Supplementary data

Methods

Anonymization procedure

The images were matched with the corresponding radiologist's report before being subjected to an anonymization procedure where all patient data was removed and timestamps randomly transposed. The timestamps were transposed per patient in order to retain follow-up information. In order to avoid time sequence fingerprinting, to option to detect patient identity by identifying time examinations were performed, a random time interval was added between the images.

Technical details: We used Java with the GDCM-library for anonymizing the reports and DICOM-images.

DICOM header

The DICOM image format is a container for image data. It contains the raw data for the image combined with additional information that can be patient-, exam- or machine-specific (the x-ray device). This type of associated data is in the paper referred to as meta-data but is also known as the DICOM-header. The header has a few fields that are required but most fields are optional and there is only a vague indication to where the data resides. E.g. an image will have the patient identifier always at the same position but may have the information regarding the body part at different positions depending on which machine was used for taking the image.

Image rescaling and cropping

All five networks used have a fixed number of neurons that process the image. Since none of the networks can move within the picture, i.e. they have nothing corresponding to "eye movement", the image must be rescaled to match its expectations. The networks sizes varied between squares of 224 pixels width to 227 pixels. As the images had been rescaled to 256 x 256 the final image presented to the network is a crop consisting of a subsection of the original image. This is a technique that forces the network to cope with different positions of the same structure, thereby forcing the network to be translation-independent. We then further randomly flipped the image along the horizontal axis as we wanted the network to complicate the detection of laterality.

Technical details: We used the OpenCV-library v. 3.0 for preprocessing the image in Python.

Deep learning procedure

Network structure

The networks were pre-trained on the ImageNet dataset and converted into Torch7 using the loadcaffe package. The last fully connected layer from each network was stripped before adding our own outcomes. The outcomes were added in parallel where each outcome had its own fully connected layer using the ConcatTable structure in Torch7. The last fully connected layer is responsible for the final combination of features into object representations which is why we only replaced this with our own layer.

Training procedure

We used stochastic gradient descent with batch size of 1 during training. The learning rate was adapted at the end of each epoch and we ran the experiment for 13 epochs (1 epoch = 1 run through the entire image dataset) using identical datasets. The data was split into 70% training images, 20% validation images and 10% test images. Over-fitting was controlled via the validation dataset and only the best performing network was tested against the test images.

During training we used random cropping to desired image size while the test image was cropped to the central portion of the image according to standard procedure. The random cropping helps the network to learn how to identify the same structure regardless of its position within the current image.

Manual review and correction

The initial, or computed, network error for each category is:

$$\textit{network error} = 1 - \textit{network accuracy}$$

Manually reviewing a subset of network errors gave a correction, a network error correction (per category), as the percentage of all studied errors that were valid calls (i.e. TN+TP, correct laterality, correct body part or correct view). The *revised network accuracy* is thus:

$$\textit{revised network accuracy} = \textit{network accuracy} + (\textit{network error} \cdot \textit{network correction})$$

This revision can only increase network accuracy and not decrease it.