PNAS
www.pnas.org

# Supplementary Information for

## A Deep Neural Network Improves Fracture Detection by Clinicians

**Robert Lindsey, Aaron Daluiski, Sumit Chopra, Alexander Lachapelle, Michael Mozer, Serge Sicular, Douglas Hanel, Michael Gardner, Anurag Gupta, Robert Hotchkiss, and Hollis Potter**

**Robert Lindsey**
**E-mail: rob@imagen.ai**

**This PDF file includes:**




www.pnas.org/cgi/doi/10.1073/pnas.1806905115

## Supporting Information Text

### Model Architecture

The model takes as input a pre-processed radiograph and generates two outputs. The first output is a single number between 0 and 1 which corresponds to the model's confidence that the in input radiograph contains a visible fracture or not. The second output is a dense conditional probability map, which we refer to as a heatmap, that localizes the site of the fracture in the preprocessed input radiograph. This is achieved by assigning each pixel in the image a confidence score indicating whether that pixel lies at the site of the given fracture.

The model is a deep convolutional neural network (DCNN) whose architecture is an extension of the common U-Net (1) model. We extend the U-Net model so as to have two branches generate the above two outputs. This architectural decision is motivated by the work of (2), which automatically learns to segment object candidates in natural images. The DCNN consists of a collection of Convolution layers, rectified linear unit (ReLU) layers, Dropout layers, Max Pooling layers, Up Sampling layers, Merging layers, Global Max Pooling layers (3), Fully Connected layers, and Sigmoid layers, stacked on top of each other. Figure 2 of the manuscript shows a schematic which gives the order in which these layers are stacked. Each layer is color coded with a different color for the purpose of clarity. The branch of the model that generates the heatmap closely resembles the U-Net model. It consists of a set of successive layers which first contracts the preprocessed input of size 1024 x 512 into feature maps whose dimensions are 128 x 64. An input size of 1024 x 512 is natural for wrist radiographs and was found to be sufficient for achieving high model performance. This is followed by a set of layers which expand this contracted feature map to generate a probability map of size 1024 x 512 (the same size as the pre-processed input radiograph), utilizing skip connections as part of the expansion process. Merging is done through concatenation of the feature maps depthwise. Lastly, the confidence score is generated by passing the contracted feature map through another stack of layers to output a single number which corresponds to the probability of whether the radiograph has a fracture or not. All the convolutional layers use a kernel of size 3 x 3 with a padding of 1. The number of output feature maps generated by each convolution layer depends on where the layer is stacked and is 1, 32, 64, 128, or 256. All the Max Pooling layers use a kernel of size 2 x 2. The dropout layer uses a dropout probability of .5.

### Model Training

Producing the final trained model involved two stages, a pre-training stage and a fine-tuning stage. For pre-training, we first split the dataset of all $100,855$ non-wrist radiographs into three parts: a training, validation, and test sets. The ratio of the number of samples in these sets is $80:10:10$ respectively. We first trained the model with this non-wrist training set, randomly initializing the weights at the start of the training process. Training was done using a variant of the stochastic gradient descent algorithm called ADAM (4). To help avoid over-fitting, we used early stopping to terminate the training process after no improvement had been observed on the validation set for five epochs. Additionally, we made use of data augmentation methods during training (5–7). We simulated the effect of having a larger labeled dataset by synthetically generating randomly altered versions of the radiographs on the fly during training. The alterations included random rotations, cropping, horizontal mirroring, and lighting and contrast adjustments. L2 regularization was used with a penalty of .00001. The loss function used to train the model is the sum of the pixel-wise cross entropy loss and the image-level cross entropy loss.

Once the model was pre-trained, we fine-tuned its weights on the dataset of wrist radiographs. As before, we split this dataset into three parts in the ratio of $80:10:10$, corresponding to training, validation, and test sets. In this fine-tuning phase, we start with the model whose weights are initialized to those of the model produced in the pre-training stage, and we train the model on the wrist-only training set. As in the pre-training stage, we used early stopping and data augmentation to reduce overfitting.

### References


1. Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation in *Medical Image Computing and Computer-Assisted Intervention.* pp. 234–241.
2. Pinheiro PO, Collobert R, Dollar P (2015) Learning to Segment Object Candidates in *Arxiv.* pp. 1–10.
3. Lin M, Chen Q, Yan S (2013) Network in Network. *International Conference on Learning Representations.*
4. Kingma DP, Ba JL (2015) Adam: a Method for Stochastic Optimization. *International Conference on Learning Representations 2015* pp. 1–15.
5. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems* pp. 1–9.
6. Yaeger L, Lyon R, Webb B (1996) Effective Training of a Neural Network Character Classifier for Word Recognition. *NIPS* 9:807–813.
7. Simard P, Steinkraus D, Platt J (2003) Best practices for convolutional neural networks applied to visual document analysis. *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* 1:958–963.