**Computational mammography using deep neural networks**

**A. Dubrovinaa, P. Kisilevb, B. Ginsburgc, S. Hashoulb,d and R. Kimmela**

**aComputer Science, Technion, Haifa, Israel; bIBM Haifa Research Lab, Haifa, Israel; cNVIDIA,Santa Clara, CA, USA; dCarmel Medical Center, Haifa, Israel**

**Abstract:**

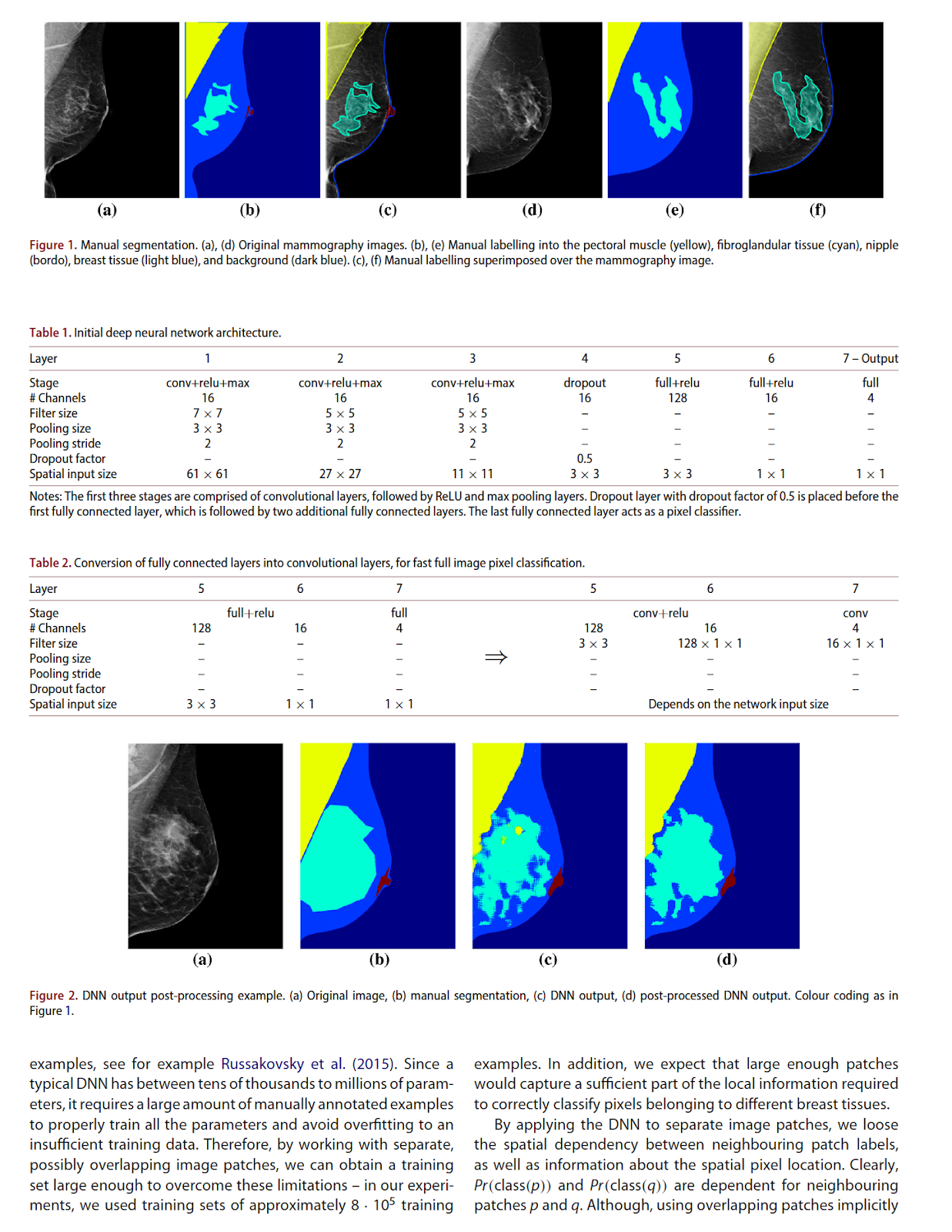
Here, we present a novelsupervised Convolutional Neural Network (CNN)-based method for breast tissue classification from mammogram images.

**Problem Statement:**

Given a digital mammography image, we wish to associate each of its pixels with one of the **four following classes: pectoral muscle, fibroglandular tissue, nipple, and the general breast tissue,** which includes fatty tissue and skin. **Our data-set consists of 40 digital mammograms of mediolateral oblique (MLO) view, manually segmented by an expert into the four regions.** While the images include a significant portion of background pixels, these pixels are easily detected in a **pre-processing step, by thresholding image intensity values with zero threshold.**

**Breast tissue classification with DNN:**

DNNs can provide discriminative image representations, sometimes referred to as features, by **successive application of linear filters, non-linear activation functions, normalisation, and pooling operations, thus, avoiding the need to design such features manually.** The proposed DNN classifier, applied to raw image pixels, provides the probability of each pixel to belong to one of the four classes described above: **Pr(class(p) = k), k = 0, 1, 2, 3.** The DNN is applied in a patch-wise manner: to classify the pixel p, the DNN is fed with a square image patch of size w Å~ w, centred at p. In our experiments, we set w = 61 pixels. **The patches are pre-processed prior to training and classification, to have a zero mean, by subtracting from them the mean of all patches in the training set. The classification accuracy is acquired by a multinomial logistic loss function.**

****

**Architecture**

It consists of three stages of convolutional layers, ReLU (rectified linear unit) activation layers, and max pooling layers, followed by three fully connected layers. To prevent overfitting, a dropout layer (Srivastava et al. 2014) with dropout factor of 0.5 was added between the convolutional and the fully connected layers. The image intensity and the normalised coordinate information can be combined after the first fully connected layer in the following manner: the two normalised patch centre coordinates and the 128-dimensional fifth layer output are concatenated into a 130-dimensional vector, passed to the second fully connected layer.

**Fast full image pixel classification**

During the classification stage, the network must be applied separately to all the **overlapping image patches**. This introduces a significant computation overhead, since both the convolutional layers, and the fully connected layers are applied multiple times to overlapping regions. we converted the proposed classification network into a **fully convolutional network.** That is, we converted the fully connected **layers #5, 6, 7, into convolutional layers,** as shown in Table 2. ***The new network is able to output dense predictions for input images of arbitrary sizes.*** Specifically, for the 829 x 640 images we used in our experiments, the new network output was 97 x 73. To obtain dense prediction for the whole image, we adopt the shift-and-stitch method of   In the proposed network, **the outputs were downsampled by a factor of 8 with respect to inputs.** Hence, by feeding the new network shifted versions of the input, **by i ∈ {0, 1, . . . , 7} pixels right and j ∈ {0, 1, . . . , 7} pixels down,** and **interlacing the obtained 64 output images,** we obtain a **dense prediction for the whole image.** The classification time of the new network is approximately 1.8 seconds as opposed to 114 seconds for the **per-patch neural network application.** Conversion to the fully connected network requires the following adjustment. Previously, the mean of all training examples was subtracted from the network input during the training and the classification. Now, a single value is subtracted from the training examples at the training stage, and **from the entire image at the classification stage, allowing a full-image classification.** In our experiments, this change had a minor effect on the classification accuracy. We used a single mean intensity value of the pixels in the mean image, computed over all the training examples

**Classifier output post-processing**

The raw DNN classifier output obtained for one of the images in our data-set is shown in Figure 2(c). Since the proposed patchbased classifier cannot incorporate constraints in relative spatial locations of different tissues, it may produce fragmented regions, as shown in Figure 2(c). Therefore, during the post-processing step, as dictated by the physiological breast structure, (i) the interior of the large pectoral muscle region adjacent to origin of the image is filled with its corresponding label, while small unconnected components of the pectoral muscle region are given the label of the general breast tissue; (ii) the outer boundary of the fibrogladular tissue region is morpholocally filled to contain the fibrogladular tissue label, and its connected components smaller than some predefined threshold, are removed; (iii) a single connected component of the nipple region, closest to the centre of the image, is retained.

**Segmentation results**

**Pre-processing**

**In our experiments, we used a data-set with 40 manually segmented MLO views. All images were aligned so that the pectoral muscle appeared on the left side of the image, for spatial consistency. A leave-one-subject-out cross validation procedure was used. We considered different 40 imagesets with 39 training images, and the remaining image was used to evaluate the classification performance. The results presented below were averaged over all 40 possible training and test image combinations. Approximately 800,000 patches were extracted to form the training set, containing an equal number of patches centred at pixels belonging to the four different regions.**

**Network training**

**The DNN was trained by stochastic gradient descent with**

**momentum. We used mini batches of 256 image patches, and**

**learning rate of 10−3, reduced by a factor of 10 twice, after 30,000**

**and 60,000 training iterations (approximately 10 and 20 epochs).**

**We used momentum 0.9,weight decay 5.10−4, and initialised the**

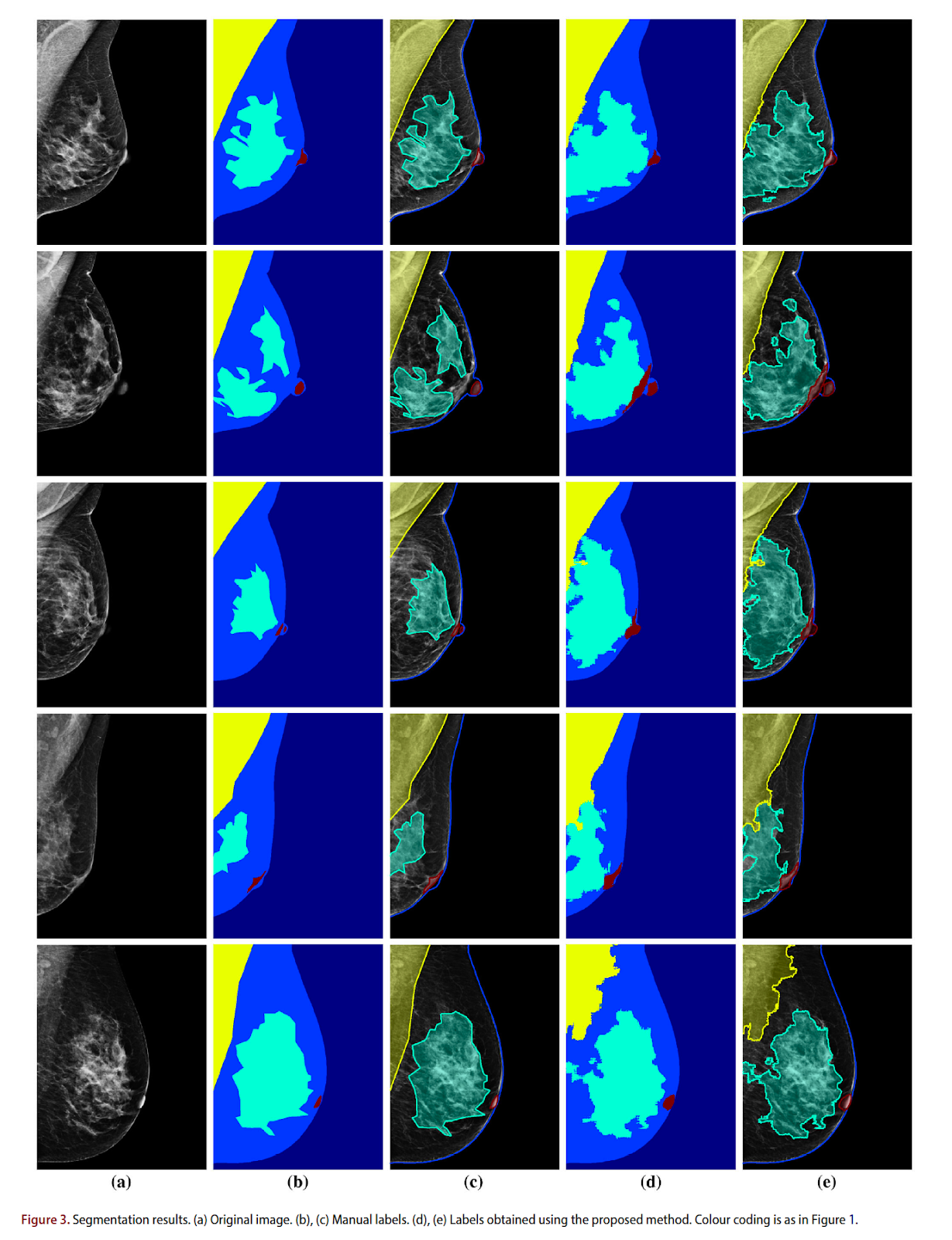
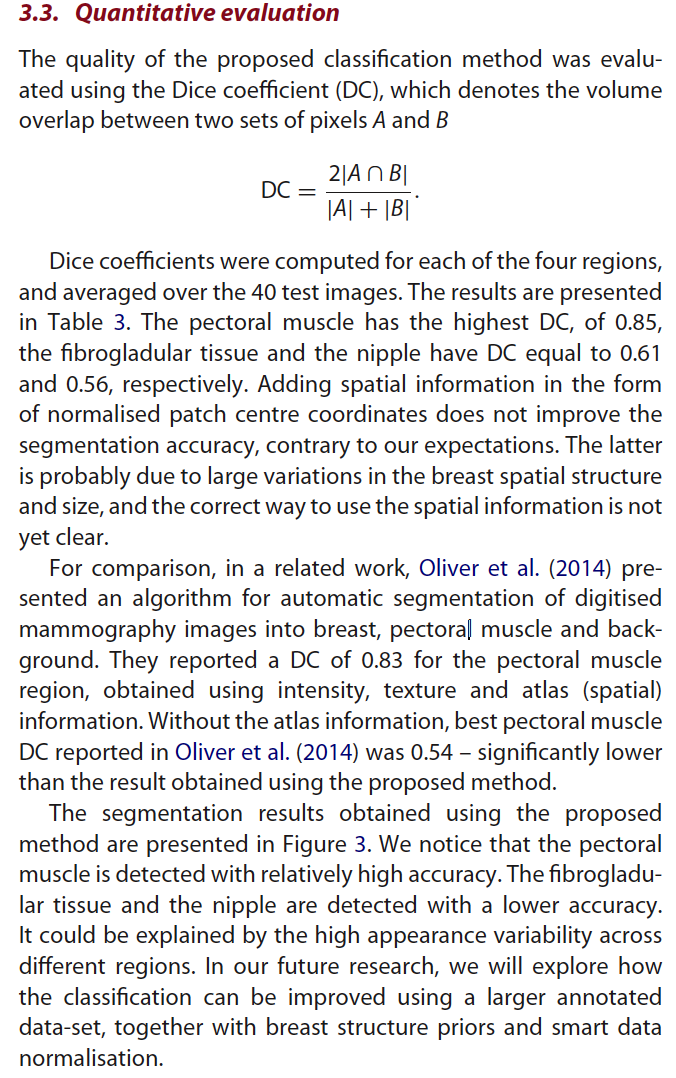
**netweights randomly, using normal distribution with zero mean**

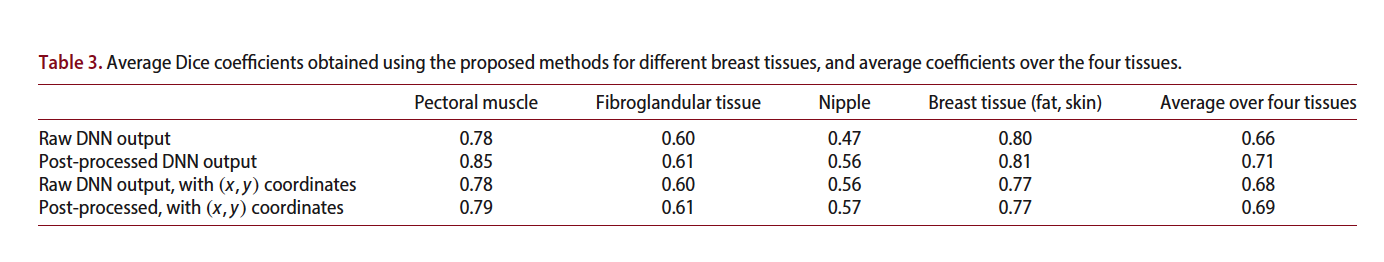
**and 0.01 and 0.1 variances for the convolutional and the fully**

**connected layers, respectively. During testing, the dropout layer**

**was removed, and the outputs of the layer #3 were multiplied by**

**a factor of 0.5.**

****

****