Image Classification on Custom Dataset

1st Dias Aimyshev
School of Engineering and Digital Sciences
Nazarbayev University
Nur-Sultan, Kazakhstan
dias.aimyshev@nu.edu.kz

2nd Miras Bakytbek
School of Engineering and Digital Sciences
Nazarbayev University
Nur-Sultan, Kazakhstan
miras.bakytbek@nu.edu.kz

3rd Agakhan Baiturov
School of Engineering and Digital Sciences
Nazarbayev University
Nur-Sultan, Kazakhstan
agakhan.baiturov@nu.edu.kz

***Abstract*—Breast cancer is a significant disease caused by the uncontrolled spread of cancer cells in the breast. [1] It is essential to find cancer traits as early as possible. The conventional way of determining through a manual scan diagnosis can be improved with the help of deep learning. In this paper, we designed a CNN model that was designed for determining whether a given scan contains cancer traits or not and provides the relevant label. Our architecture consists of 5 convolutional layers with Relu activation and further batch normalization, 3 max pool layers with dropouts, and 2 fully connected layers with Softmax at the end. After training our model we reached 85% of accuracy and compared it with the Xception model that showed 86% and an initial model that showed 83% accuracy on the same dataset.**

***Index Terms*—CNN,Xception .**

I. Motivation

Breast cancer is one of the most common cancers, with the second-highest number of deaths after lung cancer. Like most of them, breast cancer is diagnosed by CT scans first, so it is very important to correctly identify it on the scans in order to take further action. With an eye to more accurately identify the presence of cancer in the images, the practice of applying image processing algorithms has begun. In order to increase the accuracy of identification and reduce the number of false-positive and false-negative diagnoses, it is now very popular to use neural networks to process images from computed tomography. They eliminate the presence of the human factor, and modern models show impressive accuracy in excess of 90The goal of our project is the optimization of the existing algorithm, the purpose of which is to classify CT scans into those in which cancer is present and those in which it is absent. Our main focus in the project is to optimize accuracy and processing speed. Below is a summary of our current progress, planned optimization, and a timeline for our next steps.

II. Review of the literature

Manual checking or immunohistochemistry checking of sentinel lymph nodes is time-consuming, tedious and inaccurate. The deep learning-based approach uses millions of images as testing samples. To be effective at the beginning, this method suggests finding tissue on a digital image and excluding the background. This idea was implemented using threshold segmentation method by converting image into HSV colour and calculating optimal values. [2] As a result, the background was reduced by 82%. To detect metastatic on optimized image fragment paper proposed using patch-based classification and heatmap-based postprocessing. At the stage of patch-based training, the model extracts a huge amount of positive and negative from a set of training optimised images. If a patch located in the tumour region that it called positive and labelled as 1. Otherwise, it is a normal patch and gets a label of 0. Aim of the model is to distinguish between these two types of patches and embed predictions into a heatmap image. In the stage of the heatmap-based model, it generates tumour probability heatmap, where each pixel has a value between 0 and 1, which is the probability of tumour. After that, each heatmap goes through post-processing to compute a slide-based and lesion-based score. As a result, a deep learning approach combined with a human pathologist has an error rate of 0.52%. [2]

One of the previous approaches is based on CNN, trained, and tested on the BreaKHIS dataset of breast cancer histopathological images. The dataset consists of 3-channel RGB images divided into benign malignant tumours, with 4 different levels of magnification factors, where 70% used for training and 30% used for testing the model. The CNN chosen is based on AlexNet, a network used to classify CIFAR images. [3] The training was conducted using different strategies to extract patches and using different classifiers for 2 cases: taking account of patient information and not. The results show a comparison of different approaches, reaching the best accuracy of 90% for the patient level. [3]

Breast Cancer Multi-classification is another model, where deep learning showed good results. The paper focuses on increasing accuracy by implementing class structure-based deep convolutional neural networks (CSDCNN). It's a non-linear learning model with no feature extraction steps into feature learning. [4] Another advantage of CSDCNN is that it can learn semantic and discriminative hierarchical features from low-level to high-level. The CSDCNN can automatically learn hierarchical feature representations and reinforce the multi-classification method. So, basically, CSDCNN is a deep learning model with multiple hidden layers. Input layer resizes the image as 256x256. Then, convolutional layer extracts features, and on the last layer, there are 64 sets of weights which are convolved with the input, which initialize from Gaussian distributions with a standard deviation of 0.0001. After that weights are passed through ReLu. Lastly, pooling layers reduce dimension and noise. [4]

This article discusses one of the most recent breast cancer screening methods using mammographic screening. In this technique, a patch classifier was applied to divide the screenings into subareas. Also, the study compared 2 popular CNN structures - VGG and Resnet. For the training of CNN, "end-to-end" approach was used, and the training process itself was divided into 2 stages: 1) Train the patch classifier using the popular ImageNet database. 2) Training of the classifier itself based on the CBIS-DDSM database, from which the mammographic screenings were selected. According to the results of independent testing based on the Digital Database for Screening Mammography (CBIS-DDSM) test set, the model showed an average accuracy of 91% and based on the full-field digital mammography (FFDM) test set, the model showed an average accuracy of 98%. [5]

Another related work is based on Back Propagation Neural Network and improved by Radial Basis Neural Network. The dataset consists of 115 training and 61 testing images of the breast tissues from Near East University Hospital. The images are converted to grayscale format, Gaussian filter is applied to reduce the noise, then approximate, horizontally detailed, vertically detailed, diagonally detailed images are received, summed to the original image, and reconstructed to the new image. This resulted in an image with more clear edges and less background noise. Despite training took 3000 epochs, due to the small size of the dataset BPNN reached accuracy only of 59.0%, adding a radial basis function (the function which depends on the distance of input from a fixed point) at the activation layer and training 80 more epochs improved accuracy to 70.4%. [6] The main idea behind all of the above articles is to identify breast cancer through different approaches, with different systems and models. As the results show, the accuracy of the models varies greatly from low (50-60%) to quite impressive (above 90%), depending on various techniques of implementation. [6]

III. Proposed deep learning model

As shown in figure 1, where BN – batch normalization layer, D – dropout layer. The initial model was a set of repeating convolutional blocks consisting of a separable-convolutional layer (kernel = 3) and batch normalization layer after it. Each subsequent block increased in the number of repetitions in the architecture by 1 (starting from 1), and the number of filters in the separable-convolutional layer increased by 2 times (starting from 32). Each convolutional block was followed by a max pooling layer (kernel = 2), and immediately after it a dropout layer. At the end there are two dense layers with the number of

Fig. 1. Initial architecture.

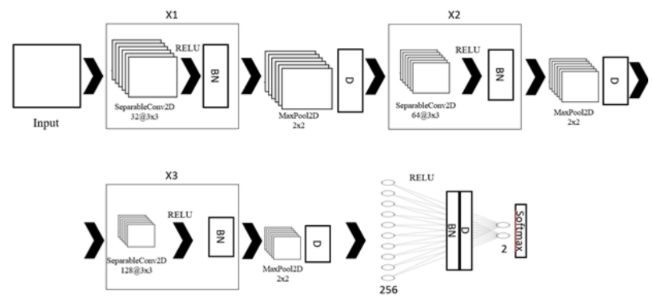
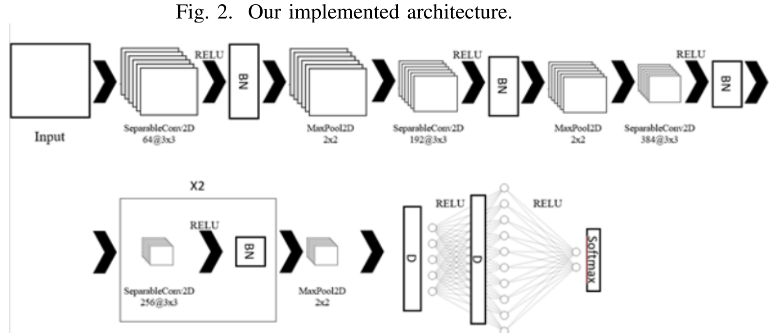


Fig. 2. Our implemented architecture.



neurons 256 and 2, respectively. In all layers, relu acts as the activation function, in the output layer - the softmax function. The hyperparameters for the model are shown in Figure 4.

To improve and optimize CNN, we decided to implement our own architecture, but not very different from the original one. As can be seen in Figure 2, our model also consists of cyclic iterations of convolutional blocks, consisting of a convolutional layer -¿ batch normalization layer -¿ max pooling layer. The number of filters can be seen in the diagram. We decided to add one density layer and increase the number of neurons. As an activation function there is also relu, and on the output layer - softmax. Also we removed all dropout layers after convolution blocks.

Also, in addition to our own model, we used the Xception architecture to compare with our model. We have not changed anything in the architecture itself, only adapted it for our purposes. Xception architecture represented in Figure 3.

IV. Experimental results with comparison to previous work

Initially, we were not satisfied with the original model performance. Despite it having only 1.2E6 parameters, the training of each epoch took 2.3E3 seconds. By implementing GPU acceleration and setting bigger batch size 128, this time was significantly reduced to 500 seconds.

The next step was implementing a different architecture to achieve better performance while being compact enough. The resulting model has more than 2 times more parameters: 2.7E6. The training still took about 500 seconds, however the performance was similar to the original model and varied in

Fig. 3. Xception architecture [7].

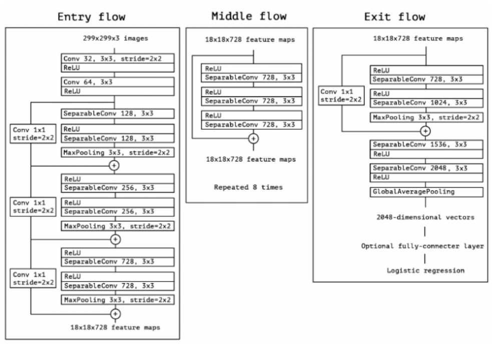


Fig. 4. Hyperparameters and functions.

Parameter/function	Values
Batch size	128
Epochs	40
Learning rate	Dynamic
Loss function	Binary cross-entropy
Optimizer	Adagrad

1-2%. The good thing was that we did not observe overfit, the training and validation scores were similar over the epochs, which can be seen on Figure 5.

Because of an unbalanced dataset, where negative tests have more images than positive tests, the difference in precision was significant: 95% and 65% respectively. After using weight-balancing technique the situation slightly improved, the precision was 92% and 70% respectively. However, sensitivity and specificity scores became more similar achieving 85% and 82% respectively in comparison with 87% and 79% for the original model. Trying different values for weight balancing didn't improve the situation, so we decided to try a completely different model.

Due to computational limitations we paid attention to lightweight CNNs. One of such was Xception. It contained 1.6E7 parameters, however because of its compactness and design the training of one epoch took only 400 seconds. Combining it with weight balancing, the accuracy achieved was at the same level 86%.

The result of the experiments is that within the given dataset and with computational limitations it was not possible to achieve much higher results. What was achieved is faster training, slightly more balanced precision score for each label, and more balanced specificity and sensitivity scores. The results are shown in Figure 6. Based on results it can be seen that Exception with weight balance showed itself slightly better in the given task. Training the models on a better dataset would probably increase the performance. In the current state it can be used to predict the negative cancer test, so if the patient did not have breast cancer, it would be confirmed. Otherwise, additional tests by specialists are strictly required.

Fig. 5. Training and validation metrics for our model.

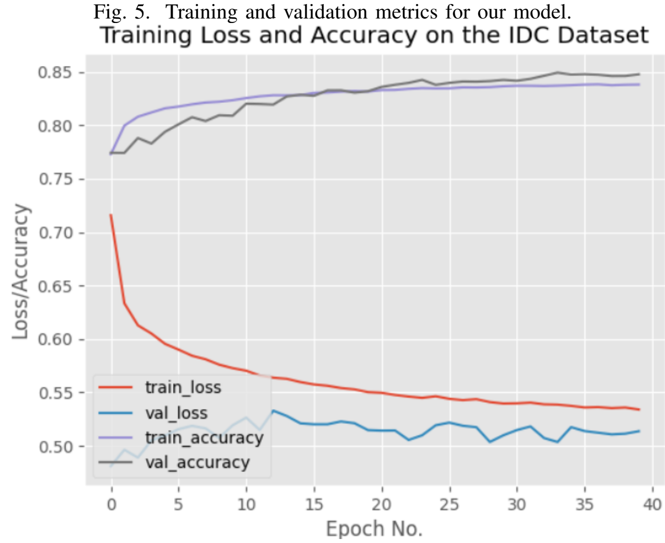


Fig. 6. Summary of all results.

Metrics	Original Model	Our Model	Weight Balance	Xception
Time to train (s)	490	500	500	400
Parameters (1E6)	1.235	2.756	2.756	16.097
Overfit	No	No	No	No
Accuracy	85%	83%	85%	86%
Precision of labels	91% and 72%	95% and 65%	92% and 70%	93% and 72%
Specificity	87%	88%	85%	87%
Sensitivity	79%	80%	82%	84%

V. Conclusion

In this paper, we described the development of CNN aimed for breast cancer detection. Our model is a combination of Convolutional layers that use ReLu activation function and batch normalization, which increases accuracy. One of the main challenges that we faced is a large unbalanced dataset with low-resolution images. This bound precision to a certain percentage, which is in case of our model is 86 %. Solution for that was weight balancing and accurate hyperparameters set up. Finally, we constructed the Xception model and initial model. After that models were trained with the same dataset. Results showed that accuracy increased after reconstruction of the initial model and close to the Xception model.

References

[1] T. J. Key, P. K. Verkasalo, and E. Banks, "Epidemiology of breast cancer," *The lancet oncology*, vol. 2, no. 3, pp. 133–140, 2001.

[2] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep learning for identifying metastatic breast cancer," *arXiv preprint arXiv:1606.05718*, 2016.

[3] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "Breast cancer histopathological image classification using convolutional neural networks," in *2016 international joint conference on neural networks (IJCNN)*. IEEE, 2016, pp. 2560–2567.

[4] Z. Han, B. Wei, Y. Zheng, Y. Yin, K. Li, and S. Li, "Breast cancer multi-classification from histopathological images with structured deep learning model," *Scientific reports*, vol. 7, no. 1, pp. 1–10, 2017.

[5] L. Shen, L. R. Margolies, J. H. Rothstein, E. Fluder, R. McBride, and W. Sieh, "Deep learning to improve breast cancer detection on screening mammography," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.
[6] S. Kaymak, A. Helwan, and D. Uzun, "Breast cancer image classification using artificial neural networks," *Procedia computer science*, vol. 120, pp. 126–131, 2017.
[7] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.