ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

MSc THESIS

Ahmed AL-KARAWI

A NOVEL FRAMEWORK FOR SEVERITY LEVEL DETECTION OF DIABETIC RETINOPATHY BASED ON MOBILE EDGE COMPUTING AND DEEP LEARNING ENSEMBLES

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ADANA-2021

ABSTRACT

MSc THESIS

A NOVEL FRAMEWORK FOR SEVERITY LEVEL DETECTION OF DIABETIC RETINOPATHY BASED ON MOBILE EDGE COMPUTING AND DEEP LEARNING ENSEMBLES

Ahmed AL-KARAWI

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Supervisor : Assist. Prof. Dr. Ercan AVŞAR
Year: 2021, Pages:55
Jury : Asst. Prof. Dr. Ercan AVŞAR
: Asst. Prof. Dr. Ahmet AYDIN
: Asst. Prof. Dr. Mümine KAYA KELEŞ

In the current era of medical science, the detection of diabetic retinopathy in an automated way is very important, as it is one of the major reasons of vision loss in the middle-age population in the developed world. If it is detected at its early stages, it may be possible to alter the severe vision loss problem. Even though there are various approaches for feature extraction, the classification task for retinal images is still challenging for human experts. Utilization of internet-based intelligent systems may be useful for diagnosis of such health problems.

Widespread usage of such systems requires maintaining a high detection accuracy as well as rapid response while utilizing the bandwidth efficiently. In this thesis, a framework is proposed for detecting severity levels diabetic retinopathy images using a smartphone as an edge computing device. The framework mainly consists of preprocessing, feature extraction, and classification steps, where the first step is performed on the edge device and the other two on the cloud computer. The overall performance is calculated based on the classification accuracy, response time, and total transmitted data. The preprocessing step involves cropping, resizing and unsharp masking. The classification is based on a concatenation ensemble of three benchmark convolutional neural network architectures that are EfficientNetB7, ResNet50, and VGG19. The proposed framework achieved a test accuracy of 0.96 and the edge computing improved the server response time and bandwidth occupation.

Keywords: Convolutional neural networks, cloud computing, deep learning, diabetic retinopathy, edge computing, fundus

ÖZ

YÜKSEK LİSANS TEZİ

DIYABETİK RETİNOPATİNİN ŞİDDET SEVİYESİNİN TESPİTİ İÇİN MOBİL KENAR HESAPLAMA VE DERİN ÖĞRENME TOPLULUKLARINA DAYANAN BİR YÖNTEM

Ahmed AL-KARAWI

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Dr. Öğr. Üyesi Ercan AVŞAR
Yıl: 2021, Sayfa: 55
Jüri : Dr. Öğr. Üyesi Ercan AVŞAR
: Dr. Öğr. Üyesi Ahmet AYDIN
: Dr. Öğr. Üyesi Mümine KAYA KELEŞ

Tıp biliminin mevcut çağında, gelişmiş dünyada orta yaşlı nüfusta geri dönüşü olmayan görme kaybının önde gelen nedeni olduğu için diyabetik retinopatinin otomatik tespiti büyük önem taşımaktadır. Eğer erken teşhis edilirse, görme kaybı problemlerinin önüne geçilebilir. Öznitelik çıkarmak için çeşitli yaklaşımlar önerilmiş olmasına rağmen, retina görüntüleri için sınıflandırma görevi, uzman kişiler için bile hâlâ zor bir işlemdir. Internet tabanlı akıllı sistemler bu tip hastalıkların teşhisinde faydalı olabilir.

Bu tip sistemlerin yaygın olarak kullanılması, yüksek bir tespit oranı ve hızlı cevap özelliklerini taşırken bant genişliğini de etkin olarak kullanmayı gerektirir. Bu tez çalışmasında, akıllı telefonları bir kenar hesaplama aracı olarak kullanan ve diyabetik retinopati görüntülerinde hastalık seviyesini tespit eden bir sistem geliştirilmiştir. Sistem genel olarak ön işleme, öznitelik çıkarma ve sınıflandırma adımlarından oluşmaktadır. İlk adım kenar cihazında, diğer iki adım ise bulut bilgisayarda gerçekleştirilmiştir. Sistemin performansının ölçülmesi için sınıflandırma başarımı, cevap süresi ve toplam iletilen veri miktarı hesaplanmıştır. Ön işleme adımı kırpma, yeniden boyutlandırma ve keskin olmayan maskeleme işlemlerinden oluşmaktadır. Sınıflandırma ise EfficientNetB7, ResNet50 ve VGG19 ağlarının birleştirilmesi ile oluşan bir model ile sağlanmıştır. Önerilen sistemin test başarımı 0.96 olarak hesaplanmıştır ve kenar hesaplamanın, cevap süresi ve bant genişliği kullanımını iyileştirdiği görülmüştür.

Anahtar kelimeler: Konvolüsyonel sinir ağları, bulut hesaplama, derin öğrenme, diyabetik retinopati, kenar hesaplama, fundus

EXTENDED ABSTRACT

In the medical care field, treatment of sicknesses is more successful when recognized at its beginning phase. Diabetes is a sickness that builds the measure of glucose in the blood brought about by an absence of insulin. It influences the retina, heart, nerves, and kidneys of 425 million grownups around the world. Diabetic retinopathy (DR) is a typical disease of diabetes mellitus, which causes sores on the retina that impact vision. In recent years, the quantity of DR cases has an expanding pattern and is relied upon to arrive at pandemic extents worldwide throughout the following years. Therefore, timely detection of DR is very important.

With the recent improvements in the internet and computation technology, it has become possible to perform diagnosis of such dieses with the help of intelligent systems. Such contemporary systems are deployed according to IoT principles. As a result, next generation of smart healthcare systems features remote analysis capability of the healthcare data. The idea in such systems is to transmit the healthcare data to the clinicians at a distant location over an internet connection. Since high amount of medical data generates too much network traffic, preprocessing of the data before transmission is important. This allows for the amount of data be reduced and save bandwidth. This preprocessing step is performed at the sensing device or gateway levels that have hardware with limited computing power. Therefore, relatively light operations like filtering, compression and resizing are performed as the preprocessing. Since these operations are applied to data before transmitting it to the cloud server, they are called collectively as edge computing to denote this situation. The preprocessed data is typically further analysed on a distant cloud server where algorithms requiring heavier computation are executed. The results of the cloud computation are then evaluated by the responsible doctors for final decision and generating an appropriate treatment scheme

In this thesis work, a framework for detecting the severity level of DR is proposed. The fundus images are first preprocessed on a mobile edge device. The

preprocessing step includes cropping the uninformative parts around the image, applying unsharp masking to the images and resizing them to have dimensions appropriate for CNN models. These operations allow a smaller amount of data to be transferred to the cloud server, where an ensemble of deep learning methods performs the detection. The CNN architectures used for classification are Residual Networks (ResNet50), EfficientNetB7, and VGG19. The outputs from these feature extraction submodels are flattened and concatenated into one long vector. After concatenation of these three feature vectors, a vector of 384 components has been generated, and, then passed on to a fully connected layer of interpretation. Next, two more dense layers and a dropout layer are added to avoid overfitting and increase generalization. To determine the suitable method for detection, performances obtained with each of these models are compared with the performances of ensembles of these models.

The performance of the system is calculated based on classification accuracy, bandwidth occupation and server response time. The proposed concatenation ensemble of EfficientNetB6, ResNet50, and VGG19 architectures has achieved about 0.97 classification accuracy score on the test set. Furthermore, the experimental results have achieved minimal time, memory storage and CPU utilization. The proposed algorithm is proved as powerful and dynamic for diabetic image classification and feature extraction.

ACKNOWLEDGEMENT

I want to express my deepest gratitude to Asst. Prof. Dr. Ercan AVŞAR for his supervision, direction, motivation, constructive advice, and continued faith in me in this thesis. It was a huge honor for me to have Asst. Prof. Dr. Ercan AVŞAR as a supervisor.

I also extend my gratitude to Asst. Prof. Dr. Ahmet AYDIN and Asst. Prof. Dr. Mümine KAYA KELEŞ for their support and motivation during the academic year and for his approval to be a member of the jury.

I owe my deepest thanks to Çukurova University Engineering Faculty, Electrical and Electronics Engineering Department that enabled me to benefit from all facilities of the department during my thesis study.

I want to share my sincere thanks and gratitude to the Turkish Government and people for giving me this opportunity and for their hospitality.

I would like to express my deepest gratitude to all my friends (especially my close friend BASHAR SAAD FALIH AL-SAFFAR) for encouragement and support.

Finally, I wish to express my heartfelt thankfulness to my family for their patience, encouragement, and continuous moral support.

CONTENTS **PAGE**

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

LIST OF ABBREVIATIONS

DR : Diabetic Retinopathy
CNN : Convolutional Neural Network
CAD : Computer-aided diagnosis
FC layers: Fully-Connected Layer
LPF : Low-Pass Filter
CM : Confusion Matrix
HF : High frequency
RAM : Read And Write Memory
CPU : Center Processing Unit
TP : True positives
TN : True negatives
FP : False positives
FN : False negatives
Acc : Accuracy
Sens : Sensitivity
IoT : Internet of Things
DNN : Deep Neural Networks

1. INTRODUCTION

1.1. Research Motivation

In the medical care field, treatment of sicknesses is more successful when recognized at its beginning phase. Diabetes is a sickness that builds the measure of glucose in the blood brought about by an absence of insulin (Huang & Li, 2020). It influences the retina, heart, nerves, and kidneys of 425 million grownups around the world (Cheung et al., 2020).

Diabetic retinopathy (DR) is a typical disease of diabetes mellitus, which causes sores on the retina that impact vision. In recent years, the quantity of DR cases has an expanding pattern and is relied upon to arrive at pandemic extents worldwide throughout the following years (Zheng, He, & Congdon, 2012). If DR is not diagnosed at early stages, it may cause visual impairment. As a result, early recognition and timely treatment can eliminate the risks of this disease. The conventional way of detecting DR is to take images of an eye's rear through fundus photography and then visually inspect the images. The manual inspection of these images by ophthalmologists is a time-consuming process. Besides, it requires experience and expertise to diagnose the DR disease correctly. Computer-aided diagnosis (CAD) systems are widely used in various medical applications, and such systems may be useful in helping clinicians diagnose DR.

In deep learning, convolutional neural networks (CNN) is a widely-used method in variety of problems like segmentation, object detection, and image classification. The properties like learning complex feature in the images and automatically extracting these features make CNN popular in such applications. As expected, CAD systems that process medical images for making a diagnosis may utilize CNN methods for accurate performance. Therefore, diagnosis of DR from the fundus images may be performed using CNN methods as well.

In the current age of innovation, the Internet turns into an essential piece of human existence. Clinical science has utilized internet-based tools for quick diagnosis at distant locations. Such tools allow the experts to inspect the clinical data and initialize the appropriate treatment time without seeing the patient. Advancements in network technologies triggered the emergence of this idea. With the widespread usage of the Internet, the investments made on the network systems increased to provide a seamless and fast connection to the high number of internet users. Despite the large bandwidth available for the Internet, it should be used economically because of the simultaneous requests made by the high amount of users.

The popularity of the systems based on the Internet of things (IoT) principles is increasing every day. IoT systems can be considered a significant source for internet data traffic because they are applied to many areas, and in turn, a huge scope of data comprising images, video, sound, and text is generated. Therefore, instead of sending all the data generated by IoT systems, a preprocessing step may be useful in reducing the data amount transmitted to the cloud server and the computations made on the cloud server. Such preprocessing step is known as edge computing, which involves relatively lighter calculations that can be run on the hardware with limited computing capacity.

1.2. Deep Learning and Applications in Medical Diagnosis

In the era of medical science, medical diagnosis is the operational process used for determining the cause of diseases and their medical condition based on laboratory testing, patient interviews, and family medical report. The rapid development of machine learning and data mining towards medical diagnosis and healthcare applications are based on creating an intelligence system to optimize the current medical process. Machine learning is currently applied over various areas as daily treatment, drug identification, clinical testing, research, radiology, diabetic retinopathy, radiotherapy, and smart health care report generation (Cheung et al., 2020; Gagliano, Van Pham, Tang, Kashif, & Ban).

Most of the medical diagnosis research using machine learning has been done in the field of image recognition based on MRI, Ultrasound, and X-ray images of the patients. Machine learning is designed for the analysis of the massive medical dataset. The tools of machine learning are more reliable towards the medical diagnosis. For instance, an intelligent heart attack prediction system using the genetic algorithm based on the historical heart dataset was proposed earlier (Sharma & Saxena, 2017). Such applications require collection and analysis of variety of medical data and problem-specific solutions need to be developed using data analysis methods.

In the era of medical science, medical diagnosis is the operational process used for determining the cause of patients and their medical condition based on laboratory testing, patient's and family medical reports. The rapid development of machine learning and data mining towards medical diagnosis and healthcare applications is based on creating an intelligence system to optimize the current medical process.

Currently, machine learning enabled devices can analyse large amounts of information, and make decisions much faster than humans. We use machine learning to promote genome science, drug creation, and medical imaging. Machine learning and deep learning are better prepared to evaluate the contexts, causes, and cases than live-health employees.

Diagnosis through machine learning works when the condition can be reduced to a classification task on physiological data where the clinician has to recognize a pattern that predicts the existence or form of the condition.

1.3. Edge Computing for Internet of Things

Many contemporary systems are deployed according to IoT principles. As a result, next generation of smart healthcare systems features remote analysis capability of the healthcare data. The idea in such systems is to transmit the healthcare data to the clinicians at a distant location over an internet connection.

Since high amount of medical data generates too much network traffic, preprocessing of the data before transmission is important. This allows for the amount of data be reduced and save bandwidth. This preprocessing step is performed at the sensing device or gateway levels that have hardware with limited computing power. Therefore, relatively light operations like filtering, compression and resizing are performed as the preprocessing. Since these operations are applied to data before transmitting it to the cloud server, they are called collectively as edge computing to denote this situation. The preprocessed data is typically further analysed on a distant cloud server where algorithms requiring heavier computation are executed. The results of the cloud computation are then evaluated by the responsible doctors for final decision and generating an appropriate treatment scheme. Generic steps of a smart healthcare system configuration are presented in Figure 1.1. As a result, the processing capabilities of the edge computing hardware do not have to be as high as cloud computing hardware. Accordingly, the devices with limited processing feature like single board computers or mobile phones may be used at the edge computing layer. However, the devices in the cloud computing layer should be able to handle heavy computation tasks, so they are usually provided as virtual machines (VM) involved in infrastructure as a service (IaaS) systems.

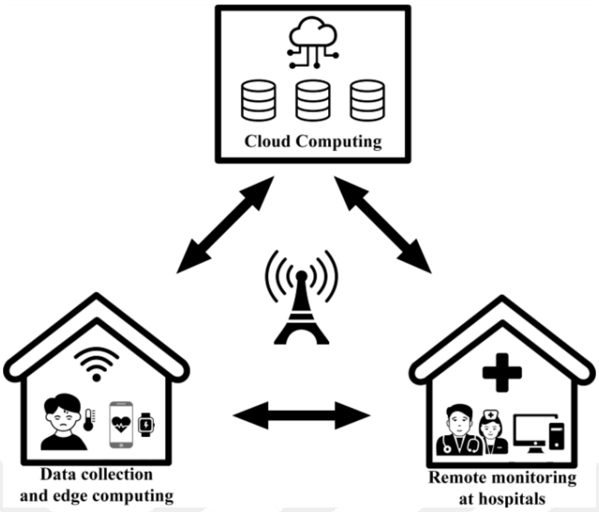


Figure 1.1. A smart healthcare system configuration

1.4. Proposed Steps and Contributions

In this thesis work, a framework for detecting the severity level of DR is proposed. The fundus images are first preprocessed on a mobile edge device. The preprocessing step includes cropping the uninformative parts around the image, applying unsharp masking to the images and resizing them to have dimensions appropriate for CNN models. These operations allow a smaller amount of data to be transferred to the cloud server, where an ensemble of deep learning methods performs the detection. The CNN architectures used for classification are Residual Networks (ResNet50), EfficientNetB7, and VGG19. To determine the suitable method for detection, performances obtained with each of these models are compared with the performances of ensembles of these models. The steps of the proposed framework are illustrated in Figure 1.2

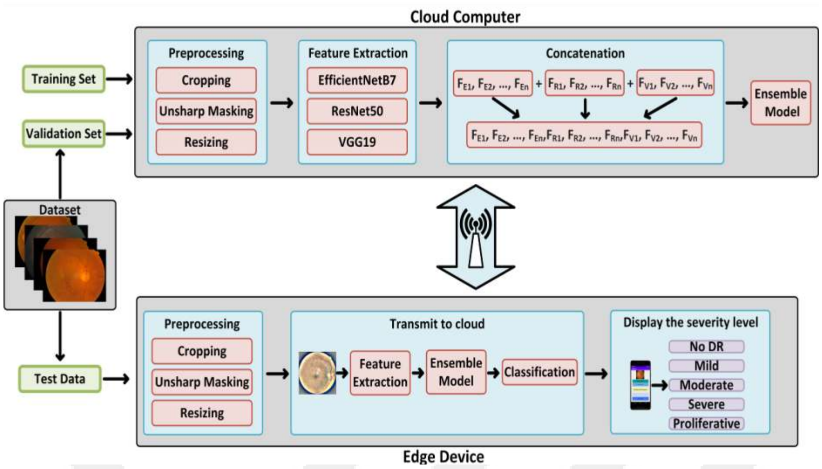


Figure 1.2. Proposed smart healthcare application framework

The contributions of this thesis can be listed as follows:

- The severity levels of DR images are determined using different CNN architectures on a cloud server.
- A mobile phone is used as an edge computing device, and the images are filtered and resized at this edge device
- Effects of edge computing on the classification performance as well as the bandwidth usage are analysed.

1.5. Outline of the Thesis

The thesis is organized as follows: Section 2 discusses diabetic retinopathy disease and its diagnosis. Section 3 discusses the related works Section 4 presents IoT techniques, the dataset details used for training and testing, experiments with basic concepts about different structures for deep learning techniques. Mobile devices for edge computing and proved how can the Edge of IoT effect on time response for the cloud. In the Section 5 the results of the experiments are presented. Finally, Section 6 concludes this work.

2. THE DIABETIC RETINOPATHY

Diabetic retinopathy (DR) is a complication of diabetes. It is a condition that occurs as a result of damage to the light-sensitive tissue at the back of the eye (retina). In DR, at first, there may be no symptoms or only mild symptoms. It could eventually cause blindness. Diabetes can develop and be diagnosed in any one who has Type 1 or Type 2 diabetes. When people have diabetes that is poorly controlled, their eye will be at greater risk of developing this complication.

The retina is the sensitive layer of the eye containing light-sensitive cells. It converts light into electric pulses by stimulating the nerve cells. The information transmitted through the optic nerves is processed by the brain into perceived images. The retina requires a continuous supply of blood. Over time, high levels of sugar in the blood can cause a reduction in blood flow to the retina. As a result, the eye attempts to produce new blood vessels to go on working appropriately. However, the new vessels do not develop properly, as a result they may leak easily (Figure 2.1).

There exist two types of DR. The first type is early DR, in which there is no new blood vessel growth in nonproliferative DR. Another one is advanced DR, in advance stage for this type known as proliferative diabetic retinopathy, the growth of new, abnormal blood vessels in the retina because of the closure of the damaged blood vessels.

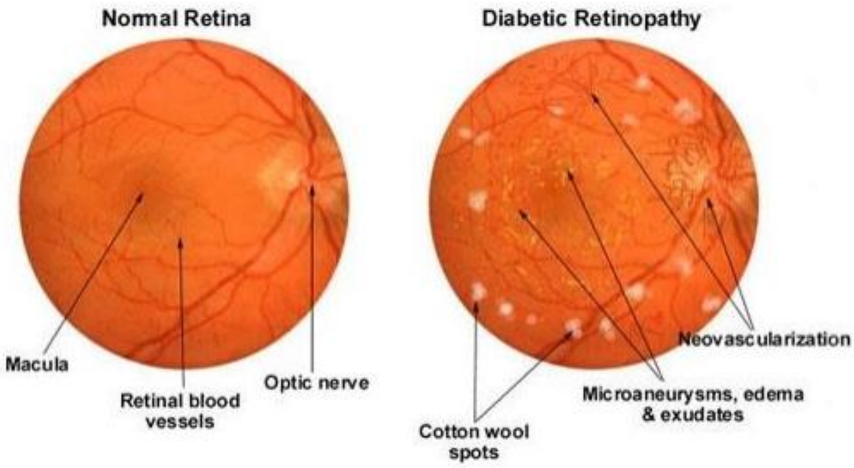


Figure 2.1. Classification of normal and DR images sample(Priya, Srinivasarao, & Sharma, 2013)

2.1. Diagnosis of Diabetic Retinopathy

Diabetes mellitus, commonly called diabetes, occurs when blood glucose levels are too high for a long time. Patients with diabetes who suffer from diabetes have a higher risk of developing diabetic retinopathy due to damage to the blood vessels caused by high blood glucose levels. (Priya et al., 2013) In order to have appropriate therapy and avoid visual loss, it is highly necessary to categorize DR depending on the severity (Priya et al., 2013).

Based upon the previous studies on the early treatment of DR, a standard that has been referred to commonly is proposed by Wilkinson (Wilkinson et al., 2003) that classified diabetic retinopathy into 5 stages including:

1. No apparent retinopathy
2. Mild Non-proliferative Diabetic Retinopathy
3. Moderate Non-proliferative Diabetic Retinopathy
4. Severe Non-proliferative Diabetic Retinopathy
5. Proliferative diabetic retinopathy

Managing type 2 diabetes requires self-monitoring of blood glucose level and following food and exercise guidelines. The glucose meter is a device used to monitor blood glucose levels, and the resulting reading is measured in milligrams per deciliter or millimoles per liter. Patients conditionally need to modify their lifestyles in order to manage their diabetes. In this case, it is most important that patients involved in decision-making for dosing and timing insulin.

One of the best strategy to delay or prevent loss vision is early diagnosis of DR. Direct ophthalmoscopy is difficult and requires practice. Patients can be screened for DR by general practitioners using nonmydriatic retinography. This cost-effective method is used to obtain digital photographs of the retina (retinograms), which can be saved in a computer for easy transport to the ophthalmologist for further examination.

When formation of new blood vessels in the retina is observed, the patients usually undergo a treatment procedure. Surgery may sometimes be necessary in cases of non proliferative diabetic retinopathy. The main symptoms of diabetic retinopathy include blurred vision or sometimes pain in the eye.

People with diabetes should have regular eye exams by a specialist who are familiar with treatment of DR. This is diagnosed by dilating pupils with eye drops and then examining the retina carefully (Priya et al., 2013). Other diagnosis methods of DR include:

Visual acuity testing: This test is concerned with how well a person can focus on objects at different distances.

Ophthalmoscopy and slit lamp exam: This test provides the doctor with a better view of the back of the eye and other structures.

Gonioscopy: It is used to diagnose cataracts and to detect eye damage. A test is administered to detect glaucoma which is a group of eye diseases which can destroy vision and cause blindness.

Tonometry: it used to monitor internal eye pressure. It is used to detect which eye is worse affected by glaucoma.

Finally, there is another way to test called an optical coherence tomography (OCT), this method used to check for fluid in retina.

2.2. Fundus Imaging

One of the other methods for inspecting the retina is the fundus imaging. It uses a specialized low power microscope optical design with built-in camera that is named as ophthalmoscope (or funduscope). The pupil of the eye is used as a point of entry for fundus camera light illumination and imaging rays. This method allows to study and follow changes inside the retina. It takes a moment for the ophthalmologic to produce a fundus image captured by ophthalmoscope. In this thesis, fundus images are analyzed for detecting the severity level of DR. An example illustration showing the steps of fundus imaging is shown in figure 2.2..

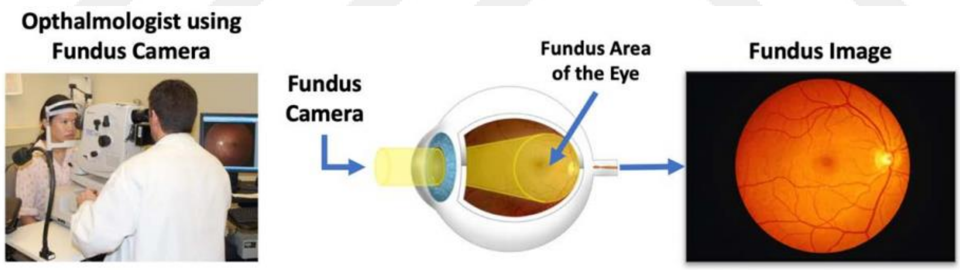


Figure 2.2. Fundus image for the eye by using Fundus imaging (Yusuf, 2019)

3. RELATED WORK

For the day-to-day follow-up and detection of diabetics, the medical experts tend to use conventional methods. These conventional techniques need more time and money than the internet-based systems that are becoming a significant part of human life due to excess technology enhancement.

Deep learning is a branch of machine learning and can be used for medical diagnosis purposes by analyzing the data collected from the patient. Therefore, fundus images can be classified according to their severity levels using deep learning methods.

On the other hand, the systems, in which the data is transmitted over internet and processed on the distant cloud computers, are becoming more widespread. As a result, the internet data traffic increases and its associated bandwidth need to be utilized efficiently. This can be accomplished by preprocessing the images and reducing the amount of total data before transmitting it. Such preprocessing steps are called as edge computing and are important part of IoT systems.

In the following subsections, the previous studies for detection of DR from fundus images and related deep learning and edge computing applications are provided.

3.1. Conventional DR Approach

In the conventional DR approach, the researchers have used the morphological features to develop detection systems. The retina vessels are used for the feature shape extraction. The morphological channel is utilized for observing the retinal vessels and their morphological changes. This method is time consuming and enhancement possibilities are very small. The study proposed by Patil includes the versatile thresholding method to exclude identification and dispenses with historical rarities from the retina construction (Patil, 2013). This method did not cover all the DR features towards the successful recognition system development.

The research progress in enhancement towards the preprocessing and feature extraction of the segmented image has been made. The watershed segmentation techniques are used to extract the optic circle from the retinal image. The shifting and differentiation operation improves the classification performance. The framework developed distinguishes the optic circle vicinity which is the area where many elements useful for a successful detection present (Hoover & Goldbaum, 2003) (Reza, Eswaran, & Dimyati, 2011).

A functioning form model, model-based way to deal with the vascular components established in the optic plate area, was proposed by (Li & Chutatape, 2004). Various statistical measures were utilized to discover the focal marginal region related with DR. The performance of DR detection using ANN achieved veins, hemorrhages-based accuracy at 93.1%, and 73.8% of precision (H. Li & Chutatape, 2004).

The research proposed on detection of different stages of DR includes analysis of 124 retinal images and four distinct classes were identified by using a three-layer feedforward neural network (Yun et al., 2008). The features were extracted from each image using the histogram equalization and morphological operations were used. by using Morphological features includes perimeters of red, green and blue layers. The proposed method achieved more than 90% sensitivity with the specificity equal to 100%.

Generally, the traditional approach can be developed for DR detection and classification as shown in figure 3.1. Firstly, the image data is collected and preprocessed to remove possible noise. Then the important regions of the image are segmented for a better feature extraction in the next step. The features representing the classes are extracted manually and then a classifier model is trained using these features.

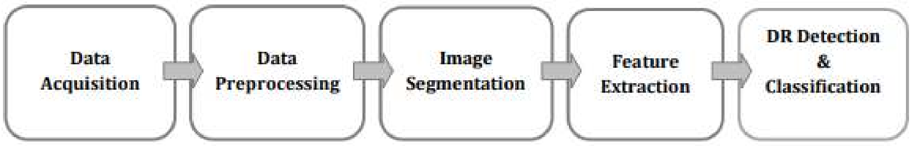


Figure 3.1. Conventional DR detection and classification process

3.2. Deep learning for DR Classification

Deep learning is an artificial intelligence (AI) technique that has various application areas including computer vision, speech recognition, and image classification (Yun et al., 2008).

Deep learning has come to be a lively instrument that goes about as a part of AI and the best technique to apply AI in many fields. The deep learning models have number levels of processing units consisting of a multilayered structure. Prevalently, in medical image analysis and classification, deep learning has various applications for performing diagnosis including DR detection and its corresponding severity level (Shankar, Zhang, Liu, Wu, & Chen, 2020).

In the field of computer-aided diagnostic tasks, the deep learning plays an important role since it includes convenient steps for complex tasks like image classification, detection, and segmentation. In particular, convolutional neural networks (CNN) makes it possible to accomplish such tasks by the convolutional layers in its structure. The major advantage of these layers is enabling the automated extraction of the features that can be used directly by the conventional machine learning methods.

There are various types and structures of CNN for detection and classification of DR. In an earlier work, some pre-trained networks were used in a grading system for DR, which is capable of classifying images based-on the disease pathologies. For binary classification, different architectures evaluated to determine the best performance. Next, the networks were evaluated for a multi-class problem. To increase effective sample size for the dataset and improve the test accuracy, various data preprocessing and data augmentation methods were applied. The result for GoogLeNet and AlexNet achieved the test accuracies of 74.5%, 68.8%, respectively (Lam, Yi, Guo, & Lindsey, 2018).

Other examples of the benchmark CNN architectures is Residual Networks (ResNet) (He, Zhang, Ren, & Sun, 2016) and, Inception-v3 (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016) that can be utilized as pretrained networks for applying transfer learning. On the other hand, it is also possible to build and train a CNN model from scratch for image classification tasks.

In a recent study for detecting severity level of DR in fundus images, a custom CNN architecture was used together with pretrained models of three benchmark architectures namely, AlexNet (Krizhevsky, Sutskever, & Hinton, 2012), SqueezeNet (Iandola et al., 2016) and VGG16 (Simonyan & Zisserman, 2014). The method was applied on 1200 images of the MESSIDOR dataset with four classes (Khan, Abbas, & Rizvi, 2019). The preproccesing steps in that study involved cropping, resizing and the histogram equalization to enhance the images. Eventually, an accuracy value of 98.15% was achieved.

CNN architecture was proposed with suitable amount of pooling layers. The model is claimed to be the modified version of AlexNet and several softmax and rectified linear unit (ReLU) functions are used as the activation functions (Shanthi & Sabeenian, 2019). Messidor database was used to validate performance. The result for this study obtained the accuracy values of 96.2%, 95.6% and 96.6% for three stages have been achieved.In a more recent study,Jayakumari proposed an algorithm for DR classification by using ImageNet model (Jayakumari, Lavanya, & Sumesh, 2020). The proposed method has been applied on a publicly available dataset. The model achieved high accuracy in DR classification, where the training accuracy was 98.6% and the No DR, mild, moderate, severe and proliferative DR achieved accuracy for each class 86.6%, 62.5%, 66.6%, 57.1%, 42.8% respectively.

Another method based on CNN was proposed by (Gulshan et al., 2016) for diabetic retinopathy detection. It was tested on two different datasets and achieved sensitivity and specificity values of 90.3% and 98.1% with EyePACS-1 dataset, and

87.0% and 98.5% using Messior-2 dataset, respectively. Also Orlando et al. has proposed a method for red lesion detection for fundus images by combining ensemble deep learning with the domain of knowledge. A CNN architecture was used to extract the features from the fundus images, and they were augmented by hand-crafted features. The Random Forest algorithm was used to classify the true lesion candidates. The result for this method was reported as the highest performance for DIARETDB1 and e-ophtha datasets (Orlando, Prokofyeva, Del Fresno, & Blaschko, 2018).

In general, the processes based on deep learning for classification of DR images begin by collecting the dataset and then pre-processing it. Next, the deep learning algorithms are applied to extract features automatically and perform classification. In case of small amount of data, an augmentation step may be applied optionally. These overall steps are shown in Figure 3.2.

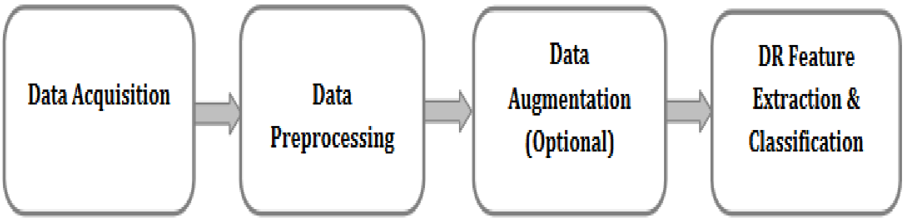


Figure 3.2. Process of Deep learning for DR classification

3.3. Deep Learning with area and IoT

Deep learning has become an important approach for computer vision, speech recognition, and bioengineering. DL is an effective tool for understanding huge amounts of complex data and it has already been applied to many IoT problems and for mobile apps.

The ability to produce the flexible computing power and storage availability, cloud platforms has become more important in application development (Mao, You, Zhang, Huang, & Letaief, 2017). Besides that, edge computing has its benefits such as reduced the latency, extend the edge node, as well as, avoiding the traffic peaks. (Yu et al., 2017). There is scope to combine cloud computing and edge technology towards new paradigm development. Using this combined approach, cloud and edge computing collectively allow for developing methods for tasks like resource management and the resource allocation (Yin, Luo, & Luo, 2018). The application for secure data storage and searching method for improving efficiency and security has been developed as well (Fu, Liu, Chao, Bhargava, & Zhang, 2018).

In recent years, many research activities that provide the importance of the edge computing for healthcare have been performed (Chen, Li, Hao, Qian, & Humar, 2018). Furthermore, the cloud platforms based on the edge have found applications in a variety of domains such as smart grid (G. Xu, Yu, Griffith, Golmie, & Moulema, 2016) and smart transportation systems (Lin et al., 2016). However, applying the various solutions found in the studies mentioned above for base station management is problematic as few consider the complexity associated with scalability, reliability, and responsiveness.

A typical deep learning model has many layers with each layer can be selected for determining specific features. This may allow for reducing the data size, eventually making it possible to be used in edge computing applications.

One of the earlier works utilizing deep learning for edge computing is about food recognition. (Tran, Hajisami, Pandey, & Pompili, 2017). In this experimental study, a mobile phone was used as the edge computing device. The proposed method shows the performance of deep leaning and reduction in response time and energy consumption.

In many medical applications, the mobile devices are widely used as the edge node. As an example, one framework proposed by (Sodhro, Luo, Sangaiah, & Baik, 2019) to improving the QoS (Quality of service), which used mobile edge computing to enable remote video streaming for operations over networks .

Edge computing is very important technology for IoT because the limitations in the network performance cause the cloud computing operations to become inefficient for processing and analyzing huge amount of data. The computing operations performed at the edge of IoT systems allows for efficient transmission of required data to the cloud to achieve an improved QoS.

3.4. Discussion on the Related Work

In this chapter, usage of deep learning in classification of DR disease has been discussed. Many of the studies related to the proposed solutions and the utilized methods were thoroughly examined. It has been understood that usage of systems based on deep learning is becoming more widespread everyday and such methods require high computational power and large datasets.

In order to improve the storage area and the CPU utilization, the deep learning based systems may require cloud computing solutions in IoT systems. The computing operations performed on the cloud side allow to achieve enhanced performance. Therefore, it has been seen that transmission of small amount of data to the cloud servers is key for economic utilization of internet bandwidth. This can be achieved by applying edge computing operations to the input data.

4. MATERIALS AND METHODS

In this chapter, the deep learning algorithms used in this thesis, IoT and edge computing methods, and the utilized dataset are mentioned. The mobile edge computing scheme has been developed for bringing the preprocessing steps closer to user. This eventually reduces the amount of data transmitted to the cloud and decreases the required bandwidth. Other details like the environment and platform used for the research and the performance measure for the system analysis is explained as well.

4.1. IoT techniques

The collection of software and hardware that has links in the physical world over the internet called IoT and its development has gained a lot of interest among researchers in various application areas (Chetoui & Akhloufi, 2020; Gangwar & Ravi, 2020). It is estimated that in 2025, there will be more than 75 billion devices connected to internet (Yusuf, 2019). As a result of this situation, utilization of low-energy systems with limited processing capability and network connection will be more widespread. The huge amount of data generated by the IoT devices can be processed by cloud computers for an efficient data analytics. However, this generates large amount of Internet traffic and eventually, some dedicated methods for efficient usage of bandwidth are required. The more significant number of connected devices may become impractical to only connect to the cloud for processing (Bhardwaj, Jain, & Sood, 2020). As a result, performing the initial data processing at the local devices is important to minimize the data loaded on the internet network. Such processing idea is called as edge computing and a mobile device is used for accomplishment of edge computing steps in this thesis. After discussing deep learning algorithms in the next section, further operations performed on edge and cloud devices are introduced in the next subsections.

4.2. Deep Learning Techniques and CNN

Deep learning has a variety of application areas, such as image classification, computer vision, and, speech recognition (Maier, Syben, Lasser, & Riess, 2019). Based on artificial neural networks (ANN), deep learning can be considered non-linear processing stages for learning patterns using the input features (Fourcade & Khonsari, 2019), where the word "deep" in deep learning denotes the extensive use of multiple layers. The input data passes through each layer where the output from the previous layer is multiplied by weights and passed through a non-linear activation functions. The output of the final layer is either a real number for regression or a discrete class label for classification problems.

A deep neural network has several layers between the input and output, and is typically designed to perform only nonlinear operations by using the matrix multiplications combined with convolution operators. There are many possible design criteria in designing a deep neural network, and this depends on different hyperparameters for the network also the purpose of the designed model (Brinker et al., 2019). The architecture of a generic deep neural network is shown in figure 7.

Most of the theoretical background of CNN derives from the achievements of a French computer scientist in the 1980s, Yann LeCun. In 2010s, usage of CNN became widespread as it allowed for achievement of high performance various pattern recognition applications (Fourcade & Khonsari, 2019).

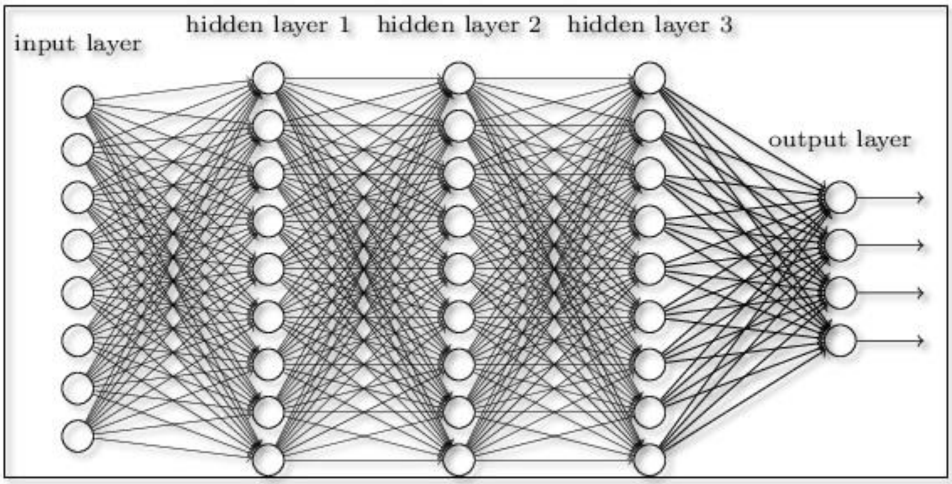


Figure 4.1**.** The architecture of a deep neural network

In the recent years, utilization of CNN has become very popular, especially in medical field such as detection and classification for DR. It can successfully extract features from input images and allow for automated assignment of related class label without needing any hand-crafted features.

CNN is among the most widely-used deep neural network methods today. It uses multilayer perceptron with a number of convolutional layers that extract the feature map of the input image. Typically, the input images are sequentially convolved with number of filters and their sizes are reduced by pooling operations. Output of the filtering process is then flattened into one single long vector which is used as a feature vector for further classification (Figure 4.2.).

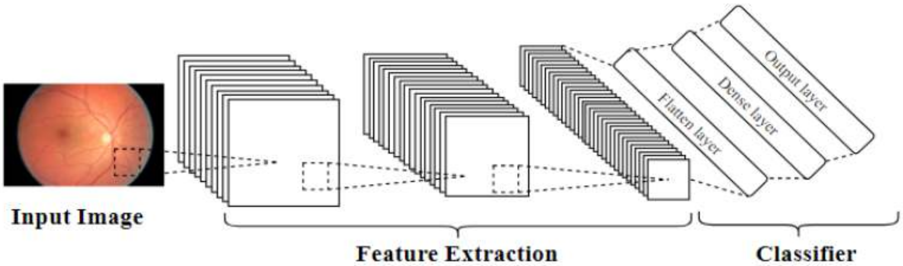


Figure 4.2. CNN model for image classification(Kandel & Castelli, 2020)

Typically, a CNN architecture has three main layers. The first one is the convolutional layers, which is the core building block for automated feature extraction. The second one is the pooling layers consisting of non-linear down sampling of the outputs of the convolutional layers. It reduces the dimensions of the feature maps with different strategies such as average pooling and max pooling. The third one is the fully connected layers to connect all the neurons of the previous layers (Lee, Gallagher, & Tu, 2016) (Figure 4.3). The number of layers, size, and the number of CNN filters may vary according to the problem to be solved with this network.

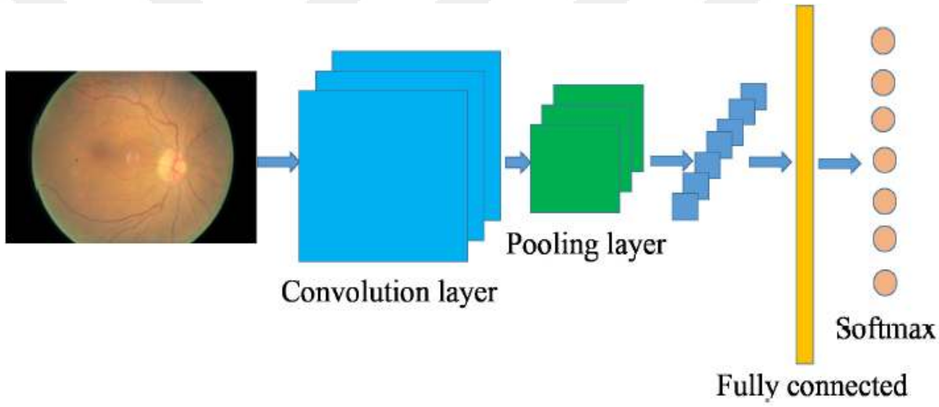


Figure 4.3. The architecture of CNN for image analysis(K. Xu, Feng, & Mi, 2017)

4.2.1. Residual Networks

In 2015, the concept of residual neural network (ResNet) was introduced by He et al (He et al., 2016), and it is an ANN-based on structures inspired from the pyramidal cells in the cerebral cortex. It uses skip connections or shortcuts to leap over specific layers and eliminates the risk of gradient vanishing. The typical ResNet model is implemented with double or triple layers. It contains non-linear ReLU activation functions and batch normalization between the first and second layer.

In this work, the ResNet model with 50 convolutional layers (ResNet50) is utilized. It is built with a feed-forward network with a shortcut connection that increases classification accuracy without the need for a larger and more complex model.

4.2.2. Efficient Net

As the CNN architectures become more complex, the overall performance of the models tends to increase. A group of CNN models known as EfficientNet is one example of this situation. There are 8 different types of EfficientNet models between 0 and 7. As the model number increases, the number of parameters increases very little, and accuracy increases considerably. EfficientNet uses a different activation function called Swish (Tan & Le QV, 1905). Unlike other CNNs, EfficientNet reduces the depth, width, and resolution of the network while reducing model complexity. Generally, it contains seven blocks, where each block may contain extra number of sub-blocks and the number of these blocks increase from EfficientNetB0 to EfficientNetB7. For example, the EfficientNetB0 model has 237 layers and the EfficientNetB7 involves 813 layers. The structure of the EfficientNetB7 model together with its computation sub-blocks is shown in Figure 4.4.

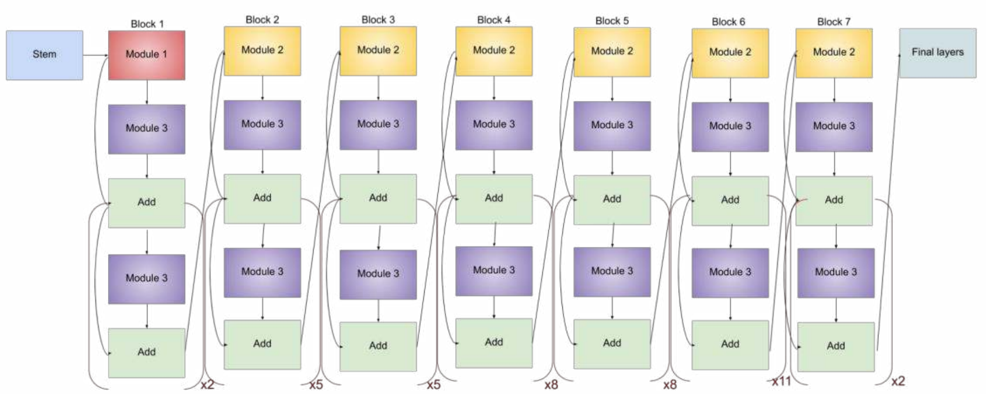


Figure 4.4. EfficientNet B7 model structure (Agarwal, 2020)

4.2.3. VGG Net

For the ImageNet competition in 2014, Visual Geometry Group (VGG) at University of Oxford proposed two CNN architectures, namely VGG16 and VGG19. VGG networks have five convolutional blocks and three dense layers, where each block contains a number of convolutional layers followed by a max pool layer. More specifically, the VGG16 and VGG19 involve total of 13 and 16 convolutional layers, respectively. These convolutional layers are followed by three fully connected layers. To convert the 3D block vector to 1D, to be inserted in fully connected layers, a flattening layer has been added. There are 4,096 neurons in the first two fully connected layers and 1000 neurons in the last fully connected layer. A softmax layer is implemented at the final layer to output the probabilities belonging to particular classes. The total number of parameters are 138M and 144M for VGG16 and VGG19 models where the increment is due to the addition of extra layers in VGG19 (Luo, Yang, Tong, Wu, & Fan, 2017) (Kandel & Castelli, 2020). In this work, VGG19 is used and its architecture is illustrated in in Figure 4.5.

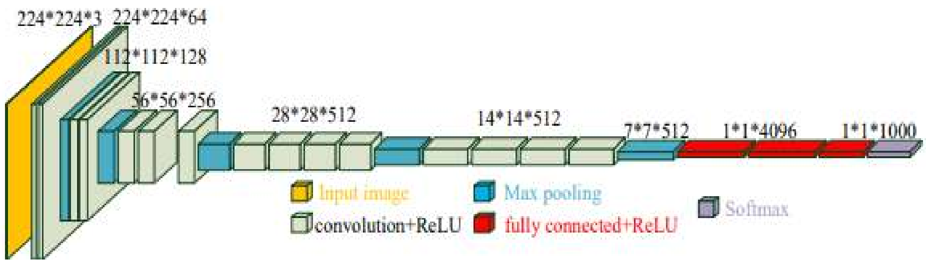


Figure 4.5. The VGG-19 model(Xiao, Wang, Cao, & Li, 2020)

4.2.4. Ensemble Learning

The idea of ensemble learning is to combine several weak learners to obtain a stronger model eventually. There are different strategies to generate ensemble models including boosting, bagging, and stacking (Ditterrich, 1997).

In this thesis, a concatenation ensemble model of three CNN architectures (VGG19, ResNet50, and EfficientNetB7) is generated. The concatenation operation was performed after the feature extraction layers of these models. In other words, the features extracted by these models are concatenated into one single feature vector which is then fed into fully connected dense layers for performing the classification task. This strategy is followed for detecting the severity level of DR images. In order to underline the effectiveness of the ensemble learning, performances of these models are evaluated individually and compared with the performance obtained by the ensemble model.

The overall structure of the proposed ensemble model involves simultaneous extraction of the features from the three CNN architectures. These architectures take the input images with size of 224x224 pixels. The outputs of the convolutional layers are then passed through max pooling and dense layers before the concatenation operation and this creates a feature vector with 128 components. The outputs from these feature extraction submodels are flattened and concatenated into one long vector. After concatenation of these three feature vectors, a vector of 384 components has been generated, and, then passed on to a fully connected layer of interpretation. Next, some two more dense layers and a dropout layer is added to avoid overfitting and increase generalization. The last final dense layer has a softmax activation function makes a multiclass classification with five outputs as showing in Figure 4.6.

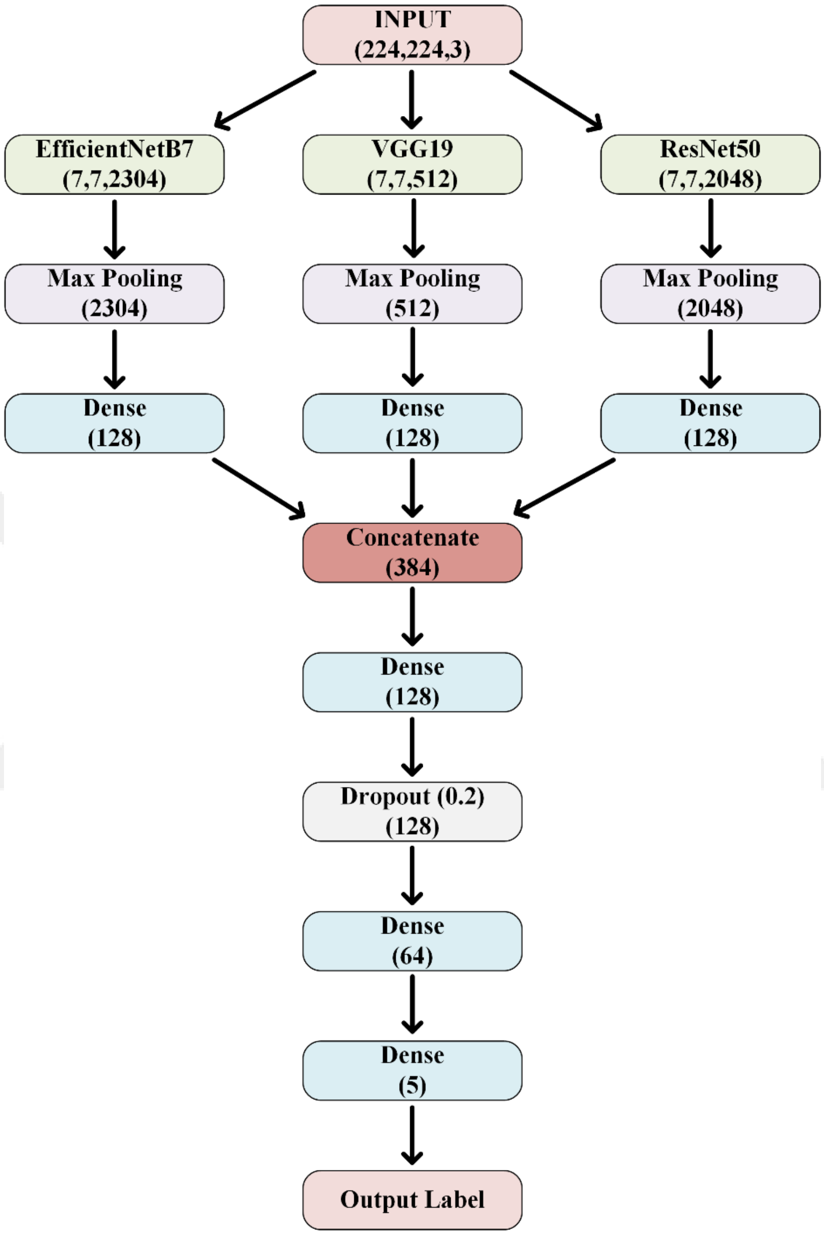


Figure 4.6. The architecture for the proposed ensemble model. The numbers in parenthesis denote the output shape of the corresponding layer

4.3. Dataset Details

There are many publicly available datasets of the retina images to detect various forms of DR. These datasets are used in various ways, including training, validation, testing, and comparing systems. In this study, the fundus image dataset generated by Asia Pacific Tele-Ophthalmology Society (APTOS) generated as a part of the challenge took part in the 4th APTOS Symposium in 2019 (Kaggle, 2021)

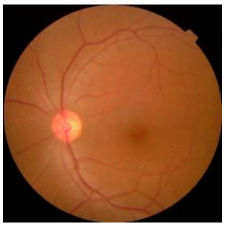
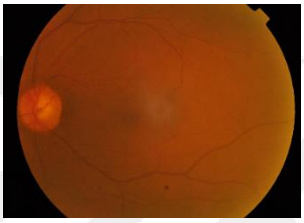
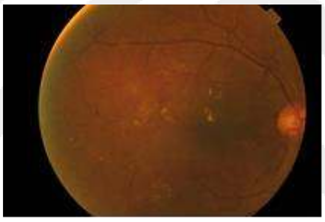
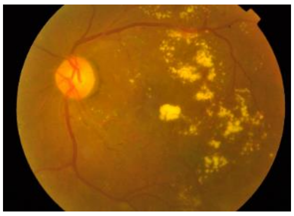
4.3.1. The APTOS 2019 Challenge Dataset

It is a publicly available dataset containing 5590 images where 3662 labelled images are in the training set and the remaining unlabelled images constitute the test set. Since only the images in the training set have the class label information, these 3662 images are used in this thesis study. The images in the dataset have various resolutions from 433x289 pixels to 5184x3456 pixels. Based on the severity level of diabetic retinopathy, the images in the dataset are collected into five classes:

1. No DR (Label 0)
2. Mild DR (Label 1)
3. Moderate DR (Label 2)
4. Severe DR (Label 3)
5. Proliferative DR (Label 4)

To eliminate the imbalance in the number of images within the classes, the data augmentation steps, rotation, horizontal flip, and vertical flip have been applied randomly to the images of the specific DR classes with relatively lower number of images. As a result, the total number of images were increased to 5388. In Table 4.1, names of the DR severity levels and distribution of images for these classes are given together with some sample images. The proportions of the training, validation, and test sets were specified as 75%, 15%, and 15%, respectively.

Table 4.1. Dataset information and sample images

Severity Level	Label No	Sample Image	Number of images in the dataset
No DR	0		1410
Mild DR	1		1288
Moderate DR	2		874
Severe DR	3		720
Proliferative DR	4		1096

4.4. Preprocessing and Data Augmentations

The training and validation of the proposed model are done in the preprocessed form of the images in the dataset. Firstly, some noisy images are excluded from the dataset and then data augmentation is applied. Next, the image preprocessing steps, cropping, unsharp masking, and resizing have been applied to images before being fed into the deep learning algorithm.

4.4.1. Noisy image exclusion and data augmentation

All of the 3662 images in the dataset were manually examined to detect and exclude noisy and blurred ones. With this step, 600 noisy images were excluded (Sikder, Chowdhury, Arif, & Nahid, 2019). Then, rotation, horizontal flip, and vertical flip have been applied as the data augmentation to the classes with relatively small number of images. In particular, these classes are Mild DR, Severe DR, and Proliferative DR. At the end of the data augmentation, a total of 5388 images were obtained. The class-bases details of the dataset after exclusion and augmentation are provided in Table 4.2, and figure 4.7. contains an example of data augmentation operations.

Table 4.2. Details of the dataset after exclusion and augmentation operations

Class	Number of image	Exclusion result	Augmentation result
0	1805	1410	1410
1	370	322	1288
2	999	874	874
3	193	180	720
4	295	274	1096
TOTAL	3662	3060	5388

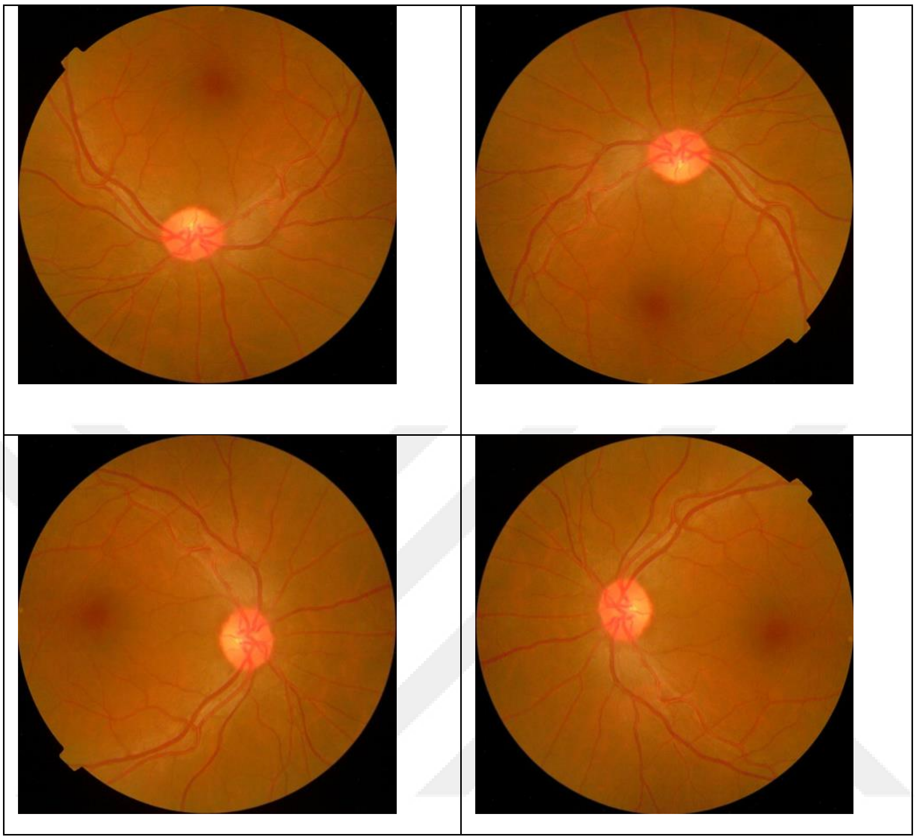


Figure 4.7. An example for data augmentation for sample image in the dataset

4.4.2. Image preprocessing

The preprocessing operations include image cropping, unsharp masking and resizing. Some of the images contain uninformative dark areas around the image of the eye. So removal of these regions is important for reducing the data size as well as keeping the informative region of interest relatively larger after resizing the image. Application of unsharp masking allows more details to become visible and eventually yield a better classification performance. Finally, the images are resized to the dimensions that are appropriate for the CNN models and this operation reduces the file size of the images considerably. The outputs of each preprocessing step have been given in Figure 4.8.

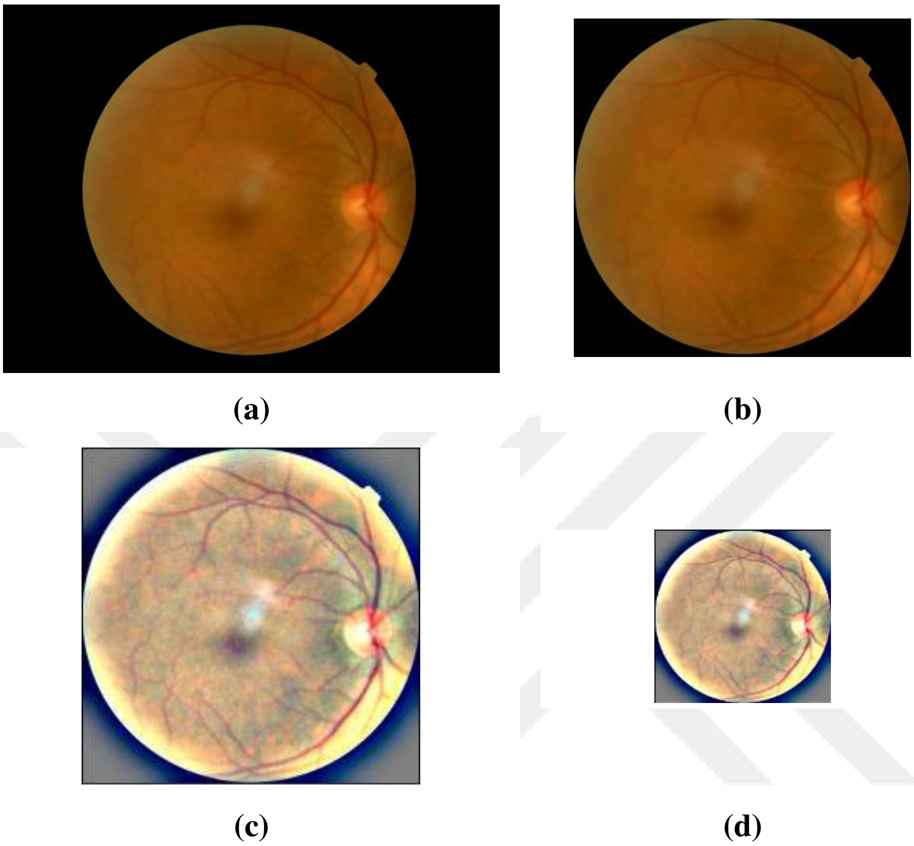


Figure 4.8. Preprocessing steps performed on the edge side . (a) A sample image (819 x 614) (b) removed the black regions around the image. (c) Unsharp masking. (d) image resized (224 x 224)

4.5. Operations on the Cloud

The operations on the cloud side have been performed using Google's Collaborator (Google Colab) platform. This platform provides a virtual environment to the users for the experimenting their algorithms. Since the services offered by Google Colab uses hardware with high computation capacity, it is a convenient tool for development of clout-based deep learning systems.

The experiments were carried out using two approaches: first, the deep learning without edge computing, and the second one is deep learning with edge computing on the mobile device. A smartphone emulation was developed for performing the preprocessing operations as the edge computing. Then the classification is done collaboratively between the edge device and the cloud servers. In the proposed system, data and initial computation are kept at the network's edge, near the users.

4.5.1. Training on the Cloud

Training of the proposed ensemble learning network was performed with the multi-label DR dataset containing five severity levels. In the multi-label classification, the 5388 images were split into training, validation and test sets with the ratios of 75%, 15%, and 15%, respectively. The hyperparameters settings were kept constant for the multi-label classification method. The proposed network has more than 7 million total parameters, which were initialized randomly at the beginning of the training process. The network was trained with Adam optimizer method (Kingma & Adam, 2018) for a fixed schedule over 30 epochs with early stopping setting in which the training is stopped if no improvement in the training accuracy is observed for a predefined number of epochs. The threshold for early stopping is set as four epochs. Several values for the learning rate was tried and it was found out that a learning rate of 0.001 achieves the best performance. The batch size for training was set as 64 and categorical crossentropy was used as the loss function to be used for parameter update between the epochs. With these settings, five CNN models were trained. Three of these models are single benchmark CNN architectures, namely, EfficientNetB7, VGG19, and ResNet50. The other two models are ensemble models where average and concatenation ensembles of the three benchmark models are generated.

4.6. Operations on the Edge

In the proposed framework, camera of the mobile device can be used as the image acquisition tool. However, since the dataset is readily available in this study, the mobile device has been utilized as an edge computing device and a gateway for transmitting the data to the cloud server over internet. This system consists of an end-user layer, an edge-layer, and the cloud layer which together form a three-layered delivery system (Figure 4.9). The edge and cloud layers involve some computational operations that are elaborated elsewhere. On the other hand, the end-user layer consists of a smartphone app with a graphical user interface (GUI) enabling user interaction (Figure 4.10). It has a simple design for selecting the image, performing preprocessing operations, and sending the image to the cloud server for assigning a class label.

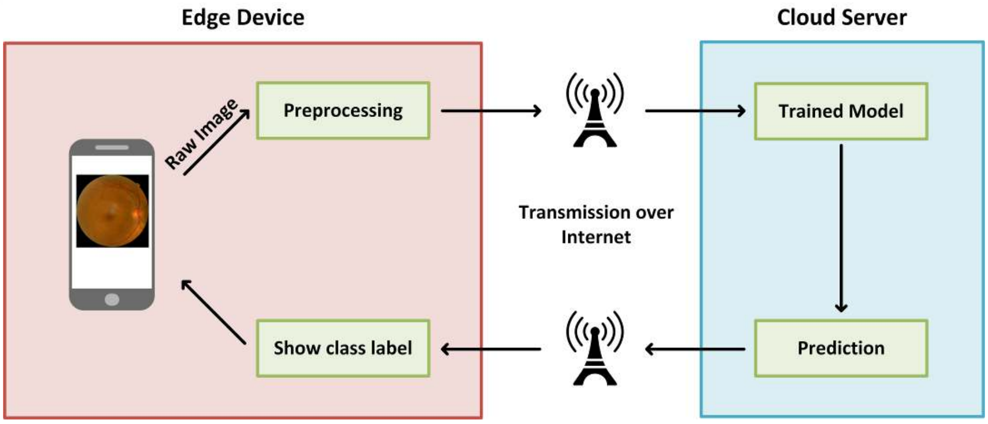


Figure 4.9. Deep learning with the Edge of IoT

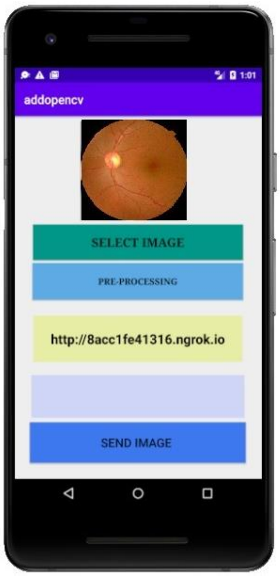


Figure 4.10. The screenshot of the Android application for edge processing

After installing the application, the GUI screen has three buttons and a single thumbnail of a fundus image. The functions of the buttons are explained as follows:

- *Select Image*: Responsible for taking an image or loading an image from the gallery. This image is then displayed on image view.
- *Preprocessing button*: This button is used to make preprocessing for the image which selected by *Select Image* button. In this step we three preprocessing steps applied firstly image cropping then followed by resizing finally applying unsharp masking.
- *Upload button*: sends the preprocessing image to the cloud server, returns the classification results and displays them in result space.

In development of the smartphone app, the android studio was used. The GUI was designed from scratch using the Java programming language with android studio platform. Android studio is a well suited software with almost all of the android devices. Okhttp3, an HTTP+HTTP/2 client package for Android and Java applications, is used for generating HTTP requests to send the data from edge to the cloud server and vice-versa. Finally, a Flask framework server was developed to handle requests and responses, which consist of the saved model to make prediction at the server. Using Flask framework with Python programming language, an API is created to receive the POST requests on the HTTP server. The received messages are dispatched by Okhttp3 client in JSON format which is then displayed on the GUI screen of the app. The performance of the system is calculated based on the overall response time.

5. RESULTS AND DISCUSSION

In this chapter, the obtained results are presented and it consists of two parts: (*i*) classification performance of the deep learning models that is performed on the cloud side, and (*ii*) effects of edge computing on the system response time and the total amount of transmitted data.

5.1. Deep Learning Classification Results

The deep learning models were evaluated by generating a multi-class confusion matrix using the test set images. The confusion matrix was then used to calculate four performance metrics that are accuracy, precision, recall, and f1-score. Also, in order to keep track of the learning performance of the model, accuracy and loss curves were plotted observed against a possible overfitting.

5.1.1. The confusion matrix

A confusion matrix is used to measure the performance of a classification model. For each of the trained models, a corresponding confusion matrix is generated and the performance metrics are calculated by referencing to each class separately. Since the highest classification accuracy is achieved with the proposed concatenation ensemble, its corresponding confusion matrix is given in Table 5.1. Each element of the CM to show the comparison between the correct label and predicted label for all images in the test set.

As can be seen from the Table 3 that the concatenation ensemble model achieved the best result for No DR, Moderate DR, and Severe DR by making a correct prediction for 212 images out of 212 images, 131 images out of 132 images with one image is missing in this class., and, 108 images out of 108 images, respectively. Whereas, the correct predicted images for Mild DR and Proliferative DR were 171, and 150 images out of 194, and, 166 images, respectively. The percentage values of these predictions are also provided in Figure 5.1.

Table 5.1. Confusion Matrix for concatenation ensemble model

		Predicted Label				
		No DR	Mild	Moderate	Severe	Proliferative
Actual Label	No DR	212	0	0	0	0
	Mild	0	171	0	0	23
	Moderate	0	1	131	0	0
	Severe	0	0	0	108	0
	Proliferative	0	14	0	2	150

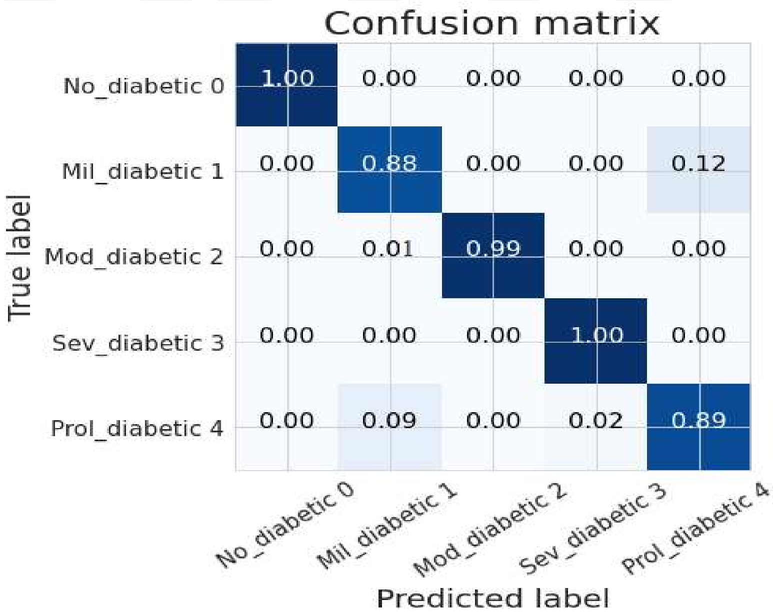


Figure 5.1. Confusion matrix

5.1.2. Loss and Accuracy Analysis

Loss and accuracy values for each of the training epochs are recorded and plotted during the analysis (Figure 5.2). These plots are generated both for training and validation sets and they provide useful information for about the success of the model training. The achieved minimum training and validation loss values are 0.0000 and 0.1237, respectively. On the other hand, the maximum corresponding accuracy values for these sets are observed to be 0.9999 and 0.9759.

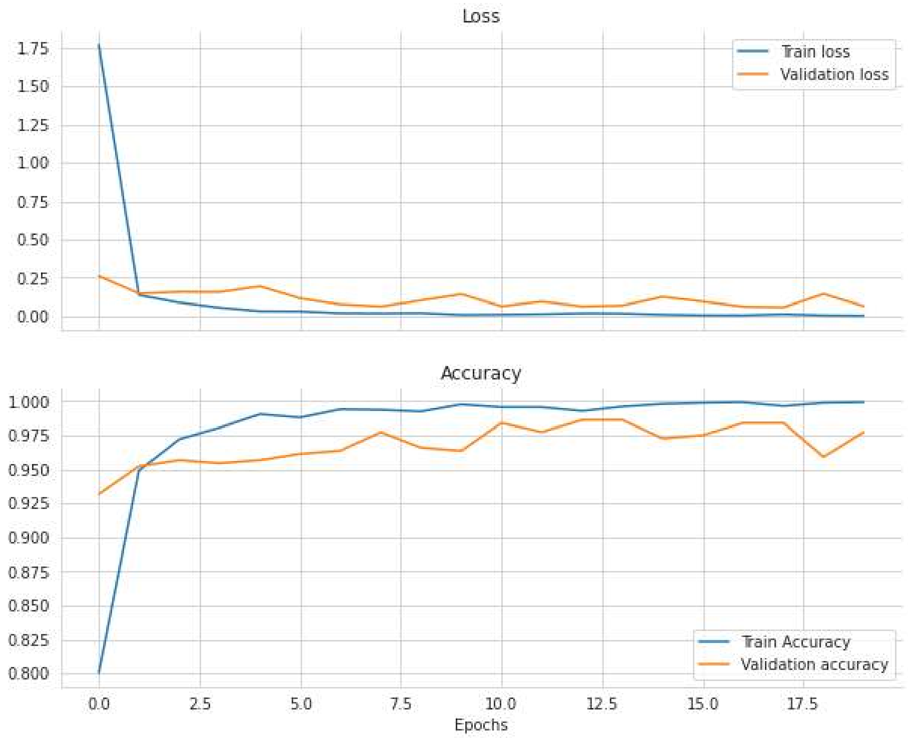


Figure 5.2. Learning curves for training and validation sets. Loss plot (top) and the accuracy plot (bottom)

5.1.3. Classification Performance Evaluation

Training and classification of unseen test samples are all performed at the cloud side of the framework. To better understand the performance of ensemble models, performances of three individual benchmark CNN classifiers (EfficientNetB7, ResNet50, and VGG19) as well as two ensemble models (average and concatenation) are calculated and their respective performances are reported (Table 5.2). Since this is a multi-class classification problem, values of the metrics like precision, recall, and f1-score may vary according to the reference class. Therefore, detailed results for all of the reference classes and for all models are provided.

Table 5.2. Classification performance of the models

		EfficientNet B7	ResNet50	VGG19	Average Ensemble	Concatenation Ensemble
0	**Train Acc.**	0.91	0.90	0.87	0.89	**0.97**
	Test Acc.	0.87	0.84	0.83	0.85	**0.96**
	Precision	0.90	0.77	0.77	0.76	**1.00**
	Recall	0.88	0.85	0.85	0.85	**1.00**
	F-Score	0.89	0.81	0.81	0.79	**1.00**
1	**Train Acc.**	0.91	0.90	0.87	0.89	**0.97**
	Test Acc.	0.87	0.84	0.83	0.85	**0.96**
	Precision	0.55	0.65	0.65	0.68	**0.92**
	Recall	0.52	0.52	0.52	0.60	**0.88**
	F-Score	0.54	0.58	0.58	0.64	**0.90**
2	**Train Acc.**	0.91	0.90	0.87	0.89	**0.97**
	Test Acc.	0.87	0.84	0.83	0.85	**0.96**
	Precision	0.85	0.73	0.73	0.74	**1.00**
	Recall	**0.99**	**0.99**	**0.99**	0.98	**0.99**
	F-Score	0.80	0.88	0.84	0.85	**0.99**
3	**Train Acc.**	0.91	0.90	0.87	0.89	**0.97**
	Test Acc.	0.87	0.84	0.83	0.85	**0.96**
	Precision	0.80	0.73	0.73	0.73	**0.98**
	Recall	0.74	0.74	0.76	0.76	**1.00**
	F-Score	0.77	0.72	0.72	0.74	**0.98**
4	**Train Acc.**	0.91	0.90	0.87	0.89	**0.97**
	Test Acc.	0.87	0.84	0.83	0.85	**0.96**
	Precision	0.79	0.72	0.75	0.73	**0.86**
	Recall	0.75	0.72	0.71	0.66	**0.89**
	F-Score	0.77	0.74	0.74	0.70	**0.88**

Since the concatenation ensemble model has the highest scores for all of the measures, it is understood that this model is suitable for detecting the severity level of DR images. On the other hand, usage of averaging operation may not be suitable for the ensemble model all the time because the accuracy of the average ensemble is smaller than EfficientNetB7 while being better than ResNet50 and VGG19.

Furthermore, images of Mild DR and Proliferative DR classes seem to be the most difficult to classify as their related precision, recall, and f-score values are relatively low. In other words, the deep learning models cannot accurately distinguish the samples from these classes. This situation is supported by the results provided in the confusion matrix given in Table 5.1.

5.1.4. Comparison of Our Results with Other Studies

To underline the effectiveness of the concatenation ensemble model, the obtained results are compared with those reported in other related studies (Table 5.3). The accuracies and recall for the previously reported studies vary between approximately 37% to 94%t and 30 %to 90%, respectively. The proposed method for multi-class DR classification has performed better than previously systems in terms of both accuracy and recall.

Table 5.3. The comparison of the current study with other related previous work

Author	Year	Classifier	No. of Classes	Accuracy	Recall
(Noronha, Acharya, Nayak, Kamath, & Bhandary, 2013)	2011	SVM	5	82%	82.50%
(Pratt, Coenen, Broadbent, Harding, & Zheng, 2016)	2016	CNN	5	75%	30%
(Sajana, Krishna, Dinakar, & Rajdeep, 2019)	2016	CNN	5	94%	76.6%
(Sathiya & Gayathri, 2014)	2016	GoogleNet	5	45%	-
(Wang, Lu, Wang, & Chen, 2018)	2018	AlexNet VGG16 InceptionNet V3	5	37% 50% 63.23%	- - -
(Chaturvedi, Gupta, Ninawe, & Prasad, 2020)	2020	DenseNet	5	94.44%	87%
Proposed method	2021	Concatenation ensemble	5	97%	95%

5.2. Effects of the Edge Computing

The overall performance of the experiment is calculated based on time response. After installing the application on android, the application it will be ready to test. The total time from the server (T_{total}) was calculated by adding the time required for transferring data to the could ($T_{Required}$), the time spent at the cloud for processing (T_{Cloud}), and the preprocessing time at the edge device (T_{Edge}).

$$T_{total} = T_{Required} + T_{Cloud} + T_{Edge}$$

Total duration of the execution on the cloud and edge sides may easily be determined through coding. Therefore, subtracting $T_{Cloud} + T_{Edge}$ from T_{Total} will result in $T_{Required}$ whose knowledge allows for making conclusions about the occupied bandwidth. It should also be noted that, $T_{Edge} = 0$ when the preprocessing is performed at the cloud side.

The effects of these steps were quantified by measuring the overall file size of the images in the set and the times taken for each image in the test. In this study, two cases are examined in detail. In the first case, the both the preprocessing and the classification are done in the cloud side. In the second case the preprocessing is performed at the edge layer. For both of these cases, response time from the cloud server is calculated for all predictions in the test set.

Some example results for six different images in the test set are given in Tables 5.4 and 5.5, where the “File size” column denote the size of the image file transmitted to the cloud.

Table 5.4. Calculated results for the first case (preprocessing performed at the cloud side)

Image no	File size (KB)	Image Width (Pixels)	Image Height (Pixels)	$T_{requird}$ (sec)	$T_{cloud} + T_{Edge}$ (sec)	T_{total} (sec)
1	4000	2588	1958	8.00	13.70	21.70
2	3405	1050	1050	6.81	11.50	18.31
3	2200	3216	2136	4.40	8.30	12.70
4	2030	2048	1536	4.06	8.10	12.16
5	4132	3388	2588	8.26	14.01	22.27
6	4602	1050	1050	9.20	14.89	24.09

Table 5.5. Calculated results for the second case (preprocessing performed at the edge device)

Image no	File size (KB)	Image Width (Pixels)	Image Height (Pixels)	$T_{requird}$(sec)	$T_{cloud} + T_{Edge}$ (sec)	T_{total} (sec)
1	640	224	224	1.28	12.33	13.61
2	520	224	224	1.04	10.64	11.68
3	404	224	224	0.81	7.35	8.16
4	410	224	224	0.82	7.57	8.39
5	553	224	224	1.11	11.71	12.82
6	598	224	224	1.20	11.20	12.40

The results in the tables show that the edge computing contributes to both total size of the transmitted images and the total transmission time, $T_{Required}$. Although there is not a significant improvement in the total processing time ($T_{Cloud} + T_{Edge}$), the faster response of the cloud servers is achieved through the reduction in $T_{Required}$. Furthermore, since the total amount of transmitted data is reduced, the network bandwidth has been utilized more efficiently.

6. CONCLUSION

In the current generation of technology, the Internet becomes a necessary part of human life. Medical science has used the Internet for fast processing and large scale data storage. The paradigm of IoT is primarily based on computational devices to produce a large scale and high quantity of data. Processing and producing data with the minute time fraction is a challenging task. The large-scale dataset consists of images, video, audio, and textual content form.

The automatic aspect of IoT-based devices improves response time based on a machine learning algorithm due to preprocessing the file earlier than importing it to the server. The current technical era is witnessing explosive usability of smart computing devices such as smartphones.

Diabetic retinopathy is a common issue in medical science. The result extraction and analysis of diabetic retinopathy need time using traditional medical devices. Most of the times, the cost and time required for diagnosis of this disease are too much. Hence, proposed research can be used in this field to provide the necessary solutions. Furthermore, utilization of deep learning with IoT in the context of medical image analysis and diagnosis has become widespread in the recent years.

In this thesis, a deep learning based framework to detect the severity level diabetic retinopathy is proposed. The framework also features edge computing capability to reduce the system response time and save bandwidth.

The proposed framework consists of three main steps that are preprocessing, feature extraction, and classification. All the experiments were performed using the APTOS 2019 dataset containing DR images of five different severity stages. The experiments included utilization of three benchmark CNN models (EfficientNetB7, VGG19, ResNet50) and their hybrid combinations. The performance of the proposed ensemble model is calculated based on accuracy, precision, recall, and f1-score measures. Precision refers to the quality of the experiment, whereas recall refers to completeness or quantity. High precision is always an excellent indicator of the successful classification of the positive classes. It was shown that the performance of the concatenation ensemble model outperforms the other four models. Therefore, it is possible to conclude that this is a suitable approach for detecting the severity level of the diabetic retinopathy from the fundus images.

The other feature of the framework is the edge computing that allows for preprocessing of the medical images before being transmitted to the cloud server. It was shown that the edge computing process reduces the time to transmit the data and response from the server. In other words, reducing the total amount of data through edge computing reduces response time of the server. This preprocessed data allows for making predictions directly without loss of time for preprocessing at the cloud side.

As future work, performance analysis of bagging and boosting methods can be compared with those obtained by the ensemble models. In addition, there are much more CNN architectures in the literature, and various combinations of them can be trained to see if there are any other models performing better. Furthermore, the parameter optimization of the models can be performed with the help of nature-inspired optimization methods such as particle swarm optimization or genetic algorithm. In that case, an optimal set of parameters with a suitable model structure can be achieved.

REFERENCES

Agarwal, A. (2020). EfficientNet B7 model Retrieved from https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142 Accessed on Thur 11 , 2021

Bhardwaj, C., Jain, S., & Sood, M. (2020). Diabetic retinopathy severity grading employing quadrant-based Inception-V3 convolution neural network architecture. *International Journal of Imaging Systems and Technology*.

Brinker, T. J., Hekler, A., Enk, A. H., Klode, J., Hauschild, A., Berking, C., . . . Fröhling, S. (2019). A convolutional neural network trained with dermoscopic images performed on par with 145 dermatologists in a clinical melanoma image classification task. *European Journal of Cancer, 111*, 148-154.

Chaturvedi, S. S., Gupta, K., Ninawe, V., & Prasad, P. S. (2020). Automated diabetic retinopathy grading using deep convolutional neural network. *arXiv preprint arXiv:2004.06334*.

Chen, M., Li, W., Hao, Y., Qian, Y., & Humar, I. (2018). Edge cognitive computing based smart healthcare system. *Future Generation Computer Systems, 86*, 403-411.

Chetoui, M., & Akhloufi, M. A. (2020). *Explainable Diabetic Retinopathy using EfficientNET.* Paper presented at the 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC).

Cheung, C. Y., Xu, D., Cheng, C.-Y., Sabanayagam, C., Tham, Y.-C., Yu, M., . . . Mitchell, P. (2020). A deep-learning system for the assessment of cardiovascular disease risk via the measurement of retinal-vessel calibre. *Nature Biomedical Engineering*, 1-11.

Ditterrich, T. (1997). Machine learning research: four current direction. *Artificial intelligence magzine, 4*, 97-136.

Fourcade, A., & Khonsari, R. (2019). Deep learning in medical image analysis: A third eye for doctors. *Journal of stomatology, oral and maxillofacial surgery, 120*(4), 279-288.

Fu, J.-S., Liu, Y., Chao, H.-C., Bhargava, B. K., & Zhang, Z.-J. (2018). Secure data storage and searching for industrial IoT by integrating fog computing and cloud computing. *IEEE Transactions on Industrial Informatics, 14*(10), 4519-4528.

Gagliano, M., Van Pham, J., Tang, B., Kashif, H., & Ban, J. Applications of Machine Learning in Medical Diagnosis.

Gangwar, A. K., & Ravi, V. (2020). Diabetic retinopathy detection using transfer learning and deep learning. In *Evolution in Computational Intelligence* (pp. 679-689): Springer.

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., . . . Cuadros, J. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama, 316*(22), 2402-2410.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Hoover, A., & Goldbaum, M. (2003). Locating the optic nerve in a retinal image using the fuzzy convergence of the blood vessels. *IEEE transactions on medical imaging, 22*(8), 951-958.

Huang, W., & Li, L. (2020). The Chronic Kidney Disease and Abnormal Retinal Blood Vessels. In *Integrative Ophthalmology* (pp. 159-172): Springer.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.

Jayakumari, C., Lavanya, V., & Sumesh, E. P. (2020). Automated Diabetic Retinopathy Detection and classification using ImageNet Convolution Neural Network using Fundus Images. *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, 577-582.

Kaggle. (2021). Retrieved from https://www.kaggle.com/c/aptos2019-blindness-detection/data. Accessed on Feb 23, 2021.

Kandel, I., & Castelli, M. (2020). Transfer Learning with Convolutional Neural Networks for Diabetic Retinopathy Image Classification. A Review. *Applied Sciences, 10*(6), 2021.

Khan, S. H., Abbas, Z., & Rizvi, S. D. (2019). *Classification of diabetic retinopathy images based on customised CNN architecture.* Paper presented at the 2019 Amity International Conference on Artificial Intelligence (AICAI).

Kingma, D. P., & Adam, B. J. (2018). a method for stochastic optimization. 2014. *arXiv preprint arXiv:1412.6980, 9*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems, 25*, 1097-1105.

Lam, C., Yi, D., Guo, M., & Lindsey, T. (2018). Automated detection of diabetic retinopathy using deep learning. *AMIA summits on translational science proceedings, 2018*, 147.

Lee, C.-Y., Gallagher, P. W., & Tu, Z. (2016). *Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree.* Paper presented at the Artificial intelligence and statistics.

Li, H., & Chutatape, O. (2004). Automated feature extraction in color retinal images by a model based approach. *IEEE Transactions on biomedical engineering, 51*(2), 246-254.

Lin, J., Yu, W., Yang, X., Yang, Q., Fu, X., & Zhao, W. (2016). A real-time en-route route guidance decision scheme for transportation-based cyberphysical systems. *IEEE Transactions on Vehicular Technology, 66*(3), 2551-2566.

Luo, H., Yang, Y., Tong, B., Wu, F., & Fan, B. (2017). Traffic sign recognition using a multi-task convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems, 19*(4), 1100-1111.

Maier, A., Syben, C., Lasser, T., & Riess, C. (2019). A gentle introduction to deep learning in medical image processing. *Zeitschrift für Medizinische Physik, 29*(2), 86-101.

Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials, 19*(4), 2322-2358.

Noronha, K., Acharya, U. R., Nayak, K. P., Kamath, S., & Bhandary, S. V. (2013). Decision support system for diabetic retinopathy using discrete wavelet transform. *Proc Inst Mech Eng H, 227*(3), 251-261. doi:10.1177/0954411912470240

Orlando, J. I., Prokofyeva, E., Del Fresno, M., & Blaschko, M. B. (2018). An ensemble deep learning based approach for red lesion detection in fundus images. *Computer methods and programs in biomedicine, 153*, 115-127.

Patil, C. M. (2013). An approach for the detection of vascular abnormalities in diabetic retinopathy. *Int. J. Data Min. Tech. Appl, 2*, 246-250.

Pratt, H., Coenen, F., Broadbent, D. M., Harding, S. P., & Zheng, Y. (2016). Convolutional neural networks for diabetic retinopathy. *Procedia computer science, 90*, 200-205.

Priya, P. V., Srinivasarao, A., & Sharma, J. (2013). Diabetic Retinopathy-Can Lead To Complete Blindness. *Int. J. Sci. Invent. Today, 2*(4), 254-265.

Reza, A. W., Eswaran, C., & Dimyati, K. (2011). Diagnosis of diabetic retinopathy: automatic extraction of optic disc and exudates from retinal images using marker-controlled watershed transformation. *Journal of medical systems, 35*(6), 1491-1501.

Sajana, T., Krishna, K. S., Dinakar, G., & Rajdeep, H. (2019). Classifying diabetic retinopathy using deep learning architecture. *Int. J. Innov. Technol. Explor. Eng, 8*(6), 1273-1277.

Sathiya, G., & Gayathri, P. (2014). Automated detection of diabetic retinopathy using GLCM. *Int. J. Appl. Eng. Res, 9*(22), 7019-7027.

Shankar, K., Zhang, Y., Liu, Y., Wu, L., & Chen, C.-H. (2020). Hyperparameter tuning deep learning for diabetic retinopathy fundus image classification. *IEEE Access, 8*, 118164-118173.

Shanthi, T., & Sabeenian, R. (2019). Modified Alexnet architecture for classification of diabetic retinopathy images. *Computers & Electrical Engineering, 76*, 56-64.

Sharma, P., & Saxena, K. (2017). Application of fuzzy logic and genetic algorithm in heart disease risk level prediction. *International Journal of System Assurance Engineering and Management, 8*(2), 1109-1125.

Sikder, N., Chowdhury, M. S., Arif, A. S. M., & Nahid, A.-A. (2019). *Early blindness detection based on retinal images using ensemble learning.* Paper presented at the 2019 22nd International Conference on Computer and Information Technology (ICCIT).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sodhro, A. H., Luo, Z., Sangaiah, A. K., & Baik, S. W. (2019). Mobile edge computing based QoS optimization in medical healthcare applications. *International Journal of Information Management, 45*, 308-318.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). *Rethinking the inception architecture for computer vision.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Tan, M., & Le QV, E. (1905). rethinking model scaling for convolutional neural networks. 2019. In.

Tran, T. X., Hajisami, A., Pandey, P., & Pompili, D. (2017). Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges. *IEEE Communications Magazine, 55*(4), 54-61.

Wang, X., Lu, Y., Wang, Y., & Chen, W.-B. (2018). *Diabetic retinopathy stage classification using convolutional neural networks.* Paper presented at the 2018 IEEE International Conference on Information Reuse and Integration (IRI).

Wilkinson, C., Ferris III, F. L., Klein, R. E., Lee, P. P., Agardh, C. D., Davis, M., . . . Verdaguer, J. T. (2003). Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. *Ophthalmology, 110*(9), 1677-1682.

Xiao, J., Wang, J., Cao, S., & Li, B. (2020). *Application of a novel and improved VGG-19 network in the detection of workers wearing masks.* Paper presented at the Journal of Physics: Conference Series.

Xu, G., Yu, W., Griffith, D., Golmie, N., & Moulema, P. (2016). Toward integrating distributed energy resources and storage devices in smart grid. *IEEE internet of things journal, 4*(1), 192-204.

Xu, K., Feng, D., & Mi, H. (2017). Deep convolutional neural network-based early automated detection of diabetic retinopathy using fundus image. *Molecules, 22*(12), 2054.

Yin, L., Luo, J., & Luo, H. (2018). Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing. *IEEE Transactions on Industrial Informatics, 14*(10), 4712-4721.

Yu, W., Liang, F., He, X., Hatcher, W. G., Lu, C., Lin, J., & Yang, X. (2017). A survey on the edge computing for the Internet of Things. *IEEE Access, 6*, 6900-6919.

Yun, W. L., Acharya, U. R., Venkatesh, Y. V., Chee, C., Min, L. C., & Ng, E. Y. K. (2008). Identification of different stages of diabetic retinopathy using retinal optical images. *Information sciences, 178*(1), 106-121.

Yusuf, R. (2019). Applying Deep Learning for Faster and Accurate Assessment of Diabetic Retinopathy (DR) for Preventing DR-Related Blindness.

Zheng, Y., He, M., & Congdon, N. (2012). The worldwide epidemic of diabetic retinopathy. *Indian journal of ophthalmology, 60*(5), 428.

CURRICULUM VITEA

He received his B.Sc. degree in information and Communications Engineering Department from Al-Khawarizmi College/University of Baghdad in 2016. His research areas are deep learning, IoT, and Digital Signal Processing.