Classifying Breast Cancer from Mammograms Using CNNs

Kiri Koppelgaard (KK, 201708294) and Signe Kirk Brødbæk (SKB, 201707519)

MSc in Information Technology (Cognitive Science), Aarhus University

Data Science, Spring 2022, May 31, 2022

(KK, SKB)[1] Worldwide, breast cancer is the one of the most common cancers with over two million annual diagnoses. Detection of malignant tumours at an advanced stage of the disease often lead to more difficult treatment with higher mortality rates. Therefore, early detection is of utmost importance. The most common breast cancer screening method is mammography, which are usually manually inspected by radiologists. Recently, it has been shown that CNN-based algorithms can exceed radiologists' interpretive accuracy (Trister et al., 2017). Potentially, CNN-based diagnosis tools could assist radiologists and help make the process of detecting breast cancer less expert dependent. In the current study, we develop three baseline CNNs and implement two transfer learning models, Inception-v3 and EfficientNetv2 to classify mammograms. All CNNs are trained and tested on the DDSM and CBIS-DDSM (Karssemeijer, 1998; Sawyer-Lee et al., 2016). The best performing model is Inception-v3 with an accuracy of 92.50 %. However, a relatively low specificity and a high sensitivity is obtained for the negative class, while the opposite is obtained for the non-negative classes. As a result, patients would be underdiagnosed by the model. Limitations include an imbalanced data set and under-regularised models. If any of these models were to be implemented as a reliable part of a CAD system, improvements must be made on the model performance.

Key words: Mammography, Breast Cancer, CNNs, Transfer Learning, DDSM, CBIS-DDSM

[1] The initials at every section denote the primary contributor for the following section. However, the entire paper has been edited by both authors. When both authors' initials appear at a section, both authors have contributed equally to that section.

Introduction

(KK) Worldwide, breast cancer is the one of the most common cancers with over two million annual diagnoses (Abdelrahman et al., 2021). Breast cancer is most common among women, though it is detected in men in rare cases (Houfani et al., 2019, p. 247). Especially, as more women reach a higher age, their risk of developing cancers increases (Houfani et al., 2019, p. 249). Breast cancer is often characterised by a lack of early symptoms, which makes early diagnosis more challenging (Milosevic et al., 2018). Detection of malignant tumours at an advanced stage of the disease often lead to more difficult treatment. Therefore, early detection is of utmost importance. Currently, the practice for realising early detection is mammography screenings. However, the analysis process can be suboptimal as it relies on qualitative expertise of the radiologists. This motivates research in machine learning (ML) techniques that has the potential to both streamline and improve the current practices (Houfani et al., 2019, p. 247) by assisting the analysis in computer-aided detection (CAD) systems. The potential of these technical advances will be the focus of this study, namely classifying mammograms to detect breast cancer using convolutional neural networks (CNNs).

What is breast cancer?

(KK) Cancer occurs due to anarchic divisions of abnormal cells, i.e., cell mutations (Houfani et al., 2019, p. 248). Usually, damaged or old cells are replaced by new ones, but in some cases, this process fails, and the damaged cells keep dividing, forming a tumour (Chaurasia & Pal, 2014). Typically, tumours are classified as either benign or malignant. Benign tumours are not considered dangerous as the cells are close to normal in appearance and do not invade nearby tissue (Houfani et al., 2019, p. 248). Malignant tumours, on the other hand, are cancerous, indicating that leaving them unchecked and untreated could lead to the cancer spreading, making it more likely to cause death. Tumours in the breast tissue can be calcifications or masses. Breast calcifications occur commonly and are calcium deposits that develop in women’s breast tissue (*Breast Calcifications*, 2021). Both masses and calcifications are usually benign but can in some cases be cancerous (Choi, 2022). Especially, postmenopausal women are considered the risk group for developing breast cancer (Houfani et al., 2019, p. 247).

Procedures for diagnosing breast cancer

(SKB) The most common breast cancer screening method is mammography (Abdelrahman et al., 2021), and is an x-ray examination of the breasts (Bigaard & Kvernrød, 2022; Thomsen, 2020), which results in images of the breast tissue showing glands, ducts, and connective tissue on a background of fatty tissue (Thomsen, 2020). The objective of these screenings is to detect changes in the breast tissue earlier when treatment is more likely to be successful (Gøtzsche & Jørgensen, 2013), and meta-analyses assessing the effect of mammography estimate a 13-17 % relative reduction in breast cancer related death (Løberg et al., 2015).

When a mammogram shows characteristic changes in the tissue, the radiologist attempts to distinguish benign from malignant changes. Benign changes usually move the breast tissue as they grow, while the malignant changes grow into the surrounding tissue (Thomsen, 2020). If a mammography screening shows signs of a tumour or changes in the tissue, the patient will be asked to come back for further examinations (Bigaard & Kvernrød, 2022). Around 1.5 % of patients will be recommended for a needle biopsy after the mammography examination (Abdelrahman et al., 2021), but 66-87 % of those biopsies will be false positives (Berg et al., 2009). This, naturally, leads to considerations on the limitations of mammography screenings. Digital mammograms, the most common screening tool, has a sensitivity of 84 % for detecting breast cancer at the time of screening (Trister et al., 2017).

The remaining undetected 16 % are partly due to human limitations (Trister et al., 2017). When a radiologist inspects a mammogram, they rely on their qualitative visual experience which introduces some subjectivity to the interpretation (Trister et al., 2017) as well as making the analysis very labour-intensive (Abdelrahman et al., 2021). Essential drawbacks of mammography screenings include overdiagnosis and overtreatment (American Cancer Society, 2022a; Løberg et al., 2015). This implies detection and treatment of tumours that might not develop to be symptomatic or life-threatening (Løberg et al., 2015). Today, all tumours are treated, since markers that can distinguish between overdiagnosed tumours and potential life-threatening tumours are lacking (American Cancer Society, 2022a; Løberg et al., 2015). If patients are overdiagnosed, they will go through anxiety and side effects without the treatment being needed.

(KK) As an alternative to manually detecting breast cancer from mammograms, CAD systems emerged in the 1990s (Abdelrahman et al., 2021). With the recent advances in computational processing power and the growth of digital health data, the potential of using ML to improve accuracy in medical diagnosing is enormous (Trister et al., 2017). Recently, it has been shown that CNN-based algorithms can exceed radiologists' interpretive accuracy and increase radiologists' efficiency (Trister et al., 2017).

Though CNN-based algorithms have proven useful as a diagnosis assisting tool, there is a wide range of ethical concerns that must be addressed. First and foremost, blindly trusting the algorithm is not advisable as absolute accuracy is unobtainable realistically and both false positives and false negatives are associated with harms (Trister et al., 2017). False negatives can lead to underdiagnosing (i.e., undetected tumours), which bears the risk of a fatal outcome. High sensitivity is therefore of essence, when implementing CNN-based algorithms in a CAD system. Though current algorithms have obtained a high level of sensitivity, there remains room for improving the specificity of these models (Abdelrahman et al., 2021). As mentioned above, overclassification of cancer tumours leads to unnecessary breast biopsies causing harm and anxiety for the patient, besides adding exorbitant expenses on the health care system (Chougrad et al., 2018). Thus, it is of high importance to increase the specificity to avoid overclassification. Yet, there is a strong ethical argument for a more efficient and early detection of cancerous tumours, which to some level outweigh the harms of overclassification using CADs (Trister et al., 2017).

CNNs

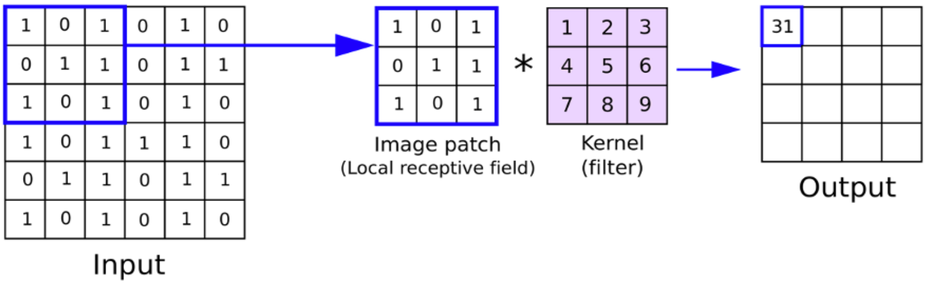
(SKB) CNNs are widely used for state-of-the-art image recognition (Nielsen, 2015; Szegedy et al., 2015). They are powerful for analysing images since they can preserve the spatial architecture of the images (Abdelrahman et al., 2021; Nielsen, 2015). CNNs consist of three main types of layers: *convolutional layers, pooling layers*, and *fully-connected layers* (IBM Cloud Education, 2020).

The convolutional layer is the core building block of CNNs, and most of the computation happens within these layers. The size of the image is decreased within these layers, without losing the relationship between different parts of the input image (Yalçın, 2018). In a convolutional layer, the input image is divided into smaller regions, e.g., 3 x 3 pixels, known as *local receptive fields*, see figure 1. The local receptive field is slid across the entire input image, usually with a stride of 1, i.e., a movement of the local receptive field of 1 pixel. For each local receptive field, there will be one neuron in the subsequent layer (Nielsen, 2015). These hidden layer neurons have the same weights and an overall bias, which define a *kernel*. When the kernel is applied to a local receptive field, the dot product is calculated between the local receptive field and the kernel (IBM Cloud Education, 2020; Reynolds, 2019). This process, known as a *convolution*, is repeated for all local receptive fields of the input image, resulting in a weighted combination of the local receptive fields and the filter (Reynolds, 2019).

In this way, the kernel acts as a feature detector and moves across the receptive fields to check whether a given feature is present. Therefore, the output of a convolutional layer is called a *feature map* (IBM Cloud Education, 2020; Yalçın, 2018). Each feature map detects one feature, which will be detectable over the entire image (Nielsen, 2015). Therefore, multiple feature maps are needed for image recognition, and a complete convolutional layer will consist of several feature maps.

Figure 1

Example of a convolution



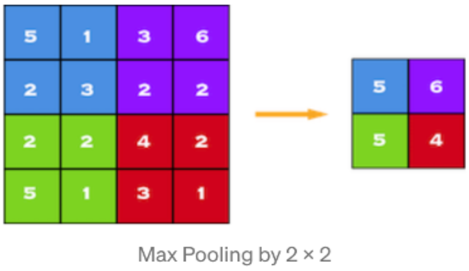
Note: The dot product of the local receptive field and the kernel (3 x 3 pixels) is fed to the output. When this process is completed for each local receptive field in the input, the output will be a feature map. From by A. H. Reynolds, 2019 (https://anhreynolds.com/blogs/cnn.html).

When a feature map has been created, the values are passed through an activation function which introduces non-linearity (Abdelrahman et al., 2021; Nelson, 2019). Typically, the rectified linear unit (ReLU) activation function, ReLU = max(0, x), is used within the hidden convolutional layers, as it is extremely computationally efficient (Abdelrahman et al., 2021; de Andrade, 2019; Nelson, 2019).

After a convolutional layer, it is common to have pooling layers (Yalçın, 2018). Pooling layers reduce the number of trainable parameters which both decreases computational complexity and helps mitigating overfitting, so the model better generalises (Abdelrahman et al., 2021; de Andrade, 2019; Yalçın, 2018). One of the most common pooling methods, *max pooling*, summarises the feature maps by taking the maximum values in a chosen 2D window of the input matrix, see figure 2. Another frequently used pooling method, *average pooling*, simply takes the average of each window.

Figure 2

Example of max pooling



Note: Example of max pooling with a pool size of 2. In the input (left), the maximum value in each coloured field is used in the output (right). From *Image Classification in 10 Minutes with MNIST Dataset*, by O.H. Yalçın, 2018, Medium (https://towardsdatascience.com/image-classification-in-10-minutes-with-mnist-dataset-54c35b77a38d)

The last layers in a CNN are the fully-connected layers that, based on the features extracted throughout the network, perform the classification task (IBM Cloud Education, 2020). In these layers, it is also common to use the ReLU activation function (O’Shea & Nash, 2015). In addition to the layers described above, other types of layers include *dropout* and *batch normalisation*. Dropout layers randomly deactivate nodes in the CNN at each training epoch by setting the weights of these nodes to zero (Abdelrahman et al., 2021). This should prevent the nodes from co-adapting too much (de Andrade, 2019) and mitigate model overfitting (Abdelrahman et al., 2021; de Andrade, 2019). Batch normalisation layers normalise values and reduce the activation function’s reliance on the parameter scales or their initial values (Abdelrahman et al., 2021).

Transfer learning

(KK) To achieve state-of-the-art accuracy with CNNs, you need either a rather large data set and computational budget or use transfer learning (Yalçın, 2021)**.** Transfer learning builds upon the principle of transferring knowledge between different subtasks within a domain (Abdelrahman et al., 2021). Rather than rebuilding a model from scratch for each specific task, transfer learning relies on pre-trained models. Hereby, the robust, discriminative filters learned by state-of-the-art networks can be applied to new data sets, known as *feature extraction,* and the network can be retrained for a new classification task, known as fine-tuning (Brownlee, 2019; Rosebrock, 2019). Thus, an advantage of transfer learning is the reduction of training time and an increased ability to transfer high-level features across domains (Abdelrahman et al., 2021). However, models lacking finetuning tend to underperform, as high-level filters are applied to a specific task without knowledge of the new data set (Abdelrahman et al., 2021).

In general, in neural networks, bottom and mid-level layers represent general features, while top layers represent more problem-specific features. Thus, when applying a pre-trained model to a new subtask, it is recommended to replace and train the top layers to your specific problem (Yalçın, 2021). This also allows you to change the top layer to match the classes in your data set. This approach was utilised by Xi et al., (2018) for mammography classification with great success. For fine-tuning, the new top layers are trained from scratch, while the pre-trained model is completely or partly frozen, i.e., the weights within these layers are not updated. A small learning rate is used for fine-tuning to prevent the network from destroying the originally learned features from the pre-trained model. (Li et al., 2019).

Current study

(SKB) In the current study, we develop three baseline CNNs and implement two transfer learning models, Inception-v3 and EfficientNetv2S, to classify mammograms. The transfer learning models are adapted and fine-tuned for the current classification task. All CNNs are trained and tested on the Digital Database for Screening Mammography (DDSM) and Curated Breast Imaging Subset of DDSM (CBIS-DDSM) (Karssemeijer, 1998; Sawyer-Lee et al., 2016). All models are trained to classify five classes: negative, benign mass, benign calcification, malignant mass, and malignant calcification.

Methods

(KK, SKB) In the following sections, we introduce the data, related work using this data, and the architecture and design consideration of the three baseline CNNs, we developed. Then, the two transfer learning models Inception-v3 and EfficientNetv2S are presented along with the details of the added top layers designed specifically for this classification task. Subsequently, we report the results of each model, and discuss these results alongside the applicability of CNNs for classifying mammograms. Furthermore, advantages and disadvantages of CNNs and ethics of CADs will be discussed. All implementations presented in this paper are available on GitHub[2].

[2] https://github.com/KiriKoppelgaard/Classifying-Breast-Cancer-from-Mammograms-Using-CNNs-and-Transfer-Learning

Data

(KK) The data set used for training and testing the CNNs are images from the DDSM (Karssemeijer, 1998, pp. 457–460) and CBIS-DDSM data sets (Sawyer-Lee et al., 2016). Both CBIS-DDSM and DDSM are commonly cited by the literature and, thereby, establish a common ground for comparison between deep learning methods (Abdelrahman et al., 2021). For the current study, a collected version of these data sets available on Kaggle was utilised[3]. The data set contains positive images from the CBIS-DDSM data set (benign and malignant calcification or mass) and negative (normal) images from the DDSM data set (*DDSM Mammography*, 2018).

DDSM is an archive of 2,620 scanned film mammography cases (Karssemeijer, 1998, pp. 457–460). Each of these cases contains two images of each breast and pathologically verified labels classifying the case. CBIS-DDSM is a subset of DDSM selected and annotated by a professional mammographer (Sawyer-Lee et al., 2016). The data set includes bounding boxes for the regions of interest, ROIs, alongside pathological information concerning tumour grade, stage, and breast mass type (Abdelrahman et al., 2021).

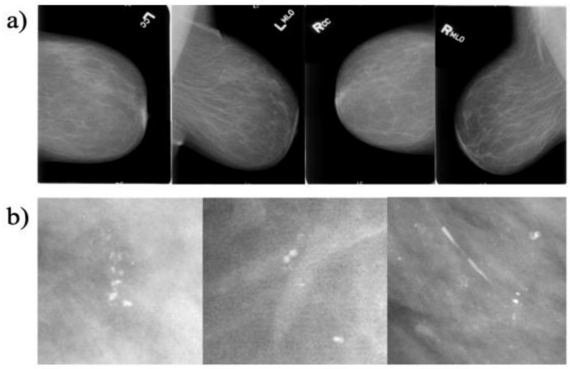
The positive images from the CBIS-DDSM data set are preprocessed by extracting the annotated ROIs with a small padding, and, subsequently, augmenting each image by randomly cropping them into 598 x 598 pixel with random rotations and flips. This was done three times for each ROI. These images are then resized into 299 x 299 pixels. The negative images were converted to 598 x 598 pixels before being resized to 299 x 299. The resulting tfrecords store the data as a sequence of binary strings ideal for processing in TensorFlow. The labels contained within the data set are as follows: 0 = negative, 1 = benign calcification, 2 = benign

[3] The collected data set, containing DDSM and CBIS-DDSM in tfrecords, can be found at the following link: https://www.kaggle.com/datasets/skooch/ddsm-mammography

mass, 3 = malignant calcification, and 4 = malignant mass (*DDSM Mammography*, 2018). For examples from DDSM and CBIS-DDSM, see figure 3.

Figure 3

Example images from DDSM and CBIS-DDSM

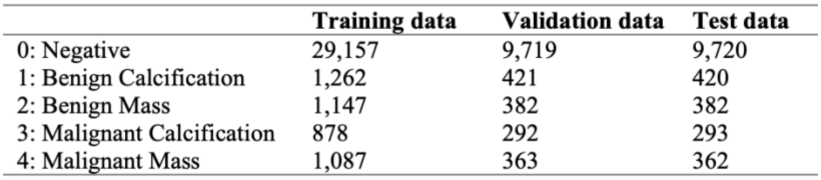


Note: a) Mammograms from DDSM - different views from the same patient. b) Sample images of calcification patches from CBIS-DDSM. From *Abnormality Detection in Mammography using Deep Convolutional Neural Networks*, by Xi et al., 2018, arXiv (http://arxiv.org/abs/1803.01906).

Of 55,885 images, 14 % are positive and 86 % are negative. We divide the data into a training (60 % = 33,531 images), a validation (20 % = 11,177 images), and a test set (20 % = 11,177 images). See table 1 for the distribution of each class in the training, validation, and test data, respectively.

Table 1

Instances in each class in the training, validation, and test data set



	Training data	Validation data	Test data
0: Negative	29,157	9,719	9,720
1: Benign Calcification	1,262	421	420
2: Benign Mass	1,147	382	382
3: Malignant Calcification	878	292	293
4: Malignant Mass	1,087	363	362

Related work utilising DDSM and CBIS-DDSM

(KK) A survey on using CNNs for breast cancer detection shows that a large body of research using these data sets already exists (Abdelrahman et al., 2021). In this survey, the highest sensitivity score reported on mass detection is 0.99 (Al-masni et al., 2018). This was obtained by using a ROI-based CNN called You Only Look Once (YOLO) for feature extraction on DDSM, before detecting masses and classifying them using a fully connected neural network. Using DDSM, Chougrad et al., (2018) developed a CAD system with CNNs and transfer learning. Testing different base models, they found that the base model, Inception-v3, achieved the best accuracy rate of 0.97 on DDSM (Abdelrahman et al., 2021).

Both of these are examples of state-of-the-art performance within the domain of detecting breast cancer from mammograms using deep learning techniques and DDSM. For more examples, see Ribli et al. (2018) or Shen et al. (2020).

Baseline CNNs

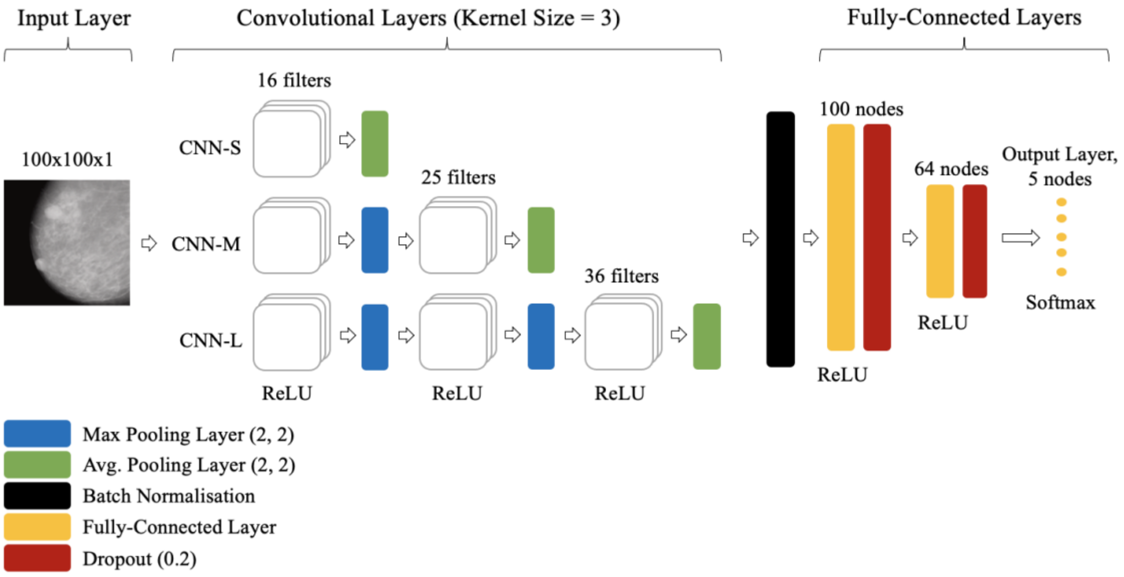
(SKB, KK) In the current study, we implement three baseline CNNs: CNN-S, CNN-M, and CNN-L. With these CNNs, we test how well simpler (and not pre-trained) networks capture the mammography data compared to the more complex and pre-trained transfer learning models. The best performing CNN, measured by a trade-off between accuracy, loss and f1-

score on the validation set, will be tested on the test set, and used as a baseline for comparison with the transfer learning models.

All CNNs were run on a Graphics Processing Unit (Nvidia Tesla T4). The input mammograms are downsized to reduce computation time, resulting in an input size of 100x100x1 pixels. Since the input images are grayscale, we keep depth = 1. Regarding the architecture of the CNNs, we use the same overall architecture in all three networks and only change the number and size of the convolutional layers. Generally, there is a trade-off between network depth, accuracy, and speed; a deeper network often gains a higher accuracy but is also more computationally heavy (Seif, 2022). Therefore, the three CNNs differ in depth to assess the most efficient and accurate architecture to balance this trade-off. The architecture of the CNNs is visualised in Figure 4.

Figure 4

Overview of the architectures of the baseline CNNs



When using convolutional layers, it is common practice to increase the number of filters for additional convolutional layers, so the model is able to learn more complex representations (Nelson, 2019). We follow the advice from Nelson (2019) and, in addition, make the size of convolutional layers powers of 2 to make them more computationally efficient on a GPU. The simplest CNN, CNN-S, includes one convolutional layer with 16 filters, while CNN-M includes two convolutional layers, the first with 16 filters and the second with 25 filters. CNN-L includes three convolutional layers; the first two identical to the layers in CNN-M, while the last layer has 36 filters. All convolutional layers had a kernel size of 3, based on Seif's (2022) argument that not much is gained from larger kernel sizes, as well as a stride of 1. All convolutional layers used the ReLU activation function, since it is computationally efficient and commonly used (Abdelrahman et al., 2021; Seif, 2022). Following a common paradigm (Seif, 2022), we implemented max pooling layers (pool size=2,2) between convolutional layers in all three CNNs. This allows us to maintain the max features throughout the network. Following Seif (2022), we implement an average pooling layer after the last convolutional layer.

After the convolutional layers, we implement a batch normalisation layer, followed by two fully-connected layers. Each of these fully-connected layers are followed by a 0.2 dropout layer. The fully-connected layers include 100 and 64 nodes, respectively. Both layers use the ReLU activation function. Finally, the activations of the second fully-connected layer are fed to the five node output layer that uses the SoftMax activation function for classification of the five classes.

All CNNs use the loss function sparse categorical cross entropy and the adaptive moment (Adam) optimiser, which dynamically updates the learning rate using gradient momentum (Abdelrahman et al., 2021). Each CNN is trained for a maximum of 200 epochs while we record accuracy and loss on the training and validation set. We implement early stopping measuring validation loss with a patience of 50 to balance over- and underfitting (Brownlee, 2018). Early stopping detects the optimal point to stop training by tracking the point when performance on the validation set starts to degrade (Brownlee, 2018), i.e. the model starts to overfit on the training data. We save the model from earlier epochs before validation performance starts to decrease. This approach can help to avoid overfitting while still ensuring that the model is trained sufficiently on the data (Brownlee, 2018).. This also has the advantage of lowering the emissions of model training, since training might be stopped earlier. The best model (i.e. the model with the lowest validation loss) from fine-tuning is saved and model performance is assessed on the test set.

Transfer learning models

The pre-trained models used in this paper are (1) Inception-v3 and (2) EfficientNetv2S. They are presented in the following sections.

Inception-v3

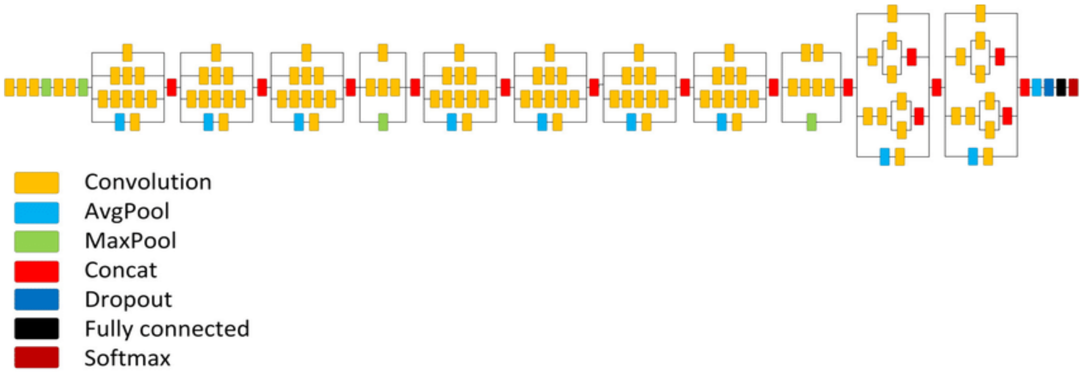
(KK) The first pre-trained CNN architecture, we implement, is Inception-v3 pre-trained on ImageNet (Szegedy et al., 2015). This architecture is designed to achieve a high accuracy with limited memory and computational power.

There are four main design principles underlying the architecture: (1) avoiding representational bottlenecks by gently decreasing representation size between layers, (2) increasing activations per tile in the CNN to disentangle features further, (3) aggregating over lower dimensional embeddings, since this will not add much or any loss in representational power, and, lastly, (4) balancing the width and depth of the network to distribute the computational cost evenly (Szegedy et al., 2015).

Compared to previous versions in the Inception family, Inception-v3 improves performance by using factorised 7 x 7 convolutions, label smoothing, and an auxiliary classifier. The factorised 7 x 7 convolutions effectively transform higher dimensional convolutions into a sequence of lower dimensional convolutions with approximately the same output, but with lower computational complexity. Label smoothing regularises the activations by introducing noise for the labels, and the auxiliary classifier pushes useful gradients to the lower layers to improve convergence (Szegedy et al., 2015). The resulting architecture is depicted in Figure 5.

Figure 5

Inception-v3 architecture



Note: From *A feasibility study of deep neural networks for the recognition of banknotes regarding central bank requirements*, by Schulte et al., 2019, arXiv (http://arxiv.org/abs/1907.07890).

EfficientNetv2S

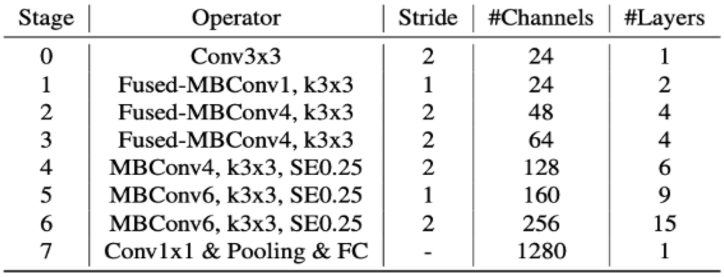
(SKB) The second transfer learning model we implement is EfficientNetV2S pre-trained on ImageNet (TensorFlow, 2022a). EfficientNetv2 was released in 2021 as an improved version of EfficientNet in terms of training speed and accuracy (Ibrahim, 2021; Tan & Le, 2021). For the development of EfficientNetv2, a combination of training-aware neural architecture search (NAS) and scaling (Tan & Le, 2021) was used. NAS aims to discover the best architecture for a neural network by automating the process of tweaking the neural network topology to learn what works well (Gey, 2021).

The improvements of EfficientNetV2 include (1) a use of both MBConv layers and fused-MBConv layers in earlier layers, (2) a smaller expansion ratio for MBConv, (3) smaller kernel sizes of 3x3 with additional layers to compensate for the smaller local receptive fields, and (4) a removal of the last stride-1 stage that was present in the original EfficientNet (Tan & Le, 2021). For the architecture of EfficientNetv2S, see table 2.

Furthermore, progressive learning was implemented for EfficientNetv2 (Tan & Dai, 2021), meaning that regularisation increased with increased image sizes.The results of these alterations is a network architecture that trains faster than previous models with a better parameter efficiency (Arora, 2021).

Table 2

Architecture of EfficientNetv2S



Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

Note: From *EfficientNetV2: Smaller Models and Faster Training*, by M. Tan & Q.V. Le, 2021, arXiv (https://doi.org/10.48550/arXiv.2104.00298).

Transfer learning implementations

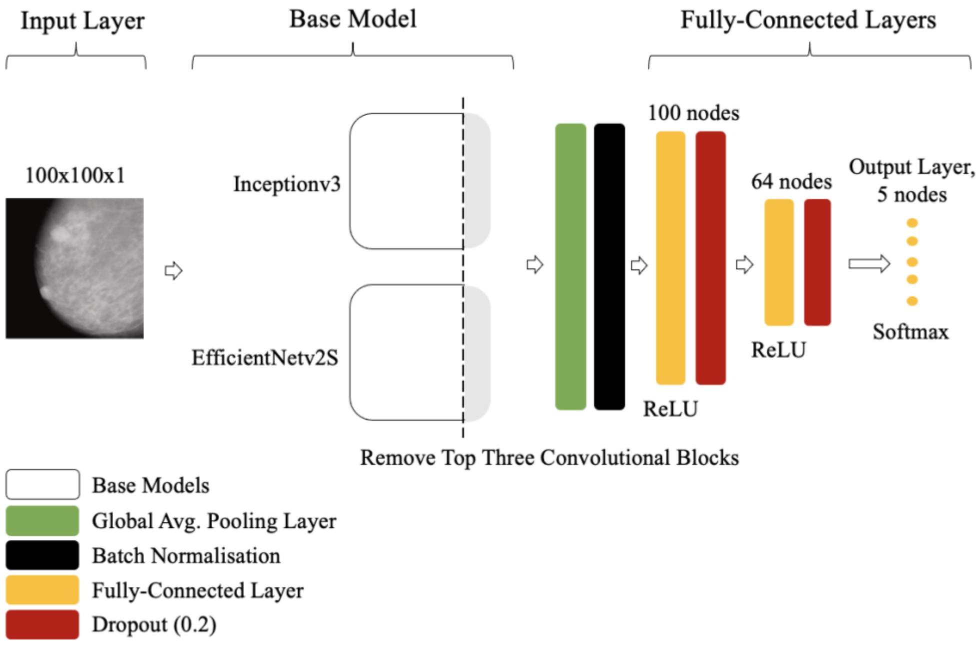
(SKB) In the current study, we utilise the Keras TensorFlow implementations of EfficientNetV2S and Inception-v3 (TensorFlow, 2022b, 2022a) as the base models for feature extraction. We then finetune them to the task of classifying breast cancer from mammograms.

The same procedure was followed for both transfer learning models. As the CNNs, both transfer learning models were run on a Graphics Processing Unit (Nvidia Tesla T4).

Following the recommendations of Yalçın (2021) and the procedure of (Xi et al., 2018), we remove the top three convolutional blocks of the base models to drop the problem-specific features from the ImageNet training. We replace them with layers specific to the current classification task: (1) a global pooling layer to reduce dimensionality, (2) a batch normalisation layer to normalise the input, and, lastly, (3) two dense layers separated by two 0.2 dropout layers to reduce overfitting. The resulting architecture is visualised in figure 6 below.

Figure 6

Architecture of top layers added to the transfer learning models, Inceptionv3 and EfficientNetv2S



Following the Keras tutorial on transfer learning and fine-tuning (Chollet, 2020), we first freeze the base model layers so the new, randomly initialised top layers can converge on our data set. Chollet (2020) emphasises the importance of this step, since mixing randomly-initialised layers with the pre-trained features, might destroy the pre-trained features due to very large gradient updates. Thus, the model might unlearn what was learned during pre-training. Next, with all base model layers frozen, we train the model for 20 epochs using the Adam optimiser and the sparse categorical cross entropy loss function. The model with the minimum validation loss from pre-training is saved for fine-tuning. We fine-tune the model by unfreezing the base model, as this can lead to incremental improvements (Chollet, 2020). Following the Keras tutorial (Chollet, 2020), we implement a very low learning rate (learning rate = 0.004) as we, at this stage, only want to readapt the weights incrementally. This procedure also helps to mitigate overfitting since the weight updates will be smaller. We fine-tune the model for 200 epochs with early stopping (patience = 50). We save the best performing model from fine-tuning and evaluate the model on the test set to enable comparison to the other presented models.

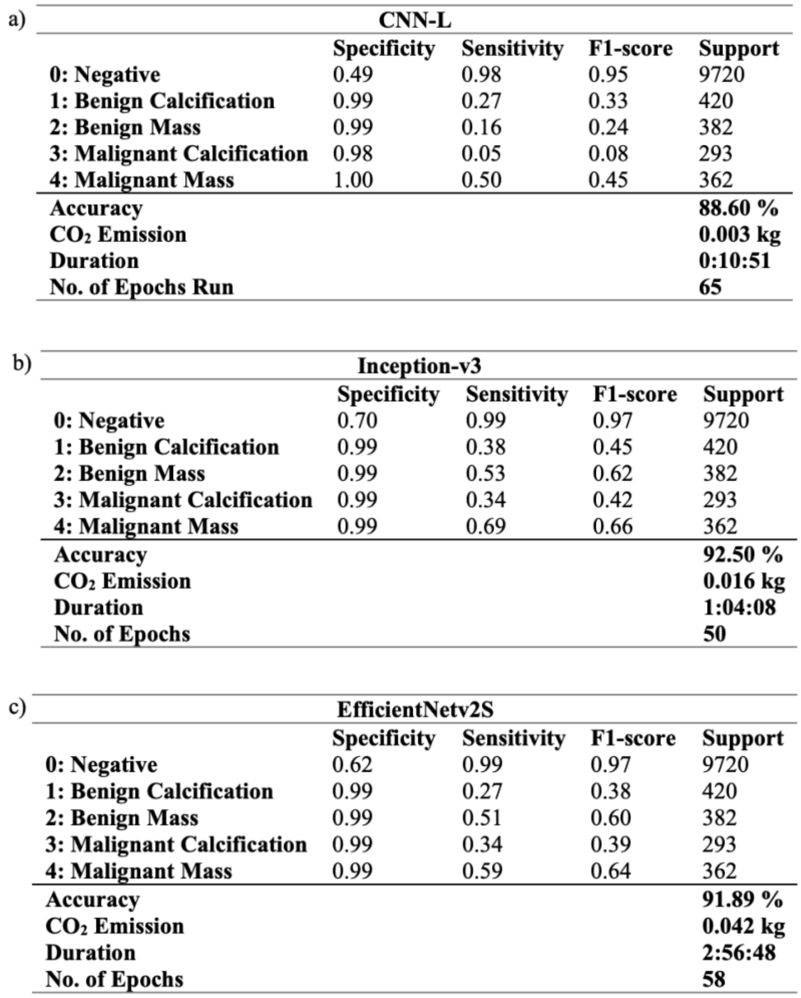
Results

(SKB, KK) In this section, the results of the three baseline CNNs and the two training models, Inceptionv3 and EfficientNetv2S, will be reported. Comparing the validation accuracy of the three baseline CNNs, we find that CNN-L achieves the best performance on the validation set, though the models' performances do not differ substantially (CNN-S_{acc} = 88.13 %, CNN-S_{loss} = 0.40, CNN-$S_{avg.\ f1\text{-}score}$= 0.32; CNN-$M_{acc}$ = 87.92 %, CNN-M_{loss} = 0.39, CNN-$M_{avg.\ f1\text{-}score}$ = 0.30; CNN-$L_{acc.}$ = 88.32 %, CNN-L_{loss} = 0.37, CNN-$L_{avg.\ f1\text{-}score}$ = 0.39). Below, we report only the test set results of the best performing baseline model, CNN-L.

On the test set, CNN-L obtains an accuracy of 88.60 % as well as an average specificity of 0.89 and sensitivity of 0.39. CNN-L obtains a lower specificity for the negative class ($specificity_{Neg.}$ = 0.49) compared to the remaining classes ($specificity_{Cal.\ and\ mass}$= 0.99-1.00). In addition, CNN-L achieves a higher sensitivity for the negative class ($sensitivity_{Neg.}$=0.98) than in the remaining classes (sensitivity$_{\text{Cal. and mass}}$ = 0.05-0.27). Lastly, the negative class has a higher f1-score (f1-score$_{\text{Neg.}}$ = 0.95) compared to the remaining classes (f1-score$_{\text{Cal. and mass}}$ = 0.08-0.45). See table 3a for classification results as well as CO_2 emissions and the duration of training. The confusion matrix of the test set predictions as well as the loss and accuracy history can be found in appendix A.

Table 3

Classification tables for the best performing baseline CNN and the two transfer learning models



a)

CNN-L				
	Specificity	**Sensitivity**	**F1-score**	**Support**
0: Negative	0.49	0.98	0.95	9720
1: Benign Calcification	0.99	0.27	0.33	420
2: Benign Mass	0.99	0.16	0.24	382
3: Malignant Calcification	0.98	0.05	0.08	293
4: Malignant Mass	1.00	0.50	0.45	362
Accuracy				**88.60 %**
CO_2 Emission				**0.003 kg**
Duration				**0:10:51**
No. of Epochs Run				**65**

b)

Inception-v3				
	Specificity	**Sensitivity**	**F1-score**	**Support**
0: Negative	0.70	0.99	0.97	9720
1: Benign Calcification	0.99	0.38	0.45	420
2: Benign Mass	0.99	0.53	0.62	382
3: Malignant Calcification	0.99	0.34	0.42	293
4: Malignant Mass	0.99	0.69	0.66	362
Accuracy				**92.50 %**
CO_2 Emission				**0.016 kg**
Duration				**1:04:08**
No. of Epochs				**50**

c)

EfficientNetv2S				
	Specificity	**Sensitivity**	**F1-score**	**Support**
0: Negative	0.62	0.99	0.97	9720
1: Benign Calcification	0.99	0.27	0.38	420
2: Benign Mass	0.99	0.51	0.60	382
3: Malignant Calcification	0.99	0.34	0.39	293
4: Malignant Mass	0.99	0.59	0.64	362
Accuracy				**91.89 %**
CO_2 Emission				**0.042 kg**
Duration				**2:56:48**
No. of Epochs				**58**

Note: a) baseline model, CNN-L, b) Inception-v3, and c) EfficientNetv2S.

Assessing the performance of the transfer learning models, Inception-v3 and EfficientNetv2S, we find that both models obtain a higher performance on the test set than the best baseline model. Inception-v3 obtains the highest accuracy of the transfer learning models (accuracy$_{\text{Inception-v3}}$ = 92.50 %, accuracy$_{\text{EfficientNetv2S}}$ = 91.89 %).

Moreover, Inception-v3 obtains the highest average specificity and specificity (avg. specificity$_{\text{Inc.}}$ = 0.93, avg. sensitivity$_{\text{Inc.}}$ = 0.58; avg. specificity$_{\text{Eff.}}$ = 0.92, avg. sensitivity$_{\text{Eff.}}$ = 0.54). Like the baseline model, both Inception-v3 and EfficientNetv2S obtain a lower specificity for the negative class (specificity$_{\text{Inc, neg.}}$ = 0.70; specificity$_{\text{Eff. neg.}}$ = 0.62) than the remaining classes (specificity$_{\text{Inc. cal. and mass}}$ = 0.99; specificity$_{\text{Eff., cal. and mass}}$ = 0.99). Furthermore, both models have a higher sensitivity for the negative class (sensitivity$_{\text{Inc. neg.}}$ = 0.99; sensitivity$_{\text{Eff., neg.}}$ = 0.99) than the remaining classes (sensitivity$_{\text{Inc., cal. and mass}}$ = 0.34-0.69; sensitivity$_{\text{Eff., cal. and mass}}$ = 0.27-0.59). Finally, both models obtain higher f1-scores for the negative class (f1-score$_{\text{Inc., neg.}}$ = 0.97; f1-score$_{\text{Eff. neg.}}$ = 0.97) compared to the remaining classes (f1-score$_{\text{Inc. cal. and mass}}$ = 0.42-0.66; f1-score$_{\text{Eff. cal. and mass}}$ = 0.38-0.64). Of all models, the training of EfficientNetv2S had the longest duration and, thus, the highest CO_2 emissions (duration = 2:56:48, CO_2 emissions = 0.042 kg). See table 3b and 3c for the classification results, as well as CO_2 emissions and duration of training.

The confusion matrix of test set predictions as well as the loss and accuracy history for Inception-v3 and EfficientNetv2S, respectively, can be found in appendix B and C.

Discussion

(KK, SKB) Overall, the best performing model is Inception-v3, which achieves both the highest average specificity, sensitivity, accuracy, and f1-scores. Although all three reported models acquire a relatively high accuracy, this is likely a result of the imbalanced data set, implying an accuracy paradox (Brownlee, 2015). Negative samples are largely overrepresented in the data set, and, consequently, the models can acquire a large accuracy by classifying cases as negative.

Regarding the negative class, all models obtain a relatively low specificity and high sensitivity and, as a result, the models tend to report false positive results for the negative class, i.e., classifying non-normal cases as normal. For the remaining classes, all models obtain high specificity and low sensitivity, which means that the models tend to report false negative results for these classes. As a result, the models will predict most mammograms as being normal, even though there is, in fact, a mass or calcification present. This leads to underdiagnosis. Again, these results could be due to the imbalance in the data set, and the models not learning enough from the data set: If the model defaults to the negative class, naturally, it will capture most negative cases and, consequently, the sensitivity would be high for the negative class and rather low for the non-negative classes. This trend is especially apparent in CNN-L, while Inception-v3 and EfficientNetv2S have higher sensitivity for the non-negative classes. When we compare the results of our best performing model, Inception-v3, to related research, it does not obtain state-of-the-art accuracy or sensitivity.

The results point to several ethical concerns were these models to be implemented in a CAD system. First, when dealing with diseases, very high performance standards are vital. Although Inception-v3 has high accuracy, it still incorrectly classifies 7.5 % of the cases, and this amount of misclassifications might still be deemed too many in this context. Second, the low sensitivity scores for the positive classes can lead to underdiagnosing with, potentially, fatal consequences. Third, the fact that the models can detect the true negatives is of little use, if they are unable to detect the positive cases. Thus, if these models were implemented as is in a CAD system, blindly trusting the classifications of the system could be detrimental. Therefore, these models, and especially the model sensitivity, need to be improved before being implemented as a diagnosis assisting tool.

Data set limitations

(KK) Before these models can be implemented for diagnosis, one of the aspects that must improve is the handling of the imbalance of classes in the data set. Though, the high number of negative cases might inform the models of the high base rate of the negative class, this leads the model to overestimate the number of negative cases. A more balanced data set could potentially force the model to acquire more information about the positive classes, rather than defaulting to the negative class. A more balanced relationship between the positive classes, might also mitigate the fact that the models in the current study tend to detect (benign and malignant) masses to a larger extent than (benign and malignant) calcifications.

A workaround of the class imbalance could be to resample the data set. This could be done either by adding copies of the underrepresented classes and, thereby, oversample the positive classes, or by deleting instances from the overrepresented class and, thereby, undersample the negative class (Brownlee, 2015).

Moreover, model training and generalisation could benefit from a larger data set. A larger data volume could have been achieved by including mammograms from other data sets such as MIAS (Suckling et al., 2015), INBreast (Moreira et al., 2012) or OPTIMAAM (Halling-Brown et al., 2020). Besides gaining a higher performance, the inclusion of other data sets could potentially increase the models' ability to generalise, as different data sets might contain cases and equipment with different qualities. Hereby, also mitigating biases in the data. An approach for testing the generalisability of models across data sets is to use one or more data set for training and another for validating and testing (e.g. Fujita et al., 2014**)**. Altogether, more data could benefit both the training process of the networks and the validity of the results. However, including more data would increase the computational costs of training the models, and, depending on the available hardware, might not be feasible.

Lastly, a limitation of the data set is that it consists of film mammograms, which does not contain the same level of detail as digital mammograms (SCCA staff, 2012). According to the National Cancer Institute as cited in SCCA staff (2012), women with denser breast tissue benefit from having digital mammograms over film mammograms, as the subtle differences between normal and abnormal tissue are easier to detect. For women with denser breast tissue, the digital mammography detects 28 % more cancers compared to film mammograms (SCCA staff, 2012). Thus, as we hope to acquire a higher performance in breast cancer detection in women with denser breast tissue, it might be beneficial to use digital mammograms with a higher level of detail.

Implementation limitations

(SKB) All models implemented in the current study tend to overfit quite fast, i.e., they are under-regularised. This is seen by the increase in training accuracy and decrease in training loss, despite the opposite pattern for the validation measures (see appendices A-C). We used 0.2 dropout after each fully-connected layer but utilising stricter regularisations might have mitigated this overfitting. Future research could experiment with 0.5 dropout layers or other regularisation methods, such as L1 or L2 regularisation added to the convolutional layers themselves (Xi et al., 2018). L1 regularisation forces some weights to zero, which can give more sparse weights, while L2 regularisation minimises the weights, but never reduces them to zero. Adding these regularisation methods might have helped to reduce overfitting. Future research could experiment with running the analysis with different regularisations within a cross-validation scheme.

A limitation of the current study is the downsizing of the input images. We downsized the mammograms to minimise computational costs and training time. However, by downsizing the mammograms, we lost some data quality, which, potentially, could have improved model performances. Another limitation lies in our design of the convolutional layers in the baseline CNNs, since these layers can have a great influence on the model's performance (de Andrade, 2019). Though we followed general recommendations for the selection hyperparameters and compared multiple models, we kept the number of convolutional layers to a minimum for the baseline CNNs (maximum depth of 3 convolutional layers in CNN-L). As mentioned above, there is a trade-off between depth and accuracy, and a maximum of three convolutional layers might not be sufficient to capture the complex patterns in mammograms. Ideally, we would experiment with additional and more complex convolutional network structures as well as experiment with different loss and activation functions. However, this process can be quite tedious and computationally heavy and, therefore, this was beyond the scope of this project. Alternatively, an automated procedure, such as NAS, could have been implemented to search for optimal architectures and hyperparameters.

(KK) Concerning the train-val-test split, there is, generally, not an optimal split percentage (Baheti, 2022; Brownlee, 2020). The optimal split depends on multiple factors, such as use case, the structure of the models, as well as the data. A common split is 80 % training data, 10 % validation data, and 10 % test data, (Baheti, 2022; Brownlee, 2020) but due to the imbalanced data, we chose a more conservative 60/20/20 split. However, most of the models' validation accuracy still varies quite a lot during training (see e.g. appendix A), indicating that the validation set was too small and/or not representative of the data set. If this is the case, this could have led to non-optimal tuning of the models (Baheti, 2022). To compensate for this, we could either have experimented with other splits or implemented k-fold cross-validation to utilise the data to a fuller extent. In k-fold cross validation, each data point in the data set would have the chance of appearing in the training and test set, which, generally, results in less biassed models and can be very helpful, when having a limited amount of data (Sanjay, 2018).

CNN limitations

(SKB) More generally, CNNs, though powerful and widely used for image classification (Hosseini et al., 2017), have some disadvantages. First, even though the shared weights and biases greatly reduce the parameter count in CNNs (Nielsen, 2015), they can be tremendously resource-heavy (O'Shea & Nash, 2015). In the current study, it would have been unfeasible to run the transfer learning models without GPU access. Furthermore, CNNs are quite vulnerable to noise in images; Goodfellow et al. (2015) reported that applying small perturbations to images in the ImageNet data set resulted in the convolutional network GoogLeNet (Szegedy et al., 2014) outputting an incorrect answer - with high confidence. In the context of the current study, this vulnerability could potentially have consequences for denser breast tissue, where more "noise" is present in the mammograms. People with denser breast tissue have a higher risk of breast cancer, while radiologists have more difficulty detecting the cancer using mammography (American Cancer Society, 2022b). If the difficulty remains - or, potentially, are enhanced - using CNNs, this could be a significant issue for the use of CNNs in CAD systems.

Conclusion

(KK, SKB) In the current study, we developed three baseline CNNs and implemented two transfer learning models, Inception-v3 and EfficientNetv2S, for a mammogram classification task using DDSM and CBIS-DDSM. The best performing model was Inception-v3, which achieved an accuracy of 95.50%. However, a relatively low specificity and a high sensitivity is obtained for the negative class, while the opposite is obtained for the non-negative classes. As a result, patients would be underdiagnosed by the model. All models acquire a relatively high accuracy, but as negative cases are largely overrepresented in the data set, the models can acquire a large accuracy by defaulting to a negative class prediction. The overrepresentation of the negative class might also explain the sensitivity-specificity pattern for the models. All models obtain a relatively low specificity and high sensitivity for the negative class, while obtaining a high specificity and low sensitivity for the non-negative classes. Resultantly, the models will tend to predict most mammograms as normal, despite the presence of a tumour. One approach to compensate for the imbalanced data set, could be to resample the data set or combine more data sets. Other potential limitations include the models' tendency to overfit, which might have been mitigated by stronger regularisation, and the selection of hyperparameters that could have been more rigorously investigated before implementation. False negatives come with a high cost, when detecting cancerous tumours. Therefore, if any of these models were to be implemented as a reliable part of a CAD system, improvements must be made on the model sensitivity.

Bibliography

Abdelrahman, L., Al Ghamdi, M., Collado-Mesa, F., & Abdel-Mottaleb, M. (2021). Convolutional neural networks for breast cancer detection in mammography: A survey. *Computers in Biology and Medicine*, *131*, 104248. https://doi.org/10.1016/j.compbiomed.2021.104248

Al-masni, M. A., Al-antari, M. A., Park, J.-M., Gi, G., Kim, T.-Y., Rivera, P., Valarezo, E., Choi, M.-T., Han, S.-M., & Kim, T.-S. (2018). Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system. *Computer Methods and Programs in Biomedicine*, *157*, 85–94. https://doi.org/10.1016/j.cmpb.2018.01.017

American Cancer Society. (2022a, January 14). *Limitations of Mammograms | How Accurate Are Mammograms?* https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection/mammograms/limitations-of-mammograms.html

American Cancer Society. (2022b, March 10). *Dense Breast Tissue | Breast Density and Mammogram Reports*. https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection/mammograms/breast-density-and-your-mammogram-report.html

Baheti, P. (2022, May). *Splitting Machine Learning Data: Train, Validation, Test Set Split*. https://www.v7labs.com/blog/train-validation-test-set, https://www.v7labs.com/blog/train-validation-test-set

Berg, W. A., Hendrick, R. E., Kopans, D. B., & Smith, R. A. (2009). *Frequently Asked Questions about Mammography and the USPSTF Recommendations: A Guide for Practitioners*. https://www.sbi-online.org/Portals/0/downloads/documents/pdfs/Detailed_Response_to_USPSTF_Guidelines-12-11-09-Berg.pdf

Bigaard, J., & Kvernrød, A.-B. (2022, March 29). *Screening for brystkræft*. Kræftens Bekæmpelse. https://www.cancer.dk/forebyg/screening/brystkraeft/

Breast calcifications: When to see a doctor. (2021, April 28). Mayo Clinic. https://www.mayoclinic.org/symptoms/breast-calcifications/basics/definition/sym-20050834

Brownlee, J. (2015, August 18). 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset. *Machine Learning Mastery*. https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

Brownlee, J. (2018, December 7). A Gentle Introduction to Early Stopping to Avoid Overtraining Neural Networks. *Machine Learning Mastery*. https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/

Brownlee, J. (2019, May 14). Transfer Learning in Keras with Computer Vision Models. *Machine Learning Mastery*. https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/

Brownlee, J. (2020, July 23). Train-Test Split for Evaluating Machine Learning Algorithms. *Machine Learning Mastery*. https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/

Chaurasia, V., & Pal, S. (2014). A Novel Approach for Breast Cancer Detection using Data Mining Techniques. *International Journal of Innovative Research in Computer and Communication Engineering, 3297,* 2320–9801.

Choi, L. (2022, March). *Breast Masses (Breast Lumps)—Gynecology and Obstetrics*. MSD Manual Professional Edition. https://www.msdmanuals.com/professional/gynecology-and-obstetrics/breast-disorders/breast-masses-breast-lumps

Chougrad, H., Zouaki, H., & Alheyane, O. (2018). Deep Convolutional Neural Networks for breast cancer screening. *Computer Methods and Programs in Biomedicine, 157,* 19–30. https://doi.org/10.1016/j.cmpb.2018.01.011

DDSM: Digital Database for Screening Mammography. (n.d.). Retrieved 26 April 2022, from http://www.eng.usf.edu/cvprg/mammography/database.html

DDSM Mammography. (2018, July 3). https://www.kaggle.com/skooch/ddsm-mammography

de Andrade, A. (2019). *Best Practices for Convolutional Neural Networks Applied to Object Recognition in Images*. https://doi.org/10.48550/ARXIV.1910.13029

Fujita, H., Hara, T., & Muramatsu, C. (2014). *Breast Imaging: 12th International Workshop, IWDM 2014, Gifu City, Japan, June 29 - July 2, 2014, Proceedings*. Springer.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). *Explaining and Harnessing Adversarial Examples* (arXiv:1412.6572). arXiv. http://arxiv.org/abs/1412.6572

Gøtzsche, P. C., & Jørgensen, K. J. (2013). Screening for breast cancer with mammography. *Cochrane Database of Systematic Reviews*, *6*. https://doi.org/10.1002/14651858.CD001877.pub5

Halling-Brown, M. D., Warren, L. M., Ward, D., Lewis, E., Mackenzie, A., Wallis, M. G., Wilkinson, L., Given-Wilson, R. M., McAvinchey, R., & Young, K. C. (2020). *OPTIMAM Mammography Image Database: A large scale resource of mammography images and clinical data* (arXiv:2004.04742). arXiv. https://doi.org/10.48550/arXiv.2004.04742

Hosseini, H., Xiao, B., Jaiswal, M., & Poovendran, R. (2017). *On the Limitation of Convolutional Neural Networks in Recognizing Negative Images* (arXiv:1703.06857). arXiv. http://arxiv.org/abs/1703.06857

Houfani, D., Slatnia, S., Kazar, O., Zerhouni, N., Merizig, A., & Saouli, H. (2019). Machine Learning Techniques for Breast Cancer Diagnosis: Literature Review. In *Advanced Intelligent Systems for Sustainable Development (AI2SD'2019): Vol. Volume 2-Advanced Intelligent Systems for Sustainable Development Applied to Agriculture and Health*. https://link.springer.com/content/pdf/10.1007/978-3-030-36664-3.pdf

IBM Cloud Education. (2020, October 20). *What are Convolutional Neural Networks?* https://www.ibm.com/cloud/learn/convolutional-neural-networks

Karssemeijer, N. (1998). *Digital mammography: Nijmegen, 1998*. Springer Science. http://site.ebrary.com/id/10649789

Li, H., Chaudhari, P., Yang, H., Lam, M., Ravichandran, A., Bhotika, R., & Soatto, S. (2019, September 25). *Rethinking the Hyperparameters for Fine-tuning*. International Conference on Learning Representations. https://openreview.net/forum?id=B1g8VkHFPH

Løberg, M., Lousdal, M. L., Bretthauer, M., & Kalager, M. (2015). Benefits and harms of mammography screening. *Breast Cancer Research*, *17*(1), 63. https://doi.org/10.1186/s13058-015-0525-z

Milosevic, M., Jankovic, D., Milenkovic, A., & Stojanov, D. (2018). Early diagnosis and detection of breast cancer. *Technology and Health Care: Official Journal of the European Society for Engineering and Medicine*, *26*(4), 729–759. https://doi.org/10.3233/THC-181277

Moreira, I. C., Amaral, I., Domingues, I., Cardoso, A., Cardoso, M. J., & Cardoso, J. S. (2012). INbreast: Toward a full-field digital mammographic database. *Academic Radiology*, *19*(2), 236–248. https://doi.org/10.1016/j.acra.2011.09.014

Nelson, D. (2019, May 28). *Image Recognition and Classification in Python with TensorFlow and Keras*. Stack Abuse. https://stackabuse.com/image-recognition-in-python-with-tensorflow-and-keras/

Nielsen, M. A. (2015). Chapter 6: Deep Learning. In *Neural Networks and Deep Learning*. Determination Press. http://neuralnetworksanddeeplearning.com

O'Shea, K., & Nash, R. (2015). *An Introduction to Convolutional Neural Networks* (arXiv:1511.08458). arXiv. http://arxiv.org/abs/1511.08458

Reynolds, A. H. (2019). *Convolutional Neural Networks (CNNs)*. Anh H. Reynolds. https://anhreynolds.com/blogs/cnn.html

Ribli, D., Horváth, A., Unger, Z., Pollner, P., & Csabai, I. (2018). Detecting and classifying lesions in mammograms with Deep Learning. *Scientific Reports*, *8*(1), 4165. https://doi.org/10.1038/s41598-018-22437-z

Rosebrock, A. (2019, May 20). Transfer Learning with Keras and Deep Learning. *PyImageSearch*. https://www.pyimagesearch.com/2019/05/20/transfer-learning-with-keras-and-deep-learning/

Sanjay, M. (2018, November 13). *Why and how to Cross Validate a Model?* Medium. https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f

Sawyer-Lee, R., Gimenez, F., Hoogi, A., & Rubin, D. (2016). *Curated Breast Imaging Subset of DDSM* (Version 1) [Data set]. The Cancer Imaging Archive. https://doi.org/10.7937/K9/TCIA.2016.7O02S9CY

SCCA staff. (2012, October 18). *The difference between digital & film mammography*. Seattle Cancer Care Alliance. https://www.seattlecca.org/blog/2012/10/the-difference-between-digital-film-mammography

Seif, G. (2022, February 11). *A Guide for Building Convolutional Neural Networks*. Medium.

https://towardsdatascience.com/a-guide-for-building-convolutional-neural-networks-e4eefd17f4fd

Shen, R., Yao, J., Yan, K., Tian, K., Jiang, C., & Zhou, K. (2020). Unsupervised domain adaptation with adversarial learning for mass detection in mammogram. *Neurocomputing*, *393*, 27–37. https://doi.org/10.1016/j.neucom.2020.01.099

Suckling, J., Parker, J., Dance, D., Astley, S., Hutt, I., Boggis, C., Ricketts, I., Stamatakis, E., Cerneaz, N., Kok, S., Taylor, P., Betal, D., & Savage, J. (2015). *Mammographic Image Analysis Society (MIAS) database v1.21* [Data set]. https://doi.org/10/250394

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). *Going Deeper with Convolutions* (arXiv:1409.4842). arXiv. http://arxiv.org/abs/1409.4842

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). *Rethinking the Inception Architecture for Computer Vision* (arXiv:1512.00567; Version 3). arXiv. http://arxiv.org/abs/1512.00567

TensorFlow. (2022a, May 18). *EfficientNetV2S*. https://www.tensorflow.org/api_docs/python/tf/keras/applications/efficientnet_v2/EfficientNetV2S

TensorFlow. (2022b, May 18). *Inceptionv3*. TensorFlow. https://www.tensorflow.org/api_docs/python/tf/keras/applications/inception_v3/InceptionV3

Thomsen, H. S. (2020, February 17). *Mammografi—Patienthåndbogen på sundhed.dk*. https://www.sundhed.dk/borger/patienthaandbogen/undersoegelser/undersoegelser/roentgen/mammografi/

Trister, A. D., Buist, D. S. M., & Lee, C. I. (2017). Will Machine Learning Tip the Balance in Breast Cancer Screening? *JAMA Oncology*, *3*(11), 1463. https://doi.org/10.1001/jamaoncol.2017.0473

Xi, P., Shu, C., & Goubran, R. (2018). *Abnormality Detection in Mammography using Deep Convolutional Neural Networks* (arXiv:1803.01906). arXiv. http://arxiv.org/abs/1803.01906

Yalçın, O. G. (2018, August 19). *Image Classification in 10 Minutes with MNIST Dataset*. Medium.

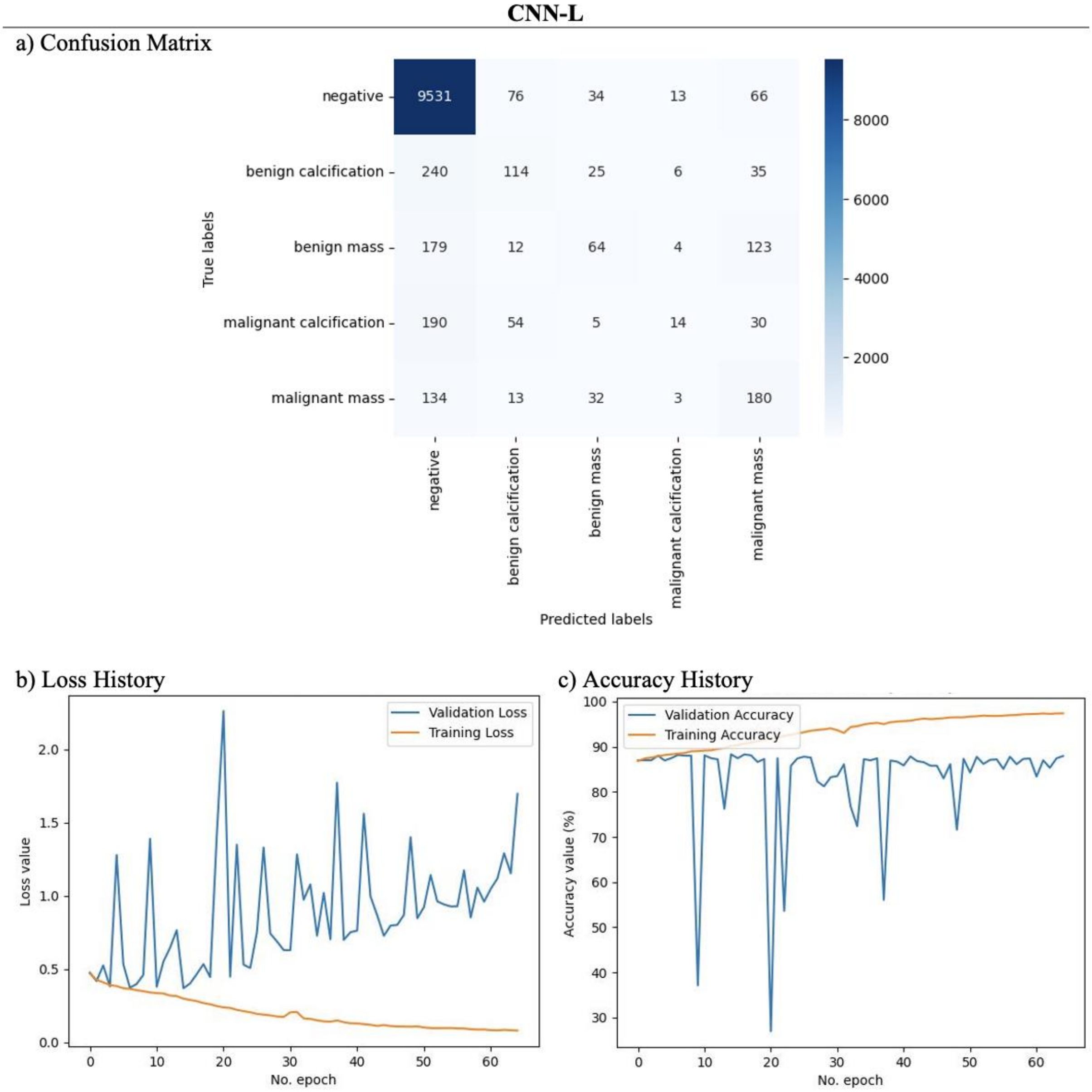
https://towardsdatascience.com/image-classification-in-10-minutes-with-mnist-dataset-54c35b77a38d

Yalçın, O. G. (2021, February 2). *4 Pre-Trained CNN Models to Use for Computer Vision with Transfer Learning*. Medium. https://towardsdatascience.com/4-pre-trained-cnn-models-to-use-for-computer-vision-with-transfer-learning-885cb1b2dfc

Appendix A

CNN-L: Confusion Matrix and validation loss and accuracy history

Figure of a) the confusion matrix for CNN-L's predictions on the test set, b) loss history on training and validation set, and c) accuracy history on the training and validation set.



CNN-L

a) Confusion Matrix

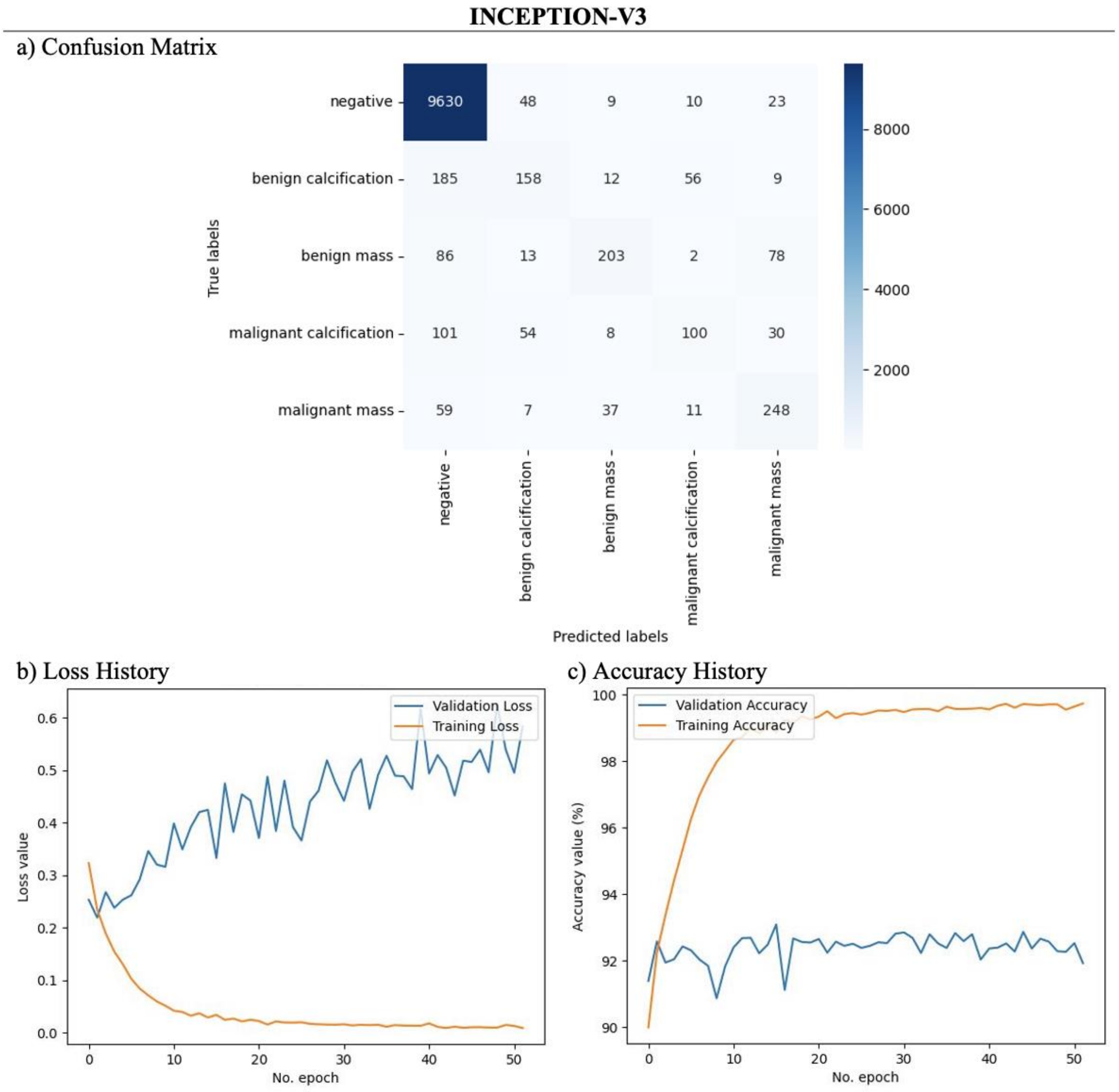
b) Loss History

c) Accuracy History

Appendix B

Inception-v3: Confusion Matrix and validation loss and accuracy history

Figure of a) the confusion matrix for Inception-v3's predictions on the test set, b) loss history on training and validation set, and c) accuracy history on the training and validation set.



INCEPTION-V3

a) Confusion Matrix

True labels \ Predicted labels	negative	benign calcification	benign mass	malignant calcification	malignant mass
negative	9630	48	9	10	23
benign calcification	185	158	12	56	9
benign mass	86	13	203	2	78
malignant calcification	101	54	8	100	30
malignant mass	59	7	37	11	248

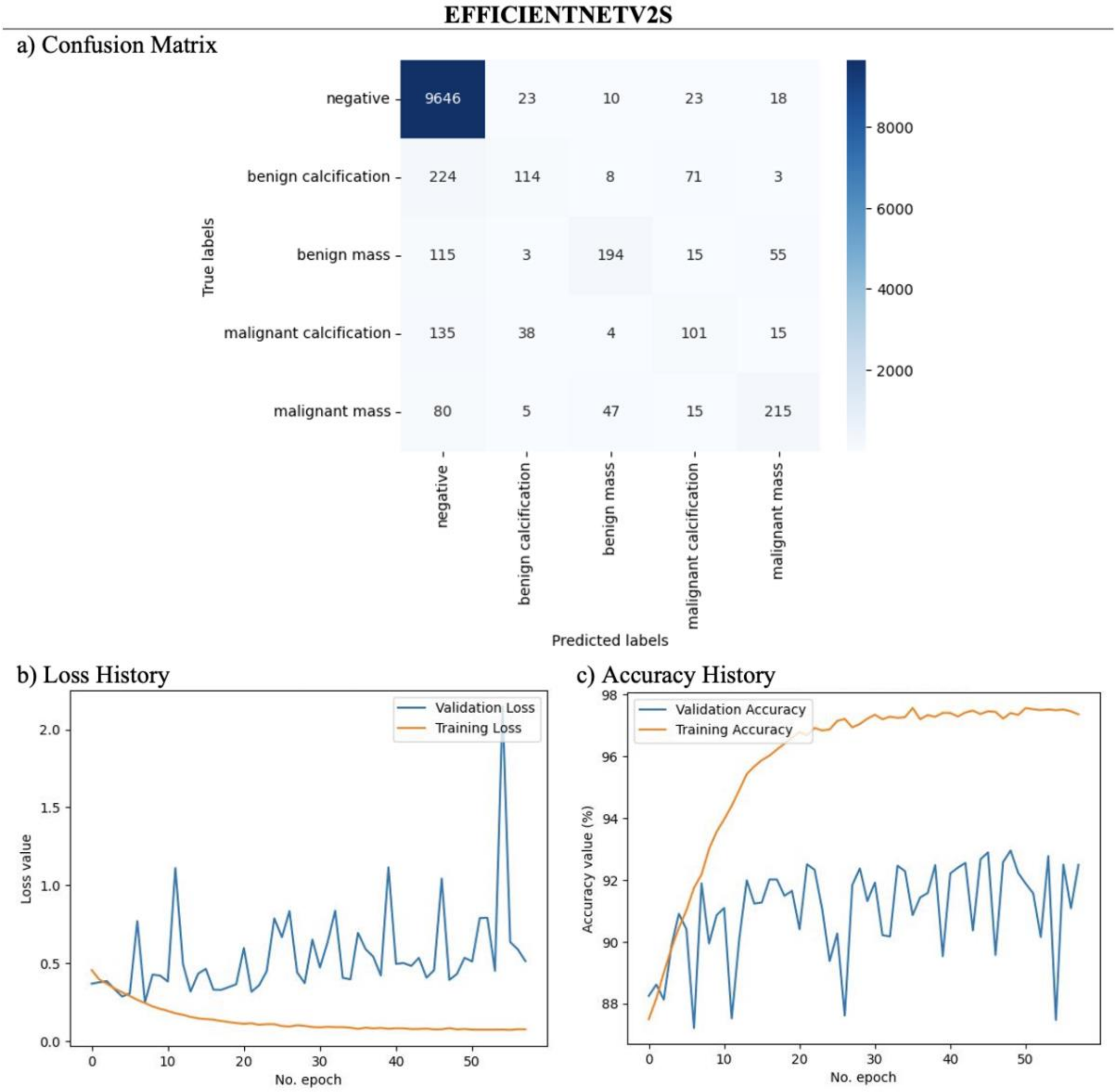
b) Loss History

c) Accuracy History

Appendix C

EfficientNetv2S: Confusion Matrix and validation loss and accuracy history

Figure of a) the confusion matrix for EfficientNetv2S's predictions on the test set, b) loss history on training and validation set, and c) accuracy history on the training and validation set.



EFFICIENTNETV2S

a) Confusion Matrix

b) Loss History

c) Accuracy History