Appendices

A. Architecture

We adopt the ResNet with bottleneck residual block for our EcNet architecture baseline. Adding our proposed ensemble-connection in the original architecture is straightforward and can easily implemented. We use the ResNet code here[^1]. There are four convolutional (conv) groups having different feature map dimensions. The first two groups have 32×32 dimension, and the rest two has 16×16 and 8×8 dimensions. We replace the identity mapping with ensemble-connection between adjacent residual blocks connecting these four groups, with the purpose to allow multi-scale representation integration and contribute to the classification. We use 1×1 shortcut to deal with feature map size reduction and dimension increase [7]. The structure of a conv group is defined as follows:

\[
\begin{bmatrix}
1 \times 1, & 16\times c/4 \times w \\
3 \times 3, & 16\times c/4 \times w \\
1 \times 1, & 16\times c \times w
\end{bmatrix},
\begin{bmatrix}
1 \times 1, & 16\times c/4 \times 2w \\
3 \times 3, & 16\times c/4 \times 2w \\
1 \times 1, & 16\times c \times 2w
\end{bmatrix} \times N-1
\]

where the left part shows the first residual block and the right part show the rest N−1 residual blocks in conv group 2 to 4, while group 1 has a single [3 × 3, 16] convolutional layer. N increases as the network depth increases. The block type is indicted in the bracket, i.e., [kernel height × kernel width, kernel depth], c is a multiplier of the kernel depth, c = {1, 2, 2} for conv group {2, 3, 4}, respectively. w determines the width of the network [39]. Since ensemble-connection doubles feature maps intrinsically, the width from the second residual block grows by a factor of 2, as indicated by 2w in brackets. For example, for our EcNet-56-12, w is equal to 12, so the last convolution layer of the network has feature map dimension 768×8×8.

For experiments on the BCIDR dataset where the input image size is 224×224, we replace the first 3×3 convolutional layer with a 7×7 convolutional layer with stride 4. For EcNet-38-8, the feature map dimension of the last convolutional layer is 512×14×14.

The source code will be released.

B. Training and testing details

Training details To train MDNet, we use Adam to optimize the language model (i.e. LSTM), with fixed learning rate 1×10^{-4} and β_1 = 0.9 and β_2 = 0.99. We use stochastic gradient descent with momentum 0.9 to optimize the image model part. The LSTM hidden layer size is 256 and input embedding size is 128. The mini-batch size is 20. Due to the mini-batch augmentation operation (see Section 4.2), this size when flowing into LSTM becomes 120. We use dropout with a ratio of 0.5 at the output layer of LSTM. In practice, we warm up the image model part for 1 epoch with learning rate 0.1, then we drop the learning rate to 0.02 and update the overall network using our optimization strategy. The learning rate is divided by 2 for every 7,500 iterations. The LSTM used in compared baseline models has the same hyperparameters with ours, which performs well based on our empirical experiments.

Report generation To generate reports at the test stage, the trained MDNet takes an image as input. The image model part generates an image feature vector. The language model takes the image feature vector as the input at time step 0 and one (out of K) special token with a specified image feature type is used as the input at time step 1. Then, we continuously sample the image model until the special END token is sampled. At each time step, we select the most likely word and also use it as the input of next step.

Sentence-image retrieval This experiment mainly follows [22]. We rank each image of the test dataset I_D based on the joint probability \( p(x^Q|I_D) \) given a query sentence \( x^Q \) and compute recall rates. A complete report including image feature descriptions and diagnostic conclusions. We remove the words related to the conclusion when we compute \( p(x^Q|I_D) \) according to our evaluation metric.

Demo Our proposed MDNet is expected to simulate the entire manual diagnosis process of medical (pathology) images: automatic generating image symptom descriptions and conclusions as a diagnostic report and visual attention maps indicting where the network looks at in real time. Our demo in the supplementary material visualizes this process.