Automatic assessment of mammographic density using a deep transfer learning method

Steven Squires,[a] Elaine Harkness,[a] Dafydd Gareth Evans,[b] and Susan M. Astley[a,*]

[a]University of Manchester, School of Health Sciences, Division of Imaging, Informatics and Data Sciences, Faculty of Biology, Medicine and Health, Manchester, United Kingdom

[b]University of Manchester, Manchester Academic Health Science Centre, School of Biological Sciences, Division of Evolution, Infection and Genomics, Faculty of Biology, Medicine and Health, Manchester, United Kingdom

Abstract

Purpose: Mammographic breast density is one of the strongest risk factors for cancer. Density assessed by radiologists using visual analogue scales has been shown to provide better risk predictions than other methods. Our purpose is to build automated models using deep learning and train on radiologist scores to make accurate and consistent predictions.

Approach: We used a dataset of almost 160,000 mammograms, each with two independent density scores made by expert medical practitioners. We used two pretrained deep networks and adapted them to produce feature vectors, which were then used for both linear and nonlinear regression to make density predictions. We also simulated an "optimal method," which allowed us to compare the quality of our results with a simulated upper bound on performance.

Results: Our deep learning method produced estimates with a root mean squared error (RMSE) of 8.79 ± 0.21. The model estimates of cancer risk perform at a similar level to human experts, within uncertainty bounds. We made comparisons between different model variants and demonstrated the high level of consistency of the model predictions. Our modeled "optimal method" produced image predictions with a RMSE of between 7.98 and 8.90 for cranial caudal images.

Conclusion: We demonstrated a deep learning framework based upon a transfer learning approach to make density estimates based on radiologists' visual scores. Our approach requires modest computational resources and has the potential to be trained with limited quantities of data.

[DOI: 10.1117/1.JMI.10.2.024502]

Keywords: deep learning; mammography; breast density; transfer learning; cancer risk.

Paper 22226R received Sep. 3, 2022; accepted for publication Mar. 6, 2023; published online Apr. 5, 2023.

1 Introduction

Studies have shown a strong relationship between breast density and the risk of developing breast cancer.[1–3] Breast density is generally defined as the proportion of fibro-glandular tissue within the breast; however, there are various different methods to estimate this measure, and these show different levels of correlation with cancer risk, with breast density assessed subjectively by radiologists demonstrating a stronger link to breast cancer than other methods.[4,5] Exactly why this subjective type of measure produces improved performance is not clear, although presumably radiologists utilize years of knowledge and experience to go beyond simple estimates of ratios of dense to nondense tissue. Their knowledge and experience can be harnessed by training an automated method to produce similar density assessments. If we can

*Address all correspondence to Susan M. Astley, sue.astley@manchester.ac.uk

accurately measure breast density over time, we can also measure whether risk-reducing interventions are working effectively.[6]

Deep learning is now the dominant method used in general image analysis tasks due to the higher accuracy it tends to achieve over more traditional methods.[7] The main alternative to deep learning is to hand-craft features and then apply traditional machine learning methods. The advantage of deep learning is that the features are automatically extracted from the data itself. This is appealing for breast density estimation as we do not completely understand why subjective expert judgement seems to outperform other methods. Deep learning does have significant downsides, such as the requirement of significant amounts of data and computing power. In addition, the interpretation of results is challenging.[8]

Medical imaging problems often have significantly smaller amounts of data than datasets that deep learning is usually trained on. One of the most commonly used nonmedical imaging datasets is ImageNet,[9] which consists of over a million images and a thousand classes. Conversely, medical imaging datasets tend to be considered large if they contain tens of thousands of images, with many datasets consisting of far fewer.[10] Even with these challenges, deep learning is increasingly being used in medical imaging studies[11] with good outcomes.

There have been many approaches for making breast density estimates. One example[12] used convolutional neural networks (CNNs) with support vector machines (SVMs) to classify breasts into four different density categories. Other work used unsupervised CNNs at different scales to extract features before finetuning on known labels.[13] Another method used fuzzed c-means before applying an SVM.[14] Deep learning has been used to make binary (dense and nondense) and four-way (fatty, scattered, heterogeneous, dense) classifications of mammograms.[15] Other classifiers of four-way density predictions have used synthetic and real mammograms.[16] Other work has involved segmentation to separate the tissue into dense and fatty before calculating the density.[17] One recent approach built[18] a deep learning model and trained it from scratch to make estimates of breast density using domain expert labeled images as targets. The authors showed that deep learning models could make estimates that correlate well with the expert labels.

Deep learning can either be performed by designing and training a model from scratch or with a transfer learning approach.[19] At the moment, there is little agreement across the medical imaging field about which option produces better outcomes.[10] In a field as new as deep learning, with little solid underlying theory and estimates being made on highly complex datasets, and using complex models, it is difficult to draw strong conclusions about which method to follow. A sensible approach is to test multiple methods on different problems and attempt to determine which models work better in certain situations. A transfer learning method to estimate breast density for low dose mammograms recently showed good performance[20] albeit on a small dataset; in this paper, we will demonstrate that transfer learning using deep networks produces good performance using a large full dose dataset.

We present a transfer learning method based upon two independent deep learning models trained on ImageNet[9] with regression models trained using a dataset with visual labels produced by domain experts.[21] These models are combined using a multilayer perceptron (MLP) to make a final ensemble prediction. We compare these results with those of a previous method trained on this dataset.[18] Each image in the dataset was previously assessed by two independent readers, which gives us the opportunity to analyze the quality of the labels themselves.[21] We will show there are challenges with using data with this level of noise both in terms of training and also in how we assess model performance.

The key contributions of this work are

- We design and implement a transfer learning-based pipeline to make estimates of breast density. Our framework consists of preprocessing, feature extraction, density mapping, and final ensemble estimation steps. The overall method produces automated breast density estimates with relatively low computational requirements.
- We analyze the dataset itself to assess how well our model performs. We demonstrate that the variability of the reader assessment means that we are limited in our ability to effectively discriminate between the quality of models once they have reached a certain accuracy. Our framework produces estimates that fall close to this range.

2 Data

The dataset is formed of full-field digital mammogram images with associated density assessments by domain experts (radiologists, advanced practitioner radiographers, and breast physicians). The images are from the predicting risk of cancer at screening (PROCAS)[21] study. All images were produced using GE mammography machines and have three different image sizes: 2294 × 1914, 3062 × 2394, and 5625 × 4095. There are four views for each woman: cranial caudal (CC) and mediolateral oblique (MLO) for the left and right breast, giving a right craniocaudal view, right mediolateral oblique view, left craniocaudal view, and left mediolateral oblique view.

In the PROCAS study, every image was independently viewed by two domain experts, out of a total pool of 19, who assigned a density value on a visual analogue scale (VAS) between 0 and 100 for each image. Therefore each woman received eight estimates of breast density across both CC and MLO and right and left breasts.

Before performing any preprocessing, we removed any images that do not have labels assigned from two readers. In Table 1, we show the number of images at the three different sizes in our dataset. The images are drawn from 39,357 women, although not all the women have the full set of image views. When considering predictions per image, we will use the entire dataset, but when analyzing density estimates per woman, we will only consider those women who have all four views. In Fig. 1, (a)–(c) show image examples of CC and (d)–(f) show image examples of MLO images with [(a), (d)] low, [(b), (e)] medium, and [(c), (f)] high densities as defined by the average of the two reader scores. These are images after preprocessing (see Algorithm 1 in Sec. 3.1 for details).

We partition the dataset into training, validation, and testing sets by woman, so that all the available views for a women are in the same partition. All the women from a previous case control set,[4] discussed below, are placed in the testing set. Otherwise the rest of the training, validation, and testing sets are chosen at random from the PROCAS dataset. In total, we have 33,011 women in the training set, 3649 women in the validation set and 2697 women in the testing set.

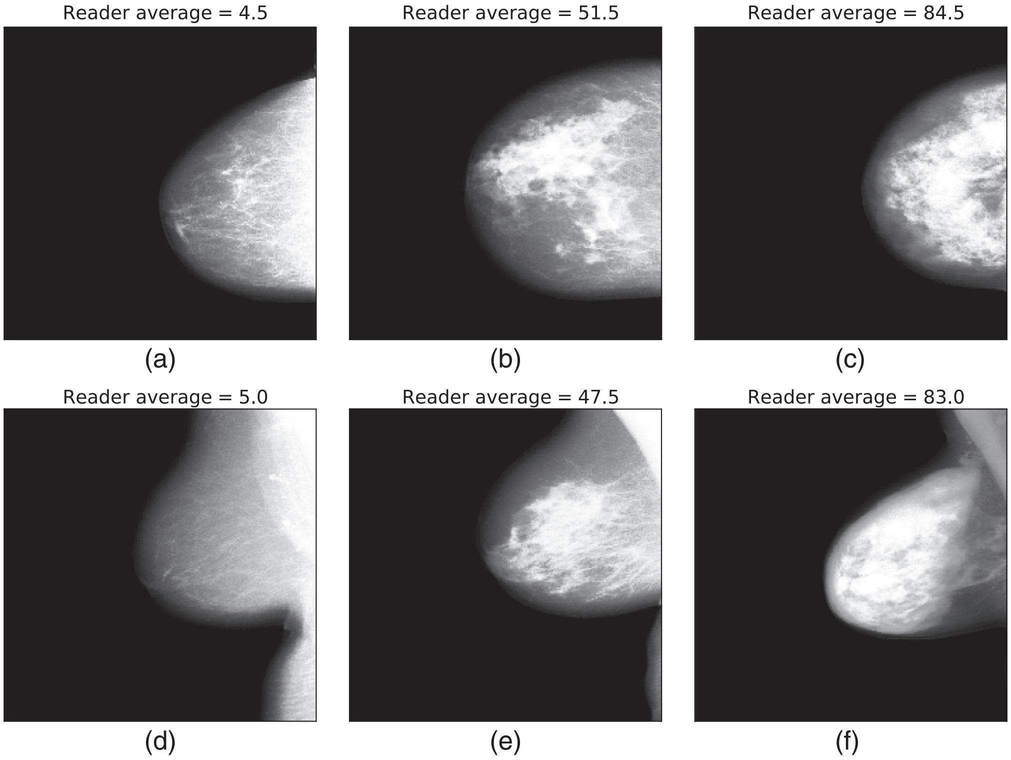
In addition, to compare with a previous study,[18] we have a second partition with 19,048 women in the training set, 769 women in the validation set, and 19,844 women in the testing set. Similar to the previous partition, all women from the previous case control set[4] are in the testing set.

We also investigated the log ratios for developing cancer from a previously created subset of the data[4] called the priors, which consist of women who did not have a cancer detected when their screen was taken but subsequently went onto develop a cancer. All these data were held in the test set and not used for training or validation.

These cancer risk predictions (on the priors) are found by taking the breast density estimates and splitting them into quintiles. Three control (no breast cancer) mammograms are matched with one breast cancer prior mammogram by matching on other known risk factors (age, body mass index, hormone replacement therapy use, menopausal status, and year of mammogram). This attempts to isolate breast density as a risk factor while controlling for potential confounding variables. Therefore it allows for estimates of the ratios of probability of developing cancer for women with high breast density compared with low breast density. The log ratio is calculated using conditional logistic regression on quintiles of the density. The higher the risk ratios at high

Table 1 Number of images at each size and aspect ratio.

Name	Height by width	Number of pixels	Aspect ratio	Number of images
Small	2294 × 1914	4,390,716	1:1.20	57,733
Medium	3062 × 2394	7,330,428	1:1.28	95,184
Large	5625 × 4095	23,034,375	1:1.37	4055
All	—	—	—	156,972



(a) (b) (c)

(d) (e) (f)

Fig. 1 Images of (a), (d) low density; (b), (e) middle density; (c), (f) high density images as defined by the average percentage density. These are the images after preprocessing (see Algorithm 1 in Sec. 3.1 for details). (a)–(c) shows the CC images and (d)–(f) shows the bottom MLO.

Algorithm 1 Preprocessing.

```
Input: a set of $n$ mammogram images, $\{Im_i \in \mathbb{R}^{a\times b}\}_{i=1}^{n}$
Output: a set of processed mammogram images, $\{I_i \in \mathbb{R}^{224\times 224}\}_{i=1}^{n}$
for $i$ to $n$ do
    Resize image using cubic interpolation from size $a \times b$ to size $224 \times 224$
    Clip all element values at 75% of maximum image value
    Subtract minimum and divide by maximum value
    Invert values
    if image is on left side then
        Flip image to be on right side
    end if
    Perform histogram equalization on image
Normalize to between 0 and 1 by subtracting minimum values and dividing by maximum
Loop Output: Processed image, $I_i$
end for
```

density compared with low density, the better the density model is at assessing risk. For further details of the approach, see Astley et al.[4]

3 Prediction Methods

The objective is to take in a mammogram image and output an estimate of the density score of that image. The procedure we follow has four parts: (1) preprocessing stage, (2) using pretrained deep learning models to extract features from the processed image, (3) mapping the features to a set of density scores, and (4) using an ensemble approach to take the multiple scores and produce a final density estimate. Throughout this paper, we will use "density mapping" to refer to the third step, where individual feature vectors are mapped to a density score and "ensemble prediction" to refer to the fourth step, which takes those density estimates and combines them into a final ensemble prediction. In Fig. 2, we show this process in a schematic format.

3.1 *Preprocessing*

We preprocessed the images to size 224 × 224 (50,176 input pixels) for processing by the feature extractors. The downscaling process reduces the number of input pixels down to 1.1%, 0.68%, and 0.22% of the original size for the small, medium, and large images, respectively (see Table 1 for the three image sizes).

We also enhance the contrast of the images to make it easier for the methods to extract information. In Algorithm 1, we lay out the steps we take to preprocess the images. We first rescale the image from its original size (see Table 1 for the three image sizes) down to 224 × 224 using cubic interpolation. We then clip all element values to 75% of the image maximum, subtract the minimum, and divide by the new maximum. The values are inverted and any image that is positioned on the left hand side is flipped horizontally. We perform histogram equalization and normalize the image to contain values between 0 and 1. We show examples of these preprocessed images in Fig. 1.

3.2 *Feature Extraction*

To perform the feature extraction part of our procedure, we use two pretrained deep networks: ResNet[22] and DenseNet.[23] Both were trained on the ILSVRC 2012 version of ImageNet,[9] a large

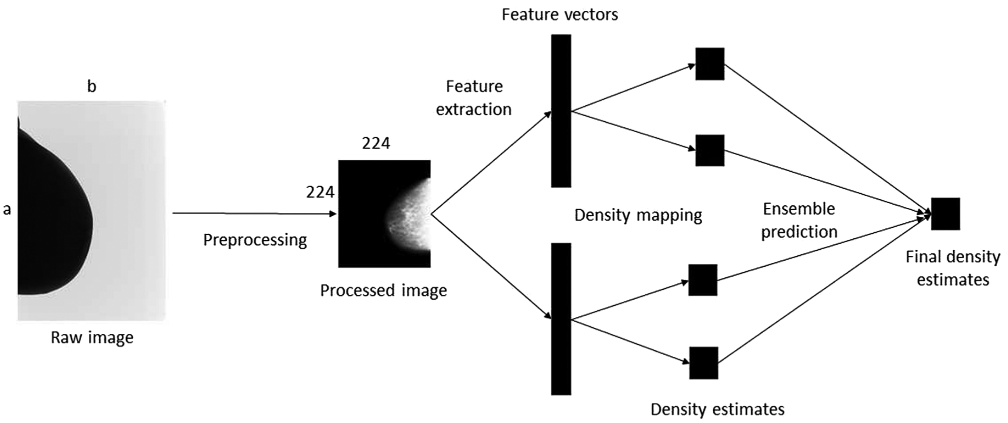


Fig. 2 Illustration of our deep learning pipeline for producing final density estimations on mammograms. The input mammogram is processed to reduce the size down from $(a \times b)$ to (224×224) and increase the contrast. The processed image is fed into the feature extractors, we use two in this paper (ResNet[22] and DenseNet[23]) but others could be included in which each produce a feature vector for that image. The density mapping, either a linear regression or MLP, is then applied to each feature vector to produce the separate density estimates. Finally, the ensemble model is applied to convert the separate density estimates into one final prediction.

Algorithm 2 Feature extraction method.

```
Input: a set of $n$ preprocessed mammogram images (see Algorithm 1), $\{I_i \in \mathbb{R}^{224\times 224}\}_{i=1}^{n}$
Output: a set of $n$, $p$-dimensional feature vectors, $\{\mathbf{v}_i \in \mathbb{R}^{p}\}_{i=1}^{n}$
Select pretrained deep model (ResNet or DenseNet) and load model with pretrained weights
Remove the necessary layers from the model and alter the model so it transforms the output into a vector
for $i$ to $n$ do
    Copy matrix $I_i \in \mathbb{R}^{224\times 224}$ into a tensor $I_{i'} \in \mathbb{R}^{224\times 224\times 3}$
    Normalize each channel to have means of (0.485, 0.456, 0.406) and standard deviations of (0.229, 0.224, 0.225), respectively.
    Run tensor through model, producing feature vector $\mathbf{v}_i$
end for
Output: $\{\mathbf{v}_i \in \mathbb{R}^{p}\}_{i=1}^{n}$
```

database of 1.2 million images across 1000 classes. ResNet and DenseNet are both popular in the literature, available as easily accessible models from PyTorch[24] and produce modest sized feature vectors, which keeps the computational requirement manageable. However, there are other potential feature extraction networks available that could be further investigated, such as VGG[25] or inception.[26]

To extract features from the preprocessed images, we remove the final fully connected classification layer from both networks, which alters the output from 1000 classes to 2208 and 512 dimensional feature vectors for DenseNet and ResNet, respectively. Details of our implementation is in Appendix A. We do not adjust the weights of the network or perform any form of fine-tuning. The training for the feature extractors was performed using ImageNet data, which consists of natural images with three channels, whereas our mammograms only have one channel. We therefore copy across the same image to make a repeated three-channel tensor. Both networks require the inputs to be normalized across the channels to have means of (0.485, 0.456, 0.406) and standard deviations of (0.229, 0.224, 0.225), respectively. In Algorithm 2, we lay out the steps of our feature extraction method.

3.3 *Density Mapping*

To map the deep feature vectors to produce a density estimation, we utilize two methods: linear regression with regularization and the application of MLPs. The linear regression approach enables us to see how well a simple model, with one consistent solution, performs. An MLP can (in principle) map any function,[27] and allows us to explore whether nonlinear mappings are necessary.

A procedure for our linear regression method is shown in Algorithm 3. During training, we add a bias term to the feature vectors from the training set, $\{\mathbf{v}_i^{\mathrm{tr}} \in \mathbb{R}^{p}\}_{i}^{n_{\mathrm{tr}}}$, then stack them into a feature matrix, $X \in \mathbb{R}^{n_{\mathrm{tr}}\times(p+1)}$. We utilize standard ridge regression, forming an objective function, $f = \|X\mathbf{w} - \mathbf{y}\|_2^2 + \lambda_2\|\mathbf{w}\|_2^2$, where $\mathbf{y} \in \mathbb{R}^{n_{\mathrm{tr}}}$ are the known labels and $\mathbf{w} \in \mathbb{R}^{p+1}$ are the weights. To find the optimal λ_2 term, we perform five-fold cross-validation on the training data, finding the value of λ, which minimizes the combined held-out error. We then retrain the weights on the entire training dataset using the optimal λ. To solve for $\mathbf{w}$ during both cross-validation and for the final optimal λ, we use pseudoinverse inversion: $\mathbf{w} = (X^T X + \lambda I)^{-1} X^T \mathbf{y}$.

The linear regression approach assumes that the feature extractors are mapping the images onto a reasonably linear space where there is no need to consider nonlinear correlations. As this assumption may not be correct, and it may be possible to improve performance with a more complex mapping, we also consider the use of the MLP.

Algorithm 3 Density mapping: linear regression training.

```
**Inputs**: a set of $n_{\mathrm{tr}}$ training feature vectors, $\{\mathbf{v}_i^{\mathrm{tr}} \in \mathbb{R}^p\}_{i=1}^{n_{\mathrm{tr}}}$ with known density labels $\mathbf{y} \in \mathbb{R}^{n_{\mathrm{tr}}}$
**Output**: a $p+1$-dimensional weight vector $\mathbf{w}$
Add bias term to each vector $\mathbf{v}$
Concatenate $n$ feature vectors to form a matrix: $X \in \mathbb{R}^{n_{\mathrm{tr}}\times(p+1)}$
Objective function: $f = \|X\mathbf{w} - \mathbf{y}\|_2^2 + \lambda_2\|\mathbf{w}\|_2^2$
Split training data to perform $K$-fold cross-validation
**for** $\lambda$ in range of $\lambda$ values **do**
  Solve for each, $k$, held-in fold by pseudoinverse inversion: $\mathbf{w}_k = (X_k^T X_k + \lambda I)^{-1} X_k^T \mathbf{y}_k$
  Find $k$ held-out errors
**end for**
Use $\lambda_{\mathrm{opt}}$ value that produces the lowest held-out error
Perform pseudoinverse inversion on full set of training data: $\mathbf{w} = (X^T X + \lambda_{\mathrm{opt}}\mathbf{I})^{-1} X^T \mathbf{y}$:
**Return w**
```

Algorithm 4 Density mapping: MLP.

```
**Inputs**: a set of $n_{\mathrm{tr}}$ training feature vectors, $\{\mathbf{v}_i^{\mathrm{tr}} \in \mathbb{R}^p\}$ with known density labels $\mathbf{y} \in \mathbb{R}^{n_{\mathrm{tr}}}$
**Output**: trained MLP model
Create MLP network: choose number of layers and number of neurons (see Appendix A for details.
**for** $L_1$, $L_{1\mathrm{Smooth}}$, MSE objective functions **do**
  **for** range of learning rates and training epochs **do**
    Train MLP network on training data
    **if** Objective function ends before convergence **then**
      Increase training epochs
    **else if** Objective function fails to converge **then**
      Reduce learning rate
    **end if**
  **end for**
  **Output**: trained MLP model
**end for**
```

In Algorithm 4, we show the procedure we follow for applying the MLP to the data. We have the same input training vectors as for the linear regression. The details of the architecture and training are in Appendix A. In summary, we train the MLP on the same training partition as the linear regression, use three objective functions: L_1, $L_{1\mathrm{Smooth}}$, mean squared error (MSE), and select the learning rates and training epochs from a trial and error approach until the training error converges reasonably smoothly to a steady state.

Algorithm 5 Ensemble method.

```
Inputs: a set of $m$ models trained on the same training set, $\{\mathrm{mod}\}_i^m$. A second training set of feature vectors $\{\mathbf{v}_i^{\mathrm{val}} \in \mathbb{R}^m\}_{i=1}^{n_{\mathrm{val}}}$ with known density labels $\mathbf{y} \in \mathbb{R}^{n_{\mathrm{val}}}$,
Output: trained ensemble MLP model
for model in $\{\mathrm{mod}\}_i^m$ do
  Make predictions, $\hat{\mathbf{y}} \in \mathbb{R}^{n_{\mathrm{val}}}$ on $\{\mathbf{v}_i^{\mathrm{val}}\}$
end for
Build new ensemble training set $\{\mathbf{v}_i^{\mathrm{ens}} \in \mathbb{R}^{n_{\mathrm{val}} \times m}\}$ from model predictions
Train new MLP (see Algorithm 4 and Appendix A for details) on the ensemble training set.
Output: MLP trained to predict ensemble density predictions from $m$ input predictions
```

3.4 *Ensemble Approach*

The two deep feature extracting models along with a linear regression and the MLP models produce different predictions for each image. In addition, as will be discussed in Sec. 5.1, we can train on different labels, producing another set of different density mapping models. Ensemble methods tend to outperform individual models[28] so if we can combine the individual predictions together, we would expect to improve the model performance. Therefore there are 16 separate predictions: two feature extractors (ResNet and DenseNet) and four density mappings (linear regression and three MLPs trained with different objective functions). Those eight models can be either trained on averaged labels or individual labels (see Sec. 5.1 for details), giving the total of 16 separate sets of predictions.

To produce our final ensemble prediction, we start by splitting the training data into two sets —a larger one that each individual model is trained upon (see Algorithms 3 and 4) and a smaller one to train the ensemble on. We train the m individual models on the first training set and then apply each to the second training set to produce n predictions. We stack the predictions into a new training set, which we then train a new MLP on to find a final model. To make predictions, we run all m models on an image, produce the output for each one and make a final prediction by feeding the m-dimensional vector into our ensemble model. In Appendix A, we provide details of the architecture and training procedure. There are many ensemble approaches available[29] including simple methods, such as averaging across individual results. We show just one approach that demonstrates that we can improve on individual results using an ensemble method.

4 Assessing Predictive Performance

4.1 *Metrics*

To assess the predictive performance of our models, we consider a range of metrics. The global measures we use to compare the quality of the VAS predictions with labels are Pearson correlation coefficient, root mean squared errors (RMSE), mean absolute error (MAE), and median absolute error (MedAE). We also show results of the risk ratios on the priors as discussed in Sec. 2.

4.2 *Perfect Predictor Estimates*

Due to label unreliability, a “perfect” model, which correctly estimates the VAS scores for all images, if compared to the noisy labels, will have a nonzero error. Therefore, we need to produce an estimate of what metrics a “perfect” model would produce so that we can assess the quality of our approach.

We take the average of the two reader scores to be the "true" values and then assume that the actual reader scores are drawn from a Gaussian distribution around the real score. The Gaussian error distribution varies for different parts of the score distribution, with smaller average errors at both low and high densities, when considering averaged reader scores. We calculate the error distribution for small ranges of densities. We then use the "true" values, the averaged reader scores, and create a pair of modeled reader scores by adding Gaussian noise to the "true" value.

In detail, we take the average reader scores and call these our perfect modeled estimates. We bin the average reader scores into small bins (4% each), which provides a reasonable number of data points to calculate the Gaussian parameters, except for above 80% where there is little data so that we use the range 80% to 100% as one bin. The distributions approximate a Gaussian (see Fig. 9 in appendix) but the Gaussian tails are longer than the real results, and there is the problem of modeled reader estimates below 0 and above 100. To correct for these two effects, we resample from the distribution if the reader estimate is outside the 0 to 100 range or if the deviation of the sample from the mean is greater than a certain threshold $\sigma_{\max}$. These corrections make the modeled distribution a plausible match for the real data. To make the model as simple as possible, the only parameter we alter is $\sigma_{\max}$. To check whether the model produces sensible estimates, we compare the differences between the pairs of real reader estimates and the modeled pair differences. We can then adjust this tuneable parameter to equalize this comparison for the different metrics we consider. In this manner, we can estimate the range of metrics that would occur for an optimal prediction method. We summarize our method in Algorithm 6.

We compare the four metrics (correlation, RMSE, MAE, and MedAE) between the pair of modeled reader scores and match up the metrics between the pair of real reader scores. For example, we alter $\sigma_{\max}$ until the modeled pair of readers have the same RMSE as the real pair of readers. We do the same for the other metrics and record the optimal metrics from the lowest to the highest. We then show the range of these values when we perform the simulation.

Algorithm 6 Perfect predictor model.

```
Inputs: $n$ average reader scores, $n$ pairs of reader scores, maximum difference $\sigma_{\max}$

Output: metrics for the optimal predictor

Bin the average reader scores into 4% bins up to 80% then one bin for 80% to 100%.
Calculate Gaussian parameters, $\mu_j$ and $\sigma_j$ for each bin, $j$ for the differences between individual and the average of scores
for average reader score $\in n$ do
  for two reader estimates do
    Draw sample, $x$ from Gaussian distribution for that bin, calculate the deviation ($\hat{\sigma}$) from the mean
    if $x > 100$ or $x < 1$ or $\hat{\sigma} > \sigma_{\max}$ then
      Resample from distribution
    end if
  end for
  Save two reader estimates alongside average reader score
end for
Original average reader score becomes optimal density estimates, $\mathbf{y}_{\text{opt}} \in \mathbb{R}^n$. Modeled reader estimates are now $\mathbf{x} \in \mathbb{R}^{n \times 2}$.
Calculate metrics by comparing $\mathbf{y}$ to average of $\mathbf{x}$
```

5 Results

We split our results section into three subsections. In Sec. 5.1, we analyze the labels to gain insight into their reliability and how we need to train our models. Then in Sec. 5.2, we analyze the performance of our models compared to the ground truth as produced by radiologist's labels of the images. Finally, in Sec. 5.3, we make a range of comparisons both between different versions of our models and between our models and previous work to try and gain further insight into our model performance.

5.1 *Reader Density Label Analysis*

In Fig. 3, we show the (a) distribution of the density scores and the (b) distribution of the absolute differences between reader scores. For the density scores, we show both the average of the two readers (averaged) and the individual reader estimates (individual). As there are twice as many individual scores as averaged scores, we normalized the distributions to make them directly comparable. The distributions of density are highly skewed, with little data at high breast densities and relatively large amounts with a density of around 20. In addition, the averaged reader score tends to compress the distribution away from both low and high densities compared with the individual scores. The reader absolute differences [Fig. 3(b)] show a distribution with a relatively long tail, with many images having similar density scores but some with large differences.

In Fig. 4, we show how RMSE between the pair of readers differs per decile of VAS scores for averaged and individual reader scores. The averaged and individual scores are used to define the deciles (bins) and the differences per decile then calculated using those images in each bin. In Table 6, the appendix scores for MAE and median absolute for the same decile are shown.

The variability of reader scores means that some of the images will be labeled inaccurately. To gain some intuition about the scale of this effect, in Figure 5, we show the distribution of the differences for the reader estimates using individual reader estimates as the scores. We bin the images into their deciles using the individual labels and then plot the distribution of the differences for each decile. The differences shown are between an individual VAS score and its pair, therefore we see skewed distributions.

In Sec. 4.2, we presented a simulation method for producing quantitative estimates of the errors for a perfect set of predictions. In Table 2, we show error measures between the modeled density scores and the reader estimates for correlation, RMSE, MAE, and the MedAE for the entire dataset (all data) and for the test set. The ranges are by adjustments of $\sigma_{\max}$ (see Sec. 4.2 and Algorithm 6). There are further plots and analysis of these results in the appendices.

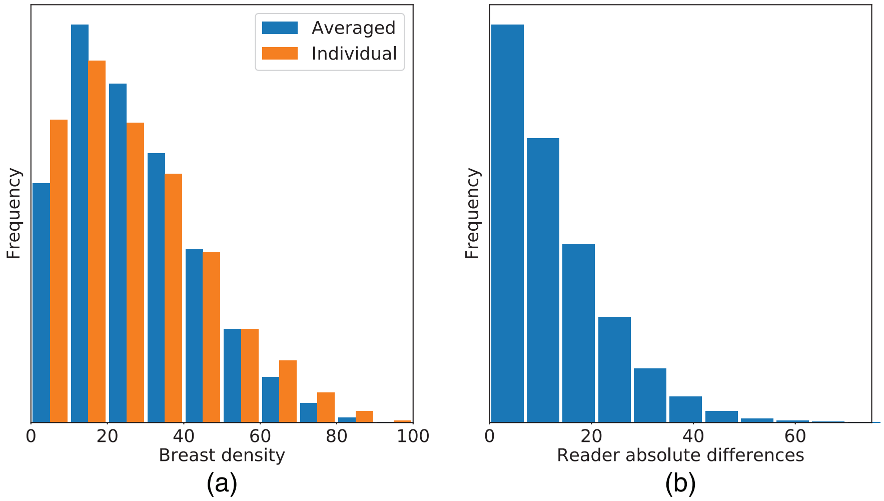


Fig. 3 (a) Distributions of the averaged and individual VAS estimates and (b) distribution of the absolute differences between reader scores.

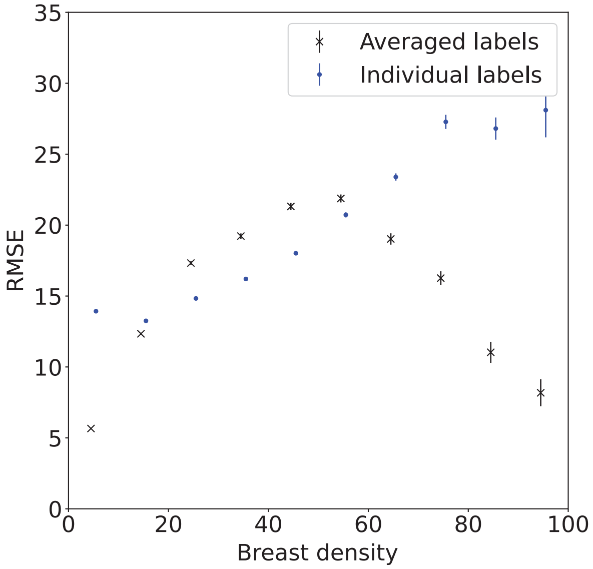


Fig. 4 The RMSE per decile with the deciles determined either with averaged (crosses) or individual (dots) labels. The errorbars are at the 95% level estimated by bootstrapping with 1000 repeats.

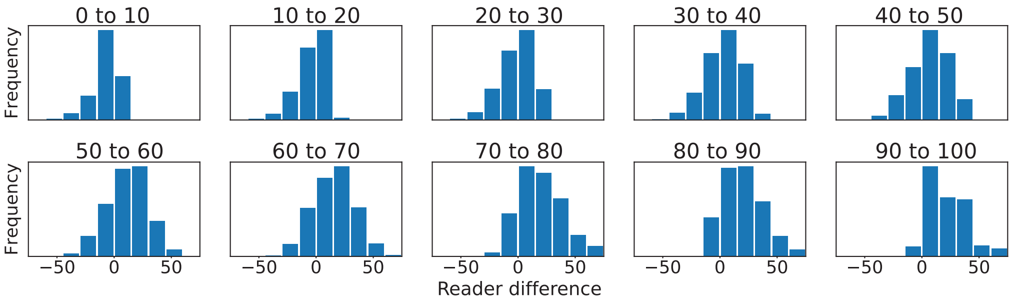


Fig. 5 The distribution of the differences for each decile, considering the individual reader estimates. The decile is shown at the top, the reader difference axis is the same for all plots while the frequency varies depending on the number of images in each decile. The lower density scores tend to have tighter distributions with fewer large differences in label annotation.

Table 2 Expected metrics per image, for a model predicting the true VAS values if the assumptions we specify are correct. This can be seen as an estimate of a set of metrics we would find if we produced a highly accurate model. The range is found by matching the metrics of the modeled pair of reader scores using $\sigma_{\max}$ for the four metrics. The best score (high correlation and low errors) is found when matching RMSE and the worst when matching the correlation.

	Correlation	RMSE	MAE	MedAE
All data				
CC	0.865 to 0.891	8.30 to 9.31	6.29 to 7.01	4.77 to 5.28
MLO	0.879 to 0.899	7.96 to 8.76	6.07 to 6.65	4.66 to 5.08
Test set				
CC	0.869 to 0.892	7.98 to 8.90	5.96 to 6.60	4.41 to 4.88
MLO	0.876 to 0.899	7.75 to 8.67	5.82 to 6.50	4.37 to 4.87

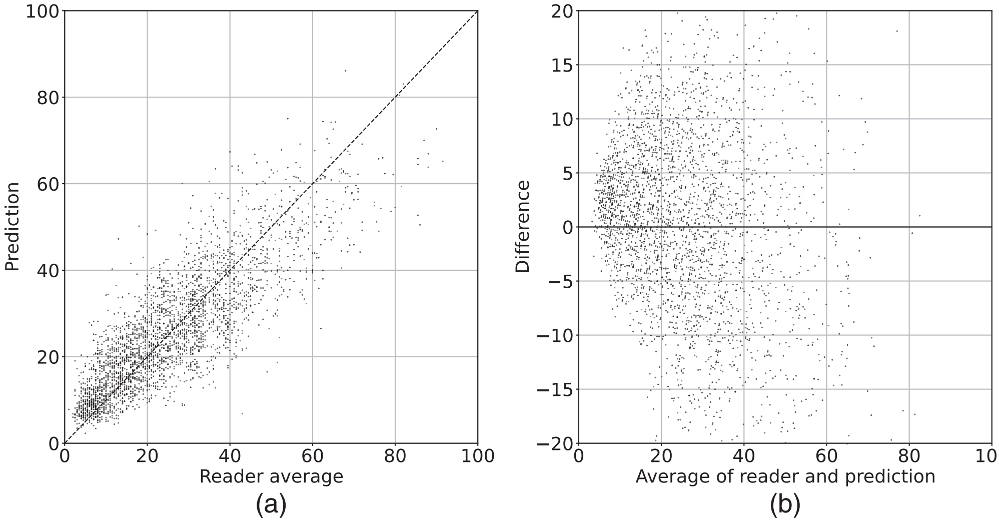


Fig. 6 Density prediction of the CC images by the ensemble model compared to the averaged reader image estimates. (a) Direct comparison between the ensemble model prediction and the reader average score. The dashed line represents equal predictions and reader average scores. (b) Bland–Altman plot of the difference between the ensemble model prediction and the reader average score against the average of the two.

5.2 *Model Predictions*

In Fig. 6, we show plots of our ensemble CC image density estimates against the reader average. Fig. 6(a) is a direct comparison of prediction against reader average, and Fig. 6(b) is a Bland–Altman plot of the difference between prediction and reader averages against the average of the two. Plots per woman are shown in the appendix in Fig. 14 and show a similar pattern, with a smaller variation. These plots allow for some intuition about the quality of the predictions compared to labels. They can also be compared to plots of the individual reader scores against one another as well as the modeled scores, all in the appendix (Fig. 11). These plots also show a considerable similarity to those of the modeled optimal plots in the appendix (Fig. 12).

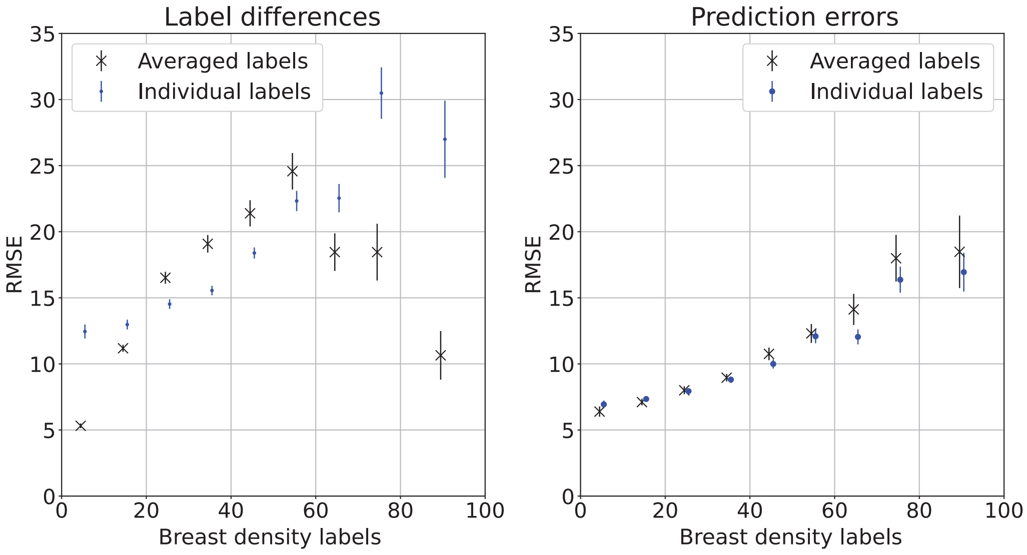
In Table 3, we show the four metrics produced by comparing our predictions against the averaged labels for the CC images. The equivalent results for the MLO images are in Table 7 in Appendix B. All the results are shown against the averaged reader scores of the test set. The label columns refers to which label was used when training the models on the training data. The general trend is for the DenseNet model to perform better than ResNet and for the MLPs to outperform the linear regression. The ensemble predictor slightly outperforms the individual models in most of the metrics.

Figure 7 (a) shows differences in labels per decile and (b) shows prediction errors per decile. The label differences are slightly different than Fig. 4 (but there are no differences in trends), as those were on the entire dataset and these on the test set. The images are binned either by their averaged labels (crosses) or individual labels (dots). The errorbars are the 95% confidence intervals found by performing 1000 sets of bootstrapping. The prediction errors are taken as the difference between the ensemble predictions and averaged labels, the individual labels are used to bin the images, not to estimate the accuracy of the model. The key point is that while the prediction errors do increase with breast density, the differences between the pairs of readers also do. At higher density, the models are both trained and compared to more variable labels.

In Fig. 8, we show the results of cancer risk predictions for the same 16 model variants and ensemble predictor as in Table 3. The odds ratios (ORs) are in comparison to the first quintile. The most relevant is $Q5$, which shows the OR of the highest density women compared with the lowest density women. All the model predictions show a substantial OR between the first and fifth quintile, with no differences outside of the uncertainty bounds. These are also comparable to those of the averaged reader VAS scores with an OR of around 4.5.[4]

Table 3 Comparison metrics between our models and the average labeled data for the CC images.

Model	Label	Obj. func.	Corr	RMSE	MAE	Median
ResNet	Average	LinReg	0.80 ± 0.01	9.50 ± 0.21	7.39 ± 0.16	5.97 ± 0.20
		L_1	0.82 ± 0.01	9.34 ± 0.22	7.08 ± 0.16	5.48 ± 0.18
		$L_{1\text{Smooth}}$	0.82 ± 0.01	9.30 ± 0.22	7.05 ± 0.16	5.48 ± 0.18
		MSE	0.82 ± 0.01	9.20 ± 0.21	7.10 ± 0.16	5.69 ± 0.16
	Individual	LinReg	0.80 ± 0.01	9.49 ± 0.21	7.38 ± 0.16	6.00 ± 0.20
		L_1	0.82 ± 0.01	9.37 ± 0.21	7.04 ± 0.16	5.34 ± 0.19
		$L_{1\text{Smooth}}$	0.82 ± 0.01	9.38 ± 0.22	7.06 ± 0.16	5.28 ± 0.20
		MSE	0.82 ± 0.01	9.13 ± 0.21	7.00 ± 0.15	5.52 ± 0.18
DenseNet	Average	LinReg	0.82 ± 0.01	9.12 ± 0.21	7.08 ± 0.15	5.72 ± 0.16
		L_1	0.83 ± 0.01	9.03 ± 0.22	6.87 ± 0.15	5.38 ± 0.22
		$L_{1\text{Smooth}}$	0.83 ± 0.01	9.04 ± 0.21	6.88 ± 0.15	5.38 ± 0.19
		MSE	0.83 ± 0.01	8.99 ± 0.21	6.94 ± 0.15	5.56 ± 0.18
	Individual	LinReg	0.82 ± 0.01	9.12 ± 0.20	7.08 ± 0.15	5.71 ± 0.16
		L_1	0.83 ± 0.01	9.05 ± 0.22	6.77 ± 0.16	5.19 ± 0.2
		$L_{1\text{Smooth}}$	0.83 ± 0.01	9.05 ± 0.23	6.77 ± 0.16	5.11 ± 0.18
		MSE	0.83 ± 0.01	9.01 ± 0.21	6.88 ± 0.15	5.39 ± 0.18
Ensemble		L_1	0.84 ± 0.01	8.79 ± 0.21	6.66 ± 0.15	5.17 ± 0.17



(a)

(b)

Fig. 7 The changes in RMSE per decile for the labels and ensemble predictions, with deciles defined by averaged and individual labels. The error bars are the 95% confidence intervals found by bootstrapping 1000 times. (a) Reader differences as measured by RMSE with data placed into deciles from averaged labels (black crosses) and individual labels (blue dots), respectively. (b) Ensemble prediction RMSE per decile with data placed into the decile using the average label (black crosses) and individual labels (blue dots).

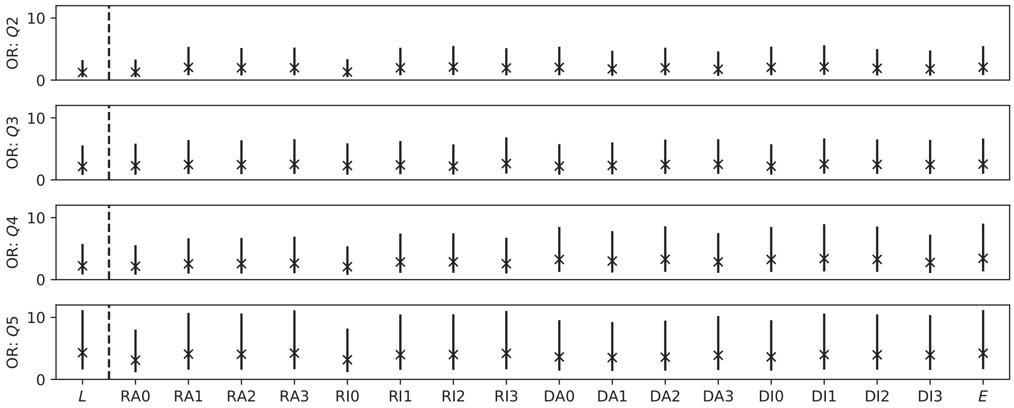


Fig. 8 Plots of the ORs calculated using the model predicted scores with uncertainties found via bootstrapping. OR: $Q2$, $Q3$, $Q4$, $Q5$ shows the ORs for the second, third, fourth, and fifth quintile, respectively. L shows the ORs found from the labels. R/D represent ResNet or DenseNet, A represents the model trained on the averaged labels, 0, 1, 2, 3 represent linear regression, the L_1, $L_{1\text{Smooth}}$, and MSE objective functions, respectively. For example, RI_2 is the ResNet feature extractor trained on the individual labels with a $L_{1\text{Smooth}}$ objective function. E is the ensemble predictor.

5.3 *Model Comparisons*

We make a comparison between models trained on averaged and individual labels and models with different feature extractors (ResNet and DenseNet) and models with density mapping from linear regression and MLPs. In addition, we compare our predictions to those of a variant of a previous method trained on this data, pVAS.[18]

In Table 4 (comparison 1), we show metrics for the differences in prediction between models trained on averaged and individual labels for the DenseNet MLP method with L1 objective function. Plots relating to these results are in the appendix (Fig. 15). The similarity in predictions is high, which might not be expected considering the substantial differences in some reader scores, implying there is a strong density signal in the data.

Metrics for the comparison between model predictions made using the feature extractors ResNet and DenseNet, both with MLPs trained on the feature vectors, are shown in Table 4 (comparison 2) (with related Fig. 16). Both models are trained on individual labels and with the L1 objective function. The differences are larger than the pair of predictions trained on the individual and averaged labels, but they also appear to be random in character; there does not appear to be much in the way of systematic variation.

We show the results for a comparison between model predictions made using the linear regression and MLP density estimators, both using the DenseNet feature extractor

Table 4 Comparison metrics between the predictions made between our density mapping models. We show standard metrics with the addition of the RMSE per quintile (labeled $Q1$ to $Q5$), with quintiles defined as the average of the two predictions. AvQ is the mean of the RMSEs per quintile. Comparison 1 is between models trained on averaged and individual labels. Comparison 2 is between MLPs trained on DenseNet and ResNet extracted features. Comparison 3 is between an MLP and linear regression model trained on the DenseNet extracted features. Comparison 4 is between our ensemble model trained on both individual and averaged labels and pVAS from a previous study.[18] The test set for this comparison is different than in the rest of this paper.

Name	Corr	RMSE	MAE	MedAE	AvQ	$Q1$	$Q2$	$Q3$	$Q4$	$Q5$
Comparison 1	0.996	1.80	1.46	1.28	2.33	1.90	1.70	1.69	2.29	4.05
Comparison 2	0.945	4.69	3.55	2.72	6.54	3.27	4.9	6.25	8.44	9.82
Comparison 3	0.983	2.60	2.04	1.68	3.86	2.66	2.32	2.97	4.35	7.02
Comparison 4	0.924	5.74	4.40	3.53	7.89	4.63	5.80	7.00	8.85	13.19

Table 5 Metrics of predictions made by our ensemble method and predictions made by pVAS[18] per image. The results are all the images, both CC and MLO. These results are from the second test set partition, discussed in Sec. 2. Uncertainties are at the 95% confidence level found via bootstrapping with 1000 repeats.

	Correlation	RMSE	MAE	MedAE
pVAS versus our method	0.924 ± 0.001	5.74 ± 0.04	4.40 ± 0.03	3.53 ± 0.03
Our method versus labels	0.839 ± 0.003	9.00 ± 0.05	6.92 ± 0.04	5.49 ± 0.05
pVAS versus labels	0.809 ± 0.003	9.78 ± 0.05	7.66 ± 0.04	6.32 ± 0.05

[Table 4 (comparison 3)] (Fig. 17 in the appendix). These are for average labels and L1 objective function for the MLP. They show some systematic differences in predictions, likely due to the linear regression underfitting to the data.

The final comparison we make is between previous work[18] labeled pVAS and the final ensemble predictions produced in this paper. Table 5 shows metrics for the pVAS estimates and our ensemble estimates. We also show these results as comparison 4 in Table 4. Plots are shown in Fig. 18. The test set is different to that for the results shown in the rest of the paper so that it coincides with the test set used by the pVAS model. Our model outperforms the pVAS model even though it is never trained end-to-end, which demonstrates the power of the representation formed by the pretrained models.

6 Discussion

Breast density, as measured on a VAS by experienced radiologists, shows a strong realtionship with breast cancer risk.[4] We designed and implemented a framework based on pretrained deep networks to find a mapping between a mammogram and its associated breast density. The core part of the system is the feature extractor, which is taken from deep networks trained on ImageNet.[9]

Reader density scores are known to be variable.[30,31] One consequence is that we are training our models on data with interreader variability, and the second is that if we were to produce a highly accurate predictor, when we compared it to the labels, it would look like it produced inaccurate predictions. We analyzed the labels, in Sec. 5.1, to gain insight into how this variability affects our conclusions.

An important consideration is how the variability changes with the density scores. The average of the two reader estimates (Fig. 4 and Table 6) appear to show significantly smaller differences between the readers at both lower and higher densities with a maximum difference in the middle. For example, in the 50 to 60 decile, we have a mean difference of 18.0 and in the 0 to 10 decile a mean difference of 4.5. This might imply that the readers are more consistent at the two extremes and produce more variable results in the middle. However, if we look at the individual reader estimates, the average differences between the readers tend to increase with increasing density across the deciles.

The reason the differences appear to peak in the middle of the density distribution and are lower at the two ends for the averaged reader estimates is likely to be at least partially a statistical artefact. If we consider the first decile (1 to 10) and take an example of an averaged reader score of 5.0, the maximum difference possible for this average result is one reader marking 1 and the other marking 9, with a difference of 8. Conversely, the maximum possible difference for an average value of 50.0 is one reader marking 1 and the other marking 99, giving a difference of 98. Part of the reason that the average differences appear to be lower in the low and high densities is because, when considering the average scores, the results with the larger differences would not result in averages with low or high densities. There might be more label consistency at the low and higher ends of the density distribution, but it is hard to separate that out from this statistical artefact.

Table 6 The fraction of density scores (number fraction) and the average (both mean and median) absolute differences between reader estimates per decile. These are all shown for the deciles defined either by averaged reader scores (averaged) or individual reader scores (individual).

	Number fraction		Mean differences		Median differences	
Decile	Averaged	Individual	Averaged	Individual	Averaged	Individual
1 to 10	0.151	0.197	4.5	9.3	4.0	5.0
11 to 20	0.251	0.228	9.9	9.7	8.0	7.0
21 to 30	0.214	0.189	13.9	11.7	12.0	10.0
31 to 40	0.170	0.157	15.1	13.1	13.0	12.0
41 to 50	0.109	0.108	16.8	14.8	14.0	13.0
51 to 60	0.059	0.059	18.0	17.2	16.0	16.0
61 to 70	0.029	0.039	15.1	19.1	12.0	17.0
71 to 80	0.012	0.019	13.0	21.5	11.0	18.0
81 to 90	0.003	0.007	8.7	21.2	7.0	18.0
91 to 100	0.000	0.001	7.2	22.6	6.0	20.0
All	1	1	12.2	12.2	9.0	9.0

As the performance of our models is measured based upon the quality of the labels, if the label variability increases with higher density, we would expect to see an apparent fall in model performance due to the increasing variability, rather than the failure of the model. In Fig. 7, this is the effect we do see, an apparent reduction in performance of the model at higher density scores. It is likely that our models are more inaccurate at higher densities as we also see a large variability when comparing model predictions to each other at those higher densities (Table 4), but it is a smaller effect than if we simply measure the differences in predictions to the labels. These effects are further discussed in the appendix.

The errors between our density predictions (Table 3) and the labels fall close to the range of the errors we see with a modeled optimal estimator (Table 2), which can perfectly predict the density of an image. There is still space for improvement in the quality of the predictions, which may be best achieved by adapting the feature extractors to better represent mammography data either through fine-tuning or with other approaches. However, there is a limit above which it will become difficult to assess if the improved models are able to perform better as they approach the metrics shown for the simulated optimal model.

We compared our models both to a previous method, pVAS,[18] and to variants of our method. We find considerable similarities between all of our models and to pVAS. This sort of similarity implies that we are finding true structure in the data. There is greater divergence of prediction at higher VAS scores, although it is still fairly small considering the uncertainty in the labels and the small amount of data at those higher densities.

There is little difference in results when training on averaged or individual labels (Table 3), and this is an interesting finding considering the low consistency of the pairs of reader estimates. We might expect there to be a significant improvement when training on the averaged labels because they might be expected to reduce some of the noise in data. The fact that we do not see this suggests that the models are able to effectively extract a true density-related signal. The ensemble predictor produces a small improvement in performance giving slightly lower errors and higher correlations than the individual density mapping models.

The linear regression produced reasonable accuracy implying that the feature extractors are producing a mapping to a fairly linear subspace; however, there is clearly some nonlinearity required as the MLPs do perform better (Table 3). DenseNet performs better than ResNet, which may be due to the fact that the DenseNet version performs better on ImageNet than the specific ResNet version used (see Appendix A for details). Alternatively, it may be due to the larger subspace size of the DenseNet model.

The model predictors (Fig. 8) are all comparable with the VAS labels in terms of risk prediction. Although there is variation between the models in terms of the ORs found, these are all within the uncertainty bounds and we cannot make any strong statements about the quality of the predictive models compared with one another.

This usage of transfer learning allows us to leverage the long training times and large computational power of other research groups. There is a debate over whether transfer learning or learning from scratch are more appropriate.[10] One answer is that transfer learning approaches like the one we have demonstrated here is far quicker to implement than the requirement of designing and then training networks from scratch. If results from a transfer learning approach are poor, then it may be necessary to pursue other approaches.

A potential issue with a transfer learning approach in medical imaging is that the models used tend to be trained on unrelated images. It might be expected that this would result in features being extracted that are unrelated to the medical domain. However, we have shown that reasonably accurate results can be obtained using these features, even when using only a linear mapping. Perhaps with some domain adaptation, these results would improve further.

We reduced the size of the images to 224 × 224, both to match the size of images the models were originally trained on and to reduce computational requirements. If we can utilize smaller images and achieve good results, it is a significant advantage in terms of the computer power and time required. Saving on computational time is a major advantage, one benefit is that it allows researchers and groups without access to large computing resources to perform analysis. It also means that multiple different runs of algorithms can be performed, and multiple other facets of the data can be investigated.

We also did not preserve the aspect ratios of the images, do anything about the fact that there are three different image sizes that are then distorted by differing amounts, or crop the images to focus on the breast. Yet we still achieve accurate predictions, although further research is required to investigate whether results could be improved by correcting these issues, or if they do not adversely effect the quality of the models. We also do not utilize the three-channel nature of the transferred network, something that might enable more predictive capacity to be extracted from the pretrained models.

Overall, our method produces predictive accuracy close to the maximum that can be assessed with the ground truth labels we have access to. We do so using a method that requires modest computer resources both in terms of memory and time. In particular, once the feature extractors have been run, the computational requirements of the density predictors are very low, especially for the linear regression. This can enable both much faster training and also training on a small dataset or subsets of the larger dataset. We also demonstrate the, perhaps, surprising ability of deep learning models trained on a different image domain to produce good performance on this medical dataset.

There is a direct benefit of accurate prediction of breast density, and that it can be used to produce information to medical practitioners or in research projects where there is no access to radiologists. Another potential value is as an input to other automated models. Density provides considerable information that might enable models that are trying to predict cancers, perform segmentation, or other tasks that might be able to utilize to produce improved performance on their specific problem.

7 Conclusions

In this paper, we have demonstrated that using a transfer learning approach with deep features results in accurate breast density predictions. This approach is computationally fast and cheap, which can enable more analysis to be done and smaller datasets to be used. However, the deep feature models were trained on a nonmedical dataset, which would imply that the features extracted could be considerably improved. If we can train deep models on medical imaging data, then we might expect to see improvements in performance when those models are used as a transfer learning model across a wide range of medical imaging applications.

We have demonstrated the issues associated with data where readers are variable in predictions. Finding ways to reduce the variability of the labels would enable us to train on more reliable data and to better assess which models are performing better than others. If we could improve the quality of the labels, it would also mean that we could more systematically investigate what measures could improve performance.

8 Appendix A: Model Details

8.1 *Feature Extraction*

We use two networks for the feature extraction: ResNet[22] and DenseNet,[23] specifically we use the "ResNet18" and "DenseNet161" architectures downloaded with pretrained weights from the PyTorch[24] Torchvision repositories. For both, we replace the final classification layer ("fc" in Resnet and "classifier" in DenseNet) with an identify matrix. We ran the images through the two networks with a NVIDIA Quadro P400 2 GB GPU, a process that takes around 10 to 50 s per 100 images for ResNet and DenseNet, respectively. In total, we have around 160,000 images, and the total run time is around 4.5 to 22 h for ResNet and DenseNet, respectively. Once this stage is completed, there is no need to repeat as the feature vectors remain the same and can be used in any required permutation.

8.2 *MLP Density Mapping*

Our MLP is small and simple with p (512 for ResNet and 2208 for DenseNet) input neurons, followed by 200 neurons, and a rectified linear unit (ReLU),[32] then 300 neurons and a ReLU to the output. We trained them using the same NVIDIA Quadro P400 GPU on PyTorch[24] with around 200 epochs, the Adam optimizer,[33] and a starting learning rate of 1×10^{-5} and one reduction in learning rate half way through to end at 2×10^{-6}. Due to the small and simple nature of the MLP, multiple other training approaches would suffice, this one was found through trial-and-error checking that the training error was reducing as we would expect. We use three objective functions: L_1, $L_{1\text{Smooth}}$, and the MSE, all available as basic functions in PyTorch.

8.3 *Ensemble Prediction*

The ensemble MLP is built using PyTorch[24] with two hidden layers of size $5p$ and $10p$, respectively, where p is the number of density predictions fed into the system. We apply ReLU to both hidden layers and trained using the Adam optimizer.

9 Appendix B: Extra Results

In Fig. 9, we show the distribution of the differences between individual label scores for the bins we use to make our simulated predictions. The bin centers are noted on the top of each plot.

9.1 *Label Analysis*

In Fig. 10, we graphically demonstrate the challenge with having high reader uncertainty together with a skewed distribution. We plot all the binned individual reader scores per decile along with the distribution of the differences from the center of the decile, these are the blue bars. We also plot the total number of individual labels above 80% as orange bars. In the bottom right two plots, we show the entire distributions (left of the bottom right pair) and a zoomed in version (bottom right plot). The potential number of images, which are falsely labeled as high density, may be comparable to the number of real high-density images. This makes it difficult to confidently assess how well our models are performing at high VAS scores, as we cannot assess whether a high VAS score is accurately labeled. If we were to perform oversampling or data augmentation on the VAS scores classed as high by the average reader estimate, we are likely

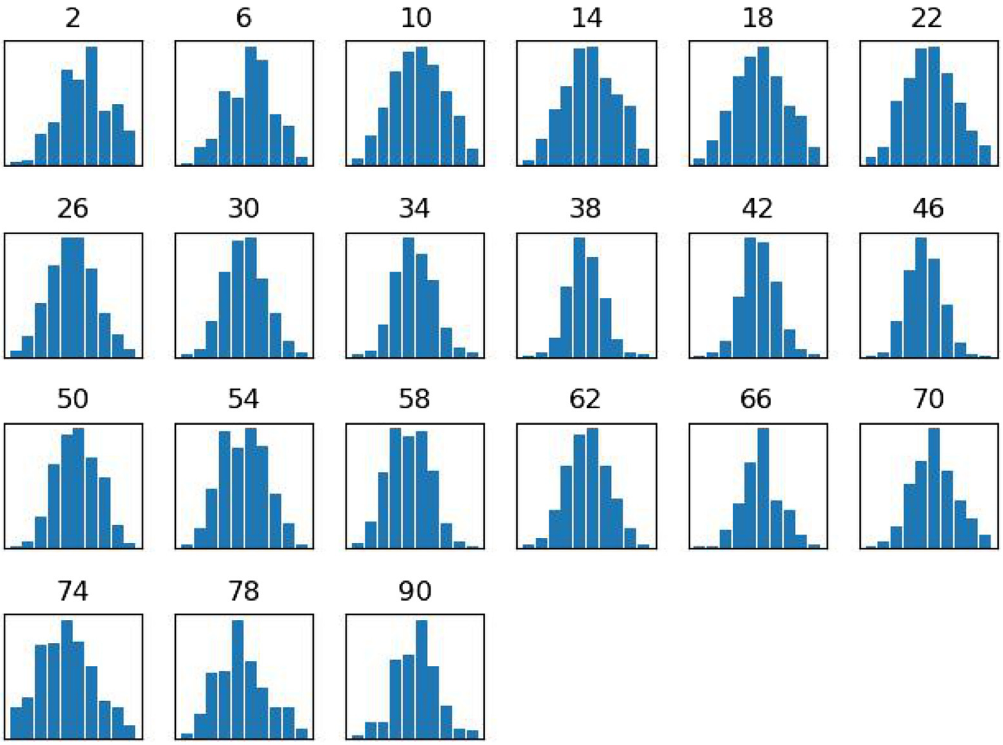


Fig. 9 Distribution of the differences within bins. Bin centers are noted on the top.

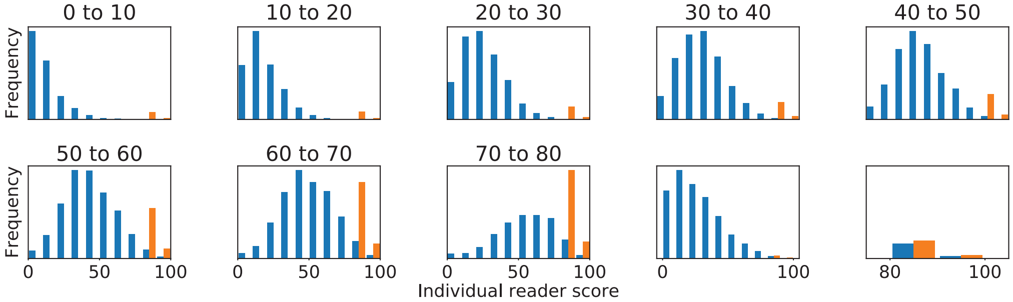


Fig. 10 For each of the first eight deciles, we show the distribution of the pair values as blue bars. For example, in the 1 to 10 decile range, we show the distribution of the reader values, which have a paired score between 1 and 10. We also show the number of reader averaged scores that are above 80. The bottom right two plots show the entire 0 to 80 distributions and a zoomed in version of the same results.

to oversample from the least reliable part of the distribution with the most falsely labeled data points and the lowest signal-to-noise ratio.

In Fig. 11, we show plots of the pairs of VAS labels for the modeled and real estimates, we show 3000 random pairs so the structure of the distribution is visible. The dotted line shows a perfect relationship between the pair of readers. Fig. 11(a) is the lower error (higher correlation) end of our modeled range, and Fig. 11(c) is the higher error (lower correlation) end of our modeled range. Fig. 11(b) show the real pairs of reader labels plotted next to one another, these points are very slightly perturbed so that individual points can be seen.

In Fig. 12, we show the average modeled optimal scores versus the average of the two modeled labels. Fig. 12(a) is a direct comparison and Fig. 12(b) is the Bland–Altman plot for the difference between the modeled optimal score and the modeled average reader scores versus the average of the optimal and modeled scores.

Previous work has shown that VAS density scores show a strong correlation to the risk of developing cancer.[4] In this paper, we have demonstrated that the variability of the reader

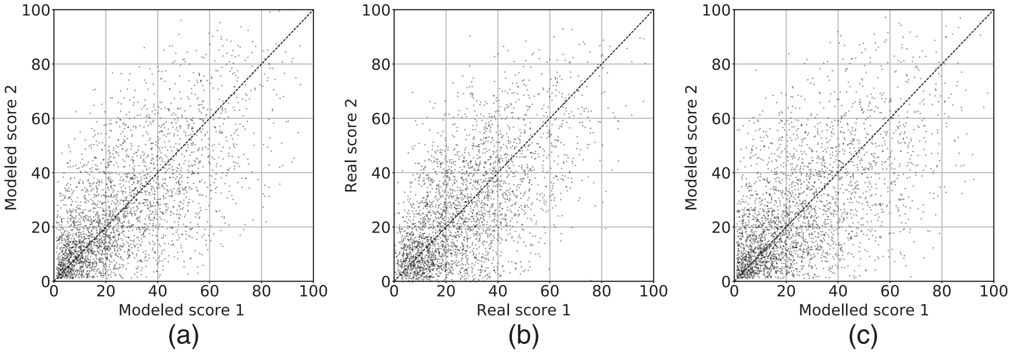


Fig. 11 Examples of the pairs of scores for the modeled and real data, the dotted line shows a perfect relationship between the pair of estimates. (a) The modeled pairs of scores for the results produced with the lowest errors. (b) The pairs of real reader scores. (c) The modeled pairs of scores produced with the highest error in the range.

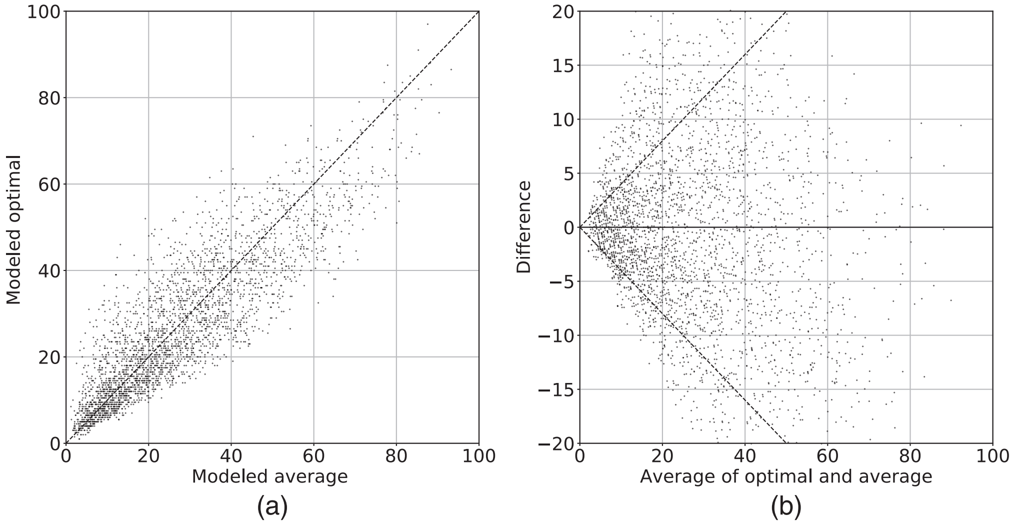


Fig. 12 Modeled optimal scores and modeled reader average scores. (a) Direct relationship between the modeled optimal scores versus the averaged modeled pairs of estimates. (b) Bland–Altman plots of the difference between the modeled optimal and modeled reader averaged scores versus the average of the modeled optimal and modeled reader average scores.

estimates means that we have to be cautious about the level of confidence we have in results derived from these labels. We therefore repeat the analysis for the Prior dataset from that previous work to calculate the ORs.[4,18] In addition, we provide an additional piece of analysis by perturbing the averaged scores. The purpose of doing so is to give some intuition about how much the ORs might change with small variations in the reader scores. We added a random amount to all the averaged scores by sampling from a Gaussian distribution with three different standard deviations of $\sigma = 1$, 2, 5. We resampled any scores that went above 100 or below 1, until we got a score within the correct range. We performed each set of perturbations five times. The perturbation we apply to the reader averaged scores is small when compared to the variability between the pairs of reader scores, which have an RMSE between readers of 16.2.

The results of the range of the ORs found are shown in Fig. 13. We show OR plots for the second ($Q2$), third ($Q3$), fourth ($Q4$), and fifth ($Q5$) quintiles. On the left side of the plots, labeled with Orig., is the nonperturbed ORs. The next five, labeled with $\sigma = 1$ show the five repeats when the perturbation is parameterized with a standard deviation of 1. The results for $\sigma = 2$ and $\sigma = 5$ are shown in the next two sections, separated by the dashed line. The crosses show the ORs and the error bars the upper and lower 95% confidence intervals found via bootstrapping.

We therefore consider two aspects of uncertainty here: via bootstrapping we see the uncertainty relating to the sampling of the data, and via the perturbations we see the uncertainty due to

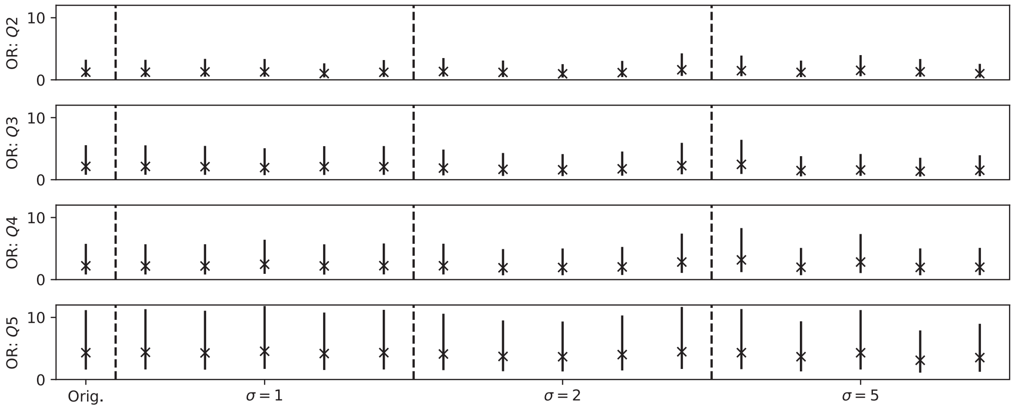


Fig. 13 Plots of the ORs calculated using the reader averaged scores with uncertainties found via bootstrapping. OR: *Q*2, *Q*3, *Q*4, *Q*5 shows the ORs for the second, third, fourth, and fifth quintile, respectively. Orig. shows the results with no perturbation, the five points around $\sigma = 1, 2, 5$ show the ORs when the reader averaged scores are perturbed by the standard deviation specified, with the multiple points being repeated applications of the perturbations across the dataset.

the unreliability of the labels due to reader variability. The uncertainty due to sampling (the bootstrap error bars) is large and show the wide range of possible ORs that could occur with a different sample. The perturbations alter both the OR and the uncertainty of it. We will see in the next section that the ORs found by the models are comparable with these ORs for the readers.

Although this is a positive result for our models, the problem is that the high level of uncertainty in the OR values means we cannot easily assess whether our models are performing better than one another. This means that we cannot bypass the metrics we considered earlier in this section to assess the quality of our models by looking directly at cancer risk, because the uncertainty involved is too large. From these perturbation results, we would not want to trust that one model is better than another without quite significant differences in estimates. However, what this also shows is that the VAS scores do produce a robust set of cancer risk prediction. In previous work, the other measures of density studied did not produce a ratio of above 3.[4] Therefore, although we cannot be confident that results in a fairly broad range are an improvement on other results, we can be reasonably confident in the overall ability of VAS to make good risk estimates in comparison to other density scores.

9.2 *Model Predictions*

In Table 7, we show the prediction results for the MLO images equivalent to Table 3.

Table 7 Comparison metrics between our models and the average labeled data for the MLO images.

Model	Label	Obj. func.	Corr	RMSE	AME	Median
ResNet	Average	LinReg	0.81 ± 0.01	9.49 ± 0.22	7.32 ± 0.17	5.87 ± 0.18
		L_1	0.82 ± 0.01	9.24 ± 0.22	7.02 ± 0.16	5.54 ± 0.19
		$L_{1\text{Smooth}}$	0.82 ± 0.01	9.35 ± 0.23	7.14 ± 0.16	5.69 ± 0.2
		MSE	0.83 ± 0.01	9.13 ± 0.21	7.04 ± 0.15	5.60 ± 0.16
	Individual	LinReg	0.81 ± 0.01	9.48 ± 0.22	7.31 ± 0.16	5.88 ± 0.18
		L_1	0.82 ± 0.01	9.25 ± 0.23	6.94 ± 0.17	5.34 ± 0.2
		$L_{1\text{Smooth}}$	0.82 ± 0.01	9.25 ± 0.23	6.95 ± 0.16	5.38 ± 0.17
		MSE	0.83 ± 0.01	9.02 ± 0.22	6.89 ± 0.15	5.45 ± 0.19

Table 7 (*Continued*).

Model	Label	Obj. func.	Corr	RMSE	AME	Median
DenseNet	Average	LinReg	0.83 ± 0.01	9.03 ± 0.22	6.96 ± 0.15	5.66 ± 0.17
		L_1	0.83 ± 0.01	8.94 ± 0.22	6.75 ± 0.16	5.21 ± 0.16
		$L_{1\text{Smooth}}$	0.83 ± 0.01	8.94 ± 0.23	6.75 ± 0.16	5.22 ± 0.17
		MSE	0.84 ± 0.01	8.82 ± 0.22	6.78 ± 0.15	5.33 ± 0.17
	Individual	LinReg	0.83 ± 0.01	9.04 ± 0.21	6.97 ± 0.15	5.66 ± 0.16
		L_1	0.83 ± 0.01	9.01 ± 0.24	6.75 ± 0.17	5.10 ± 0.18
		$L_{1\text{Smooth}}$	0.83 ± 0.01	9.03 ± 0.23	6.75 ± 0.16	5.12 ± 0.15
		MSE	0.83 ± 0.01	8.92 ± 0.21	6.77 ± 0.15	5.22 ± 0.15
Ensemble		L_1	0.84 ± 0.01	8.68 ± 0.21	6.59 ± 0.15	5.12 ± 0.17

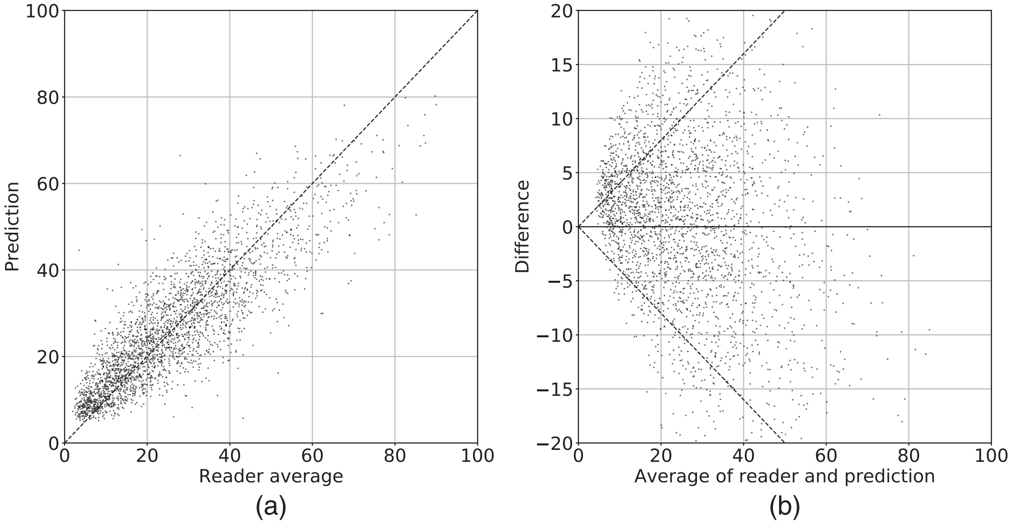
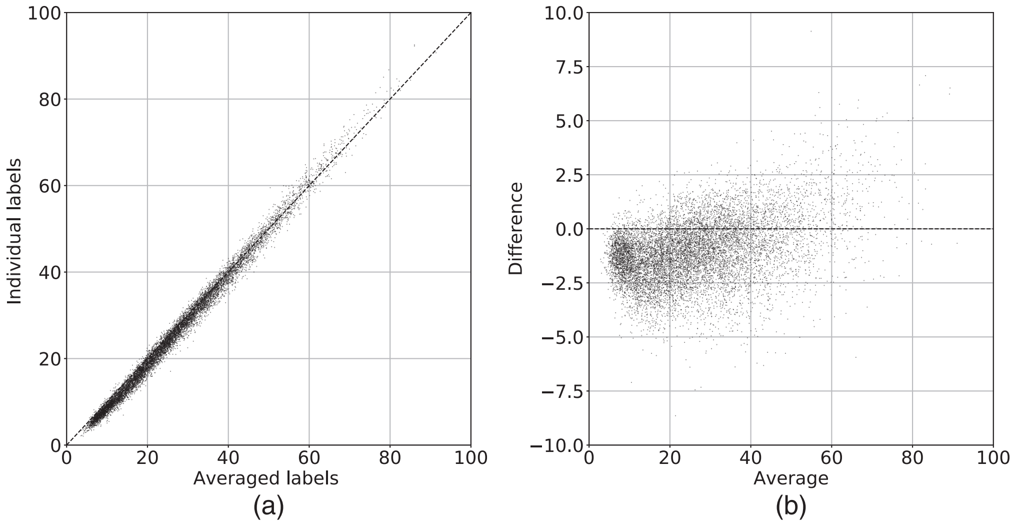


Fig. 14 Density prediction per woman by the ensemble model compared to the averaged reader woman estimates. (a) Direct comparison between the ensemble model prediction and the reader average score. (b) Bland–Altman plots of the difference between the ensemble model prediction and the reader average score against the average of the two.

Plots of the final ensemble predictions versus labels per woman are displayed in Fig. 14. There are 2682 women in the test set who have all labels and all predictions intact, 15 women are missing labels or images and are removed. We see the same general pattern as the per-image plots of Fig. 6.

9.3 *Model Comparisons*

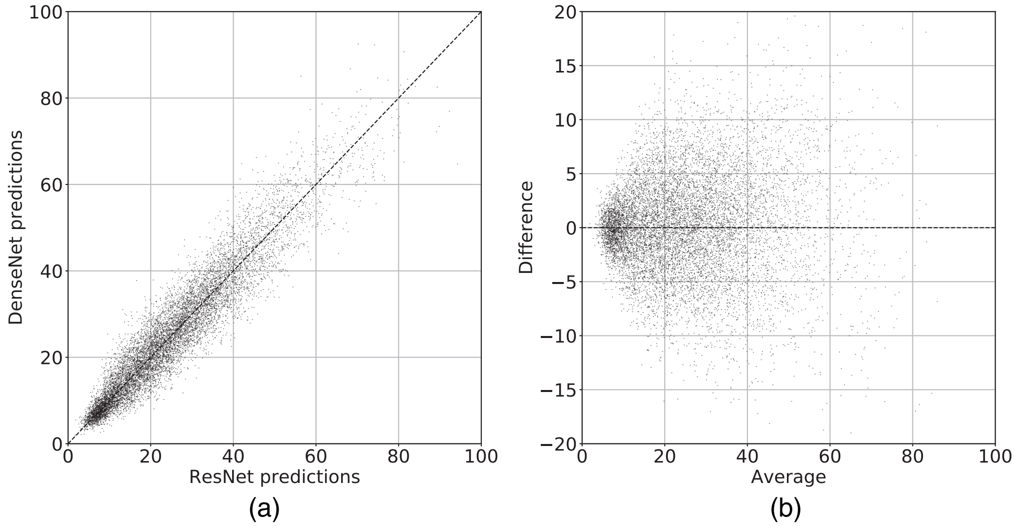
In Fig. 15 we show a comparison of predictions made by training on the individual labels compared to averaged labels, both with a DenseNet feature vector and an MLP trained using the L_1 objective function. Metrics related to these plots are shown as in Table 4 (comparison 1).



(a)

(b)

Fig. 15 Plots showing the similarities and differences between image predictions made by models trained on individual labels and averaged labels. (a) Predictions from training on individual labels against predictions from training on the averaged labels. (b) Bland–Altman plots of the same data.



(a)

(b)

Fig. 16 Differences in image prediction between MLPs trained on the DenseNet and ResNet feature vectors. (a) Predictions made using DenseNet against predictions made using ResNet. (b) Bland–Altman plots of the difference between predictions made by DenseNet and ResNet versus their average.

In Fig. 16 we show a comparison of predictions made using MLPs with the L_1 objective function on the DenseNet and ResNet feature vectors, both trained on individual labels. Metrics related to these plots are presented in Table 4 (comparison 2).

In Fig. 17 we show a comparison of predictions made using an MLP (with L_1 objective function) against linear regression, both on feature vectors from the DenseNet model, with averaged labels. Related metrics are in Table 4 (comparison 3).

In Fig. 18 we show predictions made by our ensemble predictor compared to the pVAS model. Related metrics are in Table 4 (comparison 4).

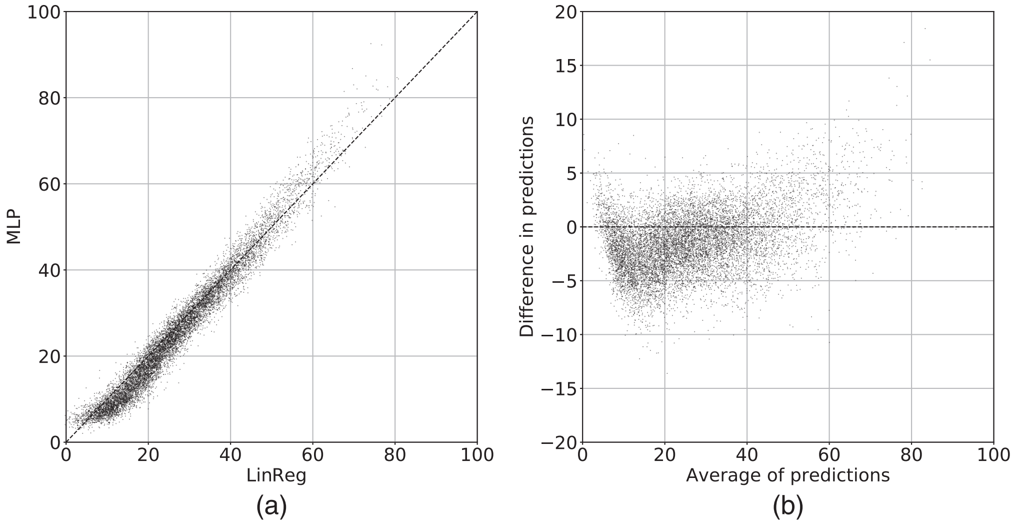


Fig. 17 Image predictions made using DenseNet extracted features with an MLP and linear regression as the density mapping functions. (a) Direct comparison. (b) Bland–Altman plots of the difference between predictions made by MLP and linear regression versus their average.

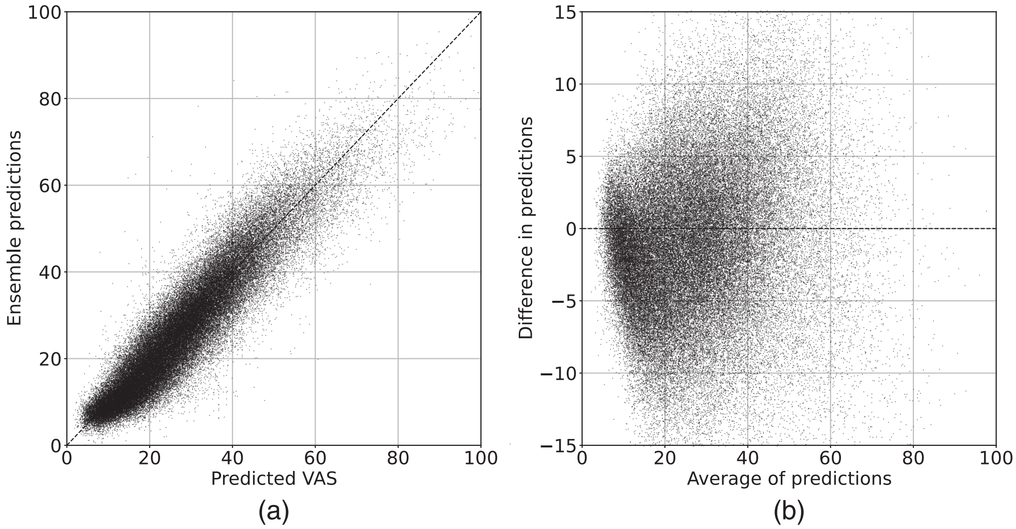


Fig. 18 A comparison between the ensemble predictions using our methods and pVAS.[18] (a) Direct comparison and (b) Bland–Altman plot.

Disclosures

The authors declare no conflicts of interest. Ethics approval for the PROCAS study was through the North Manchester Research Ethics Committee (09/H1008/81). Informed consent was obtained from all participants on entry to the PROCAS study.

Acknowledgments

Dafydd Gareth Evans, Elaine Harkness, and Susan M. Astley are supported by the National Institute for Health Research (NIHR) Manchester Biomedical Research Center (Grant No. IS-BRC-1215-20007). Steven Squires was supported by CRUK AI-informed screening (Grant No. A29024).

References

1. N. F. Boyd et al., “Breast tissue composition and susceptibility to breast cancer,” *J. Natl. Cancer Inst.* **102**(16), 1224–1237 (2010).
2. C. Huo et al., “Mammographic density-a review on the current understanding of its association with breast cancer,” *Breast Cancer Res. Treat.* **144**(3), 479–502 (2014).
3. V. A. McCormack and I. dos Santos Silva, “Breast density and parenchymal patterns as markers of breast cancer risk: a meta-analysis,” *Cancer Epidemiol. Prevent. Biomark.* **15**(6), 1159–1169 (2006).
4. S. M. Astley et al., “A comparison of five methods of measuring mammographic density: a case-control study,” *Breast Cancer Res.* **20**(1), 10 (2018).
5. A. R. Brentnall et al., “Long-term accuracy of breast cancer risk assessment combining classic risk factors and breast density,” *JAMA Oncol.* **4**(9), e180174–e180174 (2018).
6. J. Cuzick et al., “Tamoxifen-induced reduction in mammographic density and breast cancer risk reduction: a nested case–control study,” *J. Natl. Cancer Inst.* **103**(9), 744–752 (2011).
7. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Adv. in Neural Inf. Process. Syst.*, pp. 1097–1105 (2012).
8. Z. C. Lipton, “The mythos of model interpretability,” *Queue* **16**(3), 31–57 (2018).
9. J. Deng et al., “ImageNet: a large-scale hierarchical image database,” in *IEEE Conf. Comput. Vis. and Pattern Recognit.*, IEEE, pp. 248–255 (2009).
10. G. Litjens et al., “A survey on deep learning in medical image analysis,” *Med. Image Anal.* **42**, 60–88 (2017).
11. D. Shen, G. Wu, and H.-I. Suk, “Deep learning in medical image analysis,” *Annu. Rev. Biomed. Eng.* **19**, 221–248 (2017).
12. P. Fonseca et al., “Automatic breast density classification using a convolutional neural network architecture search procedure,” *Proc. SPIE* **9414**, 941428 (2015).
13. M. Kallenberg et al., “Unsupervised deep learning applied to breast density segmentation and mammographic risk scoring,” *IEEE Trans. Med. Imaging* **35**(5), 1322–1331 (2016).
14. B. M. Keller et al., “Estimation of breast percent density in raw and processed full field digital mammography images via adaptive fuzzy c-means clustering and support vector machine segmentation,” *Med. Phys.* **39**(8), 4903–4917 (2012).
15. C. D. Lehman et al., “Mammographic breast density assessment using deep learning: clinical implementation,” *Radiology* **290**(1), 52–58 (2019).
16. T. P. Matthews et al., “A multisite study of a breast density deep learning model for full-field digital mammography and synthetic mammography,” *Radiol.: Artif. Intell.* **3**(1), e200015 (2020).
17. O. H. Maghsoudi et al., “Deep-libra: an artificial-intelligence method for robust quantification of breast density with independent validation in breast cancer risk assessment,” *Med. Image Anal.* **73**, 102138 (2021).
18. G. V. Ionescu et al., “Prediction of reader estimates of mammographic density using convolutional neural networks,” *J. Med. Imaging* **6**(3), 031405 (2019).
19. S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.* **22**(10), 1345–1359 (2009).
20. S. Squires et al., “Automatic density prediction in low dose mammography,” *Proc. SPIE* **11513**, 115131D (2020).
21. D. G. R. Evans et al., “Assessing individual breast cancer risk within the uk national health service breast screening program: a new paradigm for cancer prevention,” *Cancer Prevent. Res.* **5**(7), 943–951 (2012).
22. K. He et al., “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. and Pattern Recognit.*, pp. 770–778 (2016).
23. G. Huang et al., “Densely connected convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. and Pattern Recognit.*, pp. 4700–4708 (2017).
24. A. Paszke et al., “Automatic differentiation in pytorch,” (2017).
25. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv:1409.1556 (2014).
26. C. Szegedy et al., “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. and Pattern Recognit.*, pp. 1–9 (2015).

27. K. Hornik et al., "Multilayer feedforward networks are universal approximators," *Neural Netw.* **2**(5), 359–366 (1989).
28. L. Rokach, "Ensemble-based classifiers," *Artif. Intell. Rev.* **33**(1–2), 1–39 (2010).
29. D. Opitz and R. Maclin, "Popular ensemble methods: an empirical study," *J. Artif. Intell. Res.* **11**, 169–198 (1999).
30. B. L. Sprague et al., "Variation in mammographic breast density assessments among radiologists in clinical practice: a multicenter observational study," *Ann. Internal Med.* **165**(7), 457–464 (2016).
31. M. Sperrin et al., "Correcting for rater bias in scores on a continuous scale, with application to breast density," *Stat. Med.* **32**(26), 4666–4678 (2013).
32. V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML-10)*, pp. 807–814 (2010).
33. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," arXiv:1412.6980 (2014).

Steven Squires received his PhD in machine learning at the University of Southampton, worked as a research associate on applying machine learning methods to medical imaging at the University of Manchester, and is currently working at the University of Exeter. He is a post-doctoral research fellow interested in the development and application of machine learning and statistical methods to medical problems.

Elaine Harkness: Biography is not available.

Dafydd Gareth Evans is chair of medical genetics and cancer epidemiology at the University of Manchester. He has established a national/international reputation in clinical and research aspects of cancer genetics, particularly in neurofibromatosis and breast cancer. He has published 1012 peer-reviewed research publications (first/senior author = 370), and in addition, >150 reviews and chapters. He has an ISI WoK H-index of 129 and Google Scholar H-index of 170. He is the theme leader of Manchester NIHR Biomedical Research Centre Cancer Prevention Early Detection.

Susan M. Astley is chair of intelligent medical imaging at the University of Manchester, with research interest in breast density, early detection, and the prediction of risk of breast cancer. She has published more than 290 research publications with over 6700 citations and is currently co-chair of the SPIE Medical Imaging CAD conference.