T.C.

AKSARAY UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

DEPARTMENT OF ELECTRICAL-ELECTRONIC AND COMPUTER ENGINEERING

BUILDING SMART ALGORITHM TO EXTRACT FEATURES OF TOPOGRAPHIC IMAGES OF A HUMAN EYE

MASTER OF SCIENCE THESIS

Nazar Salih ABDULHUSSEIN

SUPERVISOR

Asst. Prof. Dr. Özlem AKGÜN

AKSARAY, 2018

T.C.

AKSARAY UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE

DEPARTMENT OF ELECTRICAL-ELECTRONIC AND COMPUTER ENGINEERING

BUILDING SMART ALGORITHM TO EXTRACT FEATURES OF TOPOGRAPHIC IMAGES OF A HUMAN EYE

MASTER OF SCIENCE THESIS

Nazar Salih ABDULHUSSEIN

SUPERVISOR

Asst. Prof. Dr. Özlem AKGÜN

AKSARAY, 2018

Aksaray Üniversitesi, Fen Bilimleri Enstitüsü'nün **162335815** numaralı Yüksek Lisans öğrencisi, **Nazar Salih ABDULHUSSEIN** tarafından hazırlanan "**BUILDING SMART ALGORITHM TO EXTRACT FEATURES OF TOPOGRAPHIC IMAGES OF A HUMAN EYE**" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Dr. Öğr. Üyesi Özlem AKGÜN

AKSARAY ÜNİVERSİTESİ

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. .....

Üye: Prof. Dr. Mehmet Reşit TOLUN

BAŞKENT ÜNİVERSİTESİ

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. .....

Üye: Doç. Dr. Yunus UZUN

AKSARAY ÜNİVERSİTESİ

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. .....

Tez Savunma Tarihi: 10/01/2019

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Doç. Dr.....

Fen Bilimleri Enstitüsü Müdürü

DECLARATION

The idea for this present study was taken from the literature regarding the topics. It was announced here that whole work presented in this project has been composed and originated by myself unless otherwise specified. The study was conducted completely under the supervision of Asst. Prof. Dr. Özlem AKGÜN.

Nazar Salih ABDULHUSSEIN

ACKNOWLEDGEMENT

At first, I want to thank my supervisor Asst. Prof. Dr. Özlem AKGÜN for her invaluable advice and support throughout the year. I am also grateful to Asst. Prof. Dr. Nebras HUSSEIN from Biomedical Engineering Department, Al-Khawarizmi College of Engineering, Baghdad University, for helping me to understand the topographic maps of the cornea and how to determine if it is normal or abnormal. Special thank goes to Al-Amal Eye Private Clinic in Baghdad, Iraq for permitting the use of their eye image database. I am also thankful to my friend Mohammed Emad Al-Deen for helping me in using Python, and his helpful suggestions.

Finally, I must thank my parents and my wife for their support and encouragement over the year.

Nazar Salih ABDULHUSSEIN

AKSARAY, 2018

TABLE OF CONTENTS

M.Sc. THESIS

BUILDING SMART ALGORITHM TO EXTRACT FEATURES OF TOPOGRAPHIC IMAGES OF A HUMAN EYE

Nazar Salih ABDUHUSSIEN

Aksaray University
Graduate School Of Natural And Applied Science
Department Of Electrical-Electronic And Computer Engineering

Supervisor: Asst.Prof. Dr. Özlem AKGÜN

ABSTRACT

Precise diagnosis for a wide range of diseases infecting the human eye is a commitment. Therefore, developing new, smart algorithms is necessary to enhance doctors' diagnostic decisions. The recently invented Pentacam is a measurement system that introduces topographic maps for the cornea, measures changes upon it, and helps doctors to make a precise diagnosis. This study extracts features from corneal topographic maps to improve the Pentacam readings and further support precise diagnosing by using deep learning techniques, with an analytical view of the extracted features. A 16-layer convolutional neural network (CNN) was trained using the VGG-16 network to extract powerful features from corneal topographic maps. A sample of 732 human eyes were selected from enlarged topographic images from both genders (414 females and 318 males), divided into two groups: normal and abnormal. The patients' ages ranged from 12 to 76 years. The procedure of the study consisted of three major steps: (1) classification of the extracted features according to the refractive map type (where the estimated accuracy was 96.6%); (2) prediction of the clinical state (normal or abnormal) per individual map (where the estimated accuracy was 88.8%, 98.9%, 94.8%, and 94.5% for the sagittal map, the elevation front map, the elevation back map, and the corneal thickness map, respectively); and (3) comparison of the predicted results and clinical decision-making. The agreement between them reaches about 94.72%, which indicated the power and usefulness of the proposed algorithm.

Keywords: Artificial intelligence, Convolutional neural networks, Deep learning, Features extraction, Corneal topographic maps, VGG-16.

December, 2018; 82 pages

YÜKSEK LİSANS TEZİ

İNSAN GÖZÜNÜN TOPOĞRAFİK GÖRÜNTÜLERININ ÖZELLİKLERİNİ ELDE ETMEK İÇİN AKİLLİ ALGORİTMA OLUŞTURMAK

Nazar Salih ABDULHUSSİEN

Aksaray Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Ve Bilgisayar Mühendisliği

Danışman: Dr. Öğr. Üyesi. Özlem AKGÜN

ÖZET

İnsan gözünü etkileyen çok sayıda hastalıklar için kesin bir tanı koymak önemli bir sorumluluktur. Bu nedenle, doktorların teşhisle ilgili kararlarını geliştirmek için yeni, akıllı algoritmalar geliştirmek gereklidir. Yakın zamanda bulunan Pentacam, kornea için topoğrafik haritaları gösteren, bunlar üzerindeki değişiklikleri ölçebilen ve doktorların kesin bir teşhis yapmasına yardımcı olan bir ölçüm sistemidir. Bu çalışma, Pentacam sisteminin okumasını geliştirmek için kornea topoğrafisi haritalarından elde edilen özellikleri seçip çıkarmaktadır ve çıkarılmış özelliklerin analitik incelenmesiyle derin öğrenme tekniklerini kullanarak kesin teşhisi koymayı desteklemektedir. Kornea topoğrafik haritalarından güçlü özellikle elde etmek için VGG-16 ağı kullanılarak 16 katmanlı bir kıvrımlı sinir ağı (CNN) eğitilmiştir. Normal ve anormal olmak üzere iki gruba ayrılan her iki cinsiyetin (414 kadın, 318 erkek) genişletilmiş topoğrafik görüntülerinden 732 insan gözü örneği seçilmiştir. Hastaların yaşları 12 ile 76 arasında değişmektedir. Bu çalışmanın yöntemi üç ana adımdan oluşmaktadır: (1) Refraktif (Kırılma) harita türüne göre çıkarılmış özelliklerin sınıflandırılması (tahmin edilen doğruluk payı %96.6'dır); (2) her bir harita için klinik durumunun tahmini (normal veya anormal) (tahmin edilen doğruluk payı %88.8, %98.9, %94.8, ve Sagital harita için %94.5'dir, sırasıyla yükseklik ön haritası, yükseklik arka haritası, ve kornea yüksekliği haritası), ve (3) tahmini sonuçlar ve klinik karar vermedir. Aralarındaki uyum, önerilen algoritmanın gücünü ve kullanışlılığını gösteren %94.72 değerine ulaşmaktadır.

Anahtar Kelimeler: Yapay zeka, Kıvrımlı sinir ağları, Derin öğrenme, Özellikleri çıkarma, Kornea topoğrafi haritası, VGG-16.

Aralık, 2018; 82 sayfa

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ANNs	Artificial Neural Networks
CBR	Case-Based Reasoning
CNN	Convolutional Neural Network
CYL	Cylinder
DT	Decision Tree
FC	Fully Connected
GPU	Graphics Processing Unit
GUI	Graphical User Interface
KC	Keratoconus
KCS	Keratoconus Suspected
KPI	Keratoconus Prediction Index
MLP	Multi-Layer Perceptron
NNDTEDDS	Neural Networks Decision Trees Eye Disease Diagnosing System
RBFNN	Radial Basis Function Neural Network
ROC	Receiver Operator Characteristic
SPH	Sphere
SVM	Support Vector Machine
VGG	Visual Geometry Group
WHO	World Health Organization
ZC	Zernike Coefficients

1. INTRODUCTION

Topographic images of the human corneal anterior and posterior surfaces, created by a system called Pentacam, are becoming more important diagnostic tools than ever in the classification of healthy and unhealthy corneas. The final maps of this system carry many details that guide medical decisions. Finding hidden details and reporting the expected clinical situation are usually strongly dependent on the experience and knowledge levels of the final decision maker (ophthalmologist); however, artificial intelligence (AI) has begun to play the main role in medical imaging analysis and smart diagnosis. These unprecedented tools, accompanied by huge levels of development in computer (software and hardware) science, introduce a new area of analysis. The ability to recognize particular features of corneal topographic maps and define the clinical interrelationship among them offers new roles to artificial intelligence in the medical imaging field. The features that AI recognizes have been previously learned about and specified through a number of pre-diagnosed cases reported by ophthalmologists.

1.1 Cornea

The cornea is the outlying part of the human eye and holds significant focusing power. It is domed and transparent and is characterized by a perfect organization of tissues with neither substances nor blood vessels. It is nourished and protected from infection by two other parts, the fluid layer that resides behind it, called the aqueous humor, and the tears(Camarillo et al., 2002).

1.2 Corneal Diseases

For post-surgery cases and certain diseases, it is important to interpret the biomechanical response of the cornea. The cornea is the topmost lens of the eye and is accountable for 65-75% of sight strength; therefore, any disorder in the corneal curvature results in a sight disorder or refractive problem, including astigmatism, hyperopia, and myopia. High corneal curvature leads to Short-sightedness or myopia; low corneal curvature causes farsightedness or hyperopia (Mazen, 2015). These two refractive diseases may also occur because of a malfunction of the lens. Additionally, uneven or abnormal corneal curvature causes blurred vision or astigmatism (Maeda and Klyce, 1995).

1.3 Pentacam

Pentacam, a powerful eye segment scanner based on the Scheimpflug camera measurement, is a robust invention that helps ophthalmologists investigate the corneal anterior surface. Although the Scheimpflug technique for scanning anterior eye segments has been known since the 70s, it was not released commercially until the 80s and 90s. In 2005, it began to be put to wide clinical use (Smolek and Klyce, 1997). Figure 1.1 shows side and front view of the Pentacam.

Figure 1.1.a. Pentacam with the four refractive maps from the clinician side (URL-7).

Figure 1.1.b. Pentacam from the patient side (URL-7).

1.4 Corneal Topographic Maps

The four major topographic patterns are the pachymetry pattern, the anterior curvature sagittal pattern, and the anterior and posterior elevation patterns (Mazen, 2015), see Figure 1.2. These patterns reflect many anterior eye surface details that specialists should test.

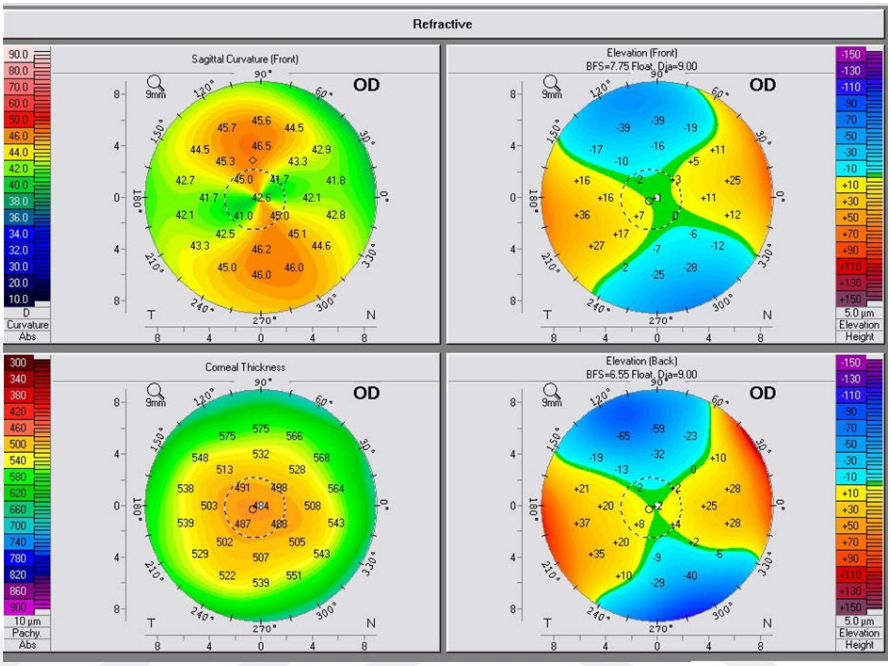


Figure 1.2. Example of four refractive maps for corneal topography from Pentacam.

1.5 Related Work

1.5.1 Artificial neural networks (ANNs)-based classifiers

As early as 1995, researchers attempted to evaluate the capabilities and usefulness of an automated system based on artificial neural networks (ANNs) in interpreting corneal topographic maps. The difficulties of corneal topographic pattern interpretation, specifically that of topographic maps' similarities among pathologies, encouraged these researchers to investigate the capabilities of an automated approach to diagnosing shape abnormalities in the cornea. The study reported that the ANNs system might require further enhancements to become an objective computerized classification tool for video-keratography. This tool would help doctors identify abnormalities in corneal topographies (Maeda and Klyce, 1995). Smolek and Klyce (1997) conducted a study to compare four video-keratographic-based keratoconus detection methods to a new ANNs-based detection method. The researchers designed a neural network approach that was able to classify keratoconus screenings automatically as "keratoconus" (KC) or "keratoconus suspected" (KCS). The

findings determined that the ANNs approach succeeded in differentiating between the topographies of the KC and KCS screenings, and even in differentiating these from other cases resembling KC. The outcome also proved that the new ANNs detection method outperformed other detection methods examined in the study in terms of specificity and accuracy. However, the researchers also demonstrated that the sensitivity of all detection methods, including ANNs, was equal (Smolek and Klyce, 1997). For classifying particular types of corneal shapes, a study was conducted in 2008 to develop and compare two different techniques-discriminant analysis and ANNs-provided that the Zernike coefficients were given. It is noteworthy that few studies have applied Zernike coefficients (ZC) as input parameters (Carvalho, 2005; Schwiegerling and Greivenkamp, 1996). Although the researchers used a comparatively small database, the study`s outcome suggests that, for the automated diagnosis of video-keratography patterns by ANNs or discriminant analysis, Zernike coefficients may be reliable input parameters as descriptors of corneal shape (Carvalho and Barbosa, 2008). In 2018, researchers at Shiley Eye Institute at the University of California San Diego Health and the University of California San Diego School of Medicine developed a new ANNs-based system to examine patients with common, treatable retinal diseases that cause blindness. The diagnoses of both age-related macular degeneration and diabetic macular edema were comparable to expert classifications and also indicated that the automated system introduced more interpretable and transparent diagnoses because it was able to determine which regions the ANNs identified (Kermany et al., 2018; LaFee, 2018).

1.5.2 Support vector machines (SVMs)-based classifiers

In 2005, a framework was developed that was built on support vector machines (SVMs) for ophthalmology pattern interpretation. Its purpose was to assist optometrists in the diagnosis of ocular refraction problems, including short-sightedness, hypermetropia, and astigmatism. The outcome of the project recommended a method for improved understanding of the eye patterns obtained from a system called Hartmann-Shack, provided that other ocular error measurements were detected and taken. The system was designed to interpret and analyze eye patterns as a whole without the need of a reference pattern to evaluate the differences between data obtained from the target pattern and the reference pattern (Netto, 2005). In 2008, some researchers attempted to classify corneal

topographies using support vector machine (SVM) technology. That study was conducted to prove that a learning algorithm using SVM technology could distinguish between corneal diseases, such as keratoconus, suspected keratoconus, hyperopic and myopic laser vision correction, orthokeratology, and pellucid marginal degeneration, from normal corneal conditions. The researchers tested the obtained topographies using the Corneal Topographer ATLAS Model 90000. The algorithm was trained and evaluated using previous data on 239 abnormal corneal cases and 85 normal cases. To train the algorithm, four-fifths of the cases were selected randomly, while the remaining one-fifth of the cases were used for the evaluation. The accuracy, sensitivity, and specificity of the algorithm were equal to or greater than 90%. The SVM-based model was able to identify the cases of abnormal corneas from normal eyes using the randomly picked training data (Bagherinia et al., 2008). In 2012, a study developed an SVM-based algorithm for subclinical keratoconus and keratoconus diagnosis using tomography and topography data. It introduced a new data classifier algorithm to detect keratoconus using corneal measurements obtained from Placido corneal topography together with a Scheimpflug camera. The SVM-based algorithm demonstrated high specificity, sensitivity, and accuracy in distinguishing four eye conditions: eyes with a previous surgery history, eyes with subclinical keratoconus or keratoconus, and normal eyes. The algorithm's accuracy was excellent for the cases regardless of whether or not data about the corneal thickness and posterior corneal surface was provided. The rate of valid predictions was >95% when provided with additional data, and 93% when no additional data was provided. It is obvious that when additional posterior corneal surface data was provided, the precision was better. The pachymetry and the additional data for the posterior and anterior corneal surfaces increased the sensitivity of the classifier (Arbelaez et al., 2012). Another study developed a new diagnosis system for vision-related problems. It presented a system that is capable of performing two tasks; it can diagnose short-sightedness (myopia) or far-sightedness (hypermetropia) and it can identify the sphere (SPH) (eyesight distance) and cylinder (CYL) (deviation) measurements. The researchers believe that it is easy for a physician to diagnose the problem if the SPH or CYL are within the normal range of measurements; however, when the measurements are not within the normal range, it takes more time for the doctor to determine the patient's eye condition. Therefore, proposed a model based

on SVM and artificial neural networks (ANNs) to detect eyesight problems and to help doctors decide how to treat those conditions appropriately and quickly. ANNs are used because they are known to be effective for checking and classifying the sightedness of the eye. The developers trained the ANNs using a Backpropagation algorithm due to its high capability to classify maps, especially nonlinear mapping. The results showed that with ANNs, the specificity of nonlinear SVMs increased from 92% to 97%. These results prove that ANNs and SVM are effective pattern recognition techniques (Kotsia and Pitas, 2007; Pontil and Verri, 1998) for detecting the state of eyesight (Noaman et al., 2014). Another study aimed to test the reliability of an SVM-based algorithm for detecting corneal patterns objectively by automatically using a set of 22 parameters provided by Pentacam measurements; it also aimed to compare the algorithm model outcomes with the results obtained from other popular keratoconus classifiers or the classification, 22 parameters were applied to an SVM algorithm developed as a computer software component based on a collection of machine-learning algorithms, called Weka. The researchers calculated and compared the accuracy through cross-validation tests for keratoconus versus normal eyes, forme fruste versus normal eyes and across all five cohorts, with other outcomes obtained from some other popular classification models. In case of the keratoconus vs. normal eyes, the algorithm's accuracy was 98.9%; its sensitivity for keratoconus detection was 99.1% and its specificity was 98.5%. For the forme fruste versus normal eyes case, the results were 93.1% for accuracy, 79.1% for sensitivity, and 97.9% for specificity. For the five cohorts, the algorithm's accuracy was 88.8%, its sensitivity was 89.0%, and its specificity was 95.2% (Ruiz Hidalgo et al., 2016).

1.5.3 Decision tree (DT)-based classifiers

In 2005, a study applied an automated classification system based on machine learning and decision tree induction to quantitatively and objectively distinguish between keratoconus and normal corneal shapes. That study compared the developed classifier with other popular classification systems. A model of a seventh-order Zernike polynomial was applied to the corneal surface. The classifier was based on the C4.5 algorithm for the decision tree classifier. The outcomes of the automated classifier were compared with several modified indices: the KISA%, the Cone Location and Magnitude classification system, the Keratoconus Prediction Index (KPI), Schwiegerling's Z3, and the Rabinowitz–McDonnell classification system;

and the standard classification thresholds were followed for each of these methods. For every classification method, the researcher also examined the area under the receiver operator characteristic (ROC) curve. The performance of the developed classification method based on the decision tree model was greater than or equal to the other classification methods that were examined. The results showed that the accuracy of the developed method was 92%, and the area under the curve of ROC was 0.97. With four of the 36 surface features of Zernike polynomial coefficients, the automated decision tree model decreased the amount of required information needed to discriminate between keratoconus and normal eyes. The study used four classification attributes: greater sagittal depth, inferior elevation, trefoil, and oblique toricity. The research findings confirmed the capability of the decision tree-based classifier to distinguish from among various corneal shapes; that the study's results also demonstrated that the decision tree-based classifier is a dependable quantitative classifier for Zernike polynomials and any device that can produce raw elevation data. The study also reported that the framework is applicable to other classification problems (Twa et al., 2005). In 2012, Kabari and Nwachukwu introduced a framework combining ANNs and decision trees to diagnose various eye conditions, such as myopia, hyperopia, and astigmatism. That study is distinguished from other related studies because it used two algorithms instead of one for data mining the ANNs and decision trees. The researchers presented a hybrid model called a Neural Networks Decision Trees Eye Disease Diagnosing System (NNDTEDDS). They asserted that both technologies are tools that can used to discover knowledge and introduce a hybrid model; they developed rules that show how the diagnosis of eye diseases is done based on the physical conditions of the eye and its related symptoms. These rules show how knowledge can be obtained from neural networks after the algorithm is trained from earlier samples of physical conditions of eyes and disease symptoms. That study's findings showed that the proposed NNDTEDDS was successful (92%) at diagnosing eye diseases; this demonstrates that combining these two technologies is an efficient and effective way to diagnose eye conditions, such as myopia, hyperopia, and astigmatism. Moreover, Kabari and Nwachukwu assured that the rules they developed are useful for teaching younger ophthalmologists how to properly diagnose eye conditions (Kabari and Nwachukwu, 2012).

1.5.4 Naive bayes-based classifiers

An experimental research study introduced an automated, Naive Bayes-based system for detecting eye diseases. The automated system uses Case-Based Reasoning (CBR) as an experience-based model for reasoning and the Naïve Bayes as an eye diseases classifier by adopting Bayes' Theorem. The findings showed that the accuracy of the automated Naïve Bayes model was 82%. Accordingly, the authors concluded that an expert system based on a combination of Naïve Bayes and CBR can be a promising technology for diagnosing eye diseases. While the Naïve Bayes classifier is capable of introducing effective diagnoses for people, more improvements and enhancements are still required (Kurniawan et al., 2014).

1.5.5 Studies comprised of several classifiers

The focus of a study conducted in 2012 was to examine the effectiveness of an artificial intelligence (AI)-based system to diagnose keratoconus. The researcher believed that the developed system would help device producers enhance their production so that the medical device could assist experts and automatically support the recognition phase of the diagnosis. They trained some classifiers, such as decision tree, SVM, ANNs, radial basis function neural network (RBFNN), and multi-layer perceptron (MLP), to recognize whether the patient has keratoconus or is suspected of having keratoconus. The classifiers were trained with part of the dataset and tested using the rest of the dataset. The output of the study indicated that the proposed algorithm's accuracy was 91%, so it is capable of discriminating between keratoconus, suspected keratoconus, and normal eyes (Toutounchian et al., 2012). In 2014, a survey was conducted to review several automated, computer-based systems to diagnose ocular diseases. The researchers focused on three types of data: imaging, clinical, and genetic. The study reported that these types of data were the most commonly used in computer-aided diagnosis of ocular diseases, such as pathological myopia, age-related macular degeneration, glaucoma, and diabetic retinopathy. The researchers asserted that, over the past several years, the diagnosis of ocular diseases based on computer-aided models has substantially improved. However, the significance of having fully-automated models that are capable of exploiting clinical knowledge and that incorporate mixed data sources still needs to be more fully investigated (Zhang et al., 2014).

1.5.6 Google’s deepmind health project

AI is extensively applied for tracing, normalizing, collecting, and storing data in the field of healthcare. In this regard, Google’s AI research branch announced a new project, DeepMind Health, to investigate the extent to which technology could be used to efficiently analyze eye scans using medical records data mining to deliver faster and better health services and improve the treatment of eye conditions. Launched in 2016, DeepMind Health is a cooperative project with the Moorfields Eye Hospital NHS Foundation Trust. Although the project is part of the early stages of AI research in health, the DeepMind Health project team has reassured that the undertaken research will have practical benefits, and the outcomes of the study will be submitted to academic journals that are peer-reviewed after being subjected to strict clinical scrutiny (Mesko, 2017).

1.6 Problem Statement

The objective of this thesis is to give diagnosing for the cornea if it is normal or abnormal to support the doctor's decision. The pre-trained convolutional neural network VGG-16 used in this project stems from the paper "Very deep convolutional networks for large-scale image recognition" (Simonyan and Zisserman, 2014).

1.7 Conclusion

It is beneficial and helpful to explore the plenty of attempts of applying AI to eye diseases diagnosis throughout the history. The exploration of related works over the past decades gives some insights as to what extent the AI-based technology supports the medical care in areas where much effort and time are required by specialists. Numerous studies have been conducted to test the accuracy and specificity of one or more of the artificial intelligence algorithms such as the ANNs, SVMs, DT, and Naive Bayes. Some researchers have tested one algorithm while others have concerned with combining two or more algorithms. All the reviewed research reported high efficiency and effectiveness of the tested algorithms.

2. THEORETICAL CONCEPTS

2.1 Introduction

The cornea is the outlying part of the human eye, of a domed shape, and transparent in nature. It holds significant focusing power (Camarillo et al., 2002), and, as indicated by the World Health Organization (WHO), corneal blindness is one of the significant factors of visual deficiency and the fourth reason for corneal darkness all around (5.1%). Frequently, reports are incomplete, but 1.5 to 2.0 million new instances of unilateral darkness are rated annually (URL-13). The Pentacam is a powerful scanner for segments of the eye, based on the Scheimpflug camera measurements. It is a robust invention that helps ophthalmologists in investigating the corneal anterior surface (Smolek and Klyce, 1997; Mazen, 2015). The four refractive maps help to detect corneal disease (the sagittal map, the pachymetric or corneal thickness map, the elevation front map, and the elevation back map). Each map gives an indication about the eye, and after reviewing all of the features from the four refractive maps, a diagnostic decision can be made (Ali et al., 2017). This work presents a diagnostic method that uses deep learning to extract features from the four refractive maps provided by the Pentacam, and then enters these features into the support vector machine (SVM) classifier for accurate diagnosing that supports the clinician's opinion. In this study, each normal indication will be written as NORMAL, and ABNORMAL will be written for the abnormal indications.

2.2 Deep Learning

Today, the classification of images is accomplished using deep learning techniques (Tan et al., 2017). It combines both classification and feature extraction. These techniques can deliver great outcomes utilizing complicated networks built with large-scale data. For this study, we trained a Convolutional Neural Network (CNN) using deep learning to recognize and classify each of these topographic maps as normal or abnormal. During training, the features were extracted from the input images to provide a powerful and deep CNN model (the VGG-16 network was used here), and unknown images were classified efficiently during the testing stage. The particular structure of the trained network is explained below.

2.3 Convolutional Neural Network (CNN)

In neural networks, CNN's are a deep learning model that contain multiple network layers. The more profound the network, the better it learns. It can self-learn furthermore self-organize without administration (Acharya et al., 2018). The majority of the convolutional models will consist of the following operations, which can be considered as the building blocks of CNN's.

2.3.1 Image input layer

This layer should be available on all networks. This layer carries the input to the network and can utilize 2-D and 3-D images. The image dimension must be introduced in this layer (Riesenhuber and Poggio, 1999).

2.3.2 Convolutional layer

A convolutional layer is the most essential element of a CNN it implements a convolution task on the previous layer. A window is moved over the input image with a stride that is indicated by the client. In the wake of performing convolution, feature maps are generated and used as input for the next layer (Cireşan et al., 2011; Scherer et al., 2010).

2.3.3 Rectified linear unit (ReLU)

The ReLU is one of the CNN layers in which, for each input element, a thresholding operation is performed; for example, a value fewer than zero is deposited to zero. This decreases data repetition and maintains the significant features intact. The size of this layer's output is similar to the past layer (Raghavendra et al., 2018).

2.3.4 Max-pooling layer

Pooling layer is implemented to every map feature to diminish its size. The value of stride is typically chosen by the client. The extreme value over the window is considered and the window is supplanted by that number. The output size of this layer is lower than that of the past layer (Cireşan et al., 2011; Scherer et al., 2010; Raghavendra et al., 2018).

2.3.5 Fully-connected layer (FC)

The neurons will be selected from the previous layer and correlated to all neurons in this layer. The yield of FC will be the number of classifications (Serre et al., 2005).

2.4 VGG-16 Network

The Visual Geometry Group (VGG) in conjunction with the University of Oxford's Department of Engineering Science, built up this design with 16 layers (Thoma, 2017; Grassmann et al., 2018). The VGG architecture has been one of the most popular CNN models since its introduction in 2014. The reason for its popularity is in its model simplicity and the use of small-sized convolutional kernels, which leads to very deep networks (Khan et al., 2018). The VGG-16 architecture strictly uses 3×3 filters with stride of 1 in convolution layer and uses 2x2 padding with stride of 2 in max-pooling layers and intermediate max-pooling layers for feature extraction and a set of three fully connected layers toward the end for classification. Each convolution layer is followed by a ReLU layer in the VGG architecture (Khan et al., 2018). The motivation for selecting the VGG-16 network was its popularity and reputation for being user-friendly. The default data size for the VGG-16 model is 224×224 pixels (URL-4). The general design of the VGG-16 network can be found in Figure 2.1.

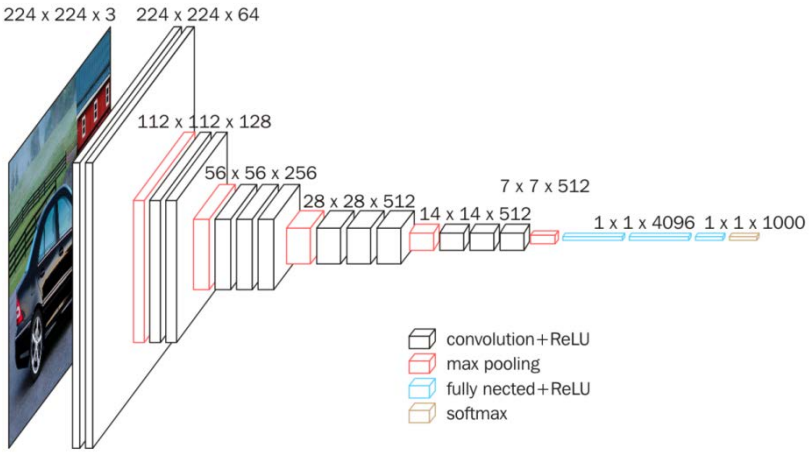


Figure 2.1. VGG-16 Model Architecture (URL-7).

2.5 Support Vector Machine (SVM)

SVM which is a binary supervised classifier; it works by finding the maximum separating hyperplane between two classes of cases, for that SVM considered better in generalizing (Smitha et al., 2011). It maximizes the edges and for that creating the largest distance between the hyperplane separation and the cases on each side of it. It has been proven to reduce un upper limits on the generalization error expectations (Kotsiantis et al., 2006).

3. METHODOLOGY

3.1 Architecture

The CNN architecture used here is VGG-16 which is introduced by Simonyan and Zisserman in their 2014 paper, "Very Deep Convolutional Networks for Large Scale Image Recognition" (Simonyan and Zisserman, 2014). The VGG-16 network is a simpler model since because of its limited use of hyperparameters. The network constantly uses 3×3 filters with a stride of 1 in the convolution layer and uses a similar padding in 2×2 pooling layers with a stride of 2. The default input size for the VGG16 model is 224x224 pixels. An overview of the VGG-16 network structure is seen below in Figure 3.1.

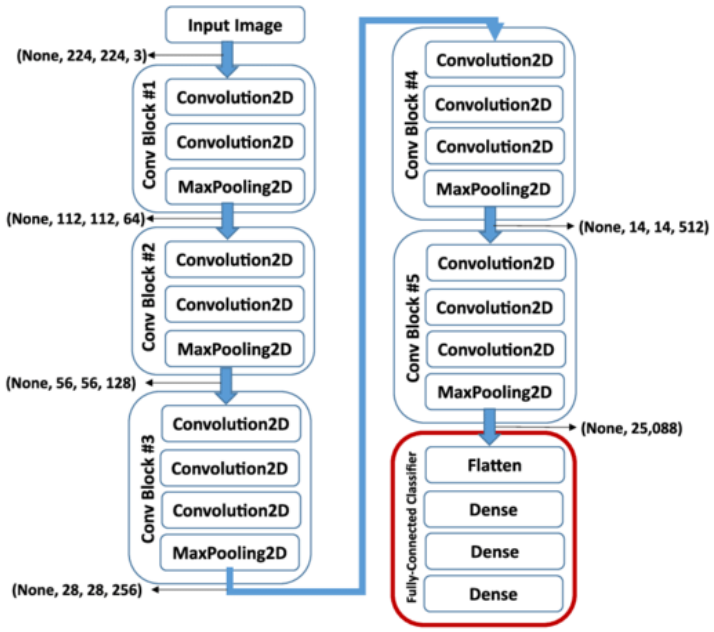


Figure 3.1. A schematic of the VGG-16 architecture (Gopalakrishnan et al., 2017).

3.2 Training and Testing

The dataset is separated into two parts: training and testing. Initially, 80% of the randomly selected samples are used for training and the remaining 20% are used for testing. The features are extracted as follows:

1- Determine the kind of refractive map (sagittal, elevation front, elevation back, corneal thickness). In this study, (732) images were used; thus, 80% (586) of the images were used for training and 20% (146) for testing.

2- Determine if these maps are normal or abnormal for each of the four refractive maps:

a. Sagittal map: 334 images were used; thus, it was applied to 80% of the sagittal images for training (268: 170 normal and 98 abnormal) and 20% for testing (67: 43 normal and 24 abnormal).

b. Elevation front map: 118 images were used; thus, it was applied to 80% of the sagittal images for training (94: 84 normal and 10 abnormal) and 20% for testing (23: 21 normal and 2 abnormal).

c. Elevation back map: 120 images were used; thus, it was applied to 80% of the sagittal images for training (96: 64 normal and 32 abnormal) and 20% for testing (24: 21 normal and 3 abnormal).

d. Corneal thickness map: 160 images were used; thus, it was applied to 80% of the sagittal images for training (128: 86 normal and 42 abnormal) and 20% for testing (32: 22 normal and 10 abnormal).

3- Finally, collect the results for all refractive maps and determines whether the final result is normal or abnormal (here were select 10 cases as a test).

3.3 Data Analysis

3.3.1 Image dataset

The topographic maps for this work were collected from Al-Amal Eye Private Clinic, Baghdad, Iraq. The four refractive maps were gathered and transferred from the Pentacam to a personal computer for analysis. The maps that were collected were of normal and abnormal corneas, as classified by the specialist. The sample four refractive maps are displayed in Figure 3.2.

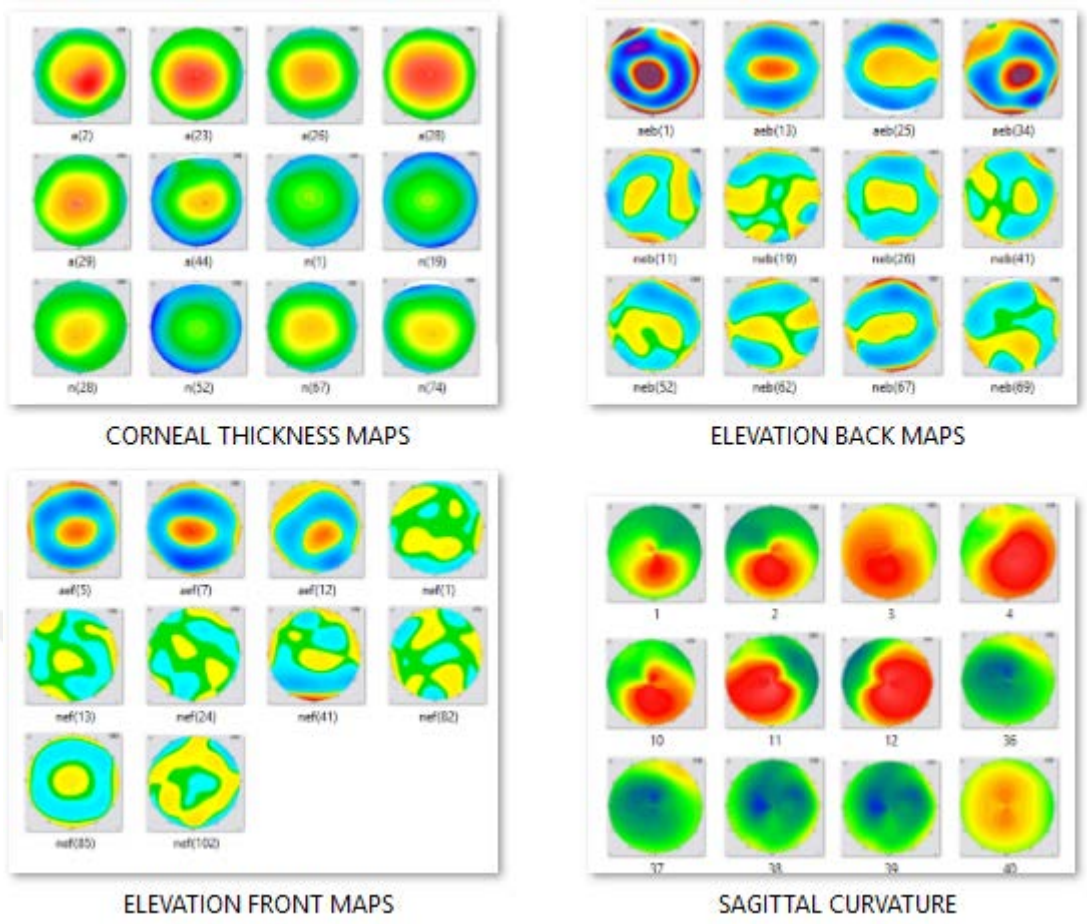


Figure 3.2. Examples of corneal topographic maps.

3.3.2 System requirements

The complete algorithm was executed using a system configuration of Intel Core™ i7-5500 CPU @ 2.40GHz, 16GB RAM. Everything was done in a windows environment and install Python 3.6.5 (URL-11), Keras which is a Python deep learning library (URL-3), OpenCV (URL-6), NumPy (URL-9), SciPy (URL-12), and Matplotlib (URL-5) through the Anaconda Distribution since Anaconda comes with all of those packages (URL-8).

3.3.3 Models creation

The methodology of the project consists of three major steps:

Step 1: Classification of the extracted features according to the refractive map type. It should create (type of map) model to predict the type of map according to the refractive map type if it is a sagittal map, elevation front map, elevation back map, corneal thickness map.

Step 2: Predict the clinical state (normal or not) per each map individually (it should create 4 models)

Step 3: The comparison between the predicted results and the clinical decision have been done. The agreement between them reaches about 94.72% Which indicated the power and usefulness of the proposed algorithm.

The steps of models creation include: create (type_of_map) model and create (sagittal, elevation front, elevation back, and corneal thickness) models, as seen below:

1- load the data set (training data) which is consists of (586) images including :

- sagittal map (268 images).
- elevation front map (118 images).
- elevation back map (120 images).
- corneal thickness map (160 images).

2- Preprocessing training data and loading into memory.

3- Extract the class label from the image path and update the label list.

4- Construct the image generator for data augmentation.

5- Build the models.

6- Train the network.

7- Save models.

After the create models, it should apply it on testing data as explained below:

1- load the data set (testing data) which is consists of (146) images including :

- sagittal map (67 images).
- elevation front map (23 images).
- elevation back map (24 images).
- corneal thickness map (32 images).

2- Loading network.

3- load the image, pre-process it, and store it in the data list.

4- Extract the class label from the image path and update the label list.

5- Get the accuracy of each model by using SVM.

3.4 Implementation

Commonly, a maximum number of images should be applied with using a VGG-16 network for real performance (Acharya et al., 2018). In this study, it was used on 732 topographic corneal images to extract features from them. The network consists of 16 layers as shown in Figure 3.1. It was trained the network for 100 epochs, achieving the extraction of features to train a VGG-16 on the corneal maps dataset to distinguish between normal and abnormal maps. A model for each map was created separately:

- Sagittal_map.model
- Elevation_front_map.model
- Elevation_back_map.model
- Corneal_thickness_map.model

And it created a model for determining the type of map:

- Type_of_map.model

The output of the training network for each model was as follows:

1- Sagittal.model

```
Epoch 95/100
8/8 [==============================] - 3s 413ms/step - loss: 0.2878 - acc: 0.8866
Epoch 96/100
8/8 [==============================] - 4s 456ms/step - loss: 0.3369 - acc: 0.8662
Epoch 97/100
8/8 [==============================] - 4s 496ms/step - loss: 0.2970 - acc: 0.8633
Epoch 98/100
8/8 [==============================] - 4s 455ms/step - loss: 0.2745 - acc: 0.9133
Epoch 99/100
8/8 [==============================] - 4s 452ms/step - loss: 0.3037 - acc: 0.8817
Epoch 100/100
8/8 [==============================] - 4s 450ms/step - loss: 0.3831 - acc: 0.8018
[INFO] serializing network...
[INFO] serializing label binarizer...
Training classifier and save
Finished training on data: dataset/data_maps/sagittal
```

Figure 3.3. 80.18% classification accuracy on the training set(sagittal map).

2- Elevation_front.model

```
Epoch 95/100
2/2 [==============================] - 1s 517ms/step - loss: 0.0141 - acc: 1.0000
Epoch 96/100
2/2 [==============================] - 1s 477ms/step - loss: 0.0722 - acc: 0.9688
Epoch 97/100
2/2 [==============================] - 1s 513ms/step - loss: 0.0778 - acc: 0.9688
Epoch 98/100
2/2 [==============================] - 1s 499ms/step - loss: 0.0223 - acc: 1.0000
Epoch 99/100
2/2 [==============================] - 1s 477ms/step - loss: 0.0579 - acc: 0.9531
Epoch 100/100
2/2 [==============================] - 1s 520ms/step - loss: 0.0277 - acc: 1.0000
[INFO] serializing network...
[INFO] serializing label binarizer...
Training classifier and save
Finished training on data: dataset/data_maps/elevation_front
```

Figure 3.4. 100 % classification accuracy on the training set(elevation front map).

3- Elevation_back.model

```
Epoch 95/100
3/3 [==============================] - 2s 502ms/step - loss: 0.1848 - acc: 0.9167
Epoch 96/100
3/3 [==============================] - 1s 492ms/step - loss: 0.1688 - acc: 0.9375
Epoch 97/100
3/3 [==============================] - 1s 497ms/step - loss: 0.1382 - acc: 0.9688
Epoch 98/100
3/3 [==============================] - 2s 503ms/step - loss: 0.1563 - acc: 0.9583
Epoch 99/100
3/3 [==============================] - 2s 505ms/step - loss: 0.1303 - acc: 0.9688
Epoch 100/100
3/3 [==============================] - 1s 499ms/step - loss: 0.1368 - acc: 0.9375
[INFO] serializing network...
[INFO] serializing label binarizer...
Training classifier and save
Finished training on data: dataset/data_maps/elevation_back
```

Figure 3.5. 93.75 % classification accuracy on the training set(elevation back map).

4- Corneal_thickness.model

```
Epoch 95/100
4/4 [==============================] - 2s 495ms/step - loss: 0.1261 - acc: 0.9609
Epoch 96/100
4/4 [==============================] - 2s 507ms/step - loss: 0.1225 - acc: 0.9688
Epoch 97/100
4/4 [==============================] - 2s 505ms/step - loss: 0.1320 - acc: 0.9688
Epoch 98/100
4/4 [==============================] - 2s 502ms/step - loss: 0.1273 - acc: 0.9531
Epoch 99/100
4/4 [==============================] - 2s 506ms/step - loss: 0.1275 - acc: 0.9531
Epoch 100/100
4/4 [==============================] - 2s 496ms/step - loss: 0.1321 - acc: 0.9375
[INFO] serializing network...
[INFO] serializing label binarizer...
Training classifier and save
Finished training on data: dataset/data_maps/corneal_thickness
```

Figure 3.6. 93.75 % classification accuracy on the training set(corneal thickness map).

5- Type_of_map.model

```
Epoch 95/100
17/17 [==============================] - 8s 484ms/step - loss: 0.0880 - acc: 0.9717 - val_loss: 0.1035 - val_acc: 0.9333
Epoch 96/100
17/17 [==============================] - 8s 480ms/step - loss: 0.0998 - acc: 0.9632 - val_loss: 0.1340 - val_acc: 0.9000
Epoch 97/100
17/17 [==============================] - 8s 476ms/step - loss: 0.0829 - acc: 0.9687 - val_loss: 0.0461 - val_acc: 1.0000
Epoch 98/100
17/17 [==============================] - 8s 474ms/step - loss: 0.0927 - acc: 0.9698 - val_loss: 0.0630 - val_acc: 1.0000
Epoch 99/100
17/17 [==============================] - 8s 475ms/step - loss: 0.0635 - acc: 0.9871 - val_loss: 0.1260 - val_acc: 0.8667
Epoch 100/100
17/17 [==============================] - 8s 490ms/step - loss: 0.0777 - acc: 0.9651 - val_loss: 0.0374 - val_acc: 1.0000
```

Figure 3.7. 100 % classification accuracy on the training set (type of map).

The training loss/accuracy plot follows:

In Figure 3.8, It was trained the model (a type of map) for 100 epochs and achieved low loss with limited overfitting. With additional training data, we could obtain higher accuracy as well.

Figure 3.8. Training and validation loss/accuracy plot for (type of map) model.

In figure 3.9, the training models (sagittal map model, elevation front model, elevation back model, corneal thickness model) obtained higher accuracy.

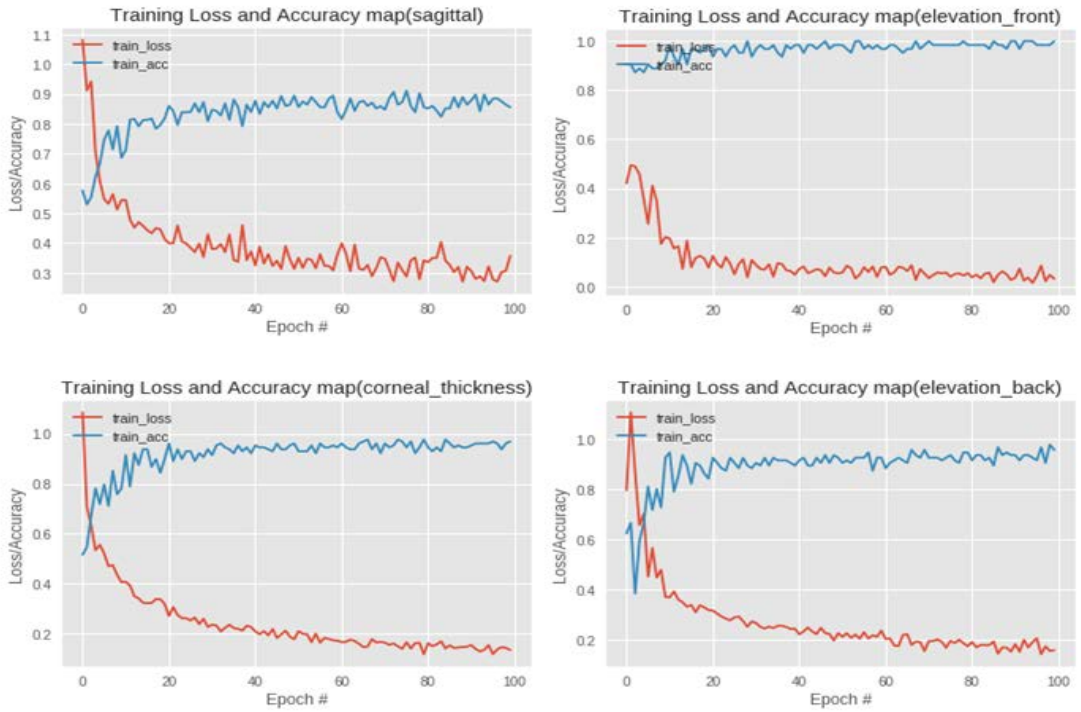


Figure 3.9. Training and validation loss/accuracy plot for four refractive maps.

3.5 Results

A sample of 732 human eyes was selected from enlarged topographic images from both genders (414 females and 318 males), divided into two groups: normal and abnormal. The age range of the patients was 12 to 76 years**.** The procedure of study consisted of three major steps: (1) classification of the extracted features according to the refractive map type (where the estimated accuracy was 96.6%). (2) prediction of the clinical state (normal or abnormal) per individual map (where the estimated accuracy was 88.8%, 98.9%, 94.8%, and 94.5% to the sagittal map, elevation front map, elevation back map, and corneal thickness map, respectively), see Table 3.1. (3) the comparison of the predicted results and clinical decision making. The agreement between them reaches about 94.72%, which indicated the power and usefulness of the proposed algorithm.

Table 3.1. The network's results**.**

Validation results	
Model	Accuracy
type_of_map.model	96.6
sagittal_map.model	88.8
elevation_front_map.model	98.9
elevation_back_map.model	94.8
corneal_thickness_map.model	94.5

3.6 Conclusion

Corneal diseases are one of the leading problems of blindness, and a large percent of the worldwide population is affected by it. The VGG-16 network obtained the highest accuracy of 96.6 % in determining the type of maps. Also, it achieved 88.8%, 98.9%, 94.8%, and 94.5% accuracy in determining whether each was normal or abnormal (sagittal map, elevation front map, elevation back map, and corneal thickness map) respectively. The trained CNN model can efficiently detect the class (normal or abnormal) and type of refractive maps of an unknown image see Figure 3.10. Hence, ophthalmologists should focus their attention more on the normal case to diminish their workload by about half. These models can help ophthalmologists to cross-check their clinical findings.

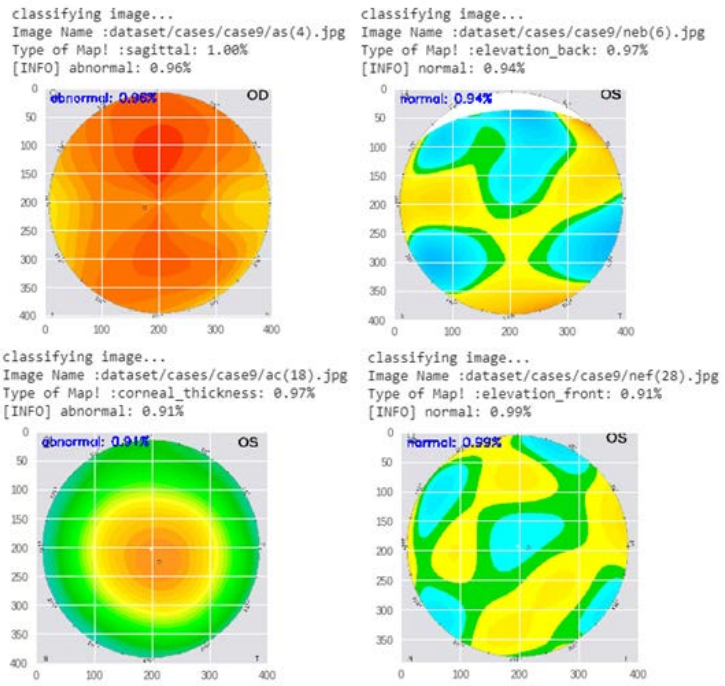


Figure3.10. Correctly classifying image and kind of map using VGG-16.

4. RESULTS AND DISCUSSION

4.1 Introduction

During this chapter, ten cases and their results that have been taken randomly from the Pentacam will be discussed and illustrated. Each case will be discussed separately within four stages; the first stage represents the input data that collected from Pentacam beside clinical decision for each map if it is normal or abnormal. The second stage describes the proposed algorithm decision VGG-16 network for each map which include kind of this map and if it is normal or abnormal. After that in stage three, it summarized the final result for each case. Finally, perform a matching between clinical decision and proposed algorithm decision.

4.2 The Work in Case Number One

4.2.1 Input data with a clinical decision

The input data to the first case and clinical decision are explained as following Figure 4.1 shows the four maps for this case:

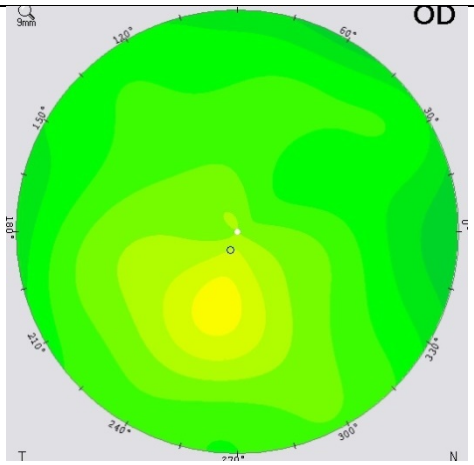
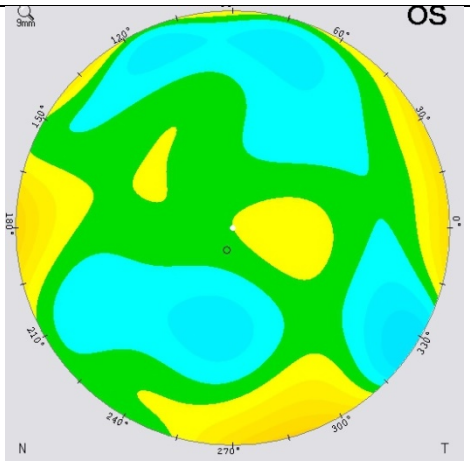
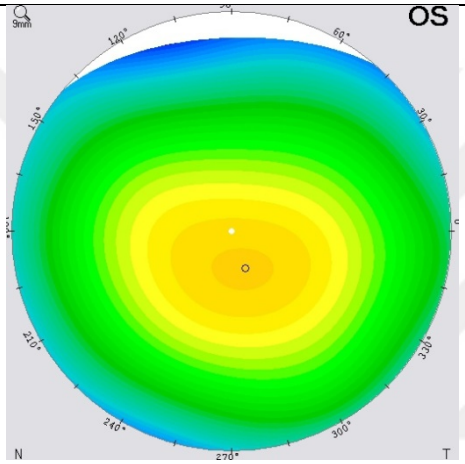
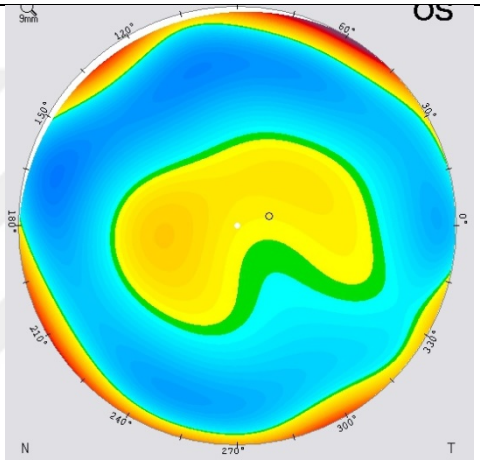
	
PENTACAM Reading: Sagittal map Clinical decision: Normal	PENTACAM Reading: Elevation front Clinical decision: Normal
	
PENTACAM Reading: Corneal thickness Clinical decision: Normal	PENTACAM Reading: Elevation back Clinical decision: Normal

Figure 4.1. Four maps for case number one with Pentacam reading and clinical decision.

4.2.2 Output data with proposed algorithm decision

4.2.2.1 Sagittal map

The output of the case number one is shown in Figure 4.2 which explain the type of map: sagittal map, and proposed algorithm decision: Normal.

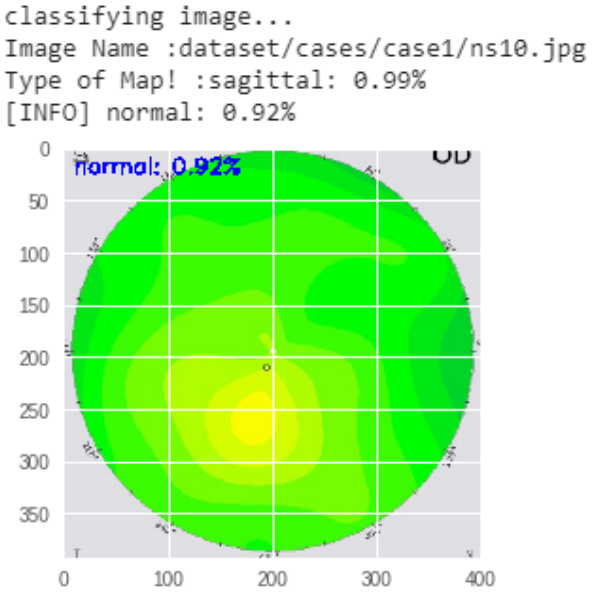


Figure 4.2. Sagittal map for case number one with the proposed algorithm decision.

4.2.2.2 Elevation front map

The output of the case number one is shown in Figure 4.3 which explain the type of map: elevation front map, and the proposed algorithm decision: Normal.

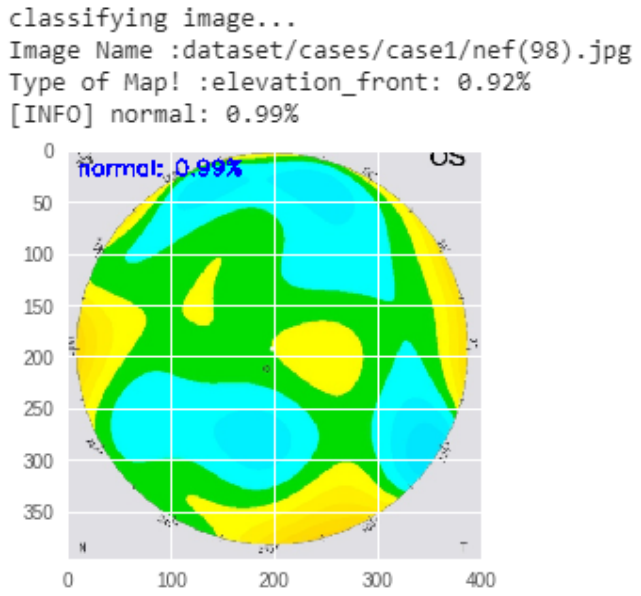


Figure 4.3. Elevation front map for case number one with the proposed algorithm decision.

4.2.2.3 Elevation back map

The output of the case number one is shown in Figure 4.4 which explain the type of map: elevation back map, and proposed algorithm decision: Normal.

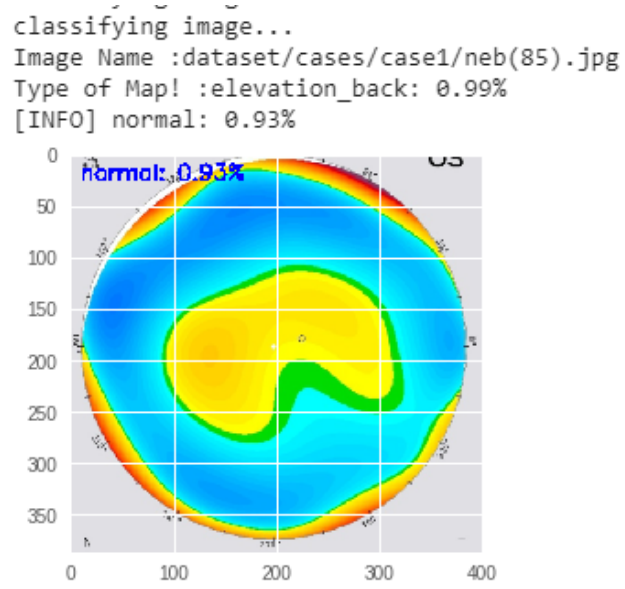


Figure 4.4. Elevation front map for case number one with the proposed algorithm decision.

4.2.2.4 Corneal thickness

The output of the case number one is shown in Figure 4.5 which explain the type of map: Corneal Thickness, and proposed algorithm decision: Normal.

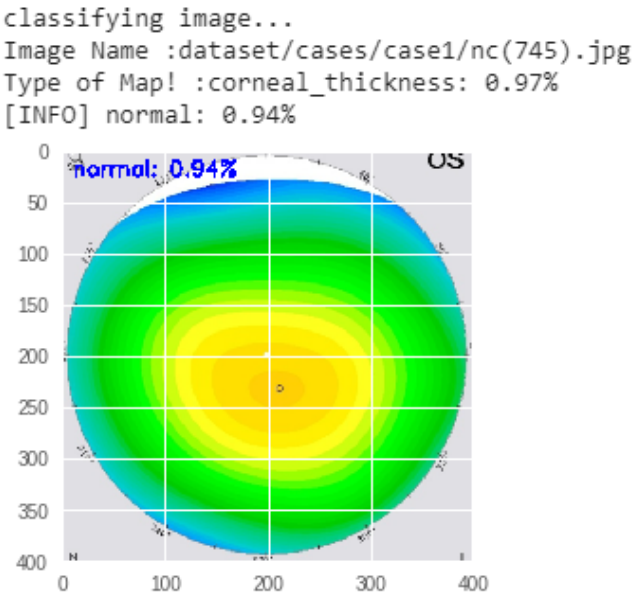


Figure 4.5. Corneal thickness for case number one with the proposed algorithm decision.

4.2.3 Final proposed algorithm result

Final proposed algorithm result for the case number one is Normal, hence: Sagittal map: normal, elevation front: normal, elevation back: normal, corneal thickness: normal.

4.2.4 Comparison between clinical decision and proposed algorithm decision

The topographic image of four refractive maps has correctly been determining the kind of map and classified as “normal” and “abnormal” by classification deep learning script.

4.3 The Work in Case Number Two

4.3.1 Input data with a clinical decision

The input data to the case number two and clinical decision are explained as following Figure 4.6: shows the four maps for this case:

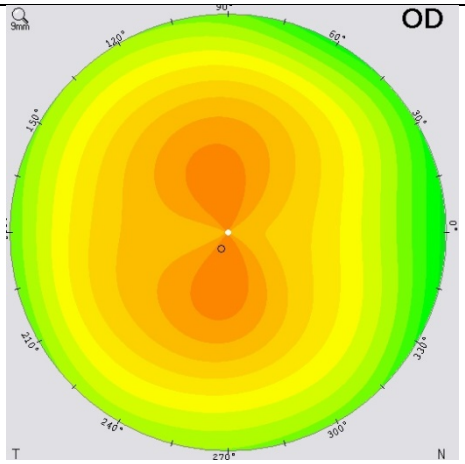
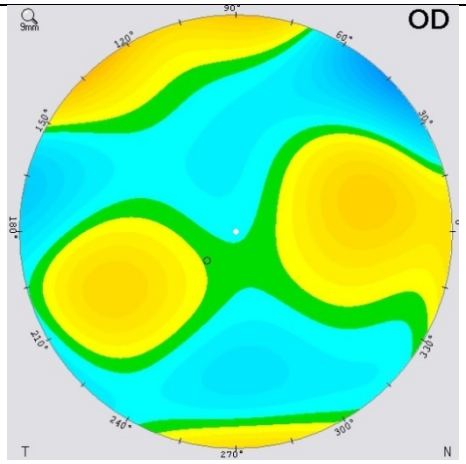
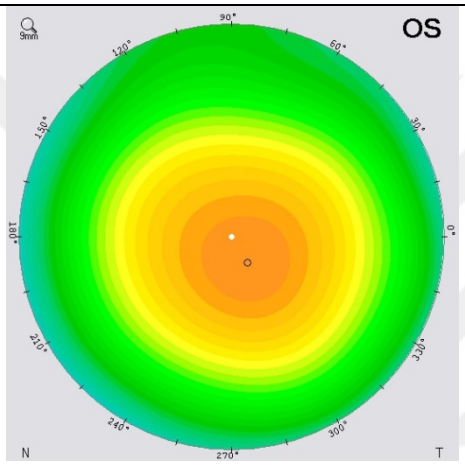
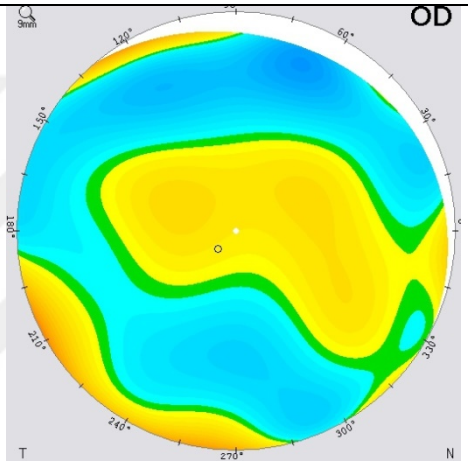
	
PENTACAM Reading: Sagittal map Clinical decision: Normal	PENTACAM Reading: Elevation front Clinical decision: Normal
	
PENTACAM Reading: Corneal thickness Clinical decision: Abnormal	PENTACAM Reading: Elevation back Clinical decision: Normal

Figure 4.6. Four maps for case number two with Pentacam reading and clinical decision.

4.3.2 Output data with proposed algorithm decision

4.3.2.1 Sagittal map

The output of the case number two is shown in Figure 4.7 which explain the type of map: sagittal map, and proposed algorithm decision: Normal.

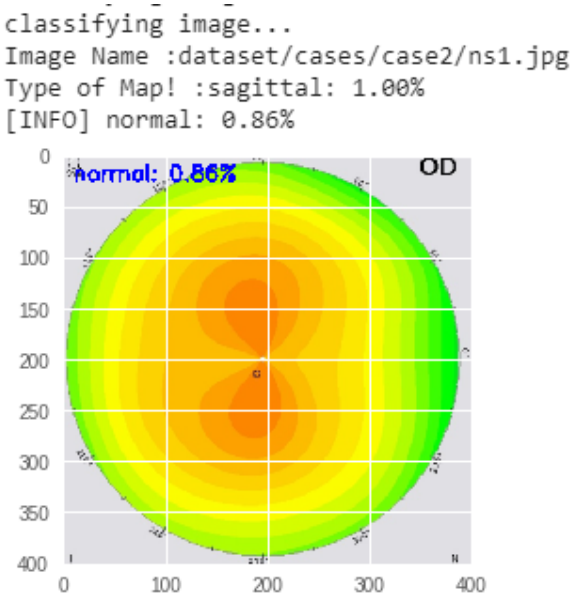


Figure 4.7. Sagittal map for case number two with proposed algorithm decision.

4.3.2.2 Elevation front map

The output of the case number two is shown in Figure 4.8 which explain the type of map: elevation front map, and proposed algorithm decision: Normal.

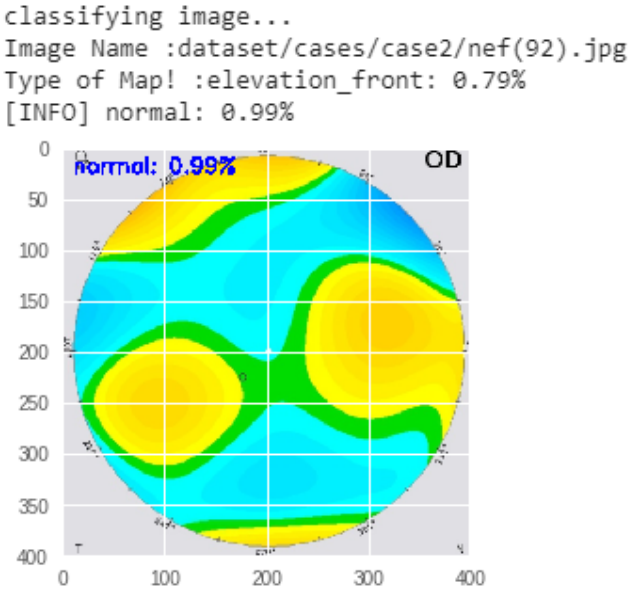


Figure 4.8. Elevation front map for case number two with proposed algorithm decision.

4.3.2.3 Elevation back map

The output of the case number two is shown in Figure 4.9 which explain the type of map: elevation back map, and proposed algorithm decision: Normal.

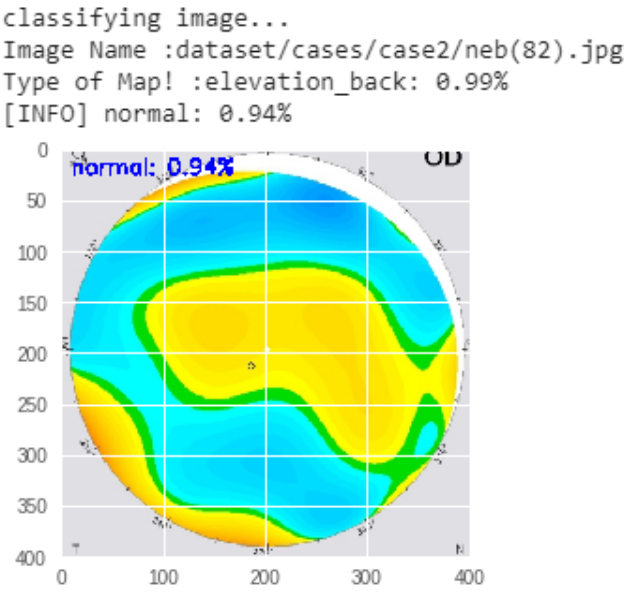


Figure 4.9. Elevation front map for case number two with proposed algorithm decision.

4.3.2.4 Corneal thickness

The output of the case number two is shown in Figure 4.10 which explain the type of map: Corneal Thickness, and proposed algorithm decision: Abnormal.

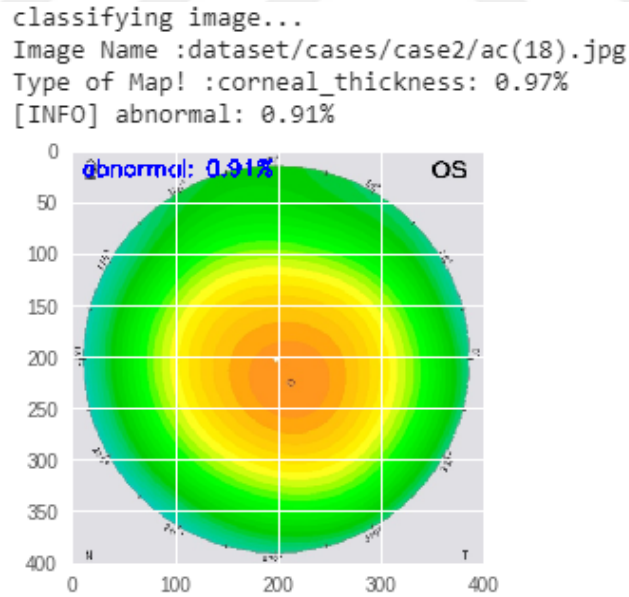


Figure 4.10. Corneal thickness for case number two with proposed algorithm decision.

4.3.3 Final proposed algorithm result

Final proposed algorithm result for the case number two is Normal, hence: Sagittal map: normal, elevation front: normal, elevation back: normal, corneal thickness: abnormal.

4.3.4 Comparison between clinical decision and proposed algorithm decision

The topographic image of four refractive maps has correctly been determining the kind of map and classified as “normal” and “abnormal” by classification deep learning script.

4.4 The Work in Case Number Three

4.4.1 Input data with a clinical decision

The input data to the case number three and clinical decision are explained as following Figure 4.11 shows the four maps for this case:

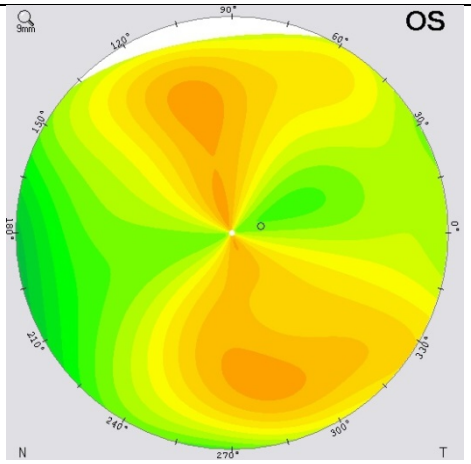
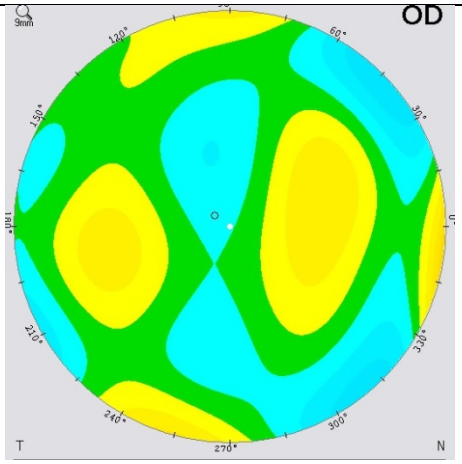
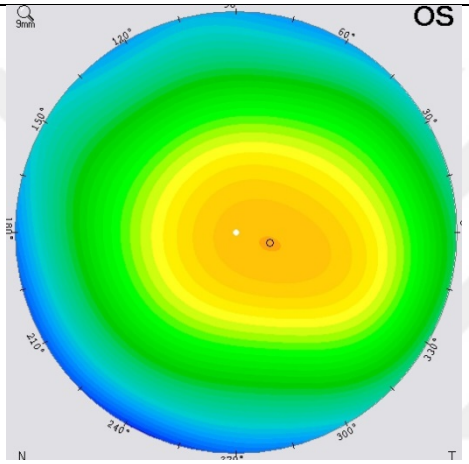
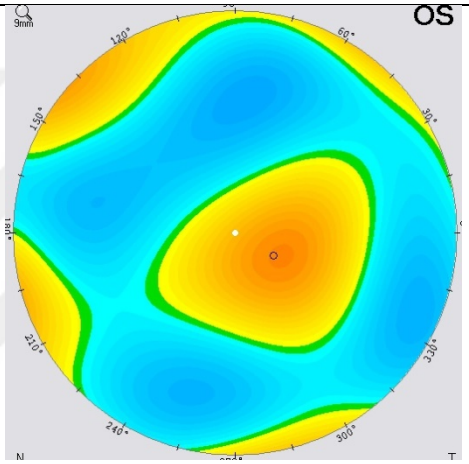
	
PENTACAM Reading: Sagittal map Clinical decision: Normal	PENTACAM Reading: Elevation front Clinical decision: Normal
	
PENTACAM Reading: Corneal thickness Clinical decision: Abnormal	PENTACAM Reading: Elevation back Clinical decision: Abnormal

Figure 4.11. Four maps for case number three with Pentacam reading and clinical decision.

4.4.2 Output data with proposed algorithm decision

4.4.2.1 Sagittal map

The output of the case number three is shown in Figure 4.12 which explain the type of map: sagittal map, and proposed algorithm decision: Normal.

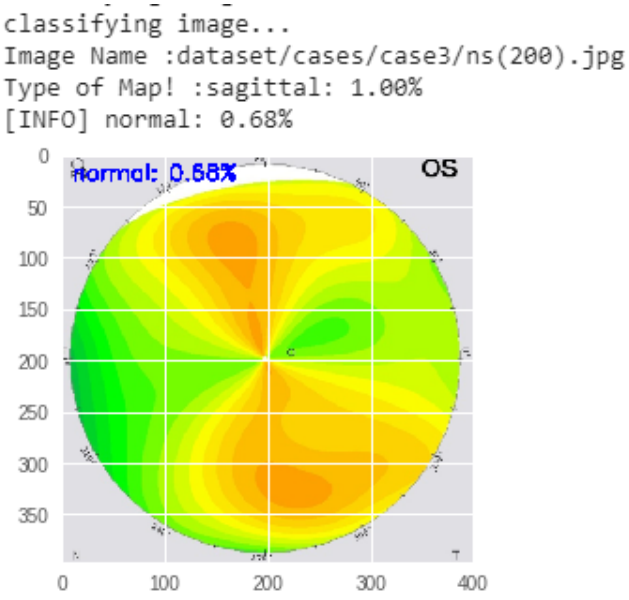


Figure 4.12. Sagittal map for case number three with proposed algorithm decision.

4.4.2.2 Elevation front map

The output of the case number three is shown in Figure 4.13 which explain the type of map: elevation front map, and proposed algorithm decision: Normal.

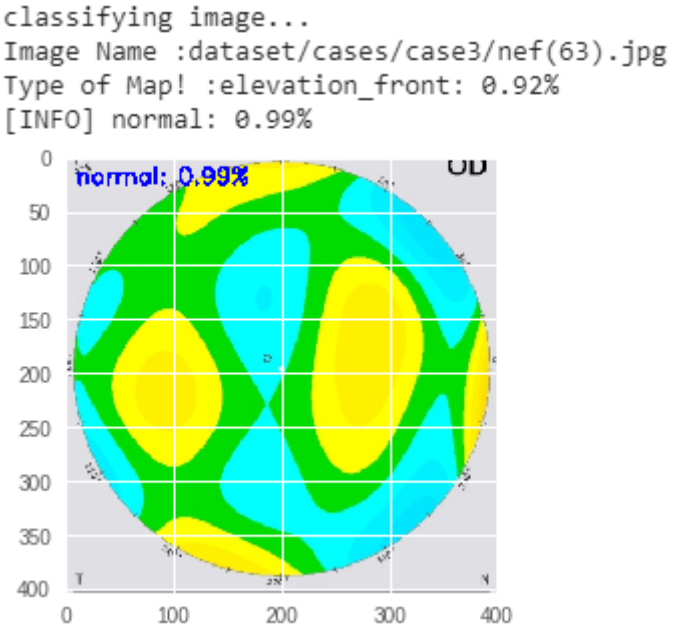


Figure 4.13. Elevation front map for case number three with proposed algorithm decision.

4.4.2.3 Elevation back map

The output of the case number three is shown in Figure 4.14 which explain the type of map: elevation back map, and the proposed algorithm decision: Abnormal.

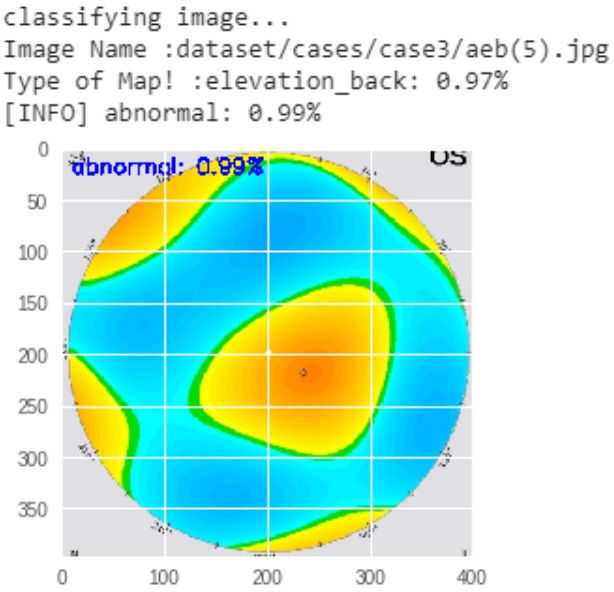


Figure 4.14. Elevation back map for case number three with proposed algorithm decision.

4.4.2.4 Corneal thickness

The output of the case number three is shown in Figure 4.15 which explain the type of map: Corneal Thickness, and proposed algorithm decision: Normal.

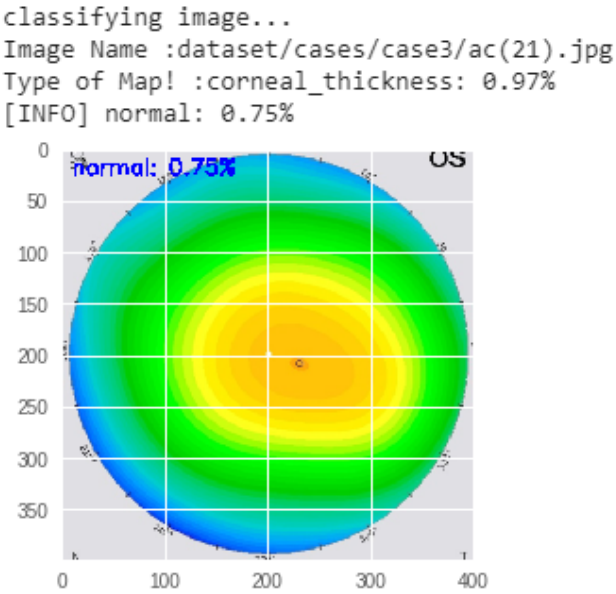


Figure 4.15. Corneal thickness for case number three with proposed algorithm decision.

4.4.3 Final proposed algorithm result

Final proposed algorithm result for the case number three is Normal, hence: Sagittal map: normal, elevation front: normal, elevation back: abnormal, corneal thickness: normal.

4.4.4 Comparison between clinical decision and proposed algorithm decision

The topographic image of four refractive maps has correctly been determining the kind of map and classified as “normal” and “abnormal” by classification deep learning script except corneal thickness map: normal but in the clinical decision it is abnormal.

4.5 The Work in Case Number Four

4.5.1 Input data with a clinical decision

The input data to the case number four and clinical decision are explained as following Figure 4.16 shows the four maps for this case:

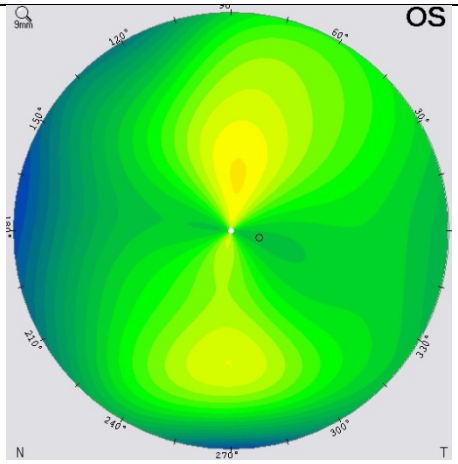
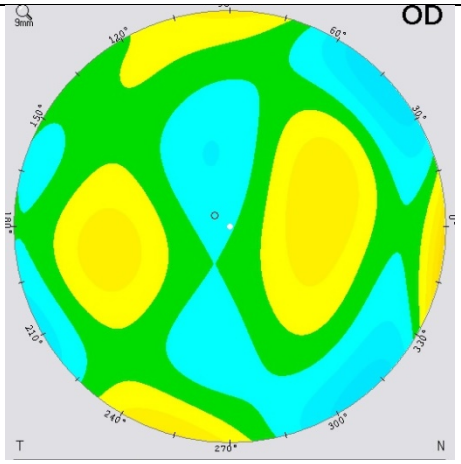
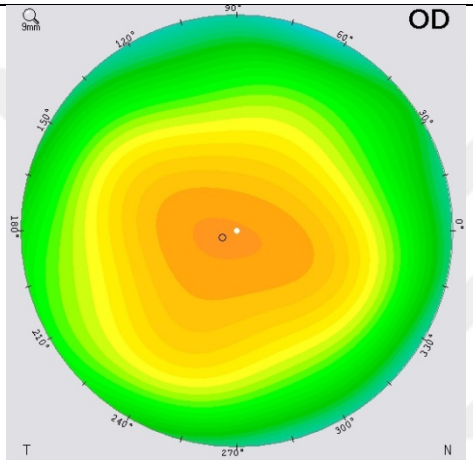
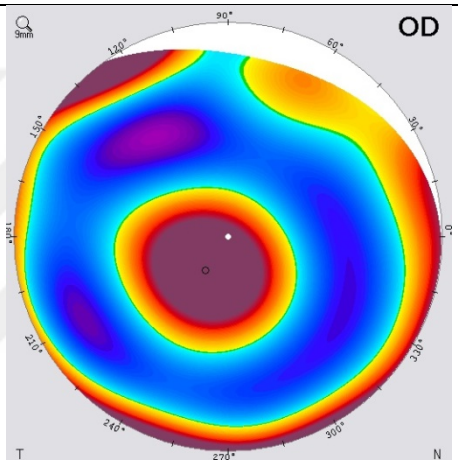
	
PENTACAM Reading: Sagittal map Clinical decision: Normal	PENTACAM Reading: Elevation front Clinical decision: Abnormal
	
PENTACAM Reading: Corneal thickness Clinical decision: Abnormal	PENTACAM Reading: Elevation back Clinical decision: Abnormal

Figure 4.16. Four maps for case number four with Pentacam reading and clinical decision.

4.5.2 Output data with proposed algorithm decision

4.5.2.1 Sagittal map

The output of the case number four is shown in Figure 4.17 which explain the type of map: sagittal map, and proposed algorithm decision: Normal.

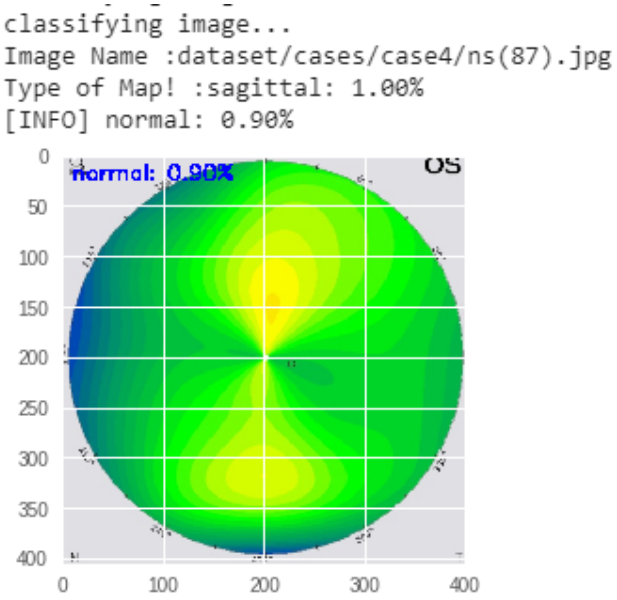


Figure 4.17. Sagittal map for case number four with proposed algorithm decision.

4.5.2.2 Elevation front map

The output of the case number four is shown in Figure 4.18 which explain the type of map: elevation front map, and proposed algorithm decision: Abnormal.

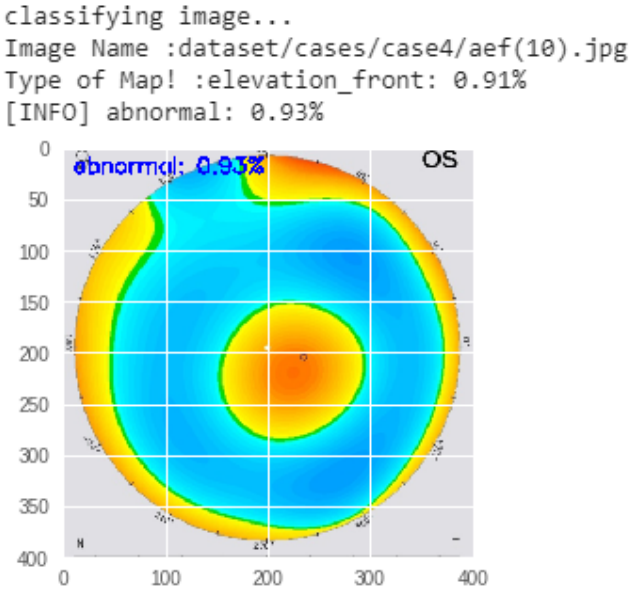


Figure 4.18. Elevation front map for case number four with proposed algorithm decision.

4.5.2.3 Elevation back map

The output of the case number four is shown in Figure 4.19 which explain the type of map: elevation back map, and the proposed algorithm decision: Abnormal.

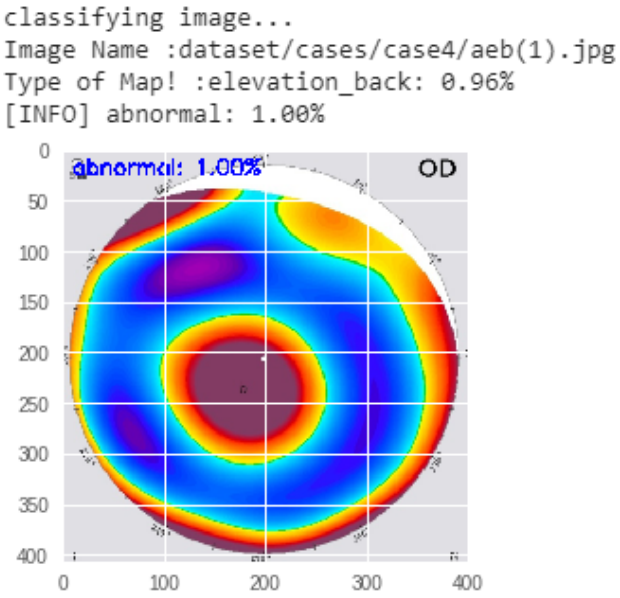


Figure 4.19. Elevation back map for case number four with proposed algorithm decision.

4.5.2.4 Corneal thickness

The output of the case number four is shown in Figure 4.20 which explain the type of map: Corneal Thickness, and proposed algorithm decision: Abnormal.

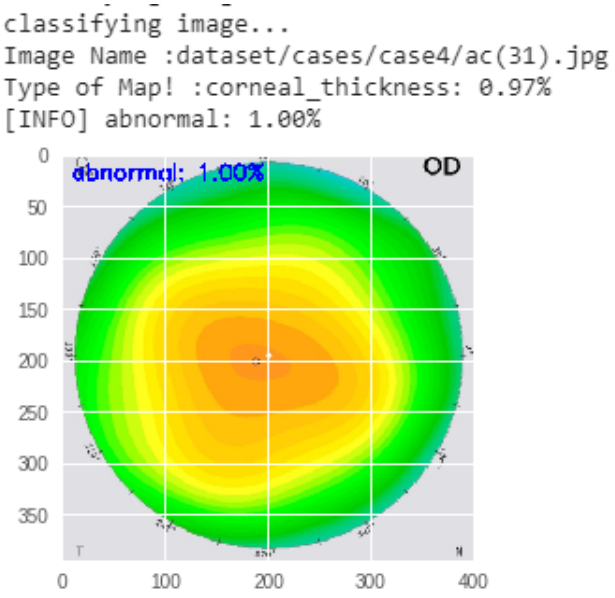


Figure 4.20. Corneal thickness for case number four with proposed algorithm decision.

4.5.3 Final proposed algorithm result

Final proposed algorithm result for the case number four is Abnormal, hence: Sagittal map: normal, elevation front: abnormal, elevation back: abnormal, corneal thickness: abnormal.

4.5.4 Comparison between clinical decision and proposed algorithm decision

The topographic image of four refractive maps has correctly been determining the kind of map and classified as “normal” and “abnormal” by classification deep learning script.

4.6 The Work in Case Number Five

4.6.1 Input data with a clinical decision

The input data to the case number five and clinical decision are explained as following Figure 4.21 shows the four maps for this case:

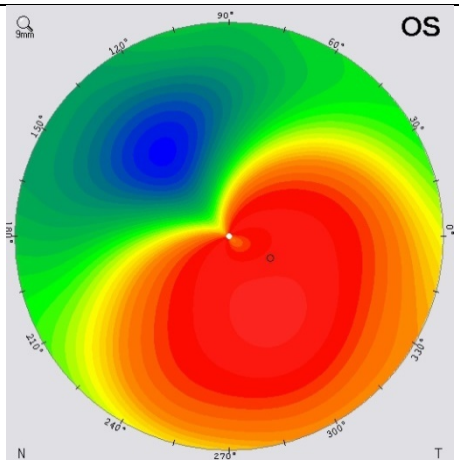
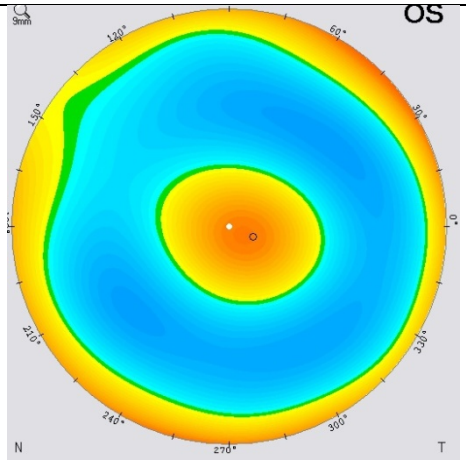
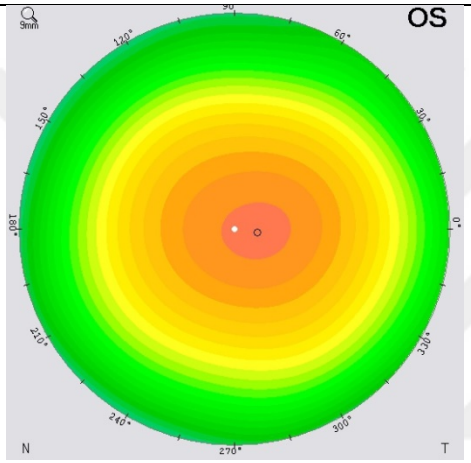
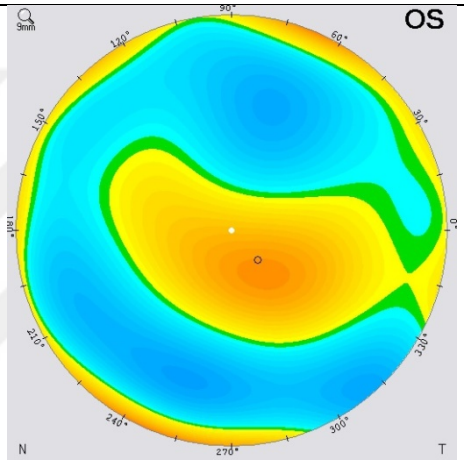
	
PENTACAM Reading: Sagittal map Clinical decision: Abnormal	PENTACAM Reading: Elevation front Clinical decision: Abnormal
	
PENTACAM Reading: Corneal thickness Clinical decision: Abnormal	PENTACAM Reading: Elevation back Clinical decision: Abnormal

Figure 4.21. Four maps for case number five with Pentacam reading and clinical decision.

4.6.2 Output data with proposed algorithm decision.

4.6.2.1 Sagittal map

The output of the case number five is shown in Figure 4.22 which explain the type of map: sagittal map, and proposed algorithm decision: Abnormal.

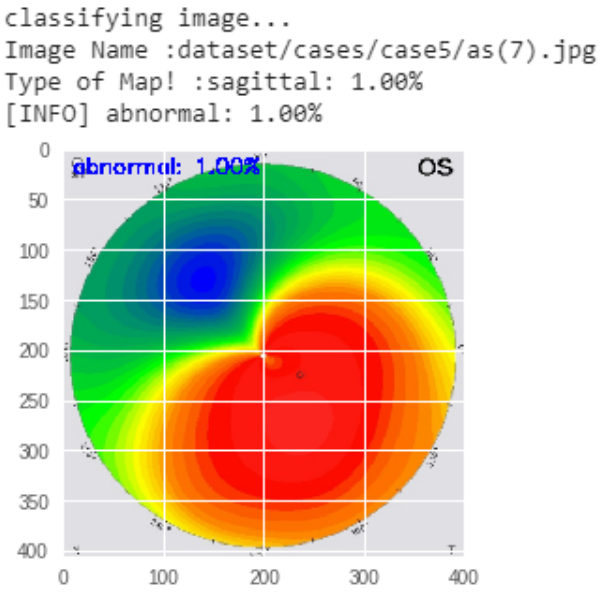


Figure 4.22. Sagittal map for case number five with proposed algorithm decision.

4.6.2.2 Elevation front map

The output of the case number five is shown in Figure 4.23 which explain the type of map: elevation front map, and the proposed algorithm decision: Abnormal.

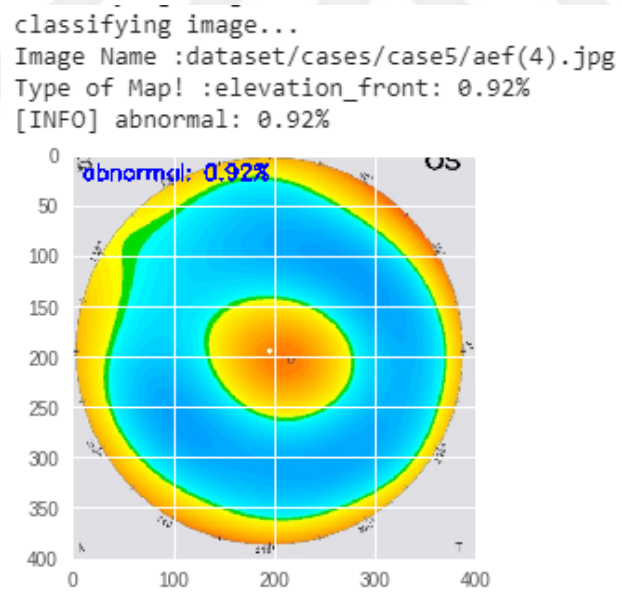


Figure 4.23. Elevation front map for case number five with proposed algorithm decision.

4.6.2.3 Elevation back map

The output of the case number five is shown in Figure 4.24 which explain the type of map: elevation back map, and proposed algorithm decision: Normal.

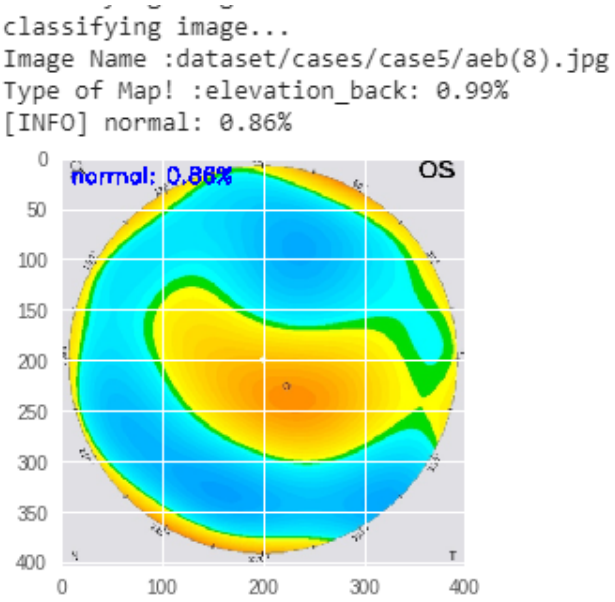


Figure 4.24. Elevation back map for case number five with proposed algorithm decision.

4.6.2.4 Corneal thickness

The output of the case number five is shown in Figure 4.25 which explain the type of map: Corneal Thickness, and proposed algorithm decision: Abnormal.

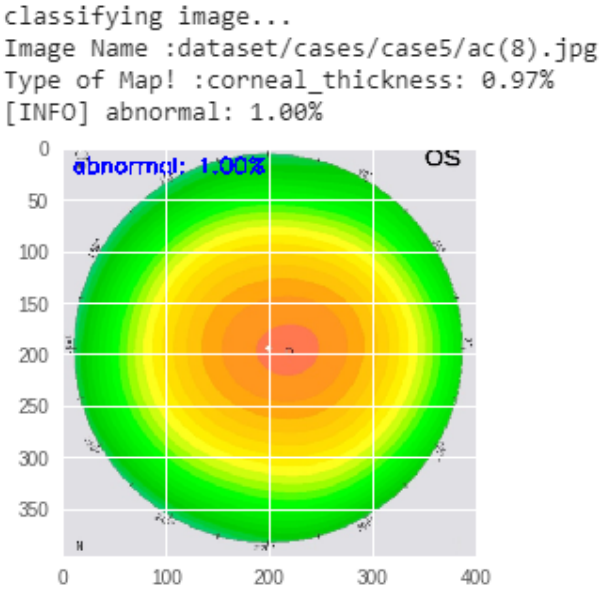


Figure 4.25. Corneal thickness for case number five with proposed algorithm decision.

4.6.3 Final proposed algorithm result

Final proposed algorithm result for the case number five is Abnormal, hence: Sagittal map: abnormal, elevation front: abnormal, elevation back: normal, corneal thickness: abnormal.

4.6.4 Comparison between clinical decision and proposed algorithm decision

The topographic image of four refractive maps has correctly been determining the kind of map and classified as “normal” and “abnormal” by classification deep learning script except elevation back map: normal but in the clinical decision it is Abnormal.

4.7 The Work in Case Number Six

4.7.1 Input data with a clinical decision

The input data to case number six and clinical decision are explained as following Figure 4.26 shows the four maps for this case:

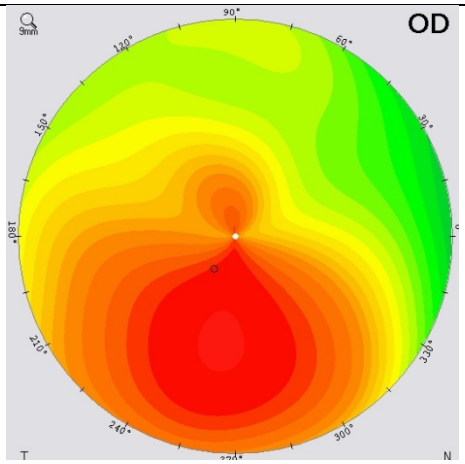
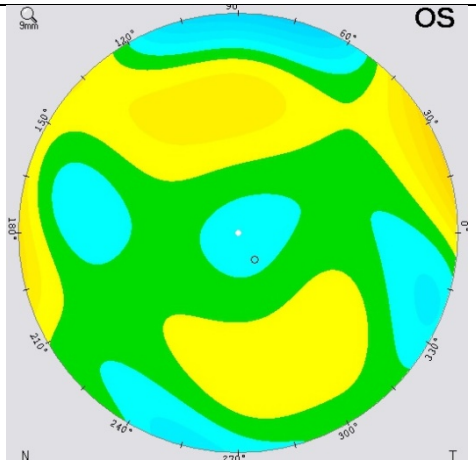
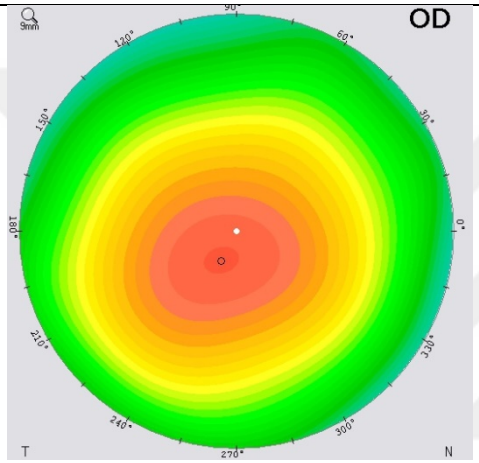
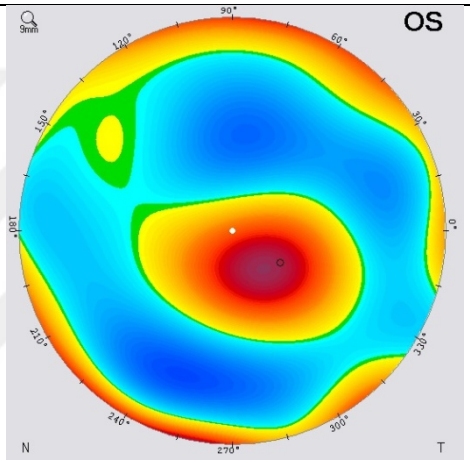
	
PENTACAM Reading: Sagittal map Clinical decision: Abnormal	PENTACAM Reading: Elevation front Clinical decision: Normal
	
PENTACAM Reading: Corneal thickness Clinical decision: Abnormal	PENTACAM Reading: Elevation back Clinical decision: Abnormal

Figure 4.26. Four maps for case number six with Pentacam reading and clinical decision.

4.7.2 Output data with proposed algorithm decision

4.7.2.1 Sagittal map

The output of the case number six is shown in Figure 4.27 which explain the type of map: sagittal map, and proposed algorithm decision: Abnormal.

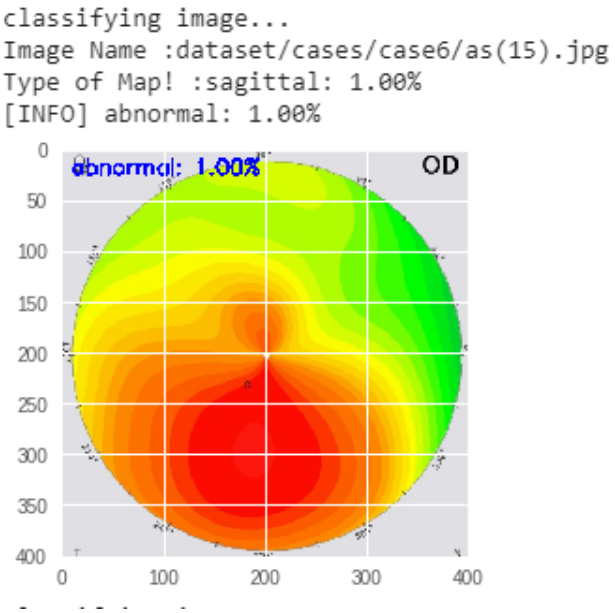


Figure 4.27. Sagittal map for case number six with proposed algorithm decision.

4.7.2.2 Elevation front map

The output of the case number six is shown in Figure 4.28 which explain the type of map: elevation front map, and proposed algorithm decision: Normal.

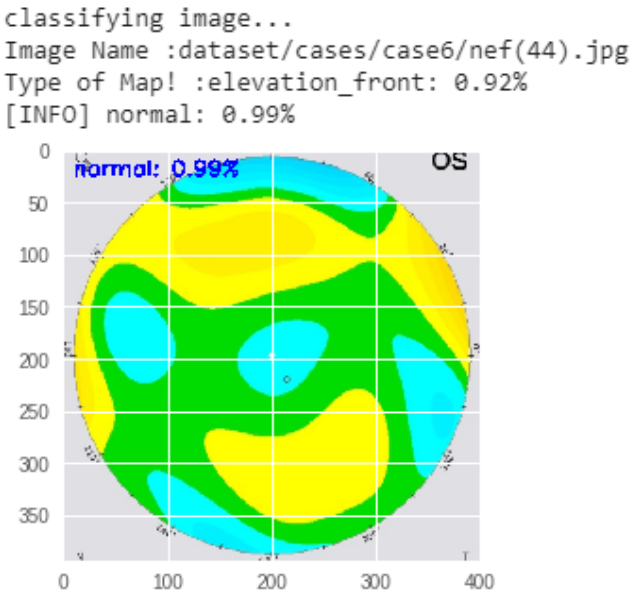


Figure 4.28. Elevation front map for case number six with proposed algorithm decision.

4.7.2.3 Elevation back map

The output of the case number six is shown in Figure 4.29 which explain the type of map: elevation back map, and the proposed algorithm decision: Abnormal.

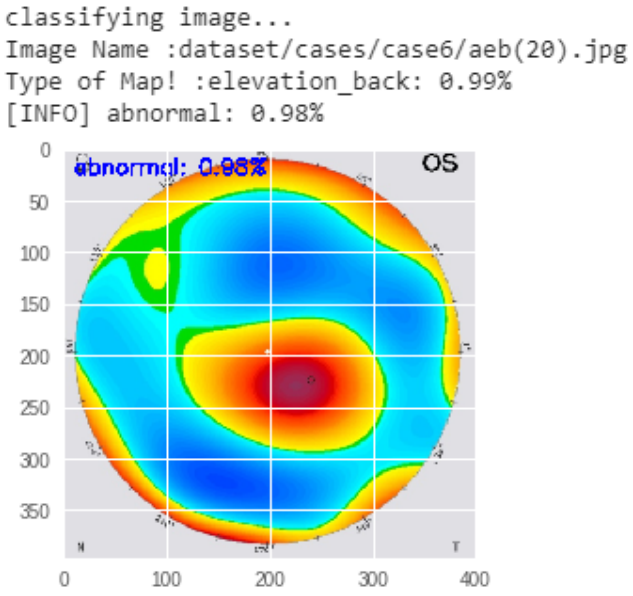


Figure 4.29. Elevation back map for case number six with proposed algorithm decision.

4.7.2.4 Corneal thickness

The output of the case number six is shown in Figure 4.30 which explain the type of map: Corneal Thickness, and proposed algorithm decision: Abnormal.

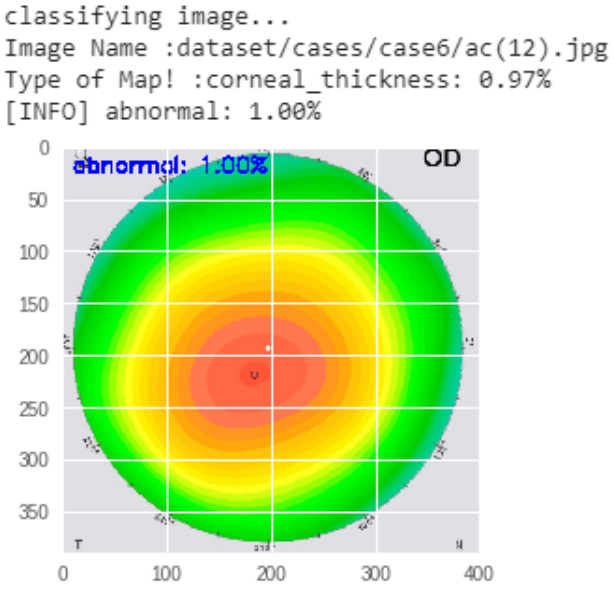


Figure 4.30. Corneal thickness for case number six with proposed algorithm decision.

4.7.3 Final proposed algorithm result

Final proposed algorithm result for the case number six is Abnormal, hence: Sagittal map: abnormal, elevation front: normal, elevation back: abnormal, corneal thickness: abnormal.

4.7.4 Comparison between clinical decision and proposed algorithm decision

The topographic image of four refractive maps has correctly been determining the kind of map and classified as “normal” and “abnormal” by classification deep learning script.

4.8 The Work in Case Number Seven

4.8.1 Input data with a clinical decision

The input data to the case number seven and clinical decision are explained as following Figure 4.31 shows the four maps for this case:

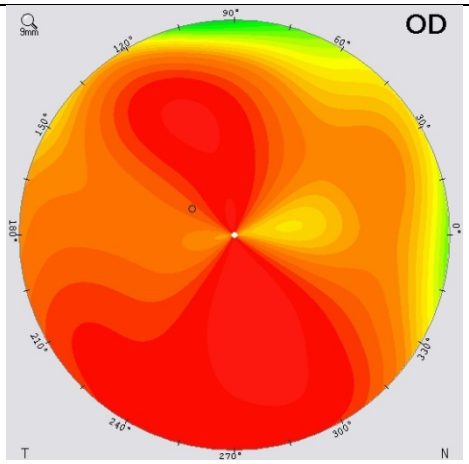
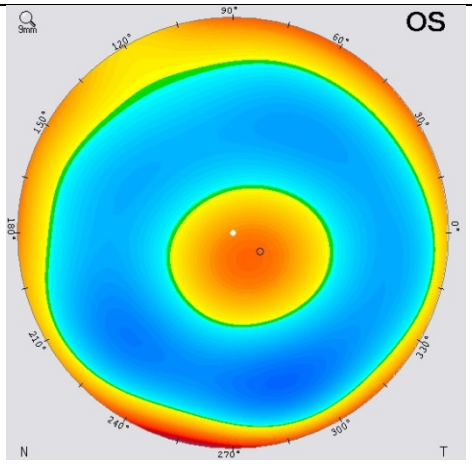
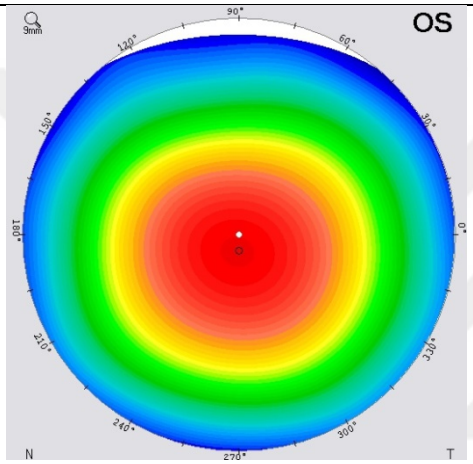
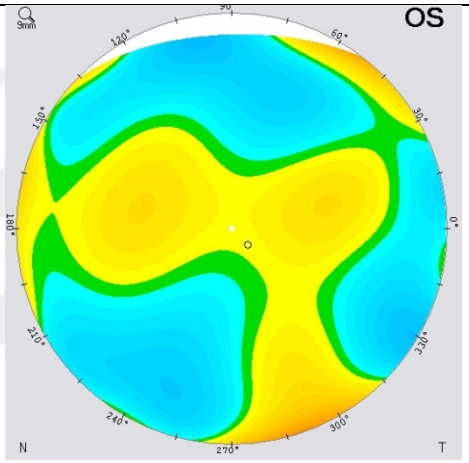
	
PENTACAM Reading: Sagittal map Clinical decision: Abnormal	PENTACAM Reading: Elevation front Clinical decision: Abnormal
	
PENTACAM Reading: Corneal thickness Clinical decision: Abnormal	PENTACAM Reading: Elevation back Clinical decision: Normal

Figure 4.31. Four maps for case number seven with Pentacam reading and clinical decision.

4.8.2 Output data with proposed algorithm decision

4.8.2.1 Sagittal map

The output of the case number seven is shown in Figure 4.32 which explain the type of map: sagittal map, and proposed algorithm decision: Abnormal.

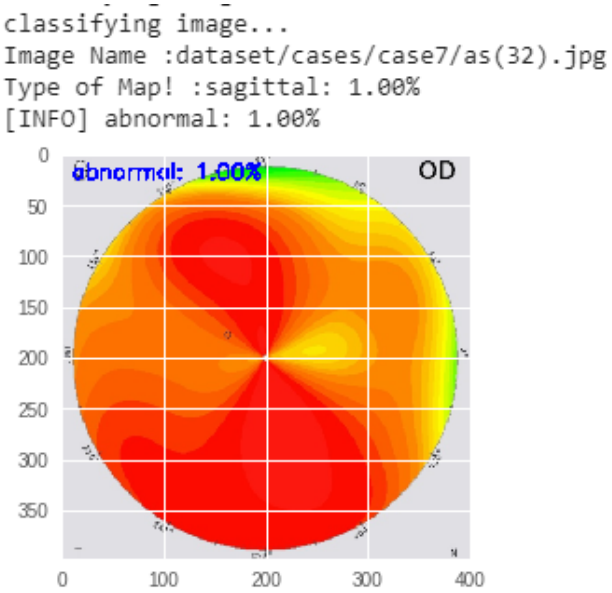


Figure 4.32. Sagittal map for case number seven with proposed algorithm decision.

4.8.2.2 Elevation front map

The output of the case number seven is shown in Figure 4.33 which explain the type of map: elevation front map, and the proposed algorithm decision: Abnormal.

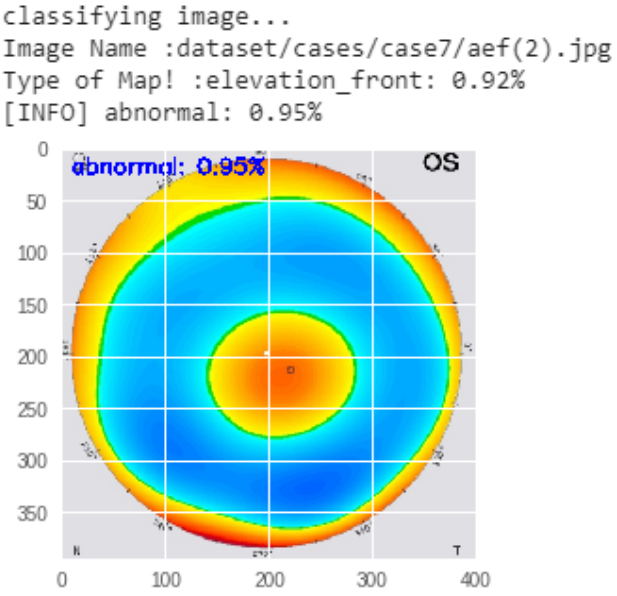


Figure 4.33. Elevation front map for case number seven with proposed algorithm decision.

4.8.2.3 Elevation back map

The output of the case number seven is shown in Figure 4.34 which explain the type of map: elevation back map, and proposed algorithm decision: Normal.

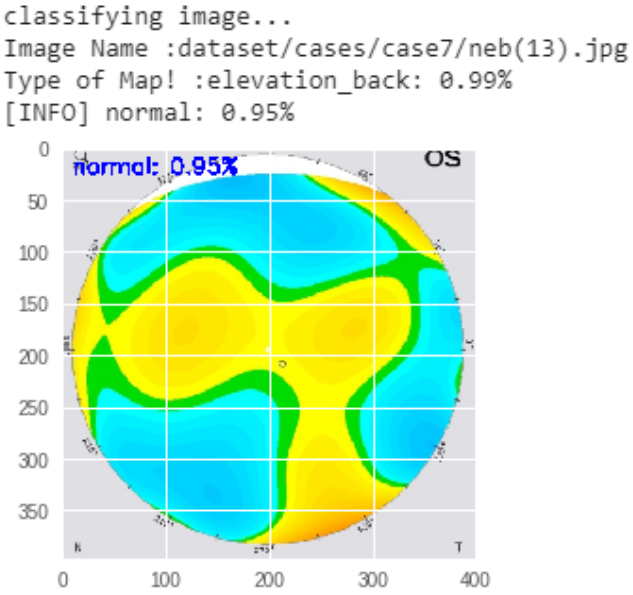


Figure 4.34. Elevation back map for case number seven with proposed algorithm decision.

4.8.2.4 Corneal thickness

The output of the case number seven is shown in Figure 4.35 which explain the type of map: Corneal Thickness, and proposed algorithm decision: Abnormal.

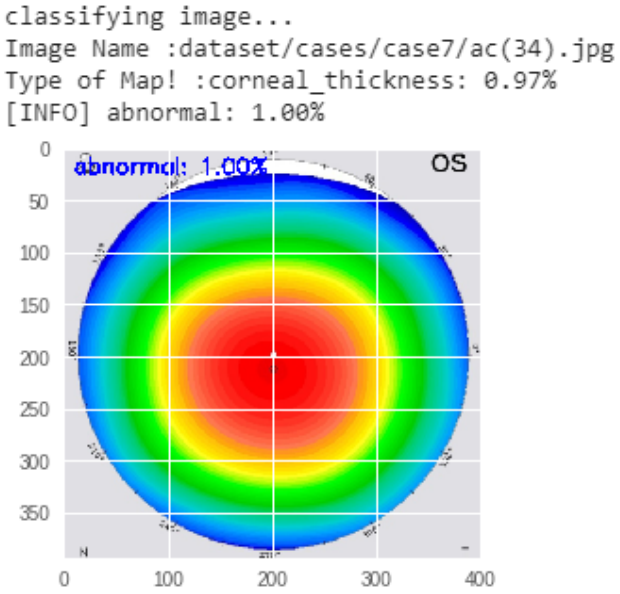


Figure 4.35. Corneal thickness for case number seven with proposed algorithm decision

4.8.3 Final proposed algorithm result

Final proposed algorithm result for the case number seven is Abnormal, hence: Sagittal map: abnormal, elevation front: abnormal, elevation back: normal, corneal thickness: abnormal.

4.8.4 Comparison between clinical decision and proposed algorithm decision

The topographic image of four refractive maps has correctly been determining the kind of map and classified as “normal” and “abnormal” by classification deep learning script.

4.9 The Work in Case Number Eight

4.9.1 Input data with a clinical decision

The input data to the case number eight and clinical decision are explained as following Figure 4.36 shows the four maps for this case:

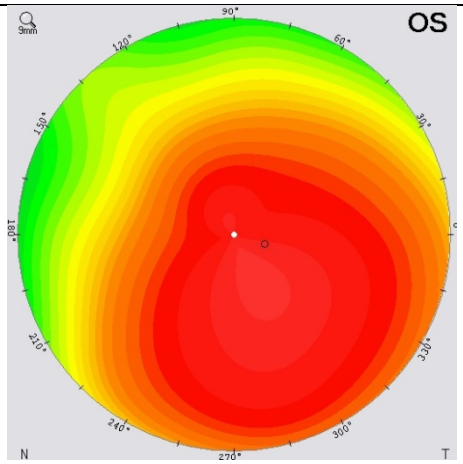
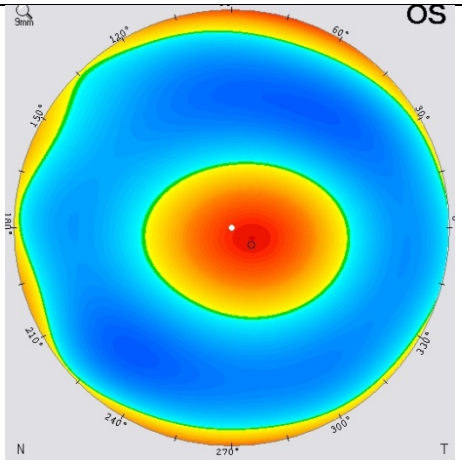
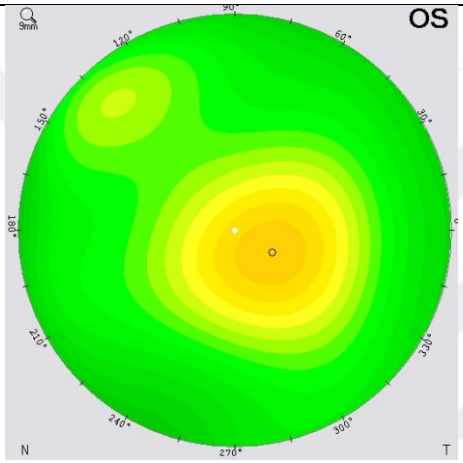
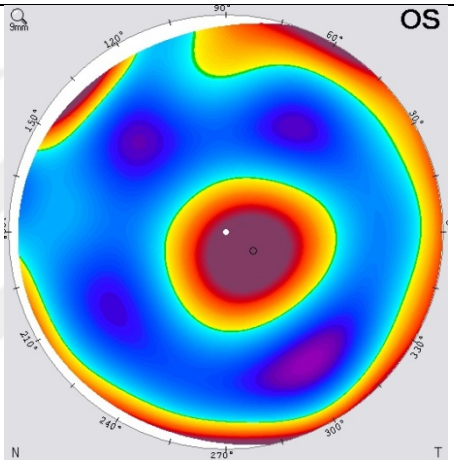
	
PENTACAM Reading: Sagittal map Clinical decision: Abnormal	PENTACAM Reading: Elevation front Clinical decision: Abnormal
	
PENTACAM Reading: Corneal thickness Clinical decision: Normal	PENTACAM Reading: Elevation back Clinical decision: Abnormal

Figure 4.36. Four maps for case number eight with Pentacam reading and clinical decision.

4.9.2 Output data with proposed algorithm decision

4.9.2.1 Sagittal map

The output of the case number eight is shown in Figure 4.37 which explain the type of map: sagittal map, and proposed algorithm decision: Abnormal.

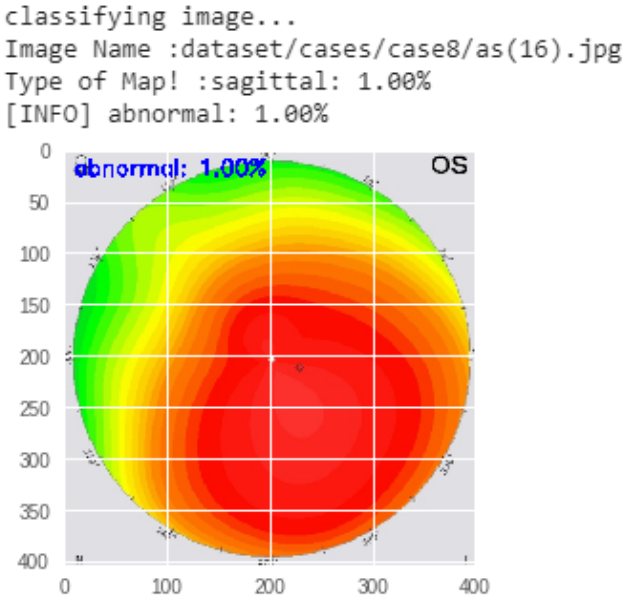


Figure 4.37. Sagittal map for case number eight with proposed algorithm decision.

4.9.2.2 Elevation front map

The output of the case number eight is shown in Figure 4.38 which explain the type of map: elevation front map, and the proposed algorithm decision: Abnormal.

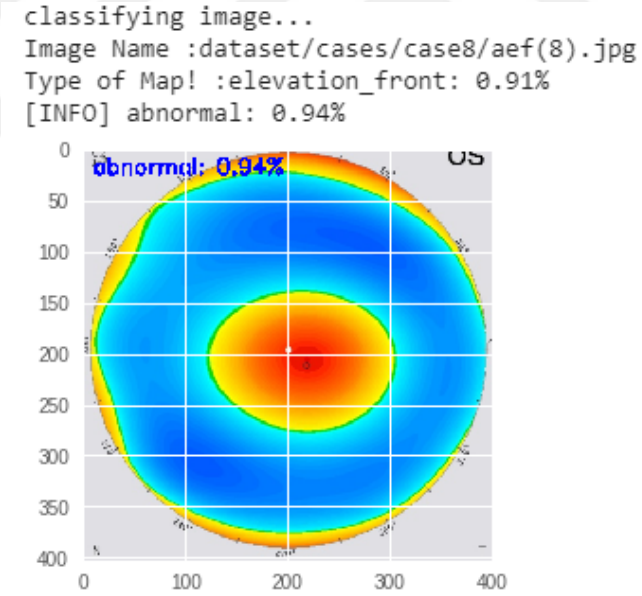


Figure 4.38. Elevation front map for case number eight with proposed algorithm decision.

4.9.2.3 Elevation back map

The output of the case number eight is shown in Figure 4.39 which explain the type of map: elevation back map, and proposed algorithm decision: Abnormal.

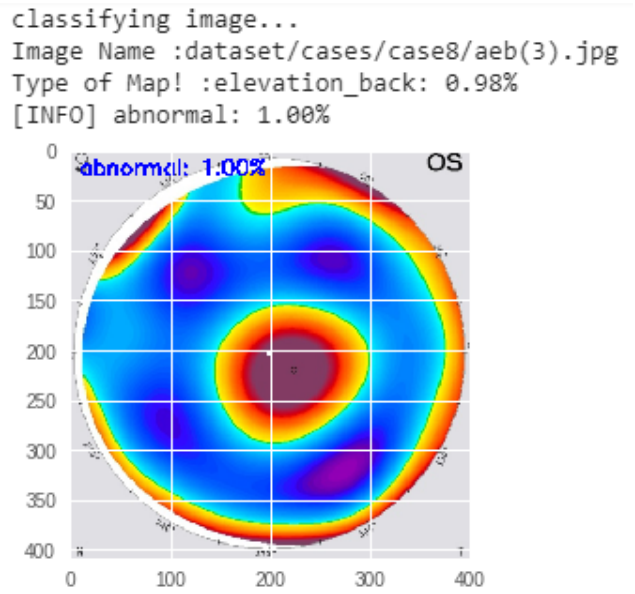


Figure 4.39. Elevation back map for case number eight with proposed algorithm decision.

4.9.2.4 Corneal thickness

The output of the case number eight is shown in Figure 4.40 which explain the type of map: Corneal Thickness, and proposed algorithm decision: Normal.

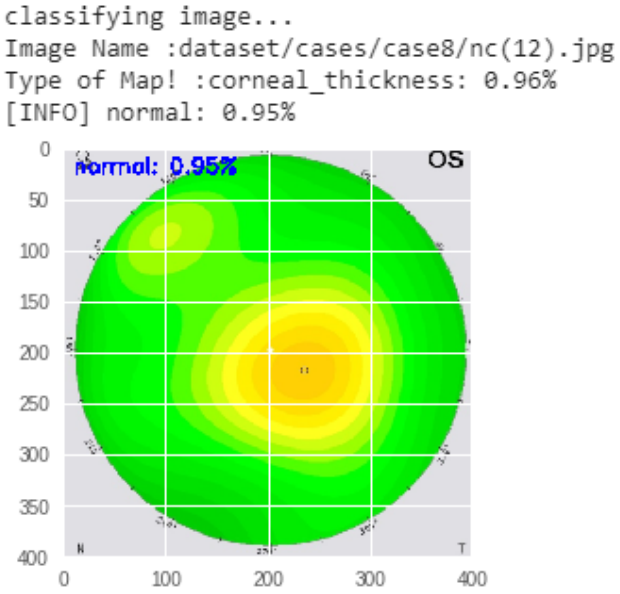


Figure 4.40. Corneal thickness for case number eight with proposed algorithm decision.

4.9.3 Final proposed algorithm result

Final proposed algorithm result for the case number eight is Abnormal, hence: Sagittal map: abnormal, elevation front: abnormal, elevation back: abnormal, corneal thickness: normal.

4.9.4 Comparison between clinical decision and proposed algorithm decision

The topographic image of four refractive maps has correctly been determining the kind of map and classified as “normal” and “abnormal” by classification deep learning script.

4.10 The Work in Case Number Nine

4.10.1 Input data with a clinical decision

The input data to the case number nine and clinical decision are explained as following Figure 4.41 shows the four maps for this case:

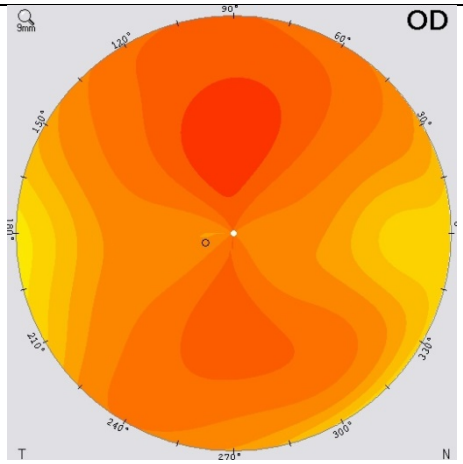
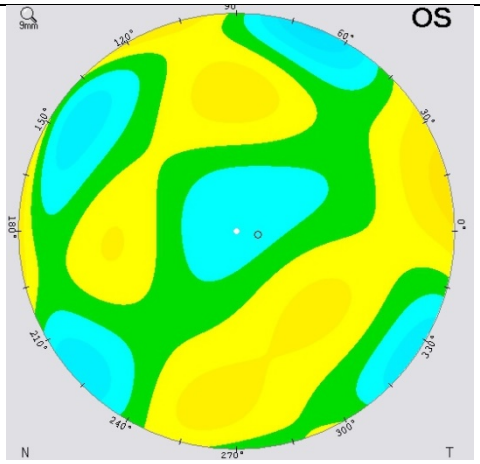
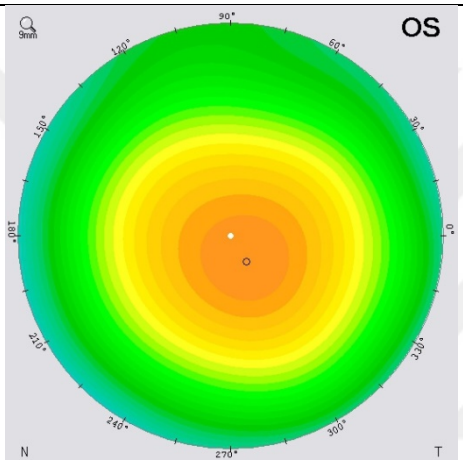
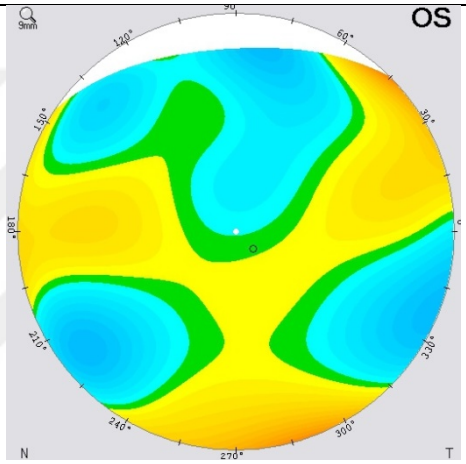
	
PENTACAM Reading: Sagittal map Clinical decision: Abnormal	PENTACAM Reading: Elevation front Clinical decision: Normal
	
PENTACAM Reading: Corneal thickness Clinical decision: Abnormal	PENTACAM Reading: Elevation back Clinical decision: Normal

Figure 4.41. Four maps for case number nine with Pentacam reading and clinical decision.

4.10.2 Output data with proposed algorithm decision

4.10.2.1 Sagittal map

The output of the case number nine is shown in Figure 4.42 which explain the type of map: sagittal map, and proposed algorithm decision: Abnormal.

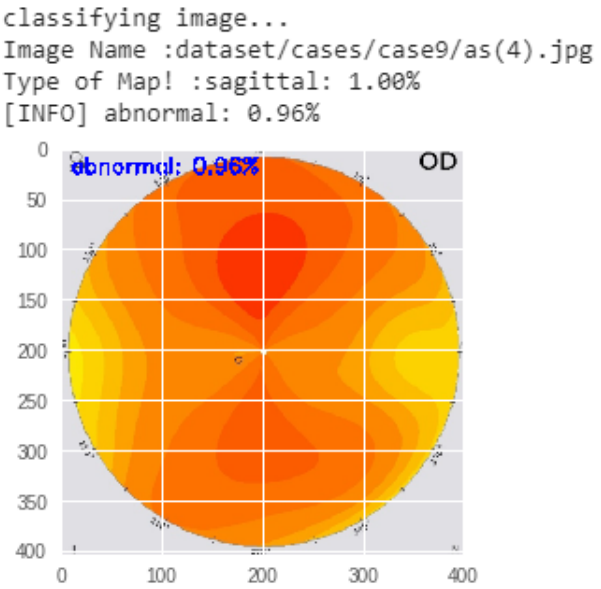


Figure 4.42. Sagittal map for case number nine with the proposed algorithm decision

4.10.2.2 Elevation front map

The output of the case number nine is shown in Figure 4.43 which explain the type of map: elevation front map, and proposed algorithm decision: Normal.

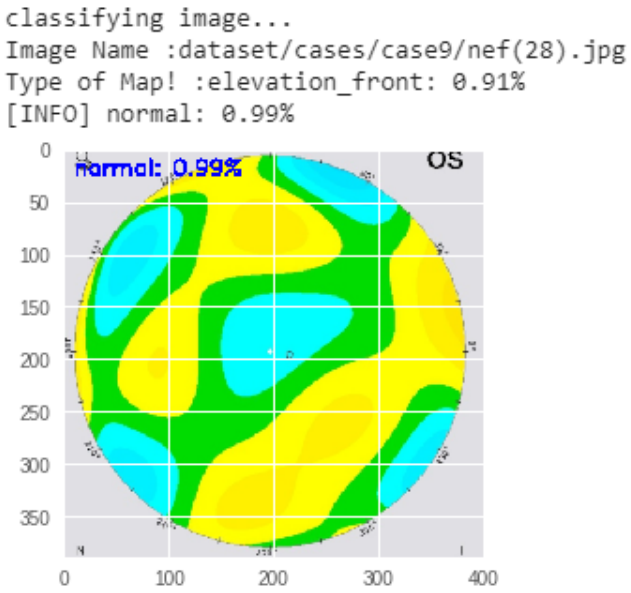


Figure 4.43. Elevation front map for case number nine with the proposed algorithm decision.

4.10.2.3 Elevation back map

The output of the case number nine is shown in Figure 4.44 which explain the type of map: elevation back map, and proposed algorithm decision: Normal.

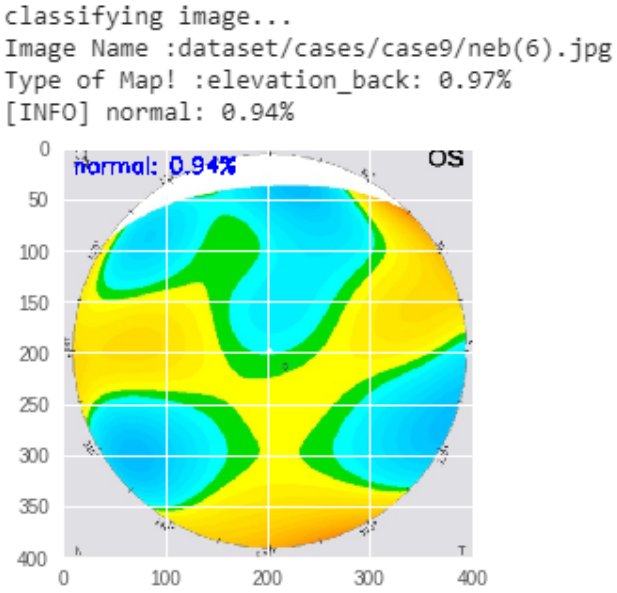


Figure 4.44. Elevation back map for case number nine with the proposed algorithm decision.

4.10.2.4 Corneal thickness

The output of the case number nine is shown in Figure 4.45 which explain the type of map: Corneal Thickness, and proposed algorithm decision: Abnormal.

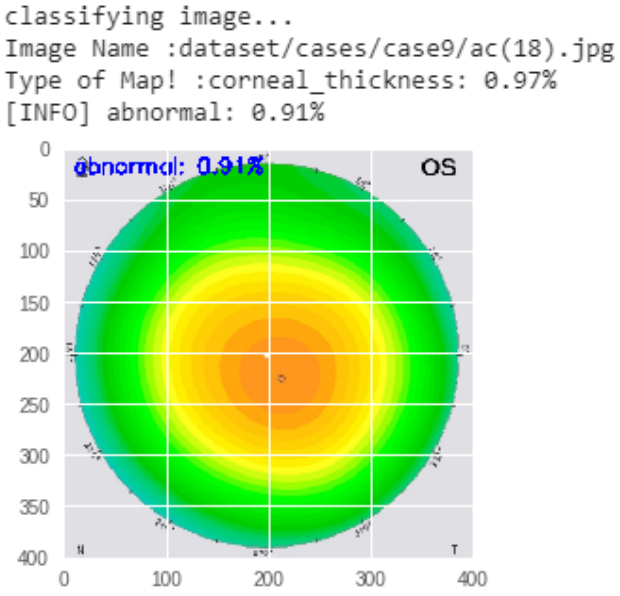


Figure 4.45. Corneal thickness for case number nine with the proposed algorithm decision.

4.10.3 Final proposed algorithm result

Final proposed algorithm result for the case number nine is Abnormal, hence: Sagittal map: abnormal, elevation front: normal, elevation back: normal, corneal thickness: abnormal.

4.10.4 Comparison between clinical decision and proposed algorithm decision

The topographic image of four refractive maps has correctly been determining the kind of map and classified as “normal” and “abnormal” by classification deep learning script.

4.11 The Work in Case Number Ten

4.11.1 Input data with a clinical decision

The input data to the case number ten and clinical decision are explained as following Figure 4.46 shows the four maps for this case:

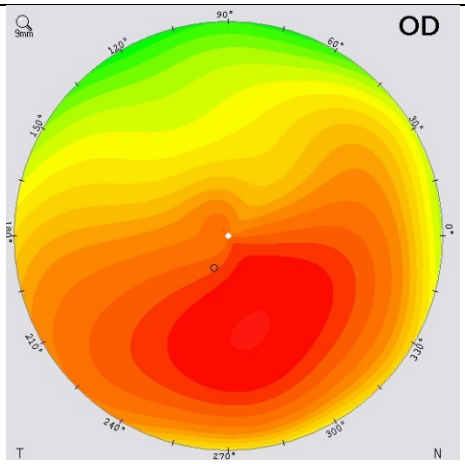
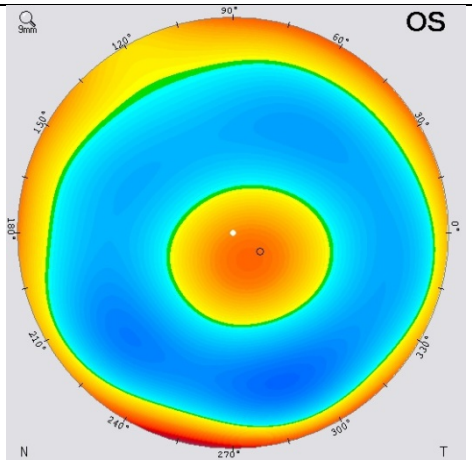
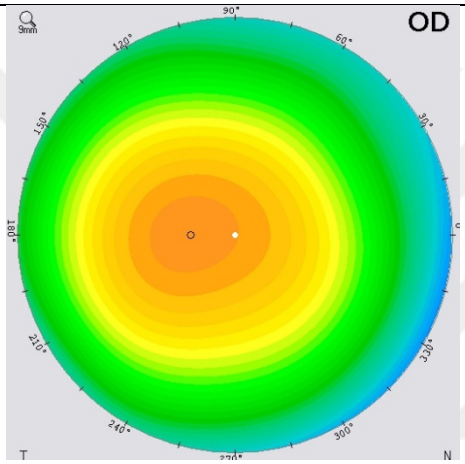
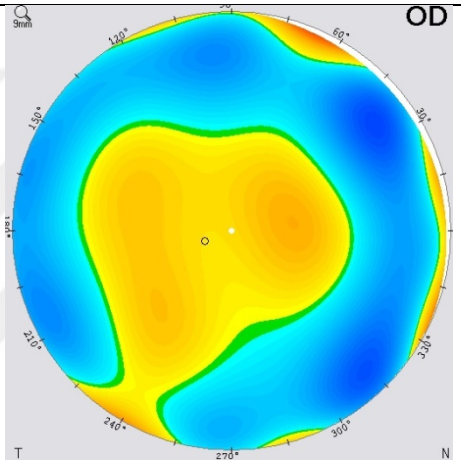
	
PENTACAM Reading: Sagittal map Clinical decision: Abnormal	PENTACAM Reading: Elevation front Clinical decision: Abnormal
	
PENTACAM Reading: Corneal thickness Clinical decision: Abnormal	PENTACAM Reading: Elevation back Clinical decision: Abnormal

Figure 4.46. Four maps for case number ten with Pentacam reading and clinical decision.

4.11.2 Output data with proposed algorithm decision:

4.11.2.1 Sagittal map

The output of the case number ten is shown in Figure 4.47 which explain the type of map: sagittal map, and proposed algorithm decision: Abnormal.

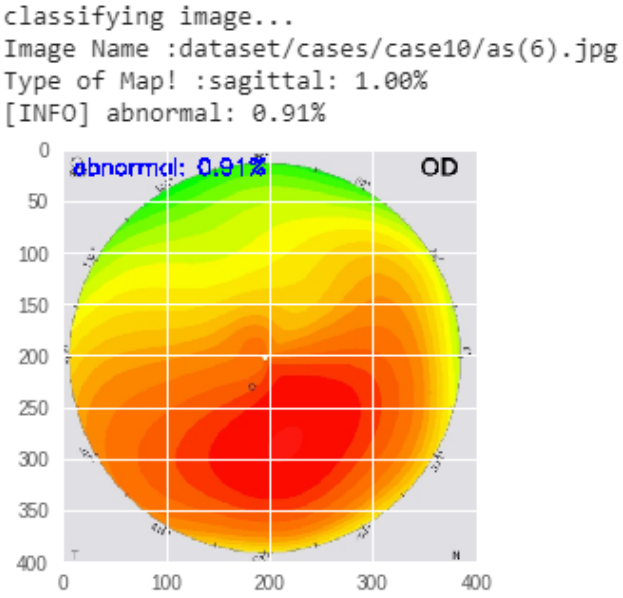


Figure 4.47. Sagittal map for case number ten with proposed algorithm decision.

4.11.2.2 Elevation front map

The output of the case number ten is shown in Figure 4.48 which explain the type of map: elevation front map, and the proposed algorithm decision: Abnormal.

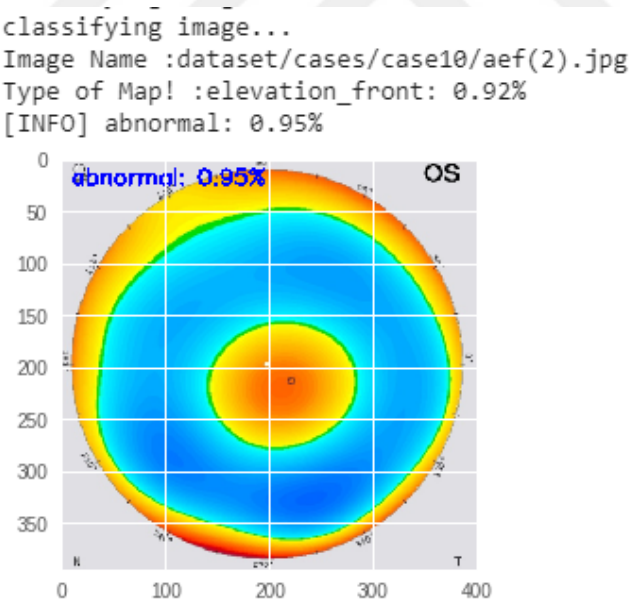


Figure 4.48. Elevation front map for case number ten with proposed algorithm decision.

4.11.2.3 Elevation back map

The output of the case number ten is shown in Figure 4.49 which explain the type of map: elevation back map, and the proposed algorithm decision: Abnormal.

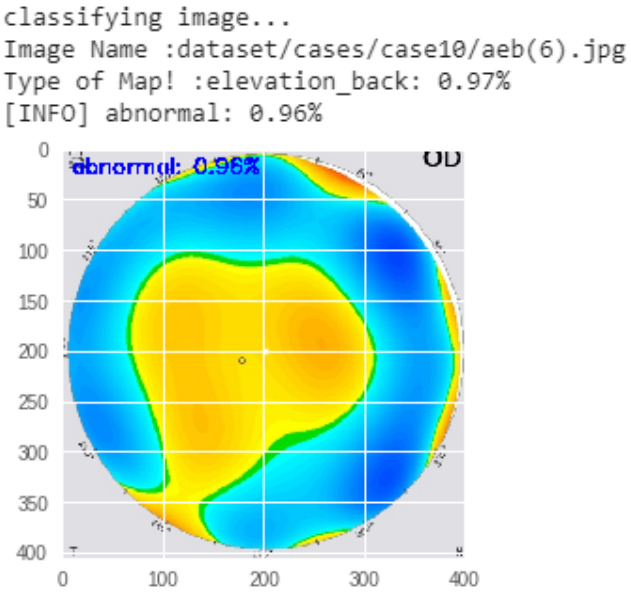


Figure 4.49. Elevation back map for case number ten with the proposed algorithm decision

4.11.2.4 Corneal thickness

The output of the case number ten is shown in Figure 4.50 which explain the type of map: Corneal Thickness, and proposed algorithm decision: Abnormal.

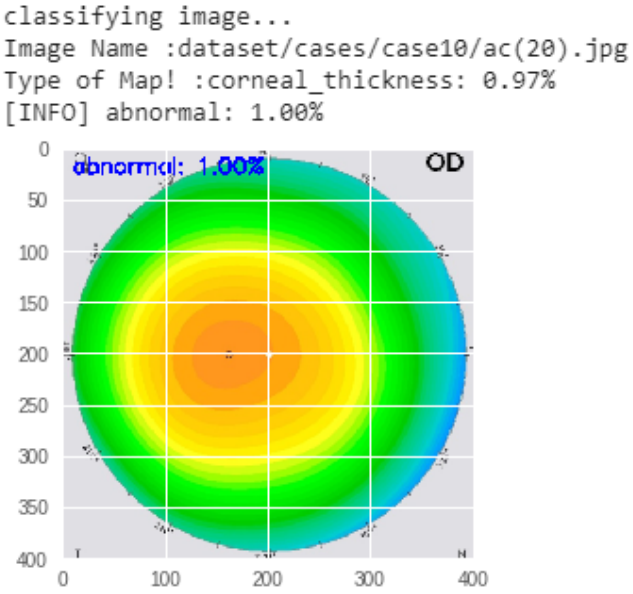


Figure 4.50. Corneal thickness for case number ten with proposed algorithm decision.

4.11.3 Final proposed algorithm result

Final proposed algorithm result for the case number ten is Abnormal, hence: Sagittal map: abnormal, elevation front: abnormal, elevation back: abnormal, corneal thickness: abnormal.

4.11.4 Comparison between clinical decision and proposed algorithm decision

The topographic image of four refractive maps has correctly been determining the kind of map and classified as "normal" and "abnormal" by classification deep learning script.

4.12 The Classification Results

SVM classifier is the classifier that has been applied to our cases to calculate the accuracy. The accuracy of SVM classifier was 96.6% for all 732 corneal topographic images in determining the type of refractive maps. Also, it achieved 88.8%, 98.9%, 94.8%, and 94.5% accuracy in determining whether each was normal or abnormal (sagittal map, elevation front map, elevation back map and corneal thickness map), respectively. Finally, the results for the four refractive maps was collect and get the final result. The clinical diagnosis is based on the experience and the reading from the Pentacam as well as the readings from other instruments in the clinic; whereas our diagnosis is based on the features that extracted from maps by using VGG-16

network. The proposed algorithm decision missed diagnosing in the case number 3 and case number 5 only when compared with the clinical decision for the discussed ten cases, but the rest of the results show a matching between both the clinical decision and the proposed algorithm decision diagnosis as shown in Table 4.1.

Table 4.1. Clinical decision compared with the proposed algorithm decision.

Case No.	Map type	Clinical Decision	Proposed Algorithm Decision	Matching
1	Sagittal	Normal	Normal	O.K
	Elevation front	Normal	Normal	O.K
	Elevation back	Normal	Normal	O.K
	Corneal thickness	Normal	Normal	O.K
2	Sagittal	Normal	Normal	O.K
	Elevation front	Normal	Normal	O.K
	Elevation back	Normal	Normal	O.K
	Corneal thickness	Abnormal	Abnormal	O.K
3	Sagittal	Normal	Normal	O.K
	Elevation front	Normal	Normal	O.K
	Elevation back	Abnormal	Abnormal	O.K
	Corneal thickness	Abnormal	Normal	Not matching
4	Sagittal	Normal	Normal	O.K
	Elevation front	Abnormal	Abnormal	O.K
	Elevation back	Abnormal	Abnormal	O.K
	Corneal thickness	Abnormal	Abnormal	O.K
5	Sagittal	Abnormal	Abnormal	O.K
	Elevation front	Abnormal	Abnormal	O.K
	Elevation back	Abnormal	Normal	Not matching
	Corneal thickness	Abnormal	Abnormal	O.K
6	Sagittal	Abnormal	Abnormal	O.K
	Elevation front	Normal	Normal	O.K
	Elevation back	Abnormal	Abnormal	O.K
	Corneal thickness	Abnormal	Abnormal	O.K
7	Sagittal	Abnormal	Abnormal	O.K
	Elevation front	Abnormal	Abnormal	O.K
	Elevation back	Normal	Normal	O.K
	Corneal thickness	Abnormal	Abnormal	O.K
8	Sagittal	Abnormal	Abnormal	O.K
	Elevation front	Abnormal	Abnormal	O.K
	Elevation back	Abnormal	Abnormal	O.K
	Corneal thickness	Normal	Normal	O.K
9	Sagittal	Abnormal	Abnormal	O.K
	Elevation front	Normal	Normal	O.K
	Elevation back	Normal	Normal	O.K
	Corneal thickness	Abnormal	Abnormal	O.K
10	Sagittal	Abnormal	Abnormal	O.K
	Elevation front	Abnormal	Abnormal	O.K
	Elevation back	Abnormal	Abnormal	O.K
	Corneal thickness	Abnormal	Abnormal	O.K

5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

From the results and discussion, it has been achieved that the proposed presented work of extract features from the corneal topographic maps was successfully efficient by a ratio of 96.6% in determining the type of refractive maps and achieved 88.8%, 98.9%, 94.8%, and 94.5% in determining whether each was normal or abnormal (sagittal map, elevation front map, elevation back map and corneal thickness map), respectively. Dataset was collected from the Pentacam. VGG-16 network was used here to extract these features.

5.2 Future Work

Our proposed work is to extract more details from the maps that are readings from Pentacam to have more information on corneal health. Our suggestions for future work are:

1- Extract more details from the four refractive maps could increase the accuracy.
2- Include other maps from another topographer device like Atlas and improve the accuracy of the final clinical decision.
3- Study the mode of abnormalities and collected further data for the classification of diseases.
4- Using the Graphical User Interface (GUI) system for more simplicity of using from the doctors and clinicians.
5- Using a mobile application to improve the doctor's decision and make it easier.
6- Applying the algorithm on the corneas maps of other devices.

REFERENCES

Acharya, U. R., Fujita, H., Lih, O. S., Adam, M., Tan, J. H., and Chua, C. K., 2017. Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network, Knowledge-Based Systems, 14, 6, 67-85.

Acharya, U. R., Fujita, H., Oh, S. L., Raghavendra, U., Tan, J. H., Adam, M., and Hagiwara, Y., 2018. Automated identification of shockable and non-shockable life-threatening ventricular arrhythmias using convolutional neural network, Future Generation Computer Systems, 79, 8, 952–959.

Ali, A. H., Ghaeb, N. H., and Musa, Z. M., 2017. Support vector machine for keratoconus detection by using topographic maps with the help of image processing techniques, IOSR Journal of Pharmacy and Biological Sciences (IOSR-JPBS), 12, 6, 50–58.

Arbelaez, M. C., Versaci, F., Vestri, G., Barboni, P., and Savini, G., 2012. Use of a support vector machine for keratoconus and subclinical keratoconus detection by topographic and tomographic data, Ophthalmology, 119, 11, 2231–2238.

Bagherinia, H., Chen, X., Flachenecker, C., Angeles, R., Burger, D., Caroline, P., and Reeder, K., 2008. Support vector machine (SVM)-based classification of corneal topography, Investigative Ophthalmology and Visual Science, 49, 13, 1023.

Camarillo, T., Choi, K., Hamilton, G., Miles, M., Muller, K., Williams, K., and Schrepel, P., 2002. Athletes as an ideal target population for orthokeratology keratoconus: improving quality of life through advancements in detection and treatment, 17, 2, 112-129.

Carvalho, L. A., 2005. Preliminary results of neural networks and zernike polynomials for classification of videokeratography maps: optometry and vision science, 82, 2, 151–158.

Cireşan, D., Meier, U., and Schmidhuber, J., 2011. A committee of neural networks for traffic sign classification, In Proceedings of the International Joint Conference on Neural Networks.

De Carvalho, L. A., and Barbosa, M. S., 2008. Neural networks and statistical analysis for classification of corneal videokeratography maps based on zernike coefficients: a quantitative comparison, Arquivos Brasileiros De Oftalmologia, 71, 3, 337–341.

Gopalakrishnan, K., Khaitan, S. K., Choudhary, A., and Agrawal, A., 2017. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection, Construction and Building Materials, 157, 8, 322–330.

Grassmann, F., Mengelkamp, J., Brandl, C., Harsch, S., Zimmermann, M. E., Linkohr, B., and Weber, B. H. F., 2018. A deep learning algorithm for prediction of age-related eye disease study severity scale for age-related macular degeneration from color fundus photography, Ophthalmology, 125, 9, 1410–

1420.

Kabari, L., and Nwachukwu, E., 2012. Neural networks and decision trees for eye diseases diagnosis, In P. Vizureanu (Ed.), Advances in Expert Systems.

Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., and Zhang, K., 2018. Identifying medical diagnoses and treatable diseases by image-based deep learning, Cell, 172, 5, 1122–1131.

Khan, S., Rahmani, H., Shah, S. A. A., and Bennamoun, M., 2018. A guide to convolutional neural networks for computer vision, Synthesis Lectures on Computer Vision (Vol. 8).

Kotsia, I., and Pitas, I., 2007. Facial expression recognition in image sequences using geometric deformation features and support vector machines, IEEE Transactions on Image Processing, 16, 1, 172–187.

Kotsiantis, S. B., Zaharakis, I. D., and Pintelas, P. E., 2006. Machine learning: A review of classification and combining techniques, Artificial Intelligence Review, 26, 3, 159–190.

Kurniawan, R., Yanti, N., and Ahmad Nazri, M. Z., 2014. Expert systems for self-diagnosing of eye diseases using Naïve Bayes (pp. 113–116). IEEE.

Maeda N., Klyce S. D., and Smolek M. K., 1995. Neural network classification of corneal topography, Invest Ophthalmol Vis Sci. 1995, 36, 7, 1327–1335.

Mazen M Sinjab., 2015. Step by step reading pentacam topography (basics and case study series), Second Edition, Jaypee Brothers Medical Publishers, New Delhi , India.

Mesko, B., 2017. The role of artificial intelligence in precision medicine, Expert Review of Precision Medicine and Drug Development, 2, 5, 239–241.

Netto, A. V., 2005. System based on computational intelligence for ophthalmology image understanding, IEEE Latin America Transactions, 3, 5, 14–22.

Noaman, K., Muharram, A., and Alqubati, I., 2014. Diagnosis of poor eyesight based on support vector machine and artificial neural networks, Journal of Emerging Trends in, 10, 5, 754–750.

Pontil, M., and Verri, A., 1998. Support vector machines for 3D object recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20, 6, 637–646.

Raghavendra, U., Fujita, H., Bhandary, S. V., Gudigar, A., Tan, J. H., and Acharya, U. R., 2018. Deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images, Information Sciences, 441, 41–49.

Riesenhuber, M., and Poggio, T., 1999. Hierarchical models of object recognition in cortex, Nature Neuroscience, 2, 11, 1019–1025.

Ruiz Hidalgo, I., Rodriguez, P., Rozema, J. J., Ni Dhubhghaill, S., Zakaria, N., Tassignon, M.-J., and Koppen, C., 2016. Evaluation of a machine-learning

classifier for keratoconus detection based on scheimpflug tomography, Cornea, 35, 6, 827–832.

Scherer, D., Müller, A., and Behnke, S., 2010. Evaluation of pooling operations in convolutional architectures for object recognition, In Lecture Notes in Computer Science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics).

Schwiegerling, J., and Greivenkamp, J. E., 1996. Keratoconus detection based on videokeratoscopic height data. Optometry and Vision Science: Official Publication of the American Academy of Optometry, 73, 12, 721–728.

Serre, T., Wolf, L., and Poggio, T., 2005. Object recognition with features inspired by visual cortex, In Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005.

Simonyan, K., and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition.

Smitha, P., Shaji, L., and Mini, M. G., 2011. A review of medical image classification techniques, International Conference on VLSI, Communication and Instrumentation, (Icvci), 34–38.

Smolek, M. K., and Klyce, S. D., 1997. Current keratoconus detection methods compared with a neural network approach, Invest Ophthalmol.Vis.Sci., 38(0146–0404 (Print)), 2290–2299.

Tan, J. H., Rajendra Acharya, U., Bhandary, S. V, Chua Chua, K., and Sivaprasad, S., 2017. Segmentation of optic disc, Fovea and Retinal Vasculature Using a Single Convolutional Neural Network.

Thoma, M., 2017. Analysis and optimization of convolutional neural network architectures, CoRR.

Toutounchian, F., Shanbehzadeh, J., Khanlari, M., and Stage, A. R., 2012. Detection of keratoconus and suspect keratoconus by machine vision, International Multiconference of Engineers and Computer Scientists, 1, 3, 14–16.

Twa, M. D., Parthasarathy, S., Roberts, C., Mahmoud, A. M., Raasch, T. W., and Bullimore, M. A., 2005. Automated decision tree classification of corneal shape: Optometry and Vision Science, 82, 12, 1038–1046.

Zhang, Z., Srivastava, R., Liu, H., Chen, X., Duan, L., Kee Wong, D. W., and Liu, J., 2014. A survey on computer aided diagnosis for ocular diseases, BMC Medical Informatics and Decision Making, 14, 12, 78-99.

URL-1 < https://deepmind.com/applied/deepmind-health/working-partners/health research-tomorrow >, Date Accessed: 02.04.2018.

URL-2 < https://health.ucsd.edu/news/releases/pages/2018-02-22-arificial-intelligence quickly-and-accurately-diagnoses-eye-diseases-and-pneumonia.aspx >, Date Accessed: 16.07.2018.

URL-3 < https://keras.io >, Date Accessed: 08.06.2018.

URL-4 < https://keras.rstudio.com/articles/applications.html >, Date Accessed: 23.06.2018.

URL-5 < https://matplotlib.org/3.0.2/index.html >, Date Accessed: 09.06.2018.

URL-6 < https://opencv.org/opencv-4-0-0.html >, Date Accessed: 01.06.2018.

URL-7 < https://www.abtosoftware.com/computer-vision-and-image-processing-solutions>, Date Accessed: 15.07.2018.

URL-8 < https://www.anaconda.com >, Date Accessed: 08.06.2018.

URL-9 < http://www.numpy.org >, Date Accessed: 07.06.2018.

URL-10 < https://www.pentacam.com/int/opticianoptometrist-without-pentacamr.html >, Date Accessed: 18.04.2018.

URL-11 < https://www.python.org/downloads/windows >, Date Accessed: 12.06.2018.

URL-12 < https://www.scipy.org/install.html >, Date Accessed: 08.10.2018.

URL-13 < http://www.who.int/blindness/causes/priority/en/index8.html >, Date Accessed: 01.06.2018.

APPENDIX LIST

APPENDIX A: The proposed code for classification of the extracted features according to the refractive map type.

APPENDIX B: The proposed code for prediction of the clinical state (normal or abnormal) per individual map.

APPENDIX C: The proposed code for comparison of the predicted results and clinical decision-making.

APPENDIX A: The proposed code for classification of the extracted features according to the refractive map type.

```
# FIRST STEP: TRAINING DATA ( TYPE OF MAP - MODEL)
dataset =    'dataset/data_train_all/'
model_name = 'models/pokedex_maps.model'
labelbin =   'models/lb_maps.pickle'
svm_model =  'models/clasfi_maps.pkl'
# initialize the number of epochs to train for, initial learning rate,
# batch size, and image dimensions
EPOCHS = 100
INIT_LR = 1e-3
BS = 32
IMAGE_DIMS = (224, 224, 3)#------------------------------line----!!!!!!!!!
# initialize the data and labels
data = []
labels = []
gen = ImageDataGenerator()
# grab the image paths and randomly shuffle them
print("[INFO] loading images...")
imagePaths = sorted(list(paths.list_images(dataset)))
random.seed(42)
random.shuffle(imagePaths)
# loop over the input images
kl = []
normal_num = 0
for imagePath in imagePaths:
# load the image, pre-process it, and store it in the data list
image = cv2.imread(imagePath)
#print(image.shape[:2])
 if image is None:
          print("continue")
          continue
           #print("imagePath_2=",imagePath)
  label = imagePath.split(os.path.sep)[-2]
  image = cv2.resize(image, (IMAGE_DIMS[0], IMAGE_DIMS[1]))
  image = img_to_array(image)
  data.append(image)
# extract the class label from the image path and update the
# labels list
  labels.append(label)
# scale the raw pixel intensities to the range [0, 1]
llv = labels
data = np.array(data, dtype="float") / 255.0
labels = np.array(labels)
print("[INFO] data matrix: {:.2f}MB".format(
          data.nbytes / (1024 * 1000.0)))
# binarize the labels
```

```
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
# partition the data into training and testing splits using 80% of
# the data for training and the remaining 20% for testing
# construct the image generator for data augmentation
aug = ImageDataGenerator(rotation_range=25, width_shift_range=0.1,
        height_shift_range=0.1, shear_range=0.2, zoom_range=0.2,
        horizontal_flip=True, fill_mode="nearest")
# initialize the model
print("[INFO] compiling model...")
model = VGG16(include_top=True, weights='imagenet',
input_tensor=None, input_shape=(224, 224,3), pooling=None)
for layer in model.layers: layer.trainable=False
x = model.layers[-2].output
output_layer = Dense(4, activation='softmax', name='predictions')(x)
model = Model(inputs=model.input, outputs=output_layer)
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(optimizer=opt,
            loss='categorical_crossentropy', metrics=['accuracy'])
(trainX, testX, trainY, testY) = train_test_split(data,
        labels, test_size=0.05, random_state=42)
####trainYY = np_utils.to_categorical(labels,num_classes=2)
# train the network
print("[INFO] training network...")
H = model.fit_generator(
        aug.flow(trainX, trainY, batch_size=BS),
        validation_data=(testX, testY),
        steps_per_epoch=len(trainX) // BS,
        epochs=EPOCHS, verbose=1)
# save the model to disk
print("[INFO] serializing network...")
model.save(model_name)
#proba = model.predict(data)
# save the label binarizer to disk
print("[INFO] serializing label binarizer...")
f = open(labelbin, "wb")
f.write(pickle.dumps(lb))
f.close()
#ext features
features = model.predict(trainX)
class_n = lb.classes_
label2 = []
for i in trainY:
  label2.append(np.argmax(i))
print('Training classifier and save')
model_svm = SVC(kernel='linear', probability=True)
model_svm.fit(features, label2)
with open(svm_model, 'wb') as outfile:
    pickle.dump((model_svm ,class_n), outfile)
#  TESTING DATA
```

```
from sklearn.metrics import accuracy_score
dataset   = 'dataset/data_test_all/'
model     = 'models/pokedex_maps.model'
labelbin  = 'models/lb_maps.pickle'
svm_model = 'models/clasfi_maps.pkl'
# initialize the number of epochs to train for, initial learning rate,
# batch size, and image dimensions
EPOCHS = 100
INIT_LR = 1e-3
BS = 32
IMAGE_DIMS = (224, 224, 3)#------------------------------line----!!!!!!!!!
# initialize the data and labels
data = []
labels = []
print("loading network...")
model = load_model(model)
print("[INFO] loading images...")
imagePaths = sorted(list(paths.list_images(dataset)))
random.seed(42)
random.shuffle(imagePaths)
# loop over the input images
kl = []
normal_num = 0
for imagePath in imagePaths:
        # load the image, pre-process it, and store it in the data list
        image = cv2.imread(imagePath)
        if image is None:
        print("continue")
        continue
        label = imagePath.split(os.path.sep)[-2]
        image = cv2.resize(image, (IMAGE_DIMS[0], IMAGE_DIMS[1]))
        image = img_to_array(image)
        data.append(image)
        # extract the class label from the image path and update the
        # labels list
        labels.append(label)
# scale the raw pixel intensities to the range [0, 1]
data = np.array(data, dtype="float") / 255.0
labels = np.array(labels)
print("[INFO] data matrix: {:.2f}MB".format(
        data.nbytes / (1024 * 1000.0)))
# binarize the labels
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
label2 = []
for i in labels:
  label2.append(np.argmax(i))
features2 = model.predict(data)
print('Testing classifier')
with open(svm_model, 'rb') as infile:
```

```
    (model_svm, class_n) = pickle.load(infile)
predictions = model_svm.predict_proba(features2)
best_class = np.argmax(predictions, axis=1)
class_probabilities = predictions[np.arange(len(best_class)), best_class]
for i in range(len(best_class)):
    print('%4d  %s: %.3f' % (i, class_n[best_class[i]], class_probabilities[i]))
#accuracy = np.mean(np.equal(best_class, labels))
predicted = model_svm.predict(features2)
accuracy = accuracy_score(label2, predicted)
print('Accuracy: %.3f' % accuracy)
```

APPENDIX B: The proposed code for prediction of the clinical state (normal or abnormal) per individual map.

```
# # SECOND STEP: TRAINING DATA (4 MAPS - MODEL)
# initialize the number of epochs to train for, initial learning rate,
# batch size, and image dimensions
EPOCHS = 100
INIT_LR = 1e-3
BS = 32
IMAGE_DIMS = (224, 224, 3)
def train_all(dataset,model_name,labelbin,svm_model):
   print("Start training on data: {}".format(dataset))
   map_name0 = dataset.split(os.path.sep)[-1]#get name map!
   # initialize the data and labels
   data = []
   labels = []
   gen = ImageDataGenerator()
   # grab the image paths and randomly shuffle them
   print("[INFO] loading images...")
   imagePaths = sorted(list(paths.list_images(dataset)))
   random.seed(42)
   random.shuffle(imagePaths)
   # loop over the input images
   kl = []
   normal_num = 0
   for imagePath in imagePaths:
         # load the image, pre-process it, and store it in the data list
    image = cv2.imread(imagePath)
    if image is None:
        print("continue")
        continue
    label = imagePath.split(os.path.sep)[-2]
    image = cv2.resize(image, (IMAGE_DIMS[0], IMAGE_DIMS[1]))
    image = img_to_array(image)
    data.append(image)
         # extract the class label from the image path and update the
    # labels list
    labels.append(label)
   # scale the raw pixel intensities to the range [0, 1]
   llv = labels
   data = np.array(data, dtype="float") / 255.0
   labels = np.array(labels)
   print("[INFO] data matrix: {:.2f}MB".format(
               data.nbytes / (1024 * 1000.0)))
   # binarize the labels
   lb = LabelBinarizer()
   labels = lb.fit_transform(labels)
   # construct the image generator for data augmentation
   aug = ImageDataGenerator(rotation_range=25, width_shift_range=0.1,
```

```
                height_shift_range=0.1, shear_range=0.2, zoom_range=0.2,
                horizontal_flip=True, fill_mode="nearest")
    # initialize the model
    print("[INFO] compiling model...")
    model = VGG16(include_top=True, weights='imagenet',
                        input_tensor=None, input_shape=(224, 224,3), pooling=None)
    for layer in model.layers: layer.trainable=False
    x = model.layers[-2].output
    output_layer = Dense(2, activation='softmax', name='predictions')(x)
    model = Model(inputs=model.input, outputs=output_layer)
    opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
    model.compile(optimizer=opt,
            loss='categorical_crossentropy', metrics=['accuracy'])
    trainYY = np_utils.to_categorical(labels)
    # train the network
    print("[INFO] training network...")
    H = model.fit_generator(
                aug.flow(data, trainYY, batch_size=BS),
                steps_per_epoch=len(data) // BS,
                epochs=EPOCHS, verbose=1)
    # save the model to disk
    print("[INFO] serializing network...")
    model.save(model_name)
    #proba = model.predict(data)
    # save the label binarizer to disk
    print("[INFO] serializing label binarizer...")
    f = open(labelbin, "wb")
    f.write(pickle.dumps(lb))
    f.close()
    #ext features
    features = model.predict(data)
    class_n = lb.classes_
    label2 = []
    for i in trainYY:
       label2.append(np.argmax(i))
    print('Training classifier and save')
    model_svm = SVC(kernel='linear', probability=True)
    model_svm.fit(features, labels)
    with open(svm_model, 'wb') as outfile:
        pickle.dump((model_svm ,class_n), outfile)
    print("Finished training on data: {}".format(dataset))
# Run training (4 maps-model)
from os.path import join as j
dir_data  = 'dataset/data_maps/'
sagittal  = 'sagittal'
front     = 'elevation_front'
back      = 'elevation_back'
thickness = 'corneal_thickness'
train_all(j(dir_data ,sagittal) ,model_sagittal ,label_sagittal ,svm_sagittal_f)
```

```
train_all(j(dir_data ,front) ,model_elevation_front ,label_elevation_front
,svm_elevation_front_f)
train_all(j(dir_data ,back) ,model_elevation_back ,label_elevation_back
,svm_elevation_back_f)
train_all(j(dir_data ,thickness) ,model_corneal_thickness ,label_corneal_thickness
,svm_corneal_thickness_f)
print("[***]")
print("[***]Finished training all")
# TESTING DATA
from sklearn.metrics import accuracy_score
# initialize the number of epochs to train for, initial learning rate,
# batch size, and image dimensions
EPOCHS = 100
INIT_LR = 1e-3
BS = 32
IMAGE_DIMS = (224, 224, 3)
def accuracy(dataset ,model ,labelbin ,svm_model):
    # initialize the data and labels
    data = []
    labels = []
    name_map = dataset.split(os.path.sep)[-1]
    print("loading network...")
    model = load_model(model)
    print("[INFO] loading images...")
    imagePaths = sorted(list(paths.list_images(dataset)))
    random.seed(42)
    random.shuffle(imagePaths)
    # loop over the input images
    kl = []
    normal_num = 0
    for imagePath in imagePaths:
    # load the image, pre-process it, and store it in the data list
        image = cv2.imread(imagePath)
        if image is None:
            print("continue")
            continue
        label = imagePath.split(os.path.sep)[-2]
        image = cv2.resize(image, (IMAGE_DIMS[0], IMAGE_DIMS[1]))
        image = img_to_array(image)
        data.append(image)
        # extract the class label from the image path and update the
        # labels list
        labels.append(label)
    # scale the raw pixel intensities to the range [0, 1]
    data = np.array(data, dtype="float") / 255.0
    labels = np.array(labels)
    print("[INFO] data matrix: {:.2f}MB".format(
        data.nbytes / (1024 * 1000.0)))
    # binarize the labels
    lb = LabelBinarizer()
```

```
    labels = lb.fit_transform(labels)
    label2 = []
    for i in labels:
        label2.append(np.argmax(i))
    features2 = model.predict(data)
    print('Testing classifier')
    with open(svm_model, 'rb') as infile:
        (model_svm, class_n) = pickle.load(infile)
    predictions = model_svm.predict_proba(features2)
    best_class = np.argmax(predictions, axis=1)
    class_probabilities = predictions[np.arange(len(best_class)), best_class]
    for i in range(len(best_class)):
        print('%4d  %s: %.3f' % (i, class_n[best_class[i]], class_probabilities[i]))
    predicted = model_svm.predict(features2)
    accuracy = accuracy_score(labels, predicted)
    print('Accuracy: %.3f' % accuracy)
    return name_map ,('%.3f' % accuracy)
from os.path import join as j
dir_data  = 'dataset/test_maps/'
sagittal  = 'sagittal'
front     = 'elevation_front'
back      = 'elevation_back'
thickness = 'corneal_thickness'
acc_stack = []
acc_stack.append(accuracy(j(dir_data     ,sagittal)     ,model_sagittal     ,label_sagittal
,svm_sagittal_f))
acc_stack.append(accuracy(j(dir_data     ,front)     ,model_elevation_front
,label_elevation_front ,svm_elevation_front_f))
acc_stack.append(accuracy(j(dir_data     ,back)     ,model_elevation_back
,label_elevation_back ,svm_elevation_back_f))
acc_stack.append(accuracy(j(dir_data     ,thickness)     ,model_corneal_thickness
,label_corneal_thickness ,svm_corneal_thickness_f))
print("[%%]Finished ")
print(acc_stack)
```

APPENDIX C: The proposed code for comparison of the predicted results and clinical decision-making.

```
# # THIRD STEP: CLASSIFICATION (10 CASES)
# ** Function to display images within the current folder **
IMAGE_DIMS = (224, 224)
def get_type_classifi(image ,modell_maps ,type_maps):
    image = cv2.imread(img_path)
    output = image.copy()
    # pre-process the image for classification
    image = cv2.resize(image, (IMAGE_DIMS[0], IMAGE_DIMS[1]))
    image = image.astype("float") / 255.0
    image = img_to_array(image)
    image = np.expand_dims(image, axis=0)
    # load the trained convolutional neural network and the label
    # binarizer
    if (type_maps == 'sagittal'):
        model = modell_maps.model_sagittal
        lb = modell_maps.label_sagittal
        model_svm = model_maps_all.svm_sagittal
        class_n   = model_maps_all.class_sagittal
    elif (type_maps == 'elevation_front'):
        model = modell_maps.model_elevation_front
        lb = modell_maps.lb_elevation_front
        model_svm = model_maps_all.svm_elevation_front
        class_n   = model_maps_all.class_elevation_front
    elif (type_maps == 'elevation_back'):
        model = modell_maps.model_elevation_back
        lb = modell_maps.lb_elevation_back
        model_svm = model_maps_all.svm_elevation_back
        class_n   = model_maps_all.class_elevation_back
    elif (type_maps == 'corneal_thickness'):
        model = modell_maps.model_corneal_thickness
        lb = modell_maps.lb_corneal_thickness
        model_svm = model_maps_all.svm_corneal_thickness
        class_n   = model_maps_all.class_corneal_thickness
    # classify the input image
    print("classifying image...")
    features3 = model.predict(image) #Extracting features use VGG16 model
    predictions = model_svm.predict_proba(features3)
    best_class = np.argmax(predictions, axis=1)
    class_probabilities = predictions[np.arange(len(best_class)), best_class]
    for i in range(len(best_class)):
        name_obj = class_n[best_class[i]]
        acc_best = class_probabilities[i]
    if name_obj.find('/') != -1:
        name_obj = label[name_obj.rfind('/')+1:len(name_obj)]
    # build the label and draw the label on the image
    label = "{}: {:.2f}%".format(name_obj, acc_best)
```

```
    output = imutils.resize(output, width=400)
    cv2.putText(output, label, (10, 25),  cv2.FONT_HERSHEY_SIMPLEX,
          0.7, (255, 0, 0), 2) #color ditc
    return output ,name_obj ,acc_best
#load data
data_dir = 'dataset/cases/' #edit then
# loda your models wati class model_maps_all
model_maps_all = model_maps_all()
IMAGE_DIMS = (224, 224) #------------------------------line----!!!!!!!!!
maps_all  = model_maps_all.model_maps
lb_maps   = model_maps_all.lb_maps
model_svm = model_maps_all.svm_all
class_n   = model_maps_all.class_all
print("waiting!")
for class_name in listdir(data_dir):
    if not isdir(join(data_dir, class_name)):
        continue
    status = {}
    abnormal_cunt = 0
    normal_cunt = 0
    print("#_________________ NEW PERSON _____________________#")
    for f in os.listdir(join(data_dir, class_name)):
      if re.match(r'.*\.(jpg|jpeg|png|bmp)', f, flags=re.I):
        img_path = os.path.join(data_dir, join(class_name, f))
        image = cv2.imread(img_path)
        output = image.copy()
        # pre-process the image for classification
        image = cv2.resize(image, (IMAGE_DIMS[0], IMAGE_DIMS[1]))
        image = image.astype("float") / 255.0
        image = img_to_array(image)
        image = np.expand_dims(image, axis=0)
        # load the trained convolutional neural network and the label
        # binarizer
        # classify the input image
        print("classifying image...")
        features3 = maps_all.predict(image)
        predictions = model_svm.predict_proba(features3)
        best_class = np.argmax(predictions, axis=1)
        class_probabilities = predictions[np.arange(len(best_class)), best_class]
        for i in range(len(best_class)):
            name_obj = class_n[best_class[i]]
            acc_best = class_probabilities[i]
        # build the label and draw the label on the image
        label = "{}: {:.2f}%".format(name_obj, acc_best)
        type_maps = name_obj
        if type_maps.find('/') != -1: #fixed name ! "datatest/data_out/sagittal"
          type_maps = type_maps[type_maps.rfind('/')+1:len(type_maps)]
          label = "{}: {:.2f}%".format(type_maps, acc_best)
        #print(type_maps)
        output , result ,acc = get_type_classifi(image ,model_maps_all ,type_maps)
```

```
    name_result = "{}: {:.2f}%".format(result, acc)
    if result == 'abnormal':
       abnormal_cunt +=1
    elif result == 'normal':
       normal_cunt +=1
    status[type_maps] = result
    print("Image Name :{}".format(img_path))
    print("Type of Map! :{}".format(label))
    # show the output image
    print("[INFO] {}".format(name_result))
    imgplot = plt.imshow(cv2.cvtColor(output, cv2.COLOR_BGR2RGB))
    plt.show()
 print("------- \nresult:")
 print("name of person:{}".format(class_name))
 if abnormal_cunt == 3 or abnormal_cunt >= normal_cunt:
    print("___ABNORMAL")
    print("___status \n" ,status)
 elif normal_cunt == 3 or normal_cunt > abnormal_cunt:
    print("___NORMAL")
    print("___status \n" ,status)
# elif normal_cunt == abnormal_cunt:
#     print("___SUSPECTED ABNORMAL")
#     print("___status \n" ,status)
 print("#-------------------------------------------------------------------------#\n            \n"
```

CURRICULUM VITAE

Name and surname **:** Nazar Salih Abdulhussein

Address **:** Bghdad- Iraq

E-mail address **:** nazar.s2009@yahoo.com

EDUCATION INFORMATION

Undergraduate **:** Babylon University, 1997-2000

Master **:** Aksaray University, 2017-2018

PROFESSIONAL EXPERIENCE AND AWARDS

1. AL-UFUQ Journal as web designer.
2. AL-QABAS Corporation for computers.
2. AL-NABAA Corporation for networks and computers.

PUBLICATIONS, PRESENTATIONS AND PATENTS

1. Human corneal state prediction from topographical maps using a deep neural network and a support vector machine, Aksaray University, Faculty of Engineering, Electrical and Electronics Engineering Department, Aksaray, TURKEY, International Journal of Current Research Vol. 10, Issue, 11, pp.75461-75467, November, 2018.

2. Artificial intelligence in corneal topography, Aksaray University, Faculty of Engineering, Electrical and Electronics Engineering Department, Aksaray, TURKEY, Journal of Intelligent Systems: Theory and Applications Vol. 2, Issue, 1, January, 2019.