INVASIVE DUCTAL CARCINOMA (IDC) CLASSIFICATION USING ALEXNET, VGG16 AND RESNET50 NEURAL NETWORKS

INVASIVE DUCTAL CARCINOMA (IDC) CLASSIFICATION USING ALEXNET, VGG16 AND RESNET50 NEURAL NETWORKS

Merve Korkmaz, Eyyüp Güzel

Graduate School of Natural and
Applied Sciences Computer
Engineering Program
Kocaeli University
{mrvekrkmz00,
}eyyupguzel47@gmail.com

Gülüzar Durmaz

Graduate School of Natural
and Applied Sciences Aviation Science
and Technology Program
Kocaeli University
{ }gulizar.durmaz@gmail.com

SUMMARY

Training a deep convolutional neural network from scratch requires large amounts of data and considerable processing power. However, collecting large amounts of data in the medical field is costly and difficult, but this can be solved with some clever tricks such as projecting, rotating and fine-tuning pre-trained neural networks. In this paper, AlexNet, VGG16, Resnet50 neural network models are trained and compared for IDC classification. The experimental results show that in the validation set of our models, epoch three(3) and 85.52%, 79.40%, 81.28% and 50.29% in the Alexnet model with a learning rate of 0.01; with epoch six(6) 84.98%, 81.13%, 83.47% accuracy.

ABSTRACT

Training a deep convolutional neural network from scratch requires large amounts of data and significant computational power. However, collecting large amounts of data in the medical field is costly and difficult, but this can be solved with some clever tricks such as mirroring, rotating and fine-tuning pre-trained neural networks. In this article, AlexNet, VGG16, Resnet50 neural network models were trained and their results were compared in order to make the IDC classification. Experimental results show that in the validation set of our models, the accuracy values with epoch three(3) respectively are 85.52% 79.40% 81.28%, and in addition, in the Alexnet model we run with a learning rate of 0.01, the accuracy came out 50.29%; It shows that 84.98% 81.13% 83.47% accuracy is achieved with epoch six (6).

1. INTRODUCTION

Breast cancer is the most common type of cancer and cause of death in women aged 20-59 years. In general, breast cancer is considered a single disease, but there are many different types. There are non-carcinoma and carcinoma-type cancers. Figure 1 shows microscopic images of breast tissue of non-carcinoma (normal and benign) and carcinoma (InSitu and invasive). The non-carcinoma class is divided into normal and benign. Benign carcinoma represents small changes in the structure of normal breast tissue that cannot be called cancerous progression. The carcinoma class is further divided into two classes, InSitu and invasive ductal carcinoma (IDC). InSitu is also called non-invasive carcinoma or Ductal carcinoma InSitu (DCIS). In the InSitu type, the abnormal tissue in the milk ducts has not spread further into nearby breast tissues. If left untreated, carcinoma becomes invasive, in which the cancer cells move from the original site to other nodes, such as lymph nodes, or other body parts. This is why invasive carcinoma is very difficult to treat.

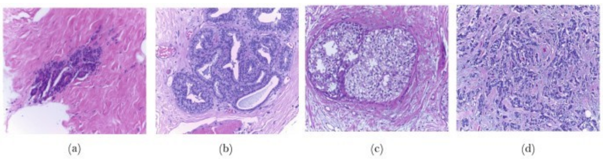


Figure 1: Microscopic images of breast tissues. (a) and (b) non-carcinoma (Normal and Benign) and (c) and (d) carcinoma tissues (InSitu and IDC) respectively [1]

Initial breast cancer detection is performed by palpation, mammography or ultrasound imaging. If these processes show growth of malignant tissue, further diagnosis is performed by breast tissue biopsy.

Breast tissue biopsies allow pathologists to examine the microscopic structure of the tissues histologically, which allows them to distinguish between non-carcinoma, carcinoma in situ or invasive ductal carcinoma. During the biopsy, these breast tissues are stained with hematoxylin and Eosin (H&E) before examination. In the next step, pathologists manually examine the (H&E) stained breast tissues, making it a misleading and time-consuming process. Furthermore, pathologists often disagree on the final diagnosis. This is because InSitu and invasive as well as normal and benign share the same type of characteristics. The disadvantages mentioned above are overcome by automating the diagnostic system. Automated diagnostic systems are robust and time efficient and are therefore very useful when there is disagreement between pathologists.

2. DATA PREPARATION

Visual data of 3 Gigabytes (GB) in size was transferred to the *Colab* platform via *the Kaggle platform* and then file paths were created using the *Pyhton Glob* library for the file location with unique ID numbers created for each patient.

Within each patient file, there are files named *0* and *1*; within each *0* and *1* file, there are texture images with *.png* extension, called *Imagepatch* in the code text. Since there are 277524 *imagepatches* in *Class0* and *Class1* files and there are processing limitations when using high-dimensional data on the Colab platform, the total number of imagepatches was reduced to 33455 by selecting patient IDs between the 60th and 90th indexes from the patient files.

In order to classify the patient images as cancerous and non-cancerous, the variable names *class0* and *class1* were determined, and the image files ending with *class0.png* in the *imagePatches* list were added to the *class0* list created at the beginning, and the image files not ending with *class0.png* were added to the *class1* list. After this process was completed, it was shown that the list lengths of *class0* and *class1* were 25252 and 8203 respectively.

In order to get sample images from the *class0* and *class1* lists, *sampled_class0* and *sampled_class1* variables were created and sample image selections were assigned to these variables with the *random.sample* method, both of which were the length of the *class1* list. Since the number of cancerous image samples, *class1*, is smaller than the number of non-cancerous image samples, *class0,* the number of reference images was prepared to be 8203. By merging the *Class0* and *Class1* lists, it was seen that their sizes were (227,227,227,3) and the length of the merged list was 16406.

In the resulting data with a list length of 16406, variables X and y were assigned to list the attributes and labels, respectively. It was shown that the lengths of both X and y variables were 16406. Using the *numpy* library and *reshape* method, the X variable containing the attributes was made four (4) dimensional in order to be suitable for the model architectures to be used.

3. MODEL TRAINING AND VALIDATION

Alexnet, VGG16 and Resnet50 neural network models from deep learning models were used in this study. The *sklearn* library was imported to start the model training and testing process and the *keras* library was imported for model selection. 75% of the prepared data was allocated for training and 25% for testing.

The model architectures of the models used were taken from the *keras* library. Alexnet, VGG16 and Resnet50 model architectures on the library include *Conv2D*, *BatchNormalization*, *MaxPool2D*, *Flatten*, *Dense* and *Dropout* common layers. *ReLU* and *Sigmoid* were used as activation functions and *Adam was* used for model optimization during model compilation. *The learning rate (lr) was* 0.000001, *binary_crossentropy* was used as the *error function*, and *accuracy* and *recall* were used as metrics.

Model training was performed with three and six epochs. In the Alexnet model training, in order to visualize the effect of varying the learning rate on the model training and test results, an additional experiment was conducted with a learning rate (lr) of 0.01 in the training with three epochs.

3.1 AlexNet

Alexnet consists of a large number of neuron-like units stacked on top of each other. It consists of multiple convolution layers, orthogonalization layers and a set of fully connected layers stacked on top of each other. The job of the convolution layers is to extract features from the input images. The convolution layer is typically followed by a partnering layer, which reduces computational complexity. [1]

The output of a neuron$f$ connected to a standard$x$ in a model$f$ (x)=tanh(x) or$f$ $(x)=(1+e^{-x})^{-1}$. In terms of training time, nonlinearities are much slower than linear ones, with *a decreasing gradient*. Based on the work of Nair and Hinton [2], these nonlinear neurons are referred to as Rectified Linear Units (ReLUs) [3]. Deep convolutional neural networks with ReLUs can be trained several times faster than their equivalents with tanh units. For this reason, ReLU and sigmoid are used as activation functions in the Alexnet architecture in this study. In addition to the Alexnet architecture shared in the article *ImageNet Classification with Deep Convolutional Neural Networks* [3] on *Figure 2,* in this study, as can be seen in *Figure 3*, the sigmoid activation function is placed in the last *dense* layer.

The main reason for using the sigmoid function is that its range of values is between 0 and 1. For this reason, it is especially used for *models that need to estimate the probability* as an output. Since the probability of anything happening is only between 0 and 1, sigmoid is the right choice. For differentiable functions, this means that the slope of the sigmoid curve can be found at any two points [4].

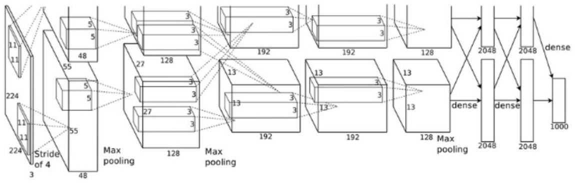
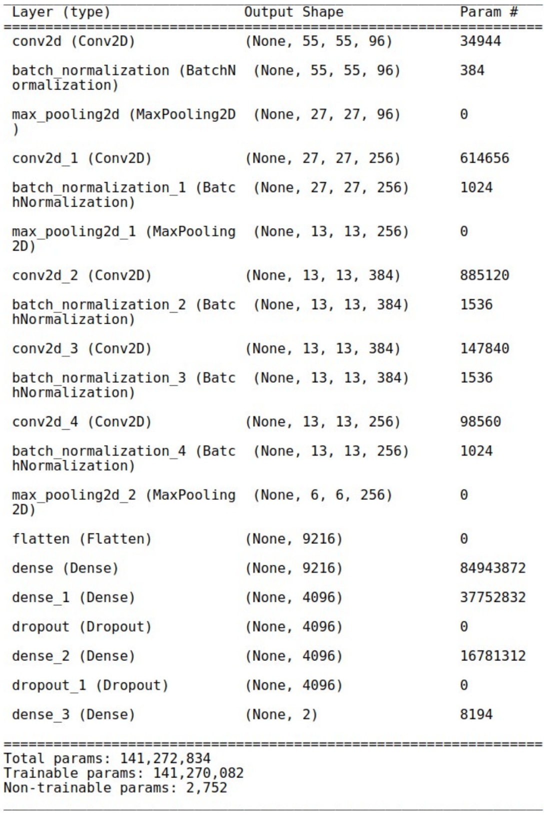


Figure 2: AlexNet Model Architecture[3]



```
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 55, 55, 96)        34944

 batch_normalization (BatchN  (None, 55, 55, 96)       384
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 27, 27, 96)       0
 )

 conv2d_1 (Conv2D)           (None, 27, 27, 256)       614656

 batch_normalization_1 (Batc  (None, 27, 27, 256)      1024
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 13, 13, 256)      0
 2D)

 conv2d_2 (Conv2D)           (None, 13, 13, 384)       885120

 batch_normalization_2 (Batc  (None, 13, 13, 384)      1536
 hNormalization)

 conv2d_3 (Conv2D)           (None, 13, 13, 384)       147840

 batch_normalization_3 (Batc  (None, 13, 13, 384)      1536
 hNormalization)

 conv2d_4 (Conv2D)           (None, 13, 13, 256)       98560

 batch_normalization_4 (Batc  (None, 13, 13, 256)      1024
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 6, 6, 256)        0
 2D)

 flatten (Flatten)           (None, 9216)              0

 dense (Dense)               (None, 9216)              84943872

 dense_1 (Dense)             (None, 4096)              37752832

 dropout (Dropout)           (None, 4096)              0

 dense_2 (Dense)             (None, 4096)              16781312

 dropout_1 (Dropout)         (None, 4096)              0

 dense_3 (Dense)             (None, 2)                 8194

=================================================================
Total params: 141,272,834
Trainable params: 141,270,082
Non-trainable params: 2,752
```

Figure 3: AlexNet Model Architecture

3.2 VGG16

In the 2014 ILSVRC competition, the most important difference between VGG-16, a simple network model developed for better results, and previous models is that binary or ternary convolution layers are followed by cooperating layers. *Figure 4* shows the VGG-16 architecture. This model consists of thirteen convolution layers, three fully connected layers, pooling, *ReLU*, *Dropout* and *Softmax* layers, totaling forty-one layers [5].

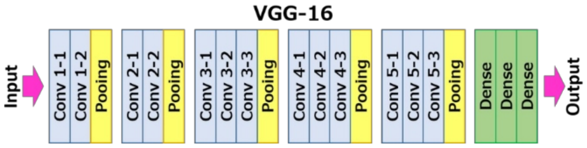
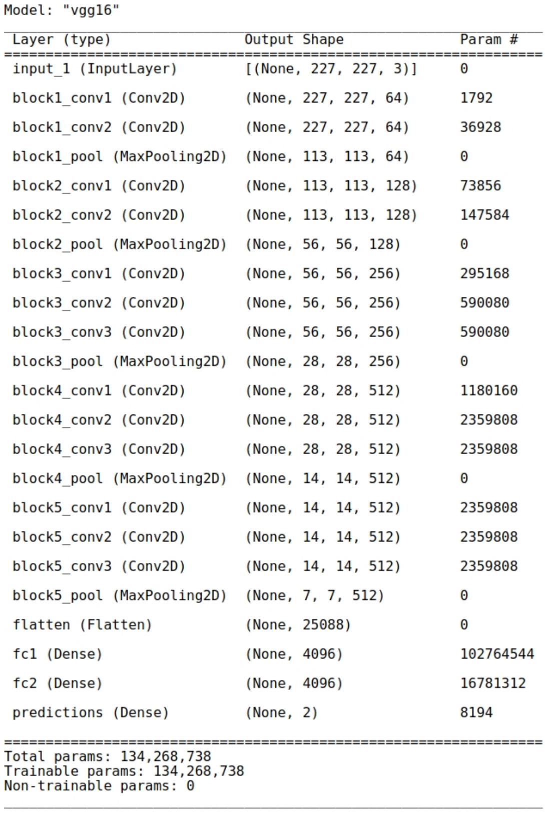


Figure 4: VGG16 Model Structure

In the VGG-16 model, convolution layers are binary or ternary is used as a convolution layer. In the fully connected layer, 7x7x512=4096 feature vectors are input. *Softmax* result with 1000 classes is calculated at the output of the fully connected layer. Approximately 138 million parameters are calculated. *The softmax* function is a more generalized logistic activation function used for multiclass classification. Since (0,1) classification will be performed in this study, the classification activation function in the last *dense* layer for the VGG16 model architecture in *Figure 5* is chosen as *a sigmoid* function.



```
Model: "vgg16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 227, 227, 3)]     0

 block1_conv1 (Conv2D)       (None, 227, 227, 64)      1792

 block1_conv2 (Conv2D)       (None, 227, 227, 64)      36928

 block1_pool (MaxPooling2D)  (None, 113, 113, 64)      0

 block2_conv1 (Conv2D)       (None, 113, 113, 128)     73856

 block2_conv2 (Conv2D)       (None, 113, 113, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 flatten (Flatten)           (None, 25088)             0

 fc1 (Dense)                 (None, 4096)              102764544

 fc2 (Dense)                 (None, 4096)              16781312

 predictions (Dense)         (None, 2)                 8194

=================================================================
Total params: 134,268,738
Trainable params: 134,268,738
Non-trainable params: 0
_________________________________________________________________
```

Figure 5: VGG16 Model Architecture

3.3 RESNET50

The ResNet50 model is a deep residual model proposed in 2015 to solve the problem of decreasing accuracy of the training set with the deepening of the network. It also has the characteristics of easy optimization and small amount of computation. The ResNet model uses the shortcut connection cross-layer link transfer method. The residual link module is shown in Figure 6. This link can combine attributes with input attributes after convolution without increasing parameters and computational complexity [6].

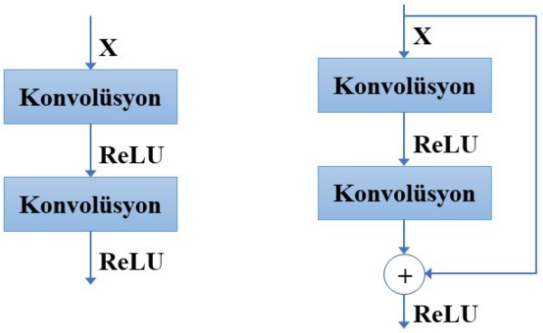


Figure 6: Standard ESA (left); shortcut connections used in ResNet architectures (right) [6].

In this study, *Binary Crossentropy* was used as the error function for the Alexnet, VGG16 and Resnet50 models.

$$H=\frac{-1}{N}\cdot\sum_{i=1}^{N} y_i\cdot \log(P(y)_i)+(1-y_i)\cdot\log(1-\boxtimes(y_i))$$

4. CLASSIFICATION RESULTS

In this section, the accuracy and error graphs, complexity matrices and classification reports of the AlexNet, VGG16 and Resnet50 neural network models trained and tested with the dataset used in the study are interpreted after three and six epochs.

In the graph shown in *Figure* 7, it can be seen that the AlexNet model, which was run with three epochs, started processing above 84% accuracy and completed the training above 85%. Test results were obtained above 85%. In the error graph shown in *Figure 8,* it is observed that the training started with a loss of less than 38% and completed with a loss of less than 37%, while the error rate decreased to 34% in the test process. The complexity matrix of the AlexNet model run with three epochs is shown in *Figure 9*. As can be seen in the classification report of the model *in Figure 10*, the average accuracy value 86%.

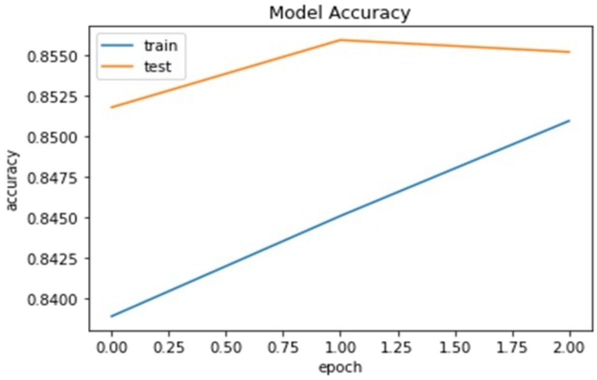


Figure 7: AlexNet Model, Epoch 3, Accuracy Graph

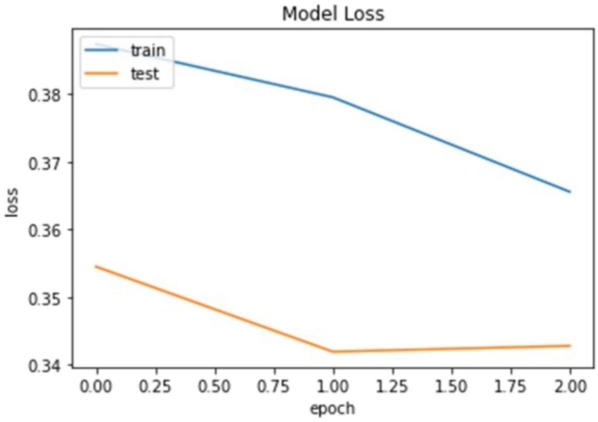


Figure 8: AlexNet Model, Epoch 3, Error Graph

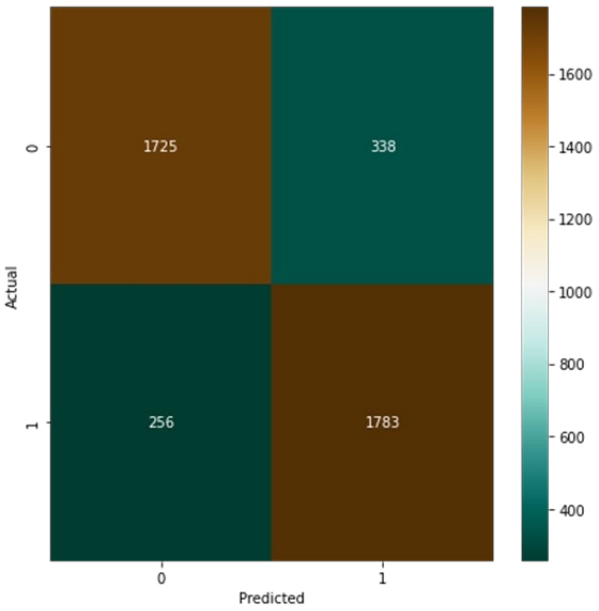
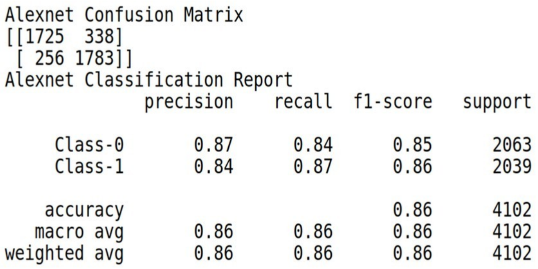


Figure 9: AlexNet Model, Epoch 3, Complexity Matrix



```
Alexnet Confusion Matrix
[[1725  338]
 [ 256 1783]]
Alexnet Classification Report
              precision    recall  f1-score   support

     Class-0       0.87      0.84      0.85      2063
     Class-1       0.84      0.87      0.86      2039

    accuracy                           0.86      4102
   macro avg       0.86      0.86      0.86      4102
weighted avg       0.86      0.86      0.86      4102
```

Figure 10: AlexNet Model, Epoch 3, Classification Report

In order to observe the results by changing one of the learning parameters on a model run with the same number of epochs, the AlexNet model run with three epochs was trained and tested once with a learning rate (lr) of 0.01. *Figure 11, Figure 12, Figure 13* and *Figure 14* show the accuracy, error graphs, complexity matrix and classification report respectively. The 0.01% learning rate was chosen randomly to observe how the model reacts when larger values are chosen. As can be seen in the accuracy graph, model training starts at 70% and

It ended by falling below 50%. It is seen that the wrong working method is used here. Depending on the learning rate used, it is observed that the decision mechanism is atrophied. In the error graph, it is observed that the error rate of the training data is increasing, but there is no change in the error rate of the test data. The model failed to make any prediction on the test data. It is observed that learning does not take place here.

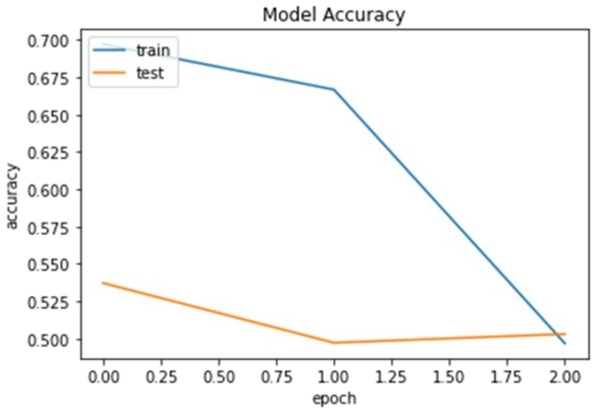


Figure 11: AlexNet Model, Epoch 3, lr: 0.01 Accuracy Graph

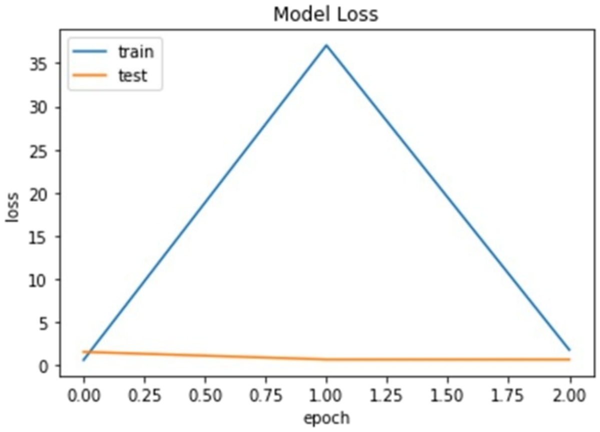


Figure 12: AlexNet Model, Epoch 3, lr:0.01, Error Graph

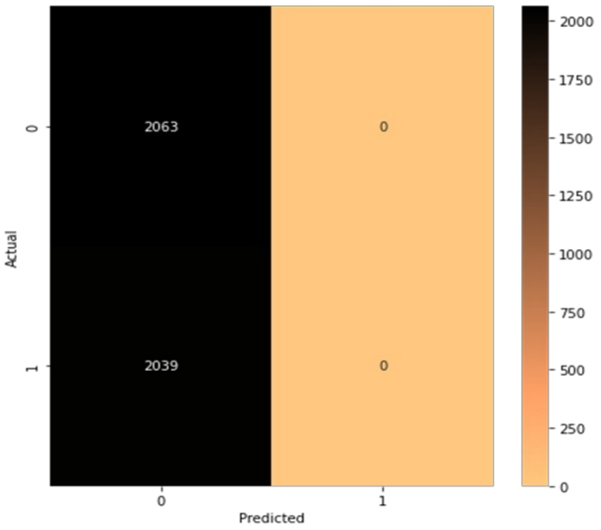
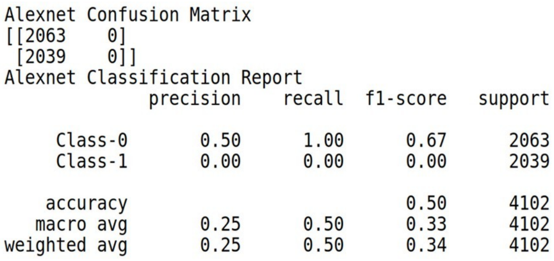


Figure 13: AlexNet Model, Epoch 3, lr:0.01, Complexity Matrix



```
Alexnet Confusion Matrix
[[2063    0]
 [2039    0]]
Alexnet Classification Report
              precision    recall  f1-score   support

     Class-0       0.50      1.00      0.67      2063
     Class-1       0.00      0.00      0.00      2039

    accuracy                           0.50      4102
   macro avg       0.25      0.50      0.33      4102
weighted avg       0.25      0.50      0.34      4102
```

Figure 14: AlexNet Model, Epoch 3, lr:0.01, Classification Report

In the graph shown in *Figure 15,* it can be seen that the VGG16 model, which was run with three epochs, started processing above 68% accuracy and completed the training above 78%. The test results were also above 78%. The error graph shown in *Figure 16* shows that the training
It is observed that it starts with a loss of 60% and finishes with a loss of less than 47%, while in the test process, the error rate decreased to 42%. The complexity matrix of the VGG16 model run with three epochs is shown in *Figure 17*. A s can be seen in the classification report of the model in *Figure 18*, the average accuracy value
79%.

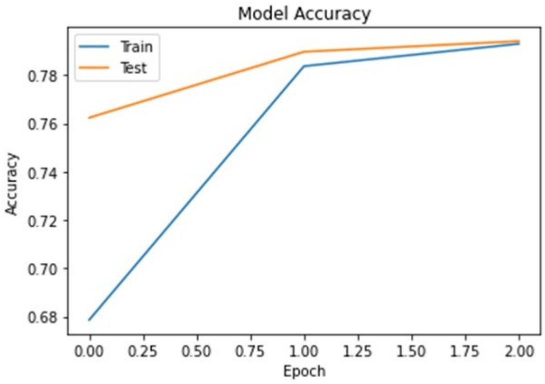


Figure 15: VGG16 Model, Epoch 3, Accuracy Graph

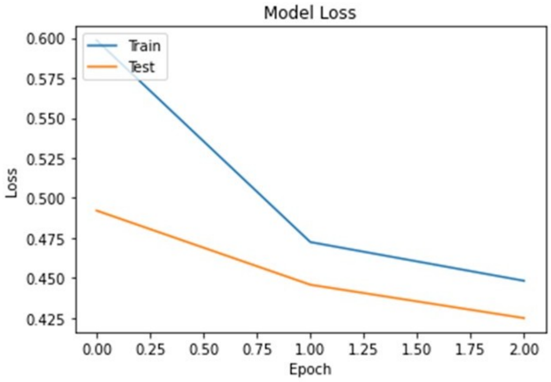


Figure 16: VGG16 Model, Epoch 3, Error Graph

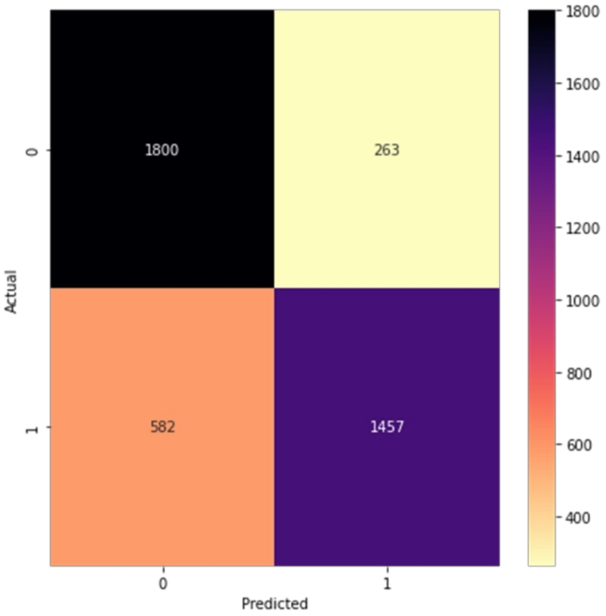
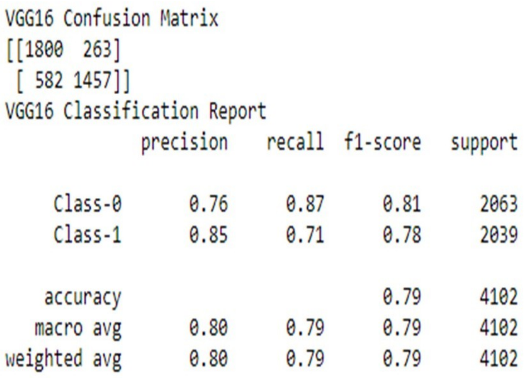


Figure 17: VGG16 Model , Epoch 3, Complexity Matrix



```
VGG16 Confusion Matrix
[[1800  263]
 [ 582 1457]]
VGG16 Classification Report
              precision    recall  f1-score   support

     Class-0       0.76      0.87      0.81      2063
     Class-1       0.85      0.71      0.78      2039

    accuracy                           0.79      4102
   macro avg       0.80      0.79      0.79      4102
weighted avg       0.80      0.79      0.79      4102
```

Figure 18: VGG16 Model, Epoch 3, Classification Report

In the graph shown in *Figure 19,* it can be seen that the ResNet50 model, which was run with three epochs, started processing above 65% accuracy and completed the training above 80%. Test results were obtained above 81%. In the error graph shown in *Figure 20,* it is observed that the training started with a loss of 60% and completed with a loss of less than 40%, while in the test process, the error rate decreased to 45%. The complexity matrix of the ResNet50 model run with three epochs is shown in *Figure 21*. As can be seen in the classification report of the model *in Figure 22*, the average accuracy value is 81%.

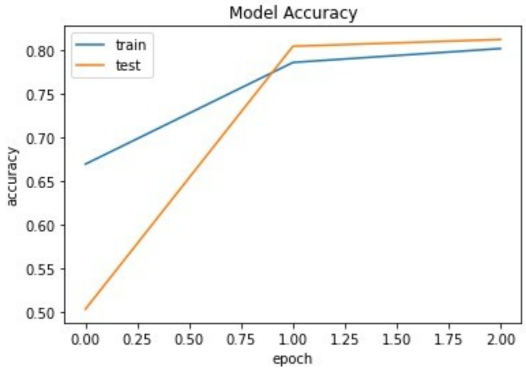


Figure 19: Resnet50 Model, Epoch 3, Accuracy Graph

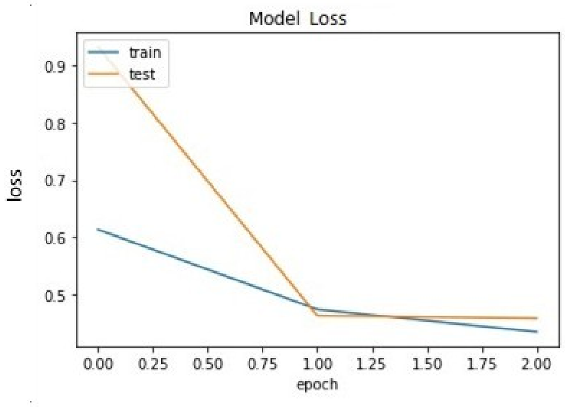


Figure 20: Resnet50 Model, Epoch 3, Error Graph

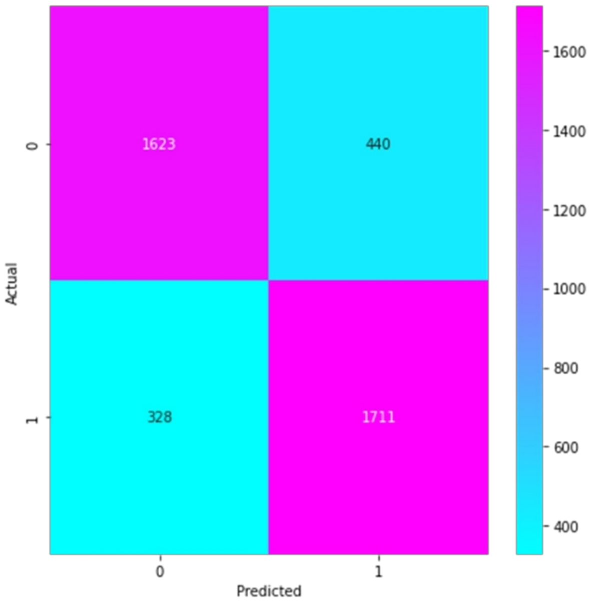
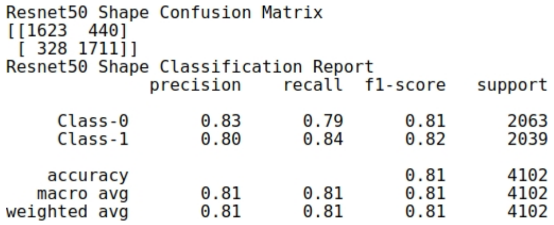


Figure 21: Resnet50 Model, Epoch 3, Complexity Matrix



```
Resnet50 Shape Confusion Matrix
[[1623  440]
 [ 328 1711]]
Resnet50 Shape Classification Report
              precision    recall  f1-score   support

     Class-0       0.83      0.79      0.81      2063
     Class-1       0.80      0.84      0.82      2039

    accuracy                           0.81      4102
   macro avg       0.81      0.81      0.81      4102
weighted avg       0.81      0.81      0.81      4102
```

Figure 22: Resnet50 Model, Epoch 3, Classification Report

The accuracy and error graphs, complexity matrix and classification report of the AlexNet model run with six epochs are shown in *Figure 23, Figure 24, Figure 25* and *Figure 26* respectively. It is seen that the accuracy values of the model are above 82% and 85% in the training and test results with six epochs. The error values are below 36% in the test results. The average accuracy of the model is 85%.

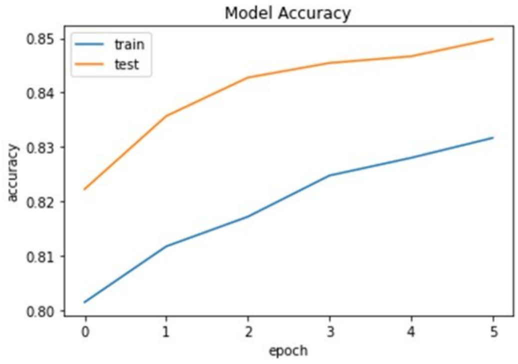


Figure 23: AlexNet Model, Epoch 6, Accuracy Graph

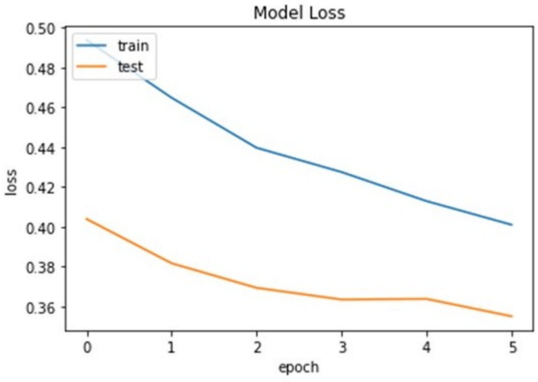


Figure 24: AlexNet Model, Epoch 6, Error Graph

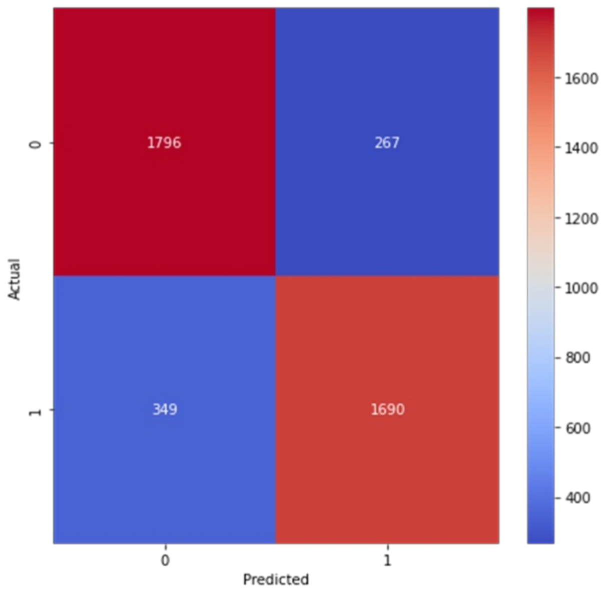
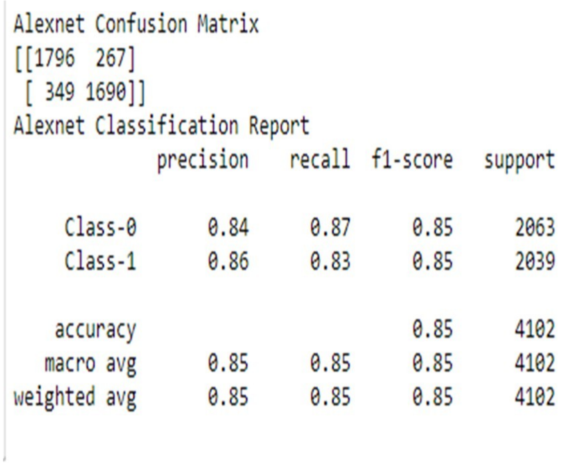


Figure 25: AlexNet Model, Epoch 6, Complexity Matrix



```
Alexnet Confusion Matrix
[[1796  267]
 [ 349 1690]]
Alexnet Classification Report
              precision    recall  f1-score   support

     Class-0       0.84      0.87      0.85      2063
     Class-1       0.86      0.83      0.85      2039

    accuracy                           0.85      4102
   macro avg       0.85      0.85      0.85      4102
weighted avg       0.85      0.85      0.85      4102
```

Figure 26: AlexNet Model, Epoch 6, Classification Report

In the graph shown in *Figure 27,* it is seen that the VGG16 model, which was run with six epochs, started to operate with an accuracy rate below 682% and completed the training above 76%. The test results were also above 80%. The error graph shown in *Figure 28* shows that the training It is observed that it starts with a loss of 60% and finishes with a loss of less than 45%, while in the test process, the error rate decreased to 42%. The complexity matrix of the VGG16 model run with six epochs is shown in *Figure 29*. A s can be seen in the classification report of the model in *Figure 30*, the average accuracy value 81%.

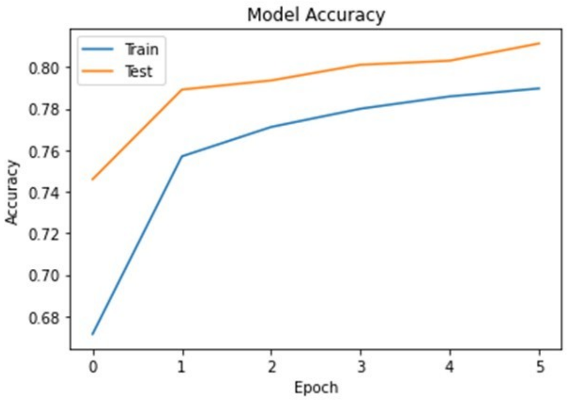


Figure 27: VGG16 Model, Epoch 6, Accuracy Graph

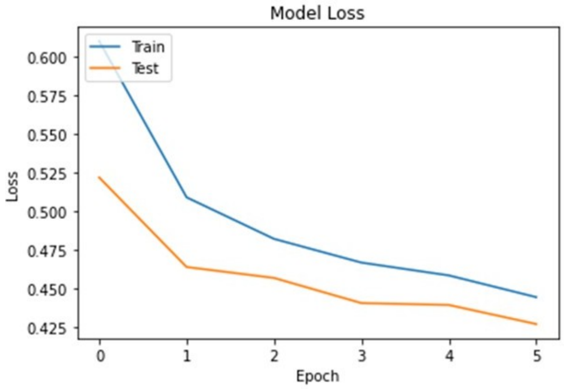


Figure 28: VGG16 Model, Epoch 6, Error Graph

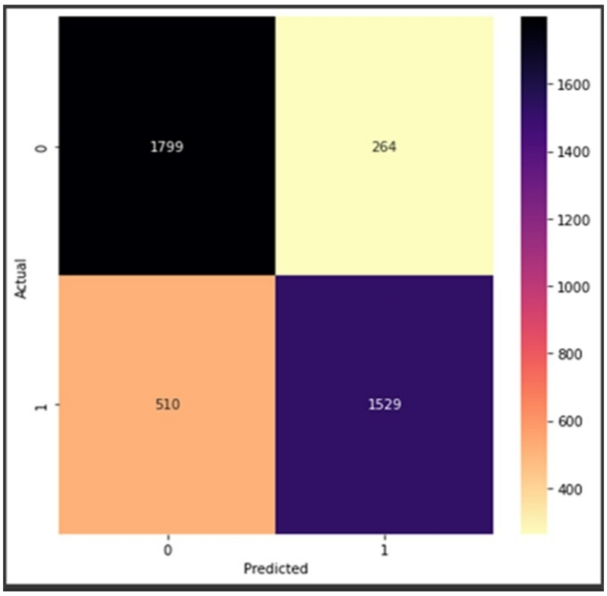
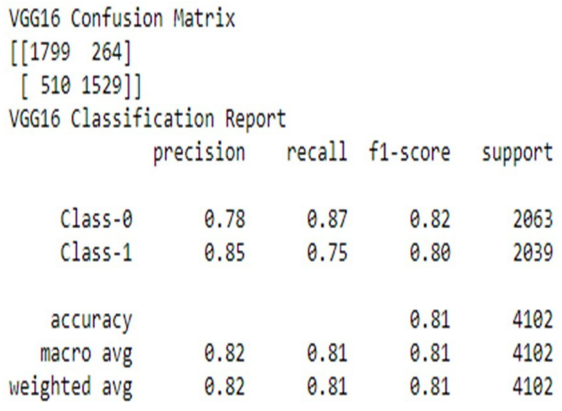


Figure 29: VGG16 Model, Epoch 6, Comparison Matrix



```
VGG16 Confusion Matrix
[[1799  264]
 [ 510 1529]]
VGG16 Classification Report
              precision    recall  f1-score   support

     Class-0       0.78      0.87      0.82      2063
     Class-1       0.85      0.75      0.80      2039

    accuracy                           0.81      4102
   macro avg       0.82      0.81      0.81      4102
weighted avg       0.82      0.81      0.81      4102
```

Figure 30 : VGG16 Model, Epoch 6, Classification Report

In the graph shown in *Figure 31,* it can be seen that the ResNet50 model, which was run with six epochs, started processing above 62% accuracy and completed the training above 78%. The test results were obtained above 82%. In the error graph shown in *Figure 32,* it is observed that the training started with a loss of less than 70% and completed with a loss of less than 45%, while the error rate decreased to 40% in the test process. The complexity matrix of the ResNet50 model run with six epochs is shown in *Figure 33*. A s can be seen in the classification report of the model in *Figure 34*, the average accuracy value
83%.

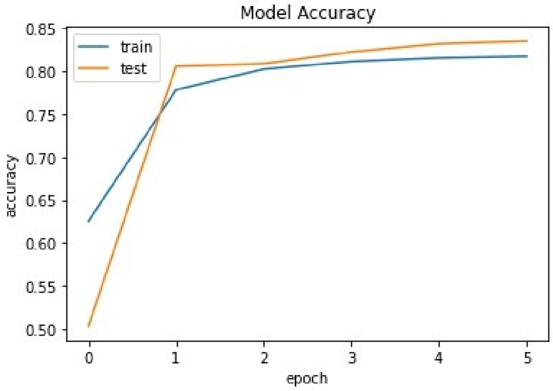


Figure 31: Resnet50 Model, Epoch 6, Accuracy Graph

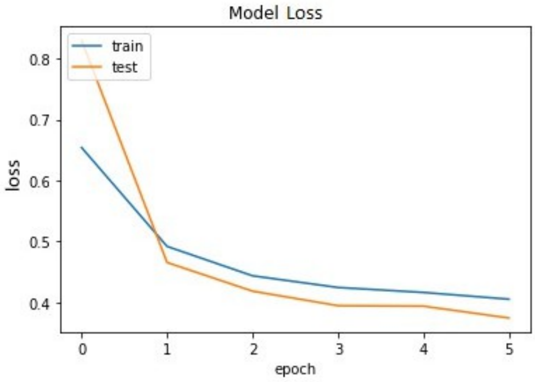


Figure 32: Resnet50 Model, Epoch 6, Error Graph

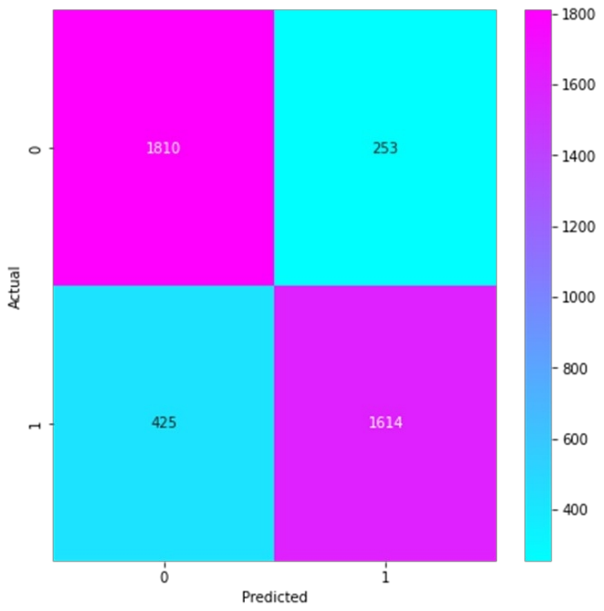
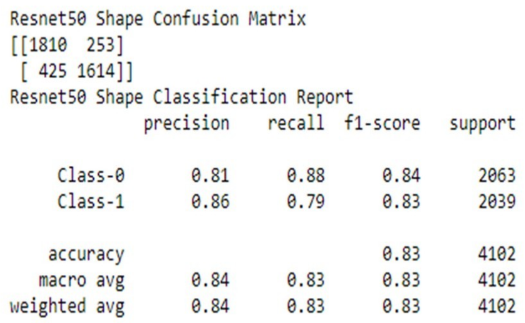


Figure 33: Resnet50 Model, Epoch 6, Comparison Matrix



```
Resnet50 Shape Confusion Matrix
[[1810  253]
 [ 425 1614]]
Resnet50 Shape Classification Report
              precision    recall  f1-score   support

     Class-0       0.81      0.88      0.84      2063
     Class-1       0.86      0.79      0.83      2039

    accuracy                           0.83      4102
   macro avg       0.84      0.83      0.83      4102
weighted avg       0.84      0.83      0.83      4102
```

Figure 34: Resnet50 Model, Epoch 6, Classification Report

5. CONCLUSION

This one Study, Using breast tissue image data obtained from https://www.kaggle.com/paultimothymooney/breast-histopathology-images/metadata, AlexNet, VGG16 and Resnet50 neural network models were used to compare the classification results of IDC and non-IDC tissues. Experimental results on models obtained from the Python.keras library show that the models used provide overall accuracy in the range of 79% to 86%. As part of our future work, we aim to extend this research and improve our performance by using deep architectures built specifically for cancer classification.

6. REFERENCES

[1] H. M. Ahmad, S. Ghuffar, and K. Khurshid, *Classification of Breast Cancer Histology Images Using Transfer Learning*. Springer International Publishing, 2019.

[2] M. J. Brown, L. A. Hutchinson, M. J. Rainbow, K. J. Deluzio, and A. R. De Asha, "A comparison of self-selected walking speeds and walking speed variability when data are collected during repeated discrete trials and during continuous walking," *J. Appl. Biomech.*, vol. 33, no. 5, pp. 384-387, 2017, doi: 10.1123/jab.2016-0355.

[3] Y. Zhang, J. Gao, and H. Zhou, "Breeds Classification with Deep Convolutional Neural Network," *PervasiveHealth Pervasive Comput. Technol. Healthc.*, pp. 145-151, 2020, doi: 10.1145/3383972.3383975.

[4] "Activation Functions in Neural Networks| by SAGAR SHARMA| Towards Data Science." https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 (accessed Jan. 03, 2022).

[5] F. Dogan and I. Turkoglu, "Deep Learning Algorithms Leaf The Comparison Of Leaf Classification Performance Of Deep Learning Algorithms," *Sak. Univ. J. Comput. Inf. Sci.*, vol. 1, no. April, pp. 10-21, 2018.

[6] B. Bayram and F. Ozoglu, "Full Text Proceedings i," no. June, 2019.