

Musa Yusuf
 Computer Science Department
 Bingham University
 Karu, Nigeria
 yusufmusa@binghamuni.edu.ng

Samuel Theophilous
 Computer Science Department
 Bingham University
 Karu, Nigeria
 s.theophilous@binghamuni.edu.ng

Jadesola Adejoke
 Computer Science Department
 Bingham University
 Karu, Nigeria
 emmanueladejoke@yahoo.co.uk

Annah B. Hassan
 Samuel Theophilous
 Computer Science Department
 Bingham University
 Karu, Nigeria
 annebjk@gmail.com

Abstract— *The alarming cases of cataract within the last decade and the projection of cataract cases within the next few decades call for urgent intervention by early diagnosis. Formal ways of detecting cataract such as physical examination, tests and diagnosis are clinic and professional bound. Hence the need for automation process. Some works have been done on Computer Aided Diagnosis (CAD) of cataract with tools such as Expert systems, which are limited to their knowledgebase thus inaccurate. Early diagnosis of cataract enables quick intervention and treatment. This paper presents a web-based Computer Aided Diagnostic for cataract detection system using Convolutional Neural Network that can be used by any nonprofessional outside the clinic environment. The systems model trained on a data set of 100 eye images using transfer learning which were retrieved from google image search results of “normal human eyes” and “human eye cataract”. It utilized ImageNet model developed in ILSVRC2012 using the Convolutional Neural Network classifier and transferred its knowledge using Transfer learning to train a new model. The new model gained the ability to classify eye images into “Normal” and “Cataractious”. The system was designed to take images as inputs and achieved a Sensitivity of 69%, a Specificity of 86%, Precision of 86%, F-Score of 56% and AUC of 84.56%. Its accuracy score was 78% which was influenced using the model trained during the ImageNet image classification using deep convolutional neural network*

Keywords—: *Computer Aided Design, Convolutional Neural Network, Cataract detection.*

I. INTRODUCTION

US department of health and human services, National eye institutes define cataract as a clouding of the lens in the eye that affect most vision [1]. Lens is the clear part of the eye that helps in focusing light to the retina. Retina is the light sensitive tissue at the back of the eye, Cataract has been noted by National Eye Institute in the United States as one the leading causes of blindness resulting from eye diseases in the world [2]. The human eye lens gets cloudy which results to the production of blurry images by the eye in a situation. This eye defect at its latter stage leads to blindness but can be corrected through eye surgery by an Ophthalmologist to replace the natural lens with a synthetic lens.[3] Stated that statistics from the Nigerian National Blindness and Visual Impairment Survey shows that 42 out of 1000 adults aged 40 and above are blind and that eye diseases like Cataract blind 2 out of 3 Nigerians. As at 2011, about 750,000 Nigerians were blinded by cataract. This shows that cataract could affect more individuals in future if it is manage properly. The need to find effective means of reducing the blindness rate is unarguably important Ease of Use

The paper proposes a CAD software that would be easy to assess and reliable enough to perform medical diagnosis. Previous researches over the years have tried creating medical diagnostic tools using techniques such as the use of Expert Systems [4]. The problems with some of the methods used is the restriction it gives the system. For instance, an Expert System is software that works through logical decision-making based on a knowledge base extracted from an Expert in the specific field. Such a system cannot make complex analysis or computation. Such a system can only be assessed by the authenticity of the information in its knowledge base.

To develop a system that is not limited by such restrictions, machine leaning methods, was implemented which were used in training a Convolutional Neural Networks (ConvNets) model that was used in the classification of cataract images. Machine learning is a branch of computer science, which gives the computer the ability to learn without the need to explicitly program the knowledge. Its learning algorithm types have been classified into various categories based on the way the learning takes place [5].

I. BACKGROUND

Recently researchers have developed much interest in using machine learning in medical diagnosis, computer aided design and other fields such as computer vision. These researches were initially started as theoretical pieces but have recently found practical applications. In this section, some charts presenting ground-truth on cataract generated from research statistics collected by NEI in 2010 are presented in Fig 1[2].

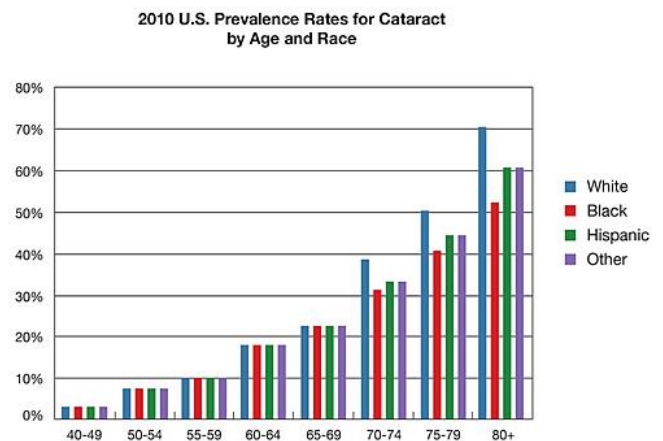


Fig 1: Prevalence Rates of Cataract in US by race as at 2010 (Cataracts, 2010)

Fig 1 shows the distribution of prevalence rate of cataract in US age wise. The older age groups have a higher percentage of cataract-infected persons. From fig 1, we can deduce that cataract usually manifest on people from age 40 but age of 80 and above have the highest prevalence rate.

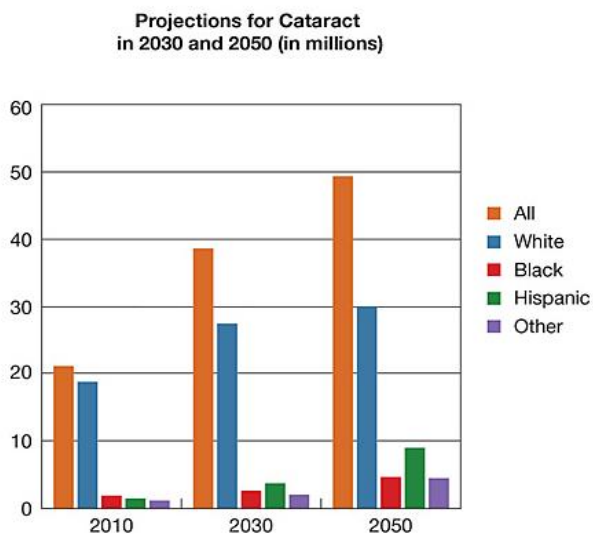


Fig 2: Projections for Cataract in 2030 and 2050 (in millions)[17]

Fig. 2 shows projected statistics of Cataract cases in the year 2030 and 2050 if the prevalence rate is not abridged. This result therefore shows how precarious cataract could be in future and therefore researchers need to foster way to tackle this challenge and therefore motivates the researcher to study this problem.

II. RELATED WORK

In recent years, there has been an increase in the studies and researches done focusing on machine learning. However, these researches have been applied in numerous fields, a good fraction of these researches aimed at improving medical diagnosis using different machine learning algorithms. Some of these researches initially started as theoretical pieces but have recently found practical applications.

[6] developed a Medical Imaging Analysis System using Artificial Neural Networks, the research showed how artificial neural networks were used for Computer-aided diagnosis systems. ANN was used in the detection and diagnosis of breast cancer and lung cancer. A computer-aided detection system identifies micro-calcification clusters automatically on full field digital mammograms using six stages (pre-processing, image enhancement, segmentation of micro-calcification candidates and false-positive reduction for individual micro-calcifications, regional clustering and false positive reduction for clustered micro calcifications. A training set of 96 images were used along with 46 images for validation. The Convolution Neural Network was used to create a model to simulate the vision of vertebrate animals and use as the simplified vision machine, which performed the classification of the regions into two output types: disease and non-disease, while a linear discriminating analysis classifier was then used to differentiate clustered micro calcifications from false positives. The ROC analysis showed the system's false-positives reduction rates were 86%, 74% and 72% at sensitivities of 70%, 80% and 90% respectively when compared to classification without the Convolutional Neural Network/Linear Discriminating Analysis approach. They also applied the detection and diagnosis of bone mineral density (BMD).

In [16] A Computerized Cataract Detection and Classification was developed. The system was able to classify 156 eye images and got 50 images that contained normal eyes, 39 images with nuclear cataract, 33 images with cortical cataract and 34 with PSC. They

gathered their dataset by capturing eye images with a Topcon SL-45 Scheimpflug camera. 90 images out of the 156 were used as the training set to create the modelled profile for the four categories. The remaining 66 images with unknown categories were classified into the four categories based on their distance from the modelled profiles in Euclidean space. The developed system was analysed and found to have sensitivity of 98% in detecting Cataract in images, specificity of 100% in identifying normal eyes and classification accuracy of 98%.

[7] developed a system that takes low-resolution images as input and produces a high-resolution image in the paper; Learning A Deep Convolutional Network for Image-Resolution. The system developed learns end-to-end mapping between low-resolutions and high-resolutions using the deep learning convolutional neural network. The methodology was compared to some existing methods and found to outclass them, but it was also easier to implement and light to run (can be executed with a CPU). They stated that just like the sparse-coding-based method, convolutional network would not be limited by internal training set but work well with external examples.

[8] carried out an ImageNet Classification with Deep Convolutional Neural Networks. In the study, a model was train using the Deep Convolutional Neural Network classifier with a dataset of 1.2 million high-resolution images. The model had 1000 different classes and when data was tested on the model, the study achieved top-1 and top-5 error rates of 37.5% and 17.0% respectively. It contained eight layers that consisted of five convolutional layers and three fully connected layers with a final 1000-way softmax. The research stated that it took six days to train the model on to GTX 580 3GB GPUs which was influenced by the dataset's size and number of layers involved in the training. The study also stated that each layer contained no more than 1% parameters required for training the model.

III. ARTIFICIAL NEURAL NETWORK

An Artificial Neural Network is an interconnection of nonlinear neurons. This is a machine learning classifier that was designed to model how the human brain works. It was created to act like the biological neurons (performing multiple tasks in a very short period of time).

From fig 3, we can define the following terms: " X_i " – is the input feature over n ; " n " – a set of features; " W " - Weights of each corresponding feature and Finally an activation function is performed on the sum and outputs are produced[9]. This can be

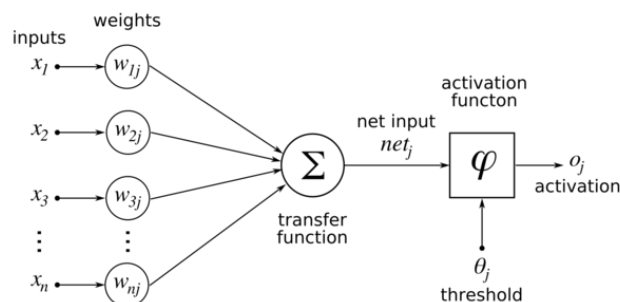


Fig 3: Showing the structure of an Artificial Neuron [15]

represented with the equation (1):

$$y = f(x_n) \dots \dots \dots (1)$$

The output y is generated from the given activation function $f(x)$ as shown in Equation (1). Where x_n is number of features over n .

$$y = \sum_{i=1}^n (w_i x_i + w_0) \dots \dots \dots (2)$$

In more appropriately mathematical notation, we can represent equation (1) as equation (2) with an addition of bias weight w_0 to the sum of product of w_i and features x_i for $i = 1, 2, \dots, n$.

The back propagation technique from equation (3) was used to train the model. Whenever features x_i are passed through the classifier, it uses the gradient descent algorithm to adjust the weights w_i of features x_i so they produce the desired output as defined in the dataset used for the training. After each step, data is fed back to the classifier to retrain and training is only said to be done when the input features lead to the expected label that is when gradient's error measure is at its minimum.

IV. CONVOLUTIONAL NEURAL NETWORK

The ConvNets is an ANN that has more than one layer and trains its models by adjusting its weights using optimization methods such as gradient descent algorithms. The first layer in the ConvNets is the "convolution layer" which uses the operation of convolution to search for patterns. Neurons in this network are used to perform the convolution operations i.e. filtering images to look for patterns. The neurons in each given filter share the same weight and bias thus giving them the ability to search for the same pattern in different parts of the image. The second layer is the "retified linear unit layer" which focuses on building up the patterns discovered by the convolution layer. The third layer in the ConvNets is the "pooling layer". This layer reduces the number of patterns allowing the neural network to focus only on important patterns. The final layer is called the fully "connected layer" which makes the ConvNets able to classify data into appropriate labels.

The accuracy of the ConvNets classifier is largely dependent on the size of the dataset it trains on, as it usually requires a very large set of training data. That was one of the major reasons why the initial ConvNets models did not perform very well (they did not have access to a very large dataset). Getting a very large dataset can be very stressful and difficult; the second problem has to train new models on large datasets from scratch every time they are being developed. This problem makes it almost impossible for researchers without heavy hardware specifications to implement ConvNets. It is for this reason that transfer learning was introduced so that knowledge collected in well-trained models can be transferred to the new models and used to build the new model while conserving hardware and time [10].

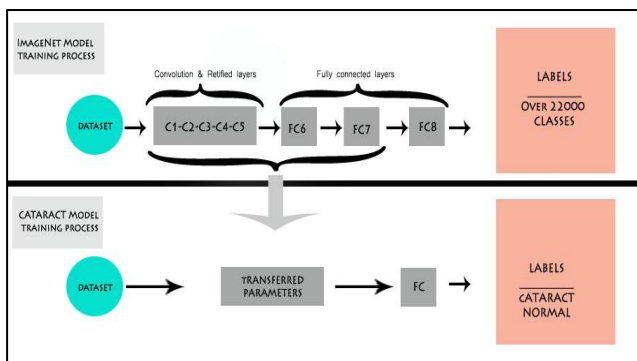


Fig 4: The system model showing how learned knowledge is being transferred from ImageNet Model to Cataract Model (Sengupta, 2003)

V. METHODOLOGY

The system developed is a web-based application that takes images as inputs. It uses the images to train a model based on the set of images it has taken as input (it does this using the ConvNets algorithm). The model created is then used as a standard to classify eye images as cataractous or normal.

The system is a three-paged web application. The first page is the landing and contains a summary and brief note on the functions of

the application and the eye centre. The second page gives an educative information on cataract and the third page has the fields used to accept inputs for evaluation.

The model trained on 1.2 million images was used in the training of the new model to reduce the large dataset and heavy hardware requirements needed to develop a model that was accurate [11]. This is possible using a technique known as transfer learning in which information from another model is used along with other dataset(s) to increase the performance of the new model and to reduce the requirements needed for the computation of the new model. The "fully connected layer" was removed and the remaining part of the model (which contained information extracted by the initial layers) is treated as a fixed feature extractor for the new dataset. It computes a 2048-dimensional vector for each image that contains the activations of the hidden layer just before the classifier. Finally, a linear classifier, was used as the classifier, is trained for the new dataset [12]. The ImageNet model used was generated using the ConvNets.

The Gradient descent algorithm was used to optimize our model by finding the smallest value of a function that minimizes the error in equation (1). Its main objective is to find a value that fits a given curve so that it gives the best minimum error for the given function over point t_0^p and y_0^p

When inputs are passed through the trainer and weights have been assigned to them, their calculations are initially not very accurate. By using the gradient descent, the weights can be adjusted until each input produces the desired output as specified in the training dataset. This process is described as the learning process. In this paper, a learning rate of 0.01 was used to descend using 4000 steps.

The solution of the gradient descent was done using the formula below:

$$E = \sum_p E^p \dots \dots \dots (1)$$

In equation (1), E computes the sum of all error as E over points P

$$E^p = \frac{1}{2} \sum (t_0^p - y_0^p)^2 \dots \dots \dots (2)$$

In equation (2), E^p is the Error measure of the slope between points t_0^p and y_0^p .

$$\text{Then the Gradient} = \frac{\partial E}{\partial W_{ij}} \dots \dots \dots (3)$$

Where W_{ij} is the synaptic interconnection from neuron j to neuron i.

To correct the weight W_{ij} we compute the change t_i and y_i

$$\Delta W_{ij} = \eta(t_i - y_i)x_i \dots \dots \dots (4)$$

$$[new]W_{ij} = [old]W_{ij} + \Delta W_{ij} \dots \dots \dots (5)$$

Where η is the learning rate, and i is the output while j is the input

The softmax regression is a linear classifier implemented on the final layer of a ConvNets, which is applied mainly for classification. The networks are trained under a cross entropy which gives a nonlinear variant of multinomial logistic regression just as in. Given that every image input in the system has only two possible label outputs, evidence needs to be gathered from the image, which determines the label it belongs. The evidences gathered are then calculated to probabilities and the image then is classified into the label with the highest probability score. A weighted sum of the pixel intensities for each input image is calculated. The calculated weight could be either negative or positive. The positivity or negativity of the weight is determined by the high intensity pixels in the image that

holds evidence that it beholds to the given label or evidence that it does not behold to the label. An extra bias is added to the evidence. This bias tries to suggest that some evidences are not dependent on the input. This results in the equation:

$$E_i = \sum_j w_{i,j}x_j + b_i \dots \dots \dots (6)$$

Where i represents a given label; j represents the index of summation of pixels of input image; x is the input image, w is the weight and the evidence is resented with E.

$$Y = \text{Softmax}(E) \dots \dots \dots (7)$$

Using equation (7), we use the Softmax function to generate an output labels Y between the range of 0s and 1s.

$$\text{Softmax}(x) = \text{normalize}(\exp(x)) \dots \dots \dots (8)$$

Using equation (8), we eliminate possibilities by computing the exponential of x_i (9)

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \dots \dots \dots (9)$$

The inputs undergo exponentiation then normalization to eliminate the possibility of having negative weights so that their sum is equal to 1 i.e. sum of probabilities equal to 1[13].

VI. SYSTEM REQUIREMENTS

The designed system has the following system requirements;

1. Run application on a Server that has python 2.7 interpreter installed. This is because python is the server-side scripting language used for the development of the web application.
2. Tensorflow python library should be installed on the server. This is the machine learning API used in this study, which was written mainly for the CNN classifier.
3. Numpy library should be available on the server. The images used in this study had to be converted to complex arrays before operations are performed on them. Numpy is a python library that allows such operations with ease.
4. OpenCV library should be available on the server. This is a computer Vision python library that would be used in this study to implement the Viola Jones' objection detection.
5. Django framework library should be available on the server. This is a python library used to develop web applications in python.
6. Material Design Lite CSS framework is going to be the library used in writing the styles for the webpages.
7. A browser needs to be installed on the user's smart device/ Computer system in order for him to gain access to use the application.
8. A Database needs to be designed so that all tests taken are recorded on the Server (in this case SQLite).
9. The server used to host the developed system was a Linux server. This is because the Tensor flow library was written for the Linux and Mac OS and did not support the Windows OS at the time this system was developed.
10. A trained model is to be made available on the server to be used for classification.

VII. EXPERIMENTS AND EVALUATION

All the images used in the training process were retrieved from google search results with searches "human eye image" and "Cataract eye image". Training the model based on the 100 images gotten from google search result will not produce a very accurate model. So in order to save time and gain high level of accuracy, the google inception model was downloaded. This model was created by google after training millions of images [12]. The images gotten were trained using the "retrain.py" file written by Google in the tensor flow examples on GitHub and used to create a new model. The file uses the inception model to start the training process before it works on the images. The use of the Tensor Flow CNN library eased the implementation and use of the classifier and model. A Total of 30 images were tested on the developed model and the system was evaluated using the ROC Curve and confusion matrix.

The Receiver Operating Characteristics Curve is a graph, which is used to evaluate the performance of classifiers. They initially had more application in the medical field to aid in some forms of decision making but have been adopted in machine learning and data mining researches to evaluate the performance of classifiers. The values used to plot an ROC curve are generated using a matrix known as confusion matrix. To plot an ROC curve to evaluate a binary classifier, a 2 x 2 confusion matrix is used which is the case with this paper.

Table 3: Confusion matrix table

	Actual Positives	Actual Negatives	Total
Predicted Positives	True Positives (TP)=11	False Positives (FP)=2	PP=TP+FP
Predicted Negatives	False Negatives (FN)=5	True Negatives (TN)=12	PN=FN+TN
Total	AP = TP +FN	AN = FP +TN	

$$\text{TPR} = \frac{\text{TP}}{\text{AP}} = \frac{11}{11+5} = 0.69$$

$$\text{TNR} = \frac{\text{TN}}{\text{AN}} = \frac{12}{2 + 12} = 0.86$$

$$\text{Precision} = \frac{\text{TP}}{\text{PP}} = \frac{11}{11 + 2} = 0.85$$

$$\text{F Score} = \text{Precision} * \text{TPR} = 0.85 * 0.69 = 0.59$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN} + \text{FN} + \text{FP}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} = \frac{11 + 12 + 5 + 2}{23} = \frac{30}{30} = 0.78$$

A Confusion matrix is a matrix, which contains information about the actual classifications and the predicted classifications performed by a classification system. It consists of four important cells:

- i. True Positive (TP): This is the number of classifications that were actually positive and were correctly classified by the system.
- ii. False Negative (FN): This is the number of classifications that were actually positive but were wrongly classified by the system.
- iii. True Negative (TN): This is the number of classifications that were actually negative and were correctly classified by the system.
- iv. False Positive (FP): This is the number of classifications were actually negative but were wrongly classified by the system.
- v. These values are analyzed and used to evaluate the system. The system used a threshold of 0.5 to classify the images. If the probability that image is in Class A (Normal) is higher than the probability that it is in Class B (Cataract), then it is classified into Class A. Class A was taken to be the positive class and Class B was taken as the negative class.

vi.” In order to get the values for the points to be plotted on an ROC Curve, the test set (in this case the images) are classified using different threshold values. At each threshold value used, a confusion matrix is constructed so as to get the “true positive rate” and “true negative rate” at that threshold. The ROC curve is finally plotted using the “true positive rate” values on the vertical axis and the “false positive rate” values on the horizontal axis [14].

A threshold of 0.5 was used in the experiment, which resulted in the generation of the confusion matrix below 0 => Cataractous, 1 => Normal

Table 1: Table showing TP, TN, FP, TPR, TNR and FPR values at different thresholds

SN	Threshold Value	TP	TN	FP	FN	TPR	TNR	FPR
1	0.0	16	0	14	0	1	0	1
2	0.1	15	6	8	1	0.94	0.43	0.57
3	0.2	15	8	6	1	0.94	0.57	0.43
4	0.3	14	8	6	2	0.88	0.57	0.43
5	0.4	11	10	4	5	0.69	0.71	0.29
6	0.5	11	12	2	5	0.69	0.86	0.14
7	0.6	11	12	2	5	0.69	0.86	0.14
8	0.7	10	12	2	6	0.63	0.86	0.14
9	0.8	9	14	0	7	0.56	1	0
10	0.9	8	14	0	8	0.50	1	0
11	1.0	0	14	0	16	0	1	0

Table 1 shows the truth table values computed for each given image, which is used in the generation of the ROC chart shown in fig 4. The curve shows how well our algorithm performs

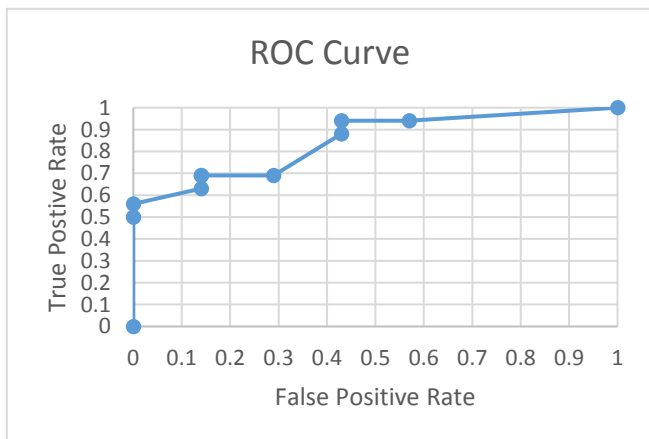


Fig 4: ROC curve generated from the values in Tables 5.3

The Trapezoidal formula, the Area under Curve (AUC) of the ROC curve was calculated as shown in the table fig 4

Table 2 shows the experiment result generated from tested some images on the developed system. Our system shows an interesting performance when tested with both cataract images and non-cataract images. From test image 5, our method had challenges in detecting cataractous eye region due to distances of the eye from the camera. The face in image 5 really have cataract but because of the nature of the face and eye size of the patient, our system was not confident in the prediction as cataract eye. On the other way image two was predicted accurately because the affected eye was captured closer to the camera.

Table 2: Result of some images tested with the proposed methodology

SN	Image sample preview	Actual Value	Predicted Value	Prob. Normal	Prob. Cataracts
1.		0	0	0.04	0.96
2.		0	0	0.01	0.99
3.		1	1	0.99	0.01
4.		1	1	0.97	0.03
5.		0	0	0.40	0.60

When non-cataract images was tested on our system, in the case of images 3 and 4, the prediction was accurate as well. The two images were actually non-cataract just as our system predicted.

Table 3 shows how the values used in the generation on the AUC for all images considered for this experiment. The proposed model generated AUC of 84%, which is better than the state-of-the-art used in detecting cataract.

VIII.” CONCLUSIONS AND FURTHER WORK

The evaluation performed on the system has beyond reasonable doubt shown how great the ConvNets Classifier is. Though the dataset was enriched with great details on the problem (images used for training were retrieved from google search results), the developed model still attained a high AUC of 84.56% and accuracy score of 78% which was influenced by the transfer learning trained on ImageNet. With a sensitivity of 69% and a specificity of 86% .This work has shown the strength of the ConvNets classifier and proves that it can be used to solve similar and even more complex problems. This technique can be used in many other fields and is not limited to the medical field as can be found in. It showed how deep learning can be used to detect and classify cataractous and normal eye images as a CAD. It worked using transfer learning to show how time and

hardware could be conserved will achieving high accuracy in the development of models.

Table 3: Table showing values used to calculate the AUC using the trapezoidal formula

i	Y-axis (TPR _i)	X-axis (FPR _i)	Area=(FPR _{i+1} -FPR _i)* (TPR _{i+1} +TPR _i)/2
1	0	0	0
2	0.5	0	0
3	0.56	0	0.0833
4	0.63	0.14	0
5	0.69	0.14	0
6	0.69	0.14	0.1035
7	0.69	0.29	0.1099
8	0.88	0.43	0
9	0.94	0.43	0.1316
10	0.94	0.57	0.4171
11	1	1	
			Total Area=0.8454
			AUC=(0.8454)*100 =84.54%

The use of the convolutional neural network can be further used to detect early cataract formation using images. In future, we intend to extend the capability of the system to be able to dictate glaucoma and other eye diseases so that CAD can reduce the rate of people blinded by eye illness.

IX." REFERENCES

[1] U. D. o. Health and H. Services, "Cataract: What You Should Know," *Washington DC: NIH Publication*, 2003.

[2] "Cataracts," in *National Eye Institute* ed, 2010.

[3] C. Obinna, "Nigerians risk more blindness from cataract, if...," in *Vanguard*, ed, 2011.

[4] G. D. Magoulas and A. Prentza, "Machine Learning in medical applications," *Springer Berlin Heidelberg*, pp. 300-307, 2001.

[5] T. O. Ayodele, "Types of Machine Learning Algorithms," University of Portsmouth, United Kingdom 2010.

[6] J. Jiang, P. Trundle, and J. Ren, "Medical image analysis with artificial neural networks," *Computerized Medical Imaging and Graphics*, vol. 34, pp. 617-631, 2010.

[7] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*, 2014, pp. 184-199.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.

[9] X. Fern. (2011), Artificial Neural Network. *Oregon State University*. Available: <http://web.engr.oregonstate.edu/~xfern/classes/cs534/notes/neuralnet-9-11.pdf>

[10] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, *Learning and Transferring Mid-Level Image Representations*. MSR, New York, USA;INRIA, Paris, France, 2014.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in neural information processing systems*, 2012.

[12] A. Karpathy. (2015), Transfer Learning. *CS231n Convolutional Neural Networks for Visual Recognition*. Available: <http://cs231n.github.io/transfer-learning/>

[13] TensorFlow, "MNIST For ML Beginners," in *TensorFlow*, ed, 2016.

[14] T. Fawcett, *ROC Graphs: Notes and Practical Considerations*: HP Laboratories Palo Alto, 2003.

[15] Knowino. (2010). Artificial Neural Network. Retrieved from Tel viv University: http://www.tau.ac.il/~tsirel/dump/Static/knowino.org/wiki/Artificial_neural_network.html

[16] Paul, A., Manuel, B., Michael, U., Benes, L., Valeria, F., & Kayoko, K. (2009). Computerized Cataract detection and Classification. *Current Eye Research*.

[17] Cataracts. (2010). Retrieved from National Eye Institute : <https://www.nei.nih.gov/eyedata/cataract>

[18] US Department of Health and Human Services. (2003). Cataract: What You Should Know. Washington DC: NIH Publication.

X." APPENDIX

This section shows the page Screenshot from the proposed system.

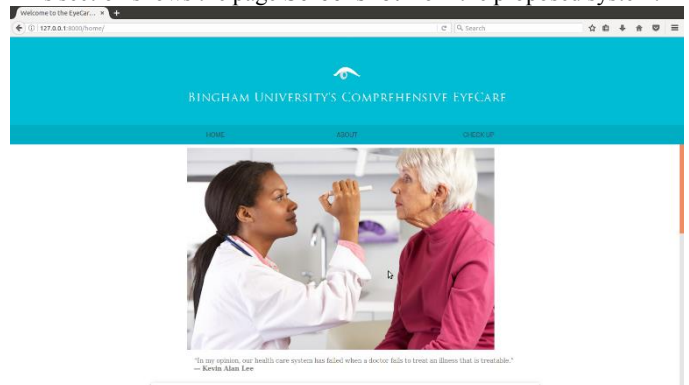


Figure 4.1: Screenshot of homepage of the Web Application

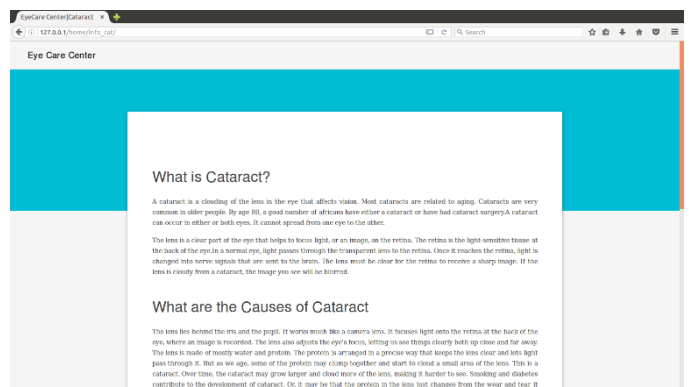


Figure 4.2: Screenshot of the about page of the web application showing information about Cataract and how to use the App.

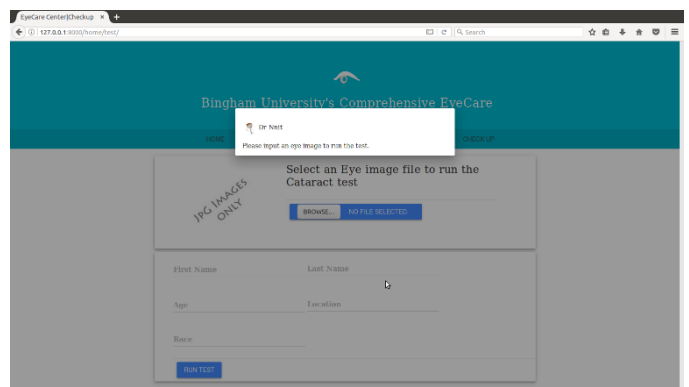


Figure 4.3: Screenshot of the Test page showing the welcome note before any test is performed

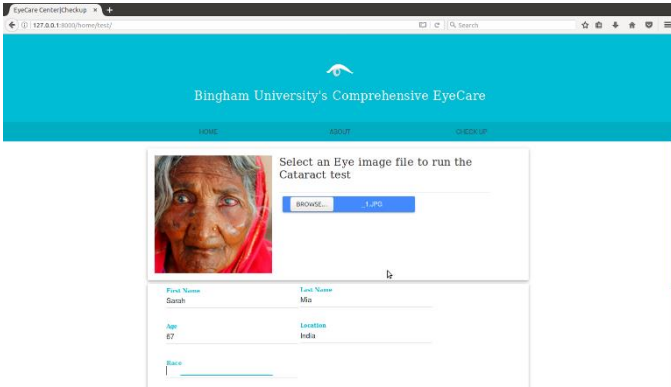


Figure 4.4: Screenshot of the Test page showing a test session and how data is entered

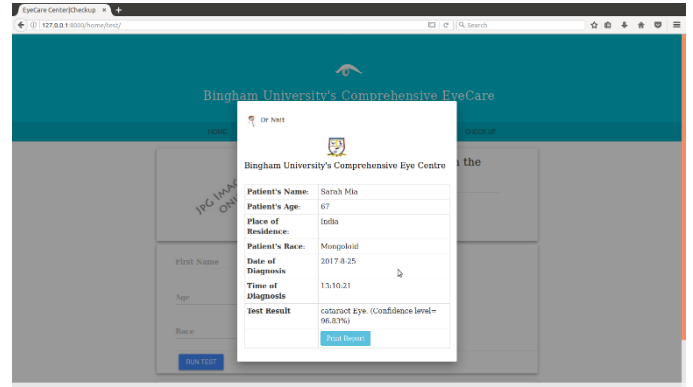


Figure 4.5: Screenshot of the Test page showing results of a diagnostic test along with the confidence level of the result.