

Automated Pterygium Detection in Anterior Segment Photographed Images using Deep Convolutional Neural Network

Assyareefah Hudaibah Saad¹, N Syahira M Zamani¹, W Mimi Diyana W Zaki¹, Aqilah Baseri Huddin¹, Aini Hussain¹

¹Center for Integrated Systems Engineering and Advanced Technologies (INTEGRA), Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Selangor, Malaysia, wmdiyana@ukm.edu.my

ABSTRACT

This paper presents an automated detection of pterygium disease in anterior segmented photographed images. The five main steps are implemented as follows; pre-processing of the ASPI database, model development using a proposed deep convolutional neural network (DCNN) architecture, evaluation of the tested models, analysis of the models' efficiencies and comparison of model performance. The proposed DCNN architecture consists of six learned layers that are three convolutional and three fully connected layers. This experimental work focuses on the identification of the best combination of hyperparameters for the proposed DCNN architecture. The optimal hyperparameter combination with learning rates of 0.001 and 0.0001 and epoch values of 20 and 25, provides the best performance of 94.09% accuracy, 93.93% F1 score, and 88.32% Matthew correlation coefficient with the highest area under the curve value for an average fold of 95%. The results prove that the proposed DCNN architecture has a promising potential to be applied in the development of future pterygium screening tools.

Key words: Anterior Segment Photographed Images, Deep Convolutional Neural Network, Pterygium.

1. INTRODUCTION

Pterygium is caused by the invasion of non-cancerous tissue on the surface of the cornea [1]. This ocular disease can occur in humans due to certain factors such as age, dryness of the eyes, dust and excessive ultraviolet light (UV) radiation reception. Reference [2] reported that dry eyes are associated with abnormalities in the corneal eyelid film and inflammatory changes following the entire ocular surface, including the adnexa, conjunctiva and cornea. Pterygium has long been known as a disease caused by the environment and excessive UV radiation for an extended length of time. These factors contribute greatly in the development of pterygium [3]. Pterygium is prevalent in people who are often exposed to the UV radiation, such as farmers, fishermen and athletes

who live near the equator, such as in Malaysia, Australia, South America and Africa.

According to [4], the high pterygium prevalence rate in the Asian continent amongst the equator population indicates that 1 out of 10 adults in the age range of 21 and above potentially suffer from this disease. This rate increases with age and engagement in active outdoor activities in sunny and hot areas. However, no significant gender differences have been observed. The prevalence rate of pterygium varies considerably depending on race, age, gender and geographical location. However, this observation differs from the current situation. The occurrence of this issue may reflect causation effect due to race (genetic) and/or the environment. In addition, the statistics shows a percentage of 23.4% of people suffering from pterygium in the tropical area of Africa located north of the equator and this percentage can be considered to be very high. This situation is believed to be caused by extremely hot weather factors.

Reference [5] revealed that chronic eye inflammation results from external elements with the involvement of pterygium pathogenesis capable of producing chronic inflammation, which causes edema inflammation, and in attempt to repair the cells, give rise to angiogenesis. Thus, pterygium can be defined as an ocular disease that is categorised under the ophthmo-heliosis group. It is the invasion and growth of fibrovascular tissue towards the cornea that is not categorised as cancerous in conjunctivitis [1]. According to the study of [5], benign growth usually has a wedge shape and does not cause problems nor require any treatment. However, when fibrovascular tissue growth has closed most of the cornea and inhibits vision, treatment for tissue removal must be carried out. According to [6], a standard classification for pterygium disease has not been fully identified to date. Normally, physical examinations are performed initially to detect the presence of pterygium by using a slit lamp camera for determining typical limbal growth in the palpebral fissure area of the eye. A complete eye examination requires an ophthalmic specialist to evaluate the condition of the patient's eyes by using the latest visual impact assessment and eye reflex index [7].

Previous studies have examined the methods of pterygium identification and classification, especially based on medical image analysis using computer-assisted systems, such as video-keratography and tomography. These techniques have a positive impact on medical research but entail relatively high costs. The anterior segment photographed image (ASPI) is the most apparent image in ophthalmic photography but is less specific. This type of photographic technique is called 'outdoor photography' [8]. ASPI is used in conventional macrophotography tools and techniques to document external but not deep images and involves a specific eye image with its structure. Outdoor photography can produce images of the conjunctiva, cornea and iris with magnification of up to two times of live sizes. The highest ASPI picture quality can be achieved by using the slit lamp photographic technique. In general, ASPI, taken with ordinary digital cameras and the slit lamp photographic techniques can improve the quality of eye disease documentation [9].

Researchers in [10] utilised ASPIs to run a pterygium screening system that employed corneal features. The study proposed four descriptive characteristics to differentiate between pterygium and normal ASPI. The combination of shape features was tested using ANN and support vector machine (SVM). The proposed system can overcome the lack of quantitative tools for pterygium screening. The investigation conducted by [11] also supported the use of image processing-based systems to examine the presence of pterygium. However, several challenges, including image variations, illumination imbalances due to the use of low-resolution cameras and variations in pterygium tissue form, needed to be addressed in the development of these systems. Besides, [12] used the ASPI approach to evaluate pterygium using the digital image processing (DIP) method. The proposed approach demonstrated good results in segmenting pterygium and normal ASPIs.

In the era of Industrial Revolution 4.0, neural networks have been widely used in machine learning, especially in the Internet of Things and cyber-physical systems. Neural networks can be defined as human-based thinker models [13]. Such networks function by accepting signals from an entry link, computing the new activation level and sending it as an output signal through the output link. The entry signals can be any raw data or other neural output results, and potential output signals constitute the final solution to the main problem or the data entry to other neurons [14]. Typical use of the neural network was applied several decades ago, and it involved AI and was known as ANN [15]. As reported in [16], there are twenty-seven types of ANN available. Each of them has a unique architecture and different techniques of modelling and learning processes. In addition, machine learning is a branch of AI that provides computers the capability to learn without being programmed as a whole

[17]. According to [18], the machine learning approach has many positive impacts on the biomedical field, and numerous studies have been conducted using this approach in medical image analyses. Many algorithms for nucleus segmentation, breast cancer image classification and other medical applications have been successfully developed. According to [19], neural networks are closely related to associative memory, and deep artificial neural network (ANN) approaches have been successfully used in machine learning since the 1940s. Deep learning in ANN provides numerical results that exhibit high precision and accuracy in solving welding problems [20]. The use of deep convolutional neural network (DCNN) in ANN has the potential to generate high-quality predictions and assumptions by altering various parameters in the production of model classes.

Deep learning is a branch of machine learning that has the potential to produce excellent results in identification and classification [19]. Deep learning has been categorised as one of ANN's learning techniques. Deep learning for ANN can provide numerical classification results rapidly and accurately [20], [21]. This method has achieved prominent success in various applications, such as automated identification and prediction of letters and words [22]. Deep learning is widely used in classifying large data sets to produce efficient findings. Applying the convolutional neural network (CNN) in deep learning can generate powerful and frequently correct assumptions by changing various parameters in development of model class [23]. CNN has been used globally in computer vision since the AlexNet model [23] made DCNN popular by defeating the ImageNet model in the ImageNet Large-scale Visual Recognition Challenge in 2012 [24]. DCNN applications, such as in robotics, automobile automation and added reality, require the computing of identification tasks to be run within a certain time frame and on a limited platform. In addition, previous research has confirmed that DCNN can be effectively applied in processing bio-medical images for classification and segmentation.

According to the previous research, many approaches in DIP and different types of neural networks can be used to achieve the objective of the current work. Based on the fact that the DCNN offers high-quality predictions by varying the parameters in the development of model class [23] in comparison with usual ANN methods, this research presents the automated pterygium detection in ASPIs using DCNN.

2. METHODOLOGY

This section is divided into three parts namely; the ASPI database, the proposed DCNN architecture and measures

utilised for training and classification purposes and lastly, the description of hardware and software used in this work.

2.1 ASPI Database

A total of 844 pterygium and normal anterior segment photographed images (ASPI) were extracted from myMata.my database, which has been acquired by researchers at the Centre of Integrated Systems and Advanced Technologies (INTEGRA) in Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia. The database consisting of ASPIs were collected locally and from the online sources [11], [25], [26]. All local ASPIs have been verified by experts from the Eye Clinic, Faculty of Health Sciences, Universiti Kebangsaan Malaysia, whereas the ground truth of the ASPIs obtained from online sources were provided. Samples of pterygium and normal ASPIs from myMata.my database are shown in Figure 1.

Data Source	Pterygium ASPI	Normal ASPI
Local Source		
		
Online Source		
		

Figure 1: Examples of pterygium and normal ASPIs from myMata.my database

2.2 Proposed DCNN architecture

The contribution of this work lies in the implementation of automated detection using DCNN which involved convolution, pooling and fully connected layer (FCL) to attain the final probabilities for detection and classification of pterygium and normal cases. The idea of implementing DCNN in this work was to overcome the problem of small ASPI database for both pterygium and normal cases. Thus, five main steps namely; ASPI data preparation, DCNN model

development, evaluation and testing of models, efficiency analysis and performance comparison of the generated models, were proposed as illustrated in Figure 2. The flow chart provides a step-by-step description of model generation, data training and testing to obtain the best hyperparameter combination for the proposed automated pterygium detection.

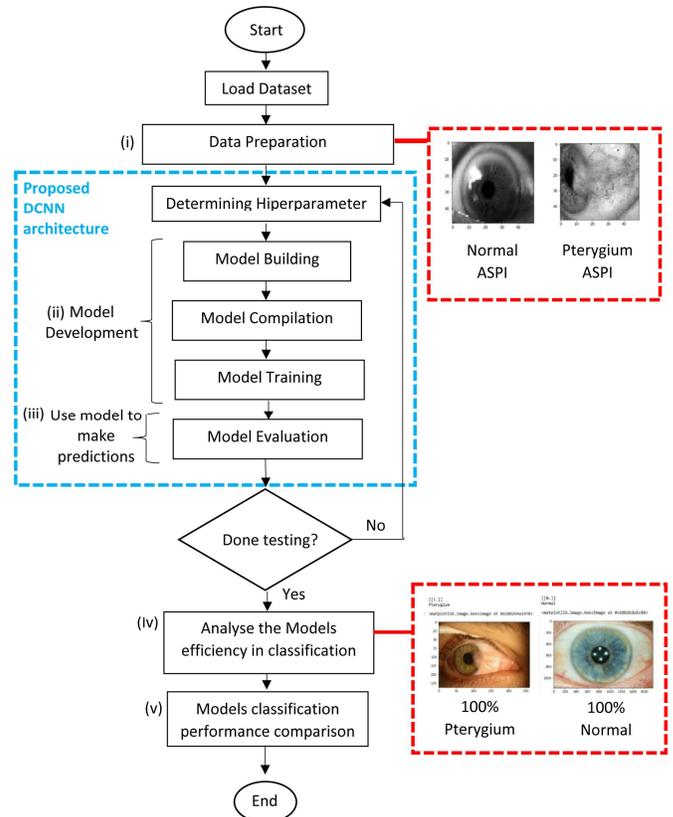


Figure 2: Flow chart for pterygium detection in ASPI using DCNN

The proposed architecture focused on its effectiveness in the detection and classification of two ASPI classes, namely, pterygium: 1 and normal: 0. Figure 3 shows the proposed DCNN architecture with six learned layers that were composed of three convolutional and three fully connected layers. In this architecture, the image dimensions of the input data were set to 50 (height) × 50 (width) × 1 (channel). The kernel, *K*, was a component involved in executing a convolutional operation and has a value of 3 × 3. The pooling layer responsible for reducing the size of the spaced feature has a value of 2 × 2 for all convolutional layers. These convolutional layers used learnable convolutional filters to extract representations from the locally connected image fraction.

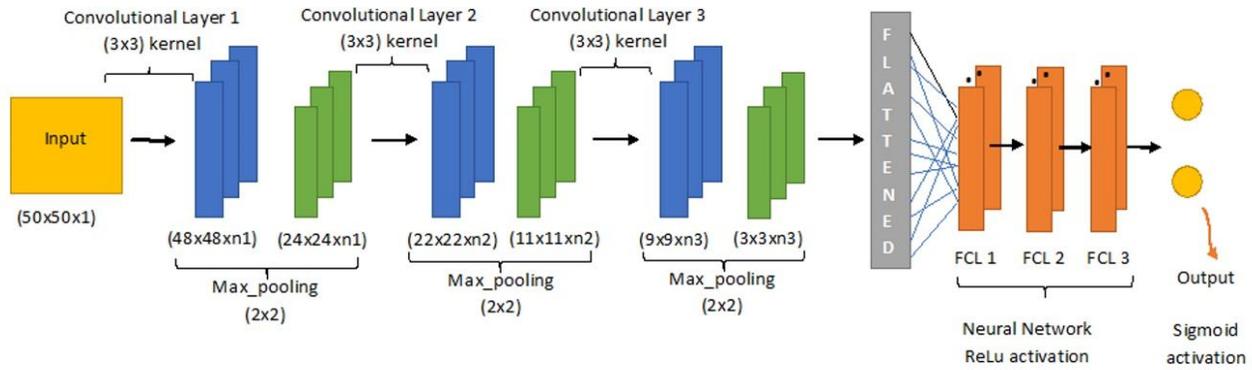
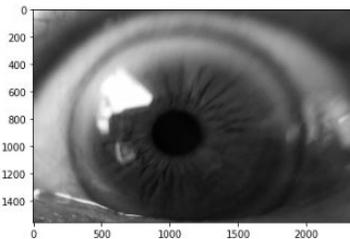


Figure 3: Proposed DCNN architecture

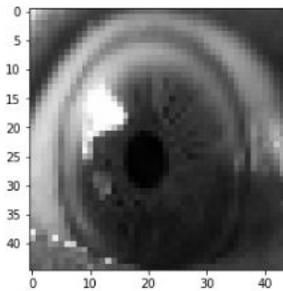
The max-pooling propagation layer was used in this architecture since it provided the maximum value from the part of the closed image by K . Furthermore, in general, it is able to retain noise that filters out all interruptions, runs the process and diminishes the dimensions [27]. Given that the image on the input data has been transformed into a suitable form for multiple perceptron levels, it was then flattened and given to the FCL. The FCL used all features for high-level classification where CNN gradually captures rich representations of input image from pixel level to content level to ensure the performance of high-level classification [28], [29]. Each training layer has backpropagation applied with each epoch. The activation function used in FCL is known as rectified linear unit (ReLU). ReLU does not activate all neurons simultaneously. Only several neurons are activated at a certain time. Thus, the network was used infrequently, making it efficient and easy in terms of computation [30]. The end function generally predicts the probability of what exists between 0 and 1 only [31], thus sigmoid activation function was used at the end of the architecture before the output data yield because this function produced only two types of classes, which were; pterygium and normal classes in this work.

A. ASPI Data Preparation

In this experimental work, the ASPIs from the local and online sources were converted into gray scale images before they were rescaled to 50x50 pixels, by using bilinear interpolation method. Figure 4 presents an example of these simple steps. The rescaled images needed to be small since training smaller number of parameters only need small amount of training data and number of iterations. Next, the hyperparameter values which consisted of the epoch, batch size, learning rate, convolutional layer and image size, were changed for each model, before the model was built.



(a) Original image



(b) The rescaled image

Figure 4: ASPI data preparation

The number of epochs varied with minimum of 5 until 25, with increment of 5, learning rate of 0.0001, 0.001 and 0.1 with fixed convolutional layer of 3 and batch size of 100. The best of all hyperparameter combination will be presented in the results section.

B. Model Development for Predictions

Next, the models were developed, arranged and trained. There were 15 models developed altogether with different hyperparameters which consisted of 5 different epoch values which were then varied with 3 values of learning rate. Subsequently, all of these models were tested for prediction & the efficiency were analysed. The six learned layers of the proposed DCNN architecture were used before the pterygium and normal ASPI were classified on the data output. A sequential type of model was generated, and this API model enabled layer-by-layer modelling of problems that

have arisen. Subsequently, the model was assessed by conducting classification and detection of pterygium and normal ASPI. The loss function used was 'binary_crossentropy', because the classification in this work involved only two classes. The type of optimisation used was 'Adam (Adaptive Moment Optimisation) optimiser' because of its efficient computing, minimal memory usage and suitability for static data. The Adam optimiser was chosen due to its capability to merge the heuristic characteristic of both RMSProp and Momentum optimiser. This optimiser computed the exponential average of gradient as well as the squares of gradient for each parameter.

C. Analyzing Model Efficiency and Model Classification Performance Comparison

The efficiency of each model produced in detecting and classifying pterygium and normal ASPI were then analysed, and the models' performance was compared. To meet the needs of this analysis, value of sensitivity, specificity, accuracy, F1 scores and the Matthew correlation coefficient (MCC) were calculated according to the classification performed in the previous step. All of these calculations were carried out outside the DCNN layer and all values were tabulated. All calculations were conducted when true positive, true negative, false positive and false negative values were obtained. Next, receiver operating characteristic (ROC) charts were tabulated, and the area under the curve (AUC) was computed for each model produced. Afterwards, the AUC values were analysed and compared to determine the best hyperparameter combination of all the resulting models and to detect and classify pterygium and normal ASPI.

On this basis, the proposed method was used to identify and detect pterygium in ASPI using the proposed DCNN architecture which comprised of six learned layers consisting of three convolutional and three fully connected layers. A test to determine the best hyperparameter for the proposed architecture was designed for each model by using different hyperparameter combinations that consisted of the epoch, batch size, learning rate, convolutional layer and image size.

2.3 Hardware and Software

The important feature of the DCNN is represented by the speed of training. As large data leads to particularly large training time, so does small data that practically takes less time in training. The important factor in deep learning is represented by technological capabilities. Thus, this work implemented both a Central Processing Unit (CPU) and a Graphic Processing Unit (GPU) that was installed on one computer. This enabled the training process to be conducted using just one machine.

The CPU used for this work has four cores embedded with an Intel (R) Core (TM) i7-6700K CPU with 16 GB of memory.

The GPU was co-installed with NVIDIA GeForce GTX 960, which has 10 GB of memory, of which 8 GB was shared with CPU usage and 2 GB was used for graphical processing. To fully utilise the GPU (other than ordinary graphics such as video play and photo editing), the NVIDIA CUDA parallel computing platform version 8.0.6.1 was installed. The NVIDIA CUDA version 8.0.6.1 allows the use of many computing cores in graphics processing to perform general-purpose mathematical calculations and achieve high-speed computing performance.

Python was used and synchronised with NVIDIA CUDA by adding NVIDIA CUDA dependency as a variable to the computer environment. Python was employed to connect Tensorflow and the Keras interface, which is an open source for a neural network library that can operate with Tensorflow. Numpy libraries were also utilised for the processing of raw data arrays. This library was widely used in the ANN architecture proposed in this work. OpenCV and matplotlib libraries were set together to present the research findings in the form of graphics and speed up the use of trained machine perceptions. Tensorflow is the best interface used as a basic library for producing a deep learning model either directly or wrapped by a library built on Tensorflow. Keras is a high-level neural network application programming interface (API) written in Python programming language. It is ideal for building a neural network model and can be built on Tensorflow.

GPU with its serial computing, Tensorflow version 1.13.1 and Keras 2.2.4 libraries were successfully installed in the computer and fully utilised in this work. The Keras library was built on Tensorflow, where Keras operates with the Tensorflow backend engine. In this research, both of these libraries performed well with the NVIDIA GeForce GTX 960 GPU hardware and the NVIDIA CUDA parallel computing version 8.0.6.1 by speeding up computing time and computation in machine learning. Figure 5 shows the interaction between Tensorflow and Keras with GPU on how the Tensorflow uses CUDA and cuDNN to regulate GPUs and boosting it.

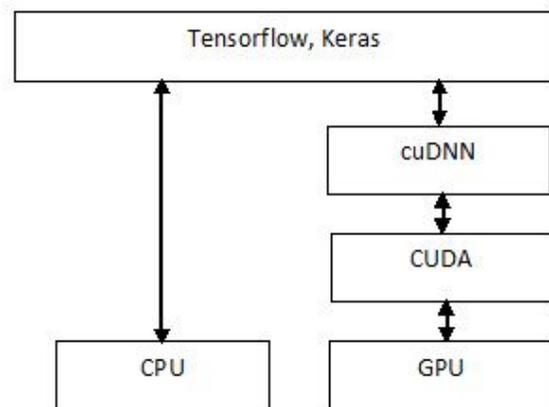


Figure 5: Interaction of Tensorflow and Keras with GPU, with the help of CUDA and cuDNN

3. RESULTS AND DISCUSSION

All ASPIs were sorted and balanced to avoid one-class detection and classification, which ultimately cause the entire framework to malfunction. A total of 70% ASPIs of both cases were used for training whilst 30% were utilised for testing.

The best hyperparameter combination of the three developed models from the experiments is presented in Table I. The detection and classification steps and the proposed DCNN architecture yielded promising results because the highest AUC yield from Experiment 2 reached 99.2%. The highest accuracy was 94.09% from Experiment 1, but AUC still had a value of 99.0%, indicating a 0.2% difference between Experiments 1 and 2. The highest F1 score and MCC were also obtained in Experiment 1. However, such an outcome does not mean that Experiment 3 was counterproductive because it also had an AUC of 99.0%. Given that the three experiments showed promising results, K-fold cross validation was performed to avoid overfitting and to determine the best hyperparameter combination. The number of folds, k , was set to five because limited data were used for training. Hence, five was the optimum number of folds that can be considered.

Figure 6 shows the ROC of the average fold of Experiments 1, 2 and 3. Experiments 2 and 3 had the highest AUC values, whilst Experiment 1 showed a difference of 1%, that was, an AUC value of 94%. Nevertheless, the three experiments shared satisfactory and promising AUC values, with the hyperparameter combination used to detect and classify pterygium and normal ASPI.

The proposed DCNN architecture and detection steps with the suitable hyperparameter combination produced promising results in pterygium and normal ASPI detection and classification. In addition to this, advanced machine learning, neural network architectures and algorithms can be implemented to improve both detection and accuracy.

4. CONCLUSION

This study developed a DCNN architecture for the detection of pterygium. In the experimental work, the three developed models worked well with the DCNN architecture to identify pterygium and normal ASPI by determining the best hyperparameter combination for application with reduced input data. Experiment 1 provided the best performance with an accuracy value of 94.09%, F1 score of 93.93% and MCC of 88.32% with the highest AUC value of an average fold of 95% at the following hyperparameter combination: a convolutional layer of 3, learning rates of 0.001 and 0.0001 and epochs of 20 and 25. In conclusion, the developed automatic pterygium detection method using the DCNN architecture with the proposed hyperparameter combination successfully detected the existence of pterygium in a small-sized ASPI database. In addition, the proposed DCNN architecture has a bright potential to be applied in the future development of pterygium screening tool.

Table 1: Results of three developed models with the best hyperparameter combinations using exhaustive cross validation

Experiment	Hyperparameter			Accuracy	F1 Score	Matthew Correlation Coefficient (MCC)	AUC of ROC
	Epoch	Learning Rate	Conv. Layer				
1	20	0.001	3	94.09	93.93	88.32	99.00
2	20	0.0001	3	88.19	86.61	78.60	99.20
3	25	0.0001	3	92.52	92.18	85.36	99.00

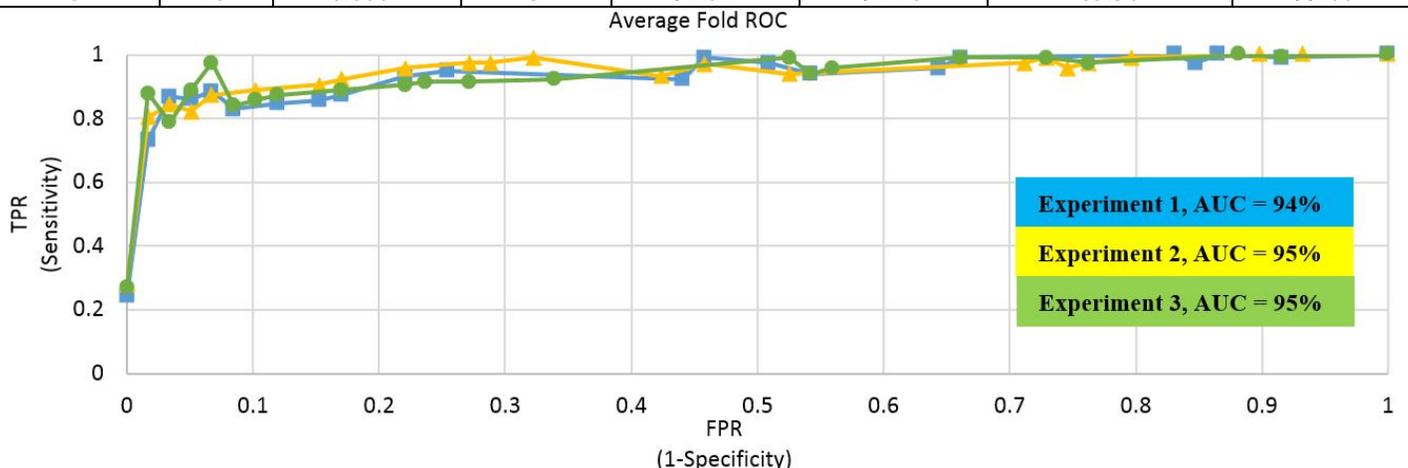


Figure 6: ROC curves of average fold for all three experiments

ACKNOWLEDGMENT

This research was supported by Dana Impak Perdana from Universiti Kebangsaan Malaysia with grant no. DIP-2018-020 and Fundamental Research Grant Scheme from the Malaysian Ministry of Higher Education (MOHE) with grant no. FRGS/1/2017/TK04/UKM/02/4.

REFERENCES

- [1] W. M. D. W. Zaki, A. Hussain, and H. A. Mutalib, **Pemrosesan Imej Digital Untuk Analisis Penyakit Mata Berkomputer**. Published by Penerbit UKM, Malaysia, 2018.
- [2] M. Hessen and E. K. Akpek, “**Dry eye: an inflammatory ocular Disease**”, *J. Ophthalmic Vis. Res.*, vol. 9, no. 2, pp. 240–250, 2014
- [3] D. J. Moran and F. C. Hollows, “**Pterygium and ultraviolet radiation: A positive correlation**”, *Br. J. Ophthalmol.*, vol. 68, no. 5, pp. 343–346, 1984
<https://doi.org/10.1136/bjo.68.5.343>
- [4] G. Gazzard *et al.*, “**Pterygium in Indonesia: prevalence, severity and risk factors**”, *Br J Ophthalmol*, vol. 86, no. 12, pp. 1341–1346, 2002
- [5] J. C. Hill, “**Pathogenesis of pterygium**”, *Eye*, vol. 3, no. 2, pp. 218–226, 1989
<https://doi.org/10.1038/eye.1989.31>
- [6] D. M. Burns, F. M. Johnston, D. G. Frazer, C. Patterson, and A. J. Jackson, “**Keratoconus: An analysis of corneal asymmetry**”, *Br. J. Ophthalmol.*, vol. 88, no. 10, pp. 1252–1255, 2004
<https://doi.org/10.1136/bjo.2003.033670>
- [7] M. Caldwell, L. Hirst, and M. A. Woodward, **Pterygium - EyeWiki**, 2015, available at <https://eyewiki.aao.org/Pterygium#References>. [Accessed: 24-Sep-2019]
- [8] T. Bennett, **Anterior Segment Photography | EYE-PIX**, 2015, available at <http://eye-pix.com/anterior-segment-photography/>. [Accessed: 24-Sep-2019].
- [9] N. Khumdat, P. Phukpattaranont, and S. Tengtrisorn, “**Development of a computer system for strabismus screening**”, *BMEiCON 2013 - 6th Biomed. Eng. Int. Conf.*, pp. 1–5, 2013
- [10] W. M. D. W. Zaki, M. M. Daud, S. R. Abdani, A. Hussain, and H. A. Mutalib, “**Automated pterygium detection method of anterior segment photographed images**”, *Comput. Methods Programs Biomed.*, vol. 154, pp. 71–78, 2018
- [11] R. G. Mesquita and E. M. N. Figueiredo, “**An algorithm for measuring pterygium’s progress in already diagnosed eyes**”, *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, pp. 733–736, 2012
<https://doi.org/10.1109/ICASSP.2012.6287988>
- [12] S. R. Abdani, W. M. Di. W. Zaki, A. Mustapha, and A. Hussain, “**Iris segmentation method of pterygium anterior segment photographed image**”, *ISCAIE 2015 - 2015 IEEE Symp. Comput. Appl. Ind. Electron.*, pp. 69–72, 2015
<https://doi.org/10.1109/ISCAIE.2015.7298330>
- [13] G. M. Shepherd and G. M. Shepherd, **Introduction to Synaptic Circuits: The Synaptic Organisation of the Brain**. Published by the Oxford University Press, New York, USA, 1990.
- [14] M. A. Nielsen, **Neural Networks and Deep Learning**. Published by the Determination Press, 2015.
- [15] L. Dormehl, **What is an artificial neural network? Here’s everything you need to know | Digital Trends**, 2019, available at <https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/>. [Accessed: 24-Sep-2019].
- [16] F. Van Veen and S. Leijnen, **A mostly complete chart of Neural Networks**, The Neural Network Zoo, 2019, available at <http://www.asimovinstitute.org/neural-network-zoo>. [Accessed: 24-Sep-2019].
- [17] A. Parvat, J. Chavan, S. Kadam, and S. Dev, “**A survey of deep-learning frameworks**”, *IEEE Int. Conf. Inven. Syst. Control*, pp. 1–7, 2017.
- [18] J. Chang, J. Yu, T. Han, H. J. Chang, and E. Park, “**A method for classifying medical images using transfer learning: A pilot study on histopathology of breast cancer**”, *2017 IEEE 19th Int. Conf. e-Health Networking, Appl. Serv. Heal. 2017*, vol. 2017-Decem, pp. 1–4, 2017
- [19] J. Schmidhuber, “**Deep learning in neural networks: An overview**”, *Neural Networks*, vol. 61, pp. 85–117, 2015
- [20] F. Ertam, “**Data classification with deep learning using tensorflow**”, *IEEE Int. Conf. Comput. Sci. Eng.*, pp. 6–9, 2017
- [21] B. H. Rasheed, M. Sivaram, D. Yuvaraj, and A. M. U. Ahamed, “**An improved novel ANN model for detection of DDoS attacks on networks**”, *Int. J. Adv. Trends Comput. Science Eng.*, vol. 8, no. 1.4, pp. 9–16, 2019
<https://doi.org/10.30534/ijatcse/2019/0281.42019>
- [22] M. Abdullah, A. Agal, M. Alharthi, and M. Alrashidi, “**Arabic handwriting recognition model based on neural network approach**”, *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 1.1, pp. 253–258, 2019
<https://doi.org/10.30534/ijatcse/2019/4581.120419>
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “**ImageNet classification with deep convolutional neural networks**”, *Adv. Neural Inf. Process. Syst.*, pp. 1097–1105, 2012
- [24] O. Russakovsky *et al.*, “**ImageNet Large Scale Visual Recognition Challenge**”, *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015
- [25] H. Proença and L. A. Alexandre, **Ubiris iris image**

- database, 2004. .
- [26] H. Proença and L. A. Alexandre, “**UBIRIS: a noisy iris image database**”, *13th Int. Conf. Image Anal. Process. - ICIAP 2005, Springer*, pp. 970–977, 2005
- [27] S. Saha, **A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way**, Towards data science, 2018, available at <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. [Accessed: 24-Sep-2019].
- [28] D. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. **Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images**, in *Proc. of the NIPS*, Lake Tahoe, NV, USA, 2012, pp. 2843–2851.
- [29] P. Moeskops, M. A. Viergever, A. M. Mendrik, L. S. de Vries, M. J. N. L. Benders, and I. Išgum. **Automatic Segmentation of MR Brain Images with A Convolutional Neural Network**, *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1252–1261, 2016. <https://doi.org/10.1109/TMI.2016.2548501>
- [30] D. Gupta, **Fundamentals of Deep Learning - Activation Functions and their use**, 2017, available at <https://www.analyticsvidhya.com/blog/2017/10/fundamentals-deep-learning-activation-functions-when-to-use-them/>. [Accessed: 24-Sep-2019].
- [31] S. Sharma, **Activation Functions in Neural Networks - Towards Data Science**, Towards data science, 2017, available at <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>. [Accessed: 24-Sep-2019].