CLASSIFICATION OF MORPHOLOGY OF SPERM CELLS WITH DEEP LEARNING

Kamil BAŞKUT, Ömer Lütfü TORTUMLU
Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi, 34220 Istanbul, Türkiye
{kamil.baskut1,omertortumlu}@gmail.com

***Abstract*—In this study, it is aimed to establish a decision mechanism in which sperm morphological analysis can be carried out autonomously. The datasets were used in the fold structure. The HuSHeM dataset consists of 4 classes, while the SMIDS dataset consists of 3 classes. In this study, the shapes of the picture in the dataset are examined. Within the scope of the project, it is planned to test the datasets in GoogleNet, Xception and Mobilenet networks. These networks will be used directly on keras and the most successful parameters will be calculated for each data and network by testing the optimizer, activation function in the last layer. The data augmentation rate will be tested to increase the success rate of the results.**

***Index Terms*—Deep Learning, CLASSIFICATION OF MORPHOLOGY OF SPERM CELLS**

I. INTRODUCTION

A. *The aim of the study*

It is aimed to establish a decision-making mechanism for autonomic testing by looking at the shape of sperm cells for the diagnosis of infertility in men. In this way, it is planned to reduce the margin of error and obtain objective results.

B. *Used Methods*

HuSHeM and SMIDS data sets are divided into 5 different fold. Each fold is divided into class structure according to the shapes of the pictures. 4 classes were created for the HuSHeM dataset and 3 classes were created for the SMIDS dataset. This dataset structure was tested for each fold of GoogleNet, Xception and Mobilenet and the average success was calculated. Data augmentation was applied to improve these results.

Possible data sets that can be used in the system *HuSHeM*,*SMIDS*.

Possible networks that can be used in the system*GoogleNet*, *Xception*, *Mobilenet*.

II. DATASET CHOOSING

In the model, HuSHeM and SMIDS datasets which are colored with chemicals in laboratory environment are used.

A. *Hushem Dataset*

It consists of 5 different fold. Each fold is divided into 4 different classes according to the shapes of the pictures. The pixel values of the images in the Hushem dataset are 131x131.

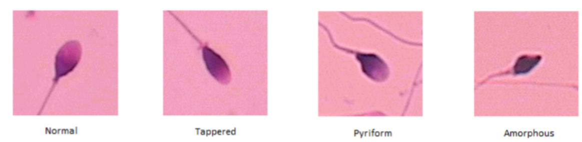

Fig. 1 HuSHeM Dataset

B. *Smids Dataset*

It consists of 5 different fold. Each fold is divided into 3 different classes according to the shapes of the pictures. The pixel values of the images in the Smids dataset are 170x170.

Fig. 2 SMIDS Dataset

III. NETWORK SELECT

The data sets were tested for each fold with GoogleNet, Xception and Mobilenet networks. Optimizer and activation function will try to determine the most successful network and parameters.

Conventional neural networks to be tested are listed below.

- GoogleNet
- Xception
- Mobilenet

Conventional neural networks are a deep learning algorithm that can take pictures as input and classify inputs with the help of learnable weights. The networks to be tested within the scope of the project are some kind of conventional neural network and will be tested in python language through the keras library. Weights of networks trained with imagenet data were used in all networks used during the study.

A. GoogleNet

GoogleNet is a kind of convolutional neural network model. It consists of numerous convolution and maximum docking steps. GoogleNet, which was trained with imagenet data in this project, was used by changing the number of classes in the last layer over keras. Input shapes are adjusted according to the pixel value of the images in the data set.

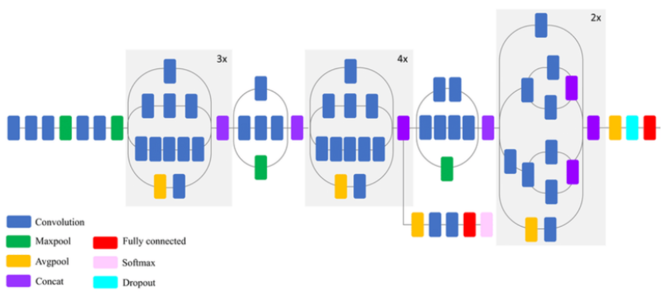

Fig. 3 GoogleNet

B. Xception

Developed in 2017 by François Chollet. The architecture of the Xception model is based on the input stream, Medium Stream, and Output Stream.

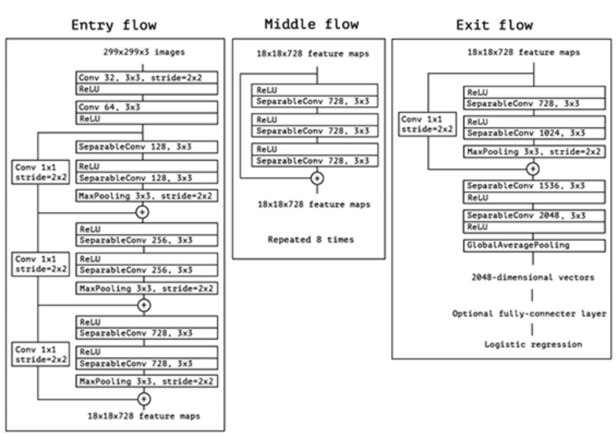

Fig. 4 Xception

C. Mobilenet

MobileNet is based on an aerodynamic architecture that uses separable curves as depth to form light deep neural networks. In this project, the Mobilenet network, which was trained with imagenet data, was used by changing the number of classes in the last layer over keras. Input shapes were adjusted according to the pixel value of the images in the data set.

	Type / Stride	Filter Shape	Input Size
	Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
	Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
	Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
	Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
	Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
	Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
	Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
	Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
	Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
	Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
	Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
	Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
	Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$	Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
	Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
	Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
	Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
	Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
	Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
	Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
	FC / s1	1024×1000	$1 \times 1 \times 1024$
	Softmax / s1	Classifier	$1 \times 1 \times 1000$

Fig. 5 MobileNet

IV. Data Augmentation

It was planned to increase the success rate by using more pictures during the training. It is planned to generate new images at a certain rate for an existing image by using Data Augmentation function.

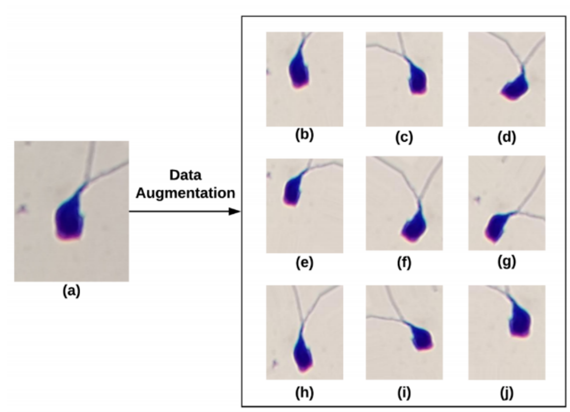

Fig. 6 Data Augmentation

V. Conclusion

In this study, networks were tested with different optimizer and activation parameters for data sets. As a result of combining all the studies, the parameters with maximum success were analyzed.

A. Optimizer Test

In this study, the networks were tested with different optimizer parameters for data sets. Used optimizers adam, adamax and sgd. The most successful optimizers in the networks tested for datasets have been recorded.

B. Activation Test

In this study, the networks were tested with different activation parameters for data sets. Results vary depending on the activation used. Activations with the highest success were recorded in the networks tested for datasets. Used activations softmax,softplus and sigmoid.

	GoogleNet-sgd	Xception-adam	MobileNet-adamax
Softmax	72	65	60
Softplus	71	65	70
Sigmoid	71	74	36
Hard-sigmoid	30	36	43

Fig. 7 Activaton test for HuSHeM dataset

	GoogleNet-sgd	Xception-adam	MobileNet-adamax
Softmax	79	82	76
Softplus	84	80	84
Sigmoid	85	74	37
Hard-sigmoid	53	36	41

Fig. 8 Activaton test for SMIDS dataset

VI. Data Augmentation

As a result of combining all the studies, the parameters with maximum success were analyzed. It is planned to increase the maximum success by multiplying the pictures. For this purpose, previously tested parameters that provide the highest success for data sets and networks were used.

VII. Confusion Matrix

The Confusion matrix was used to give information about the accuracy of the estimates. It is planned to see which class the pictures predicted with the confusion matrix belong to and which classes are compared. Confusion matrix accuracy, precision and recall calculations were used in this study.

A. Accuracy

It is the ratio of accurate estimates to the entire test class. Used to represent the success of our network in the data set.

B. Precision

It's how accurately we estimate from all classes. The model is used to calculate the rate of knowing the truth

Dataset	Network	Activation	Optimizer	Accuracy	Precision	Recall
HuSHeM	GoogleNet	Softplus	sgd	72	0.72	0.75
	Xception	Sigmoid	adam	73	0.73	0.79
SMIDS	GoogleNet	Sigmoid	sgd	87	0.87	0.88
	GoogleNet	Softplus	sgd	83	0.83	0.87

Fig. 9 Confusion matrix

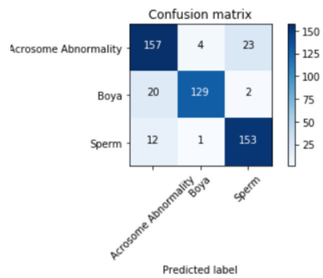

Fig. 10 SMIDS GoogleNet Sigmoid sgd

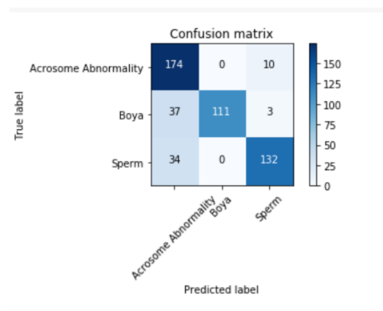

Fig. 11 SMIDS GoogleNet Softplus Sgd

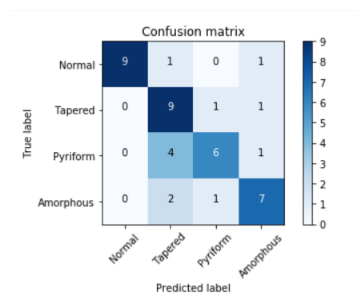

Fig. 12 HuSHeM GoogleNet Softplus Sgd

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14]

References

[1] M. LLC. (1999) Ms windows nt kernel description. [Online]. Available: http://web.archive.org/web/20080207010024

[2] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 935–942.

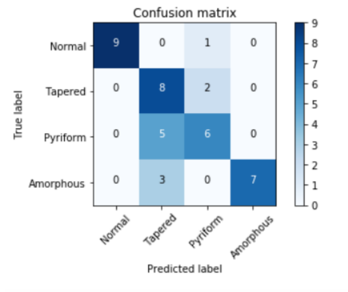

Fig. 13 HuSHeM Xception Sigmoid Adam

[3] H. Guo and H. L. Viktor, "Learning from imbalanced data sets with boosting and data generation: the databoost-im approach," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 30–39, 2004.

[4] K. N. Rao, "A novel class imbalance learning method using subset filtering," 2012.

[5] ——, "A novel class imbalance learning method using subset filtering," 2012.

[6] M. Knauff and J. Nejasmic, "An efficiency comparison of document preparation systems used in academic research and development," *PloS one*, vol. 9, no. 12, p. e115069, 2014.

[7] F. Brischoux and P. Legagneux, "Don't format manuscripts," *The Scientist*, vol. 23, no. 7, p. 24, 2009.

[8] G. Inc. (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. [Online]. Available: https://arxiv.org/pdf/1704.04861.pdf

[9] F. Chollet. (2017) Xception: Deep learning with depthwise separable convolutions. [Online]. Available: https://arxiv.org/pdf/1610.02357v3.pdf

[10] Google. (2014) Going deeper with convolutions. [Online]. Available: https://arxiv.org/pdf/1409.4842v1.pdf

[11] K. Team. The python deep learning library. [Online]. Available: https://keras.io/.

[12] A. Inc. The world's most popular data science platform. [Online]. Available: https://www.anaconda.com/

[13] JetBrains. The python ide for professional developers by jetbrains. [Online]. Available: https://www.jetbrains.com/pycharm/

[14] P. S. Foundation. Python is a programming language. [Online]. Available: https://www.python.org/