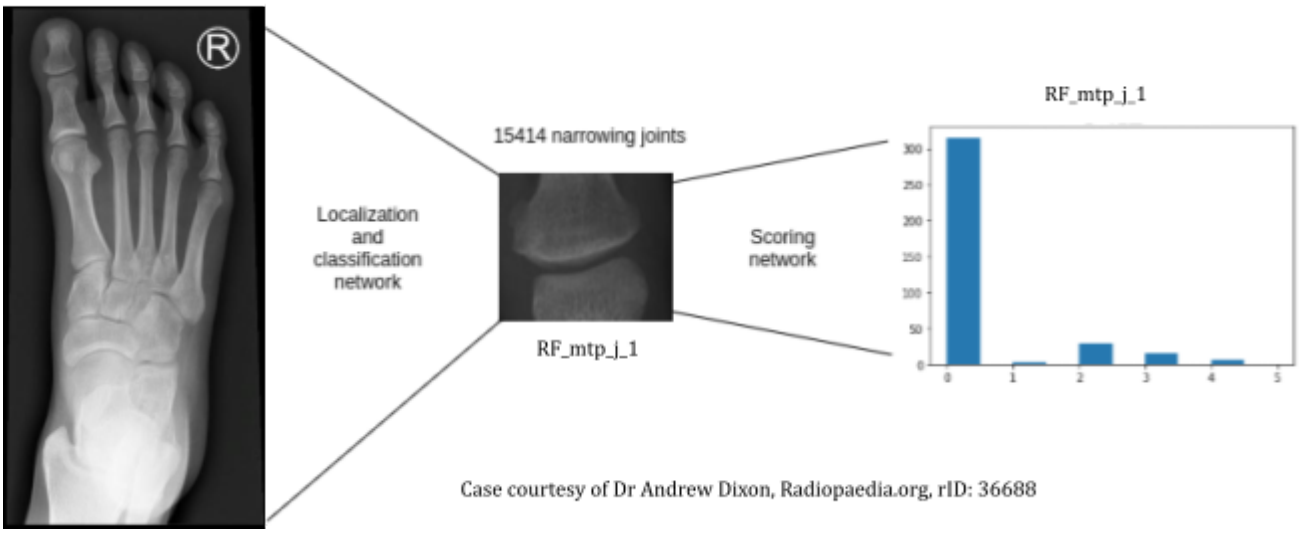
The Sharp/van der Heijde scoring method for joint space narrowing.
Source: https://www.synapse.org/#!Synapse:syn20545111/wiki/597243



The Sharp/van der Heijde scoring method for joint space erosion.
Source: https://www.synapse.org/#!Synapse:syn20545111/wiki/597243

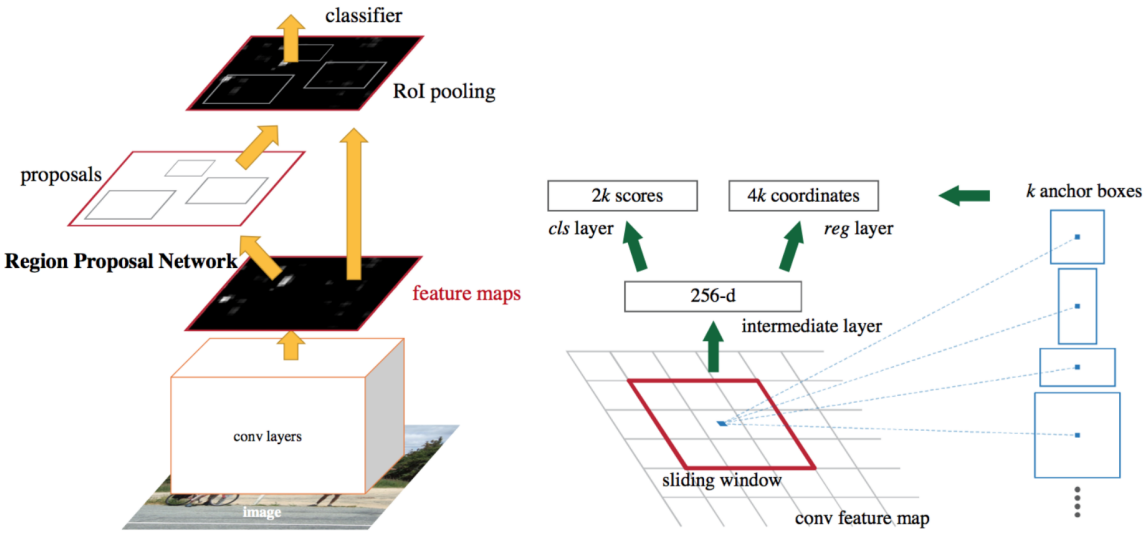


8.2 System Architecture



8.3 Network Architecture

8.3.1 Localization Network - Faster RCNN

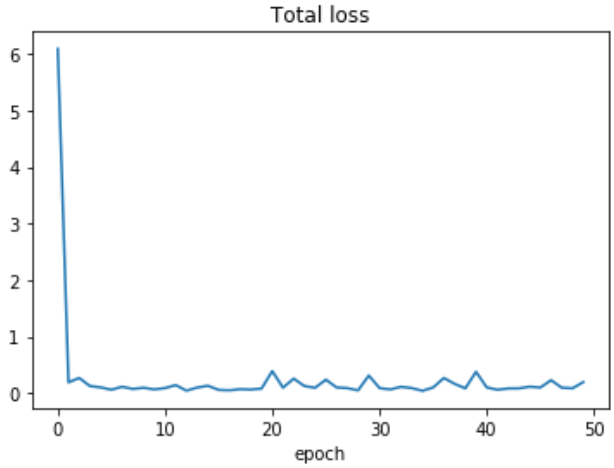


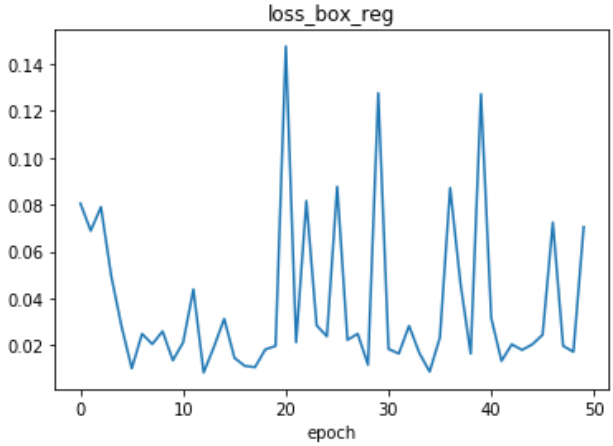
Source: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

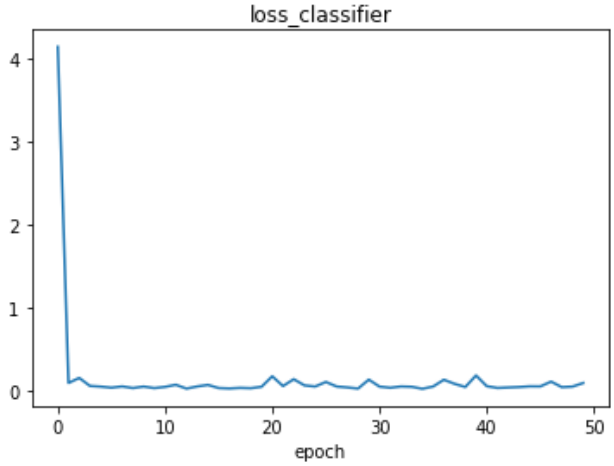
8.3.2 Scoring Network - CNN

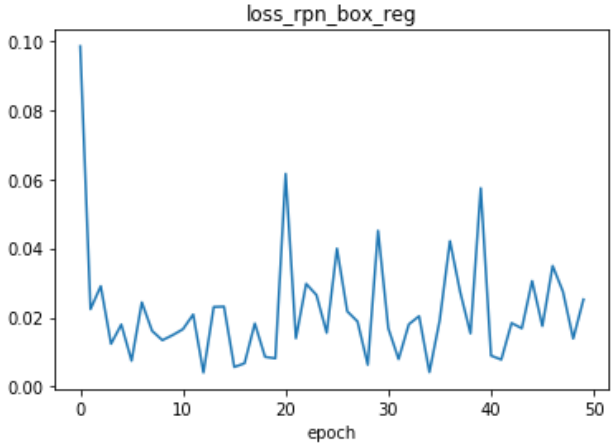
```
self.features = nn.Sequential(
        nn.Conv2d(3, 16, 3, padding = 1),
        nn.BatchNorm2d(16),
        nn.ReLU(True),
        nn.MaxPool2d(2, 2),
        nn.Conv2d(16, 64, 3, padding = 1),
        nn.BatchNorm2d(64),
        nn.ReLU(True),
        nn.MaxPool2d(2, 2),
        nn.Conv2d(64, 256, 3),
        nn.BatchNorm2d(256),
        nn.ReLU(True),
        nn.MaxPool2d(2, 2),
        nn.Conv2d(256, 1024, 3, stride = 3),
        nn.BatchNorm2d(1024),
        nn.ReLU(True),
    )
    self.classifier = nn.Sequential(
        nn.Linear(1024 * 5 * 5, 256),
        nn.ReLU(True),
        nn.BatchNorm1d(256),
        nn.Linear(256, 64),
        nn.ReLU(True),
        nn.BatchNorm1d(64),
        nn.Linear(64, num_classes)
    )
```

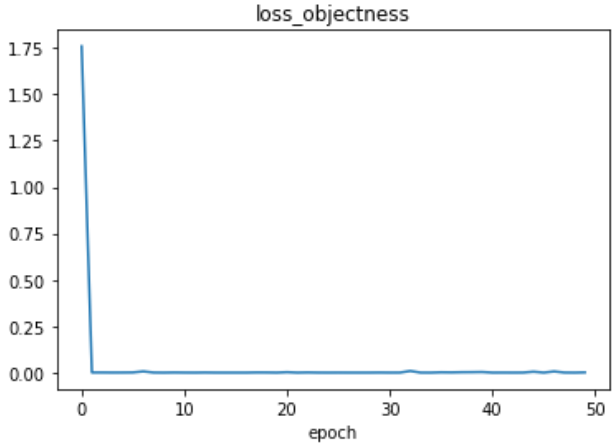
8.4.1 Training loss - Faster RCNN for Narrowing Joint



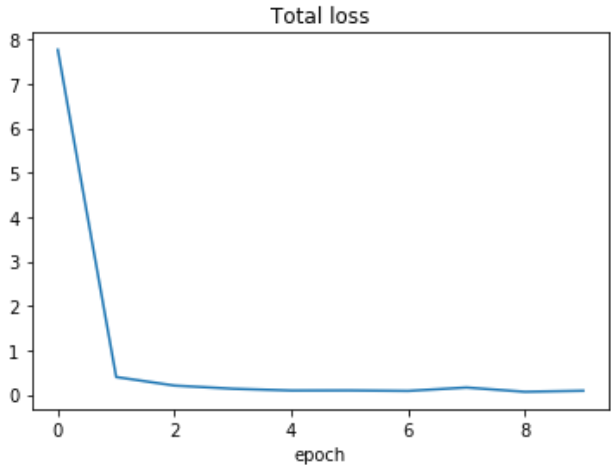


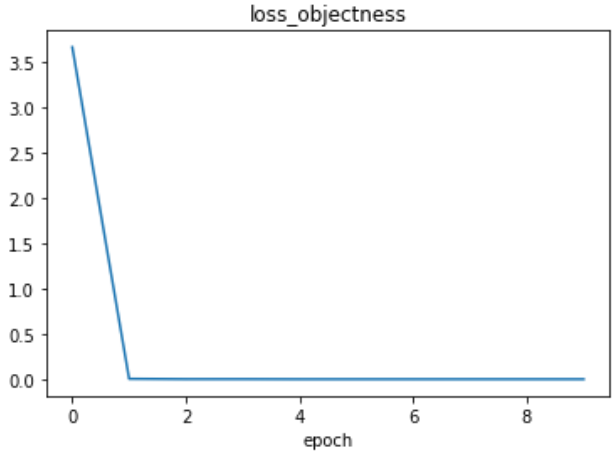


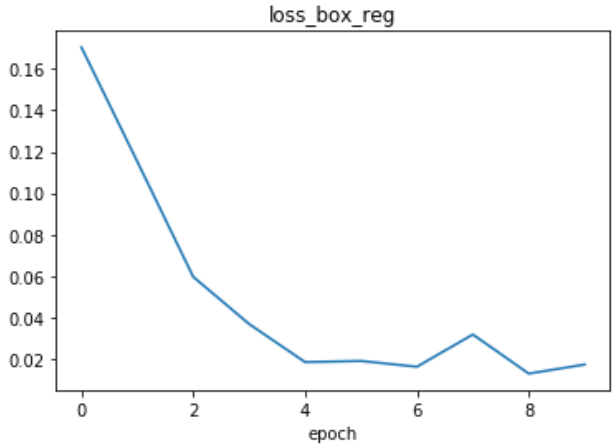


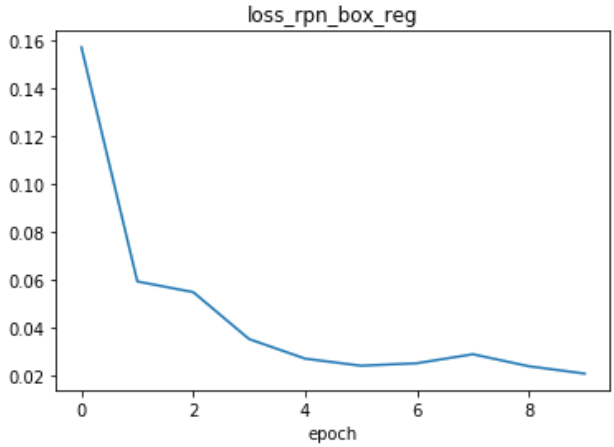


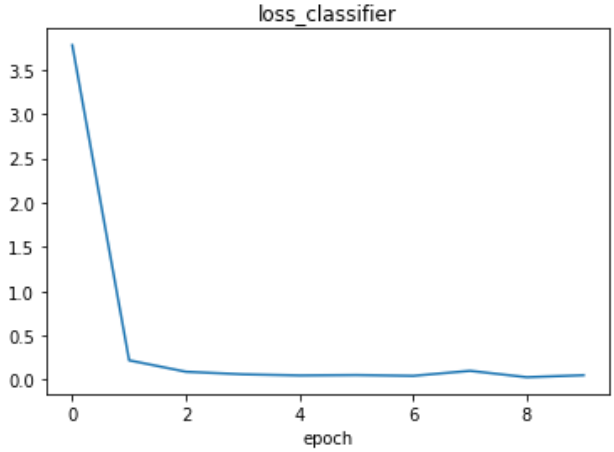
8.4.2 Training loss - Faster RCNN for Erosion Joint



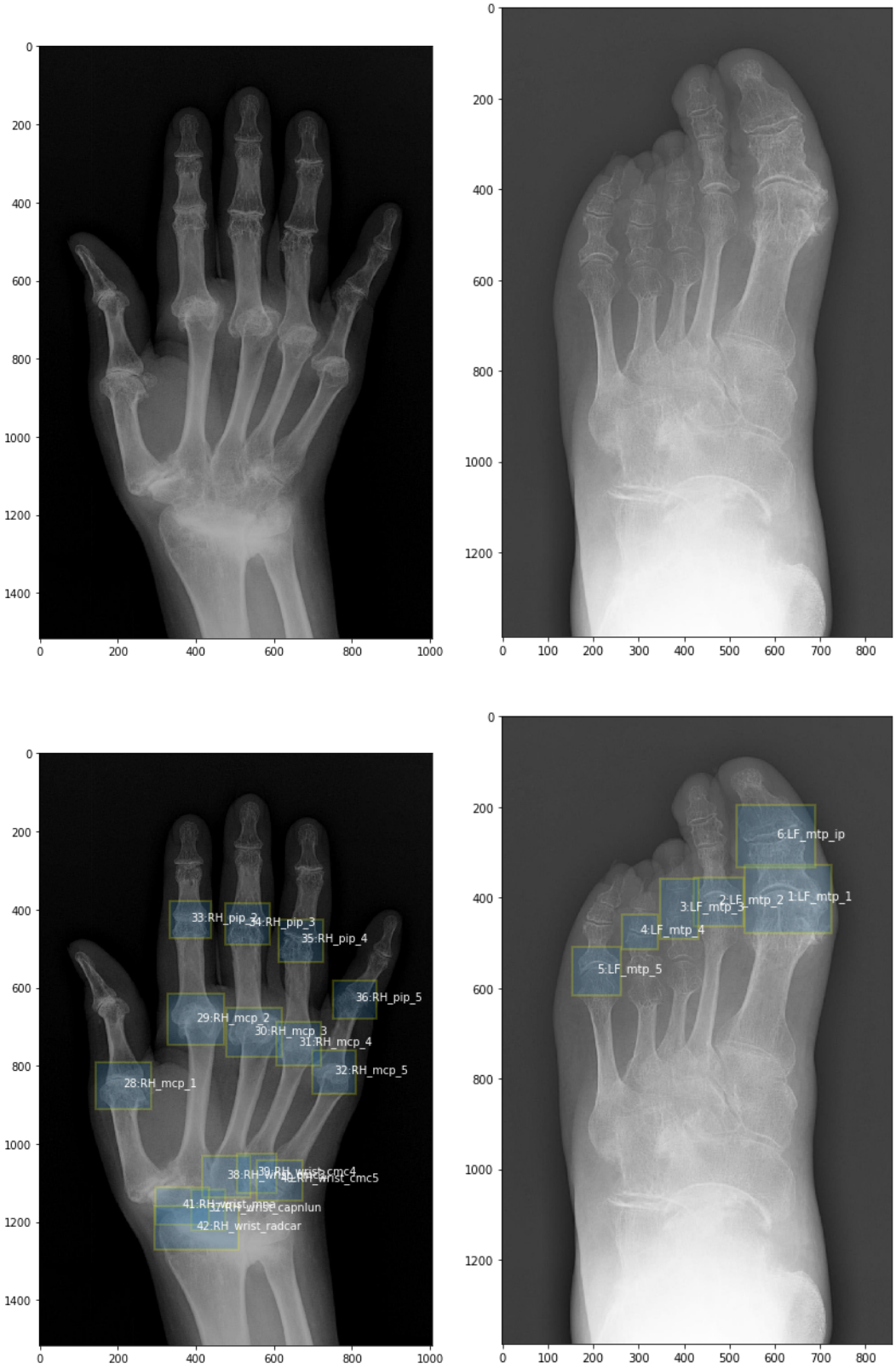




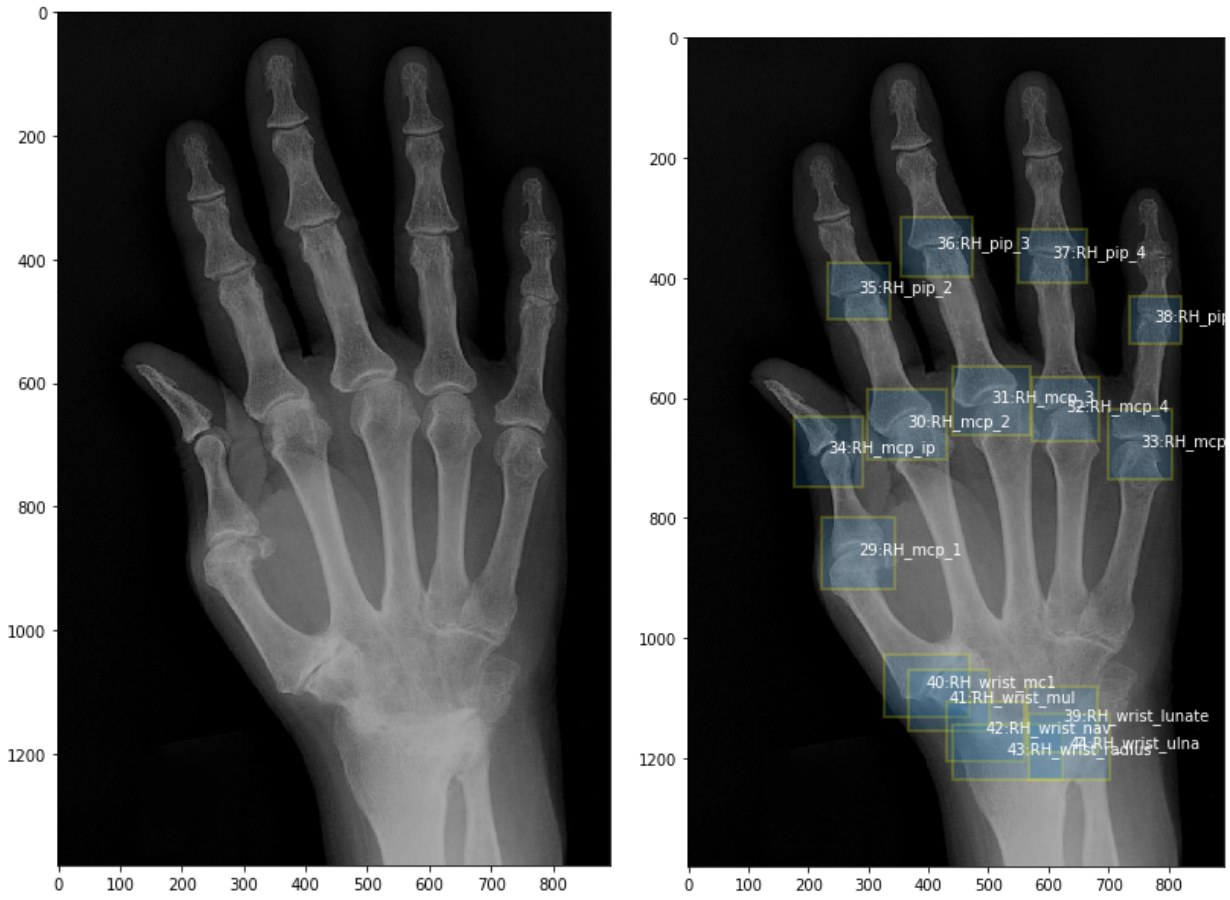




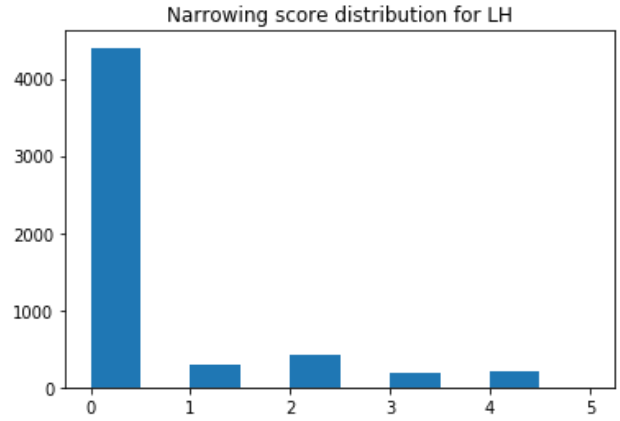
8.5.1 False Positive - Narrowing

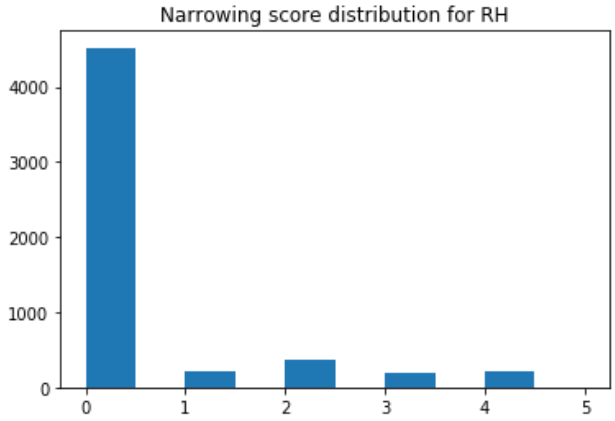


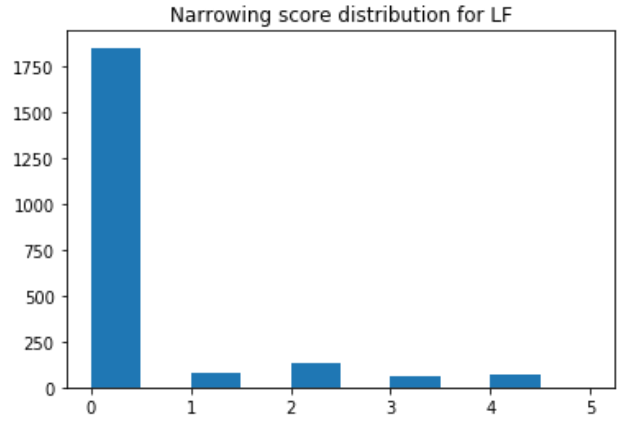
8.5.2 False Positive - Erosion

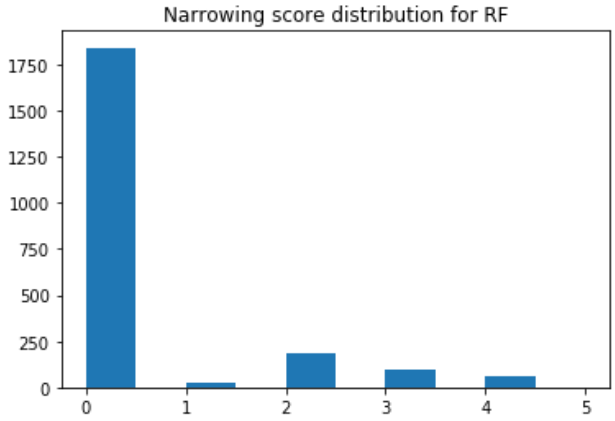


8.6.1 Narrowing score distribution for 4 limbs









8.7.1 Training vs. Validation Accuracy - CNN for Narrowing Score