ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ &
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Πατίκου Ιωάννη

Α.Μ. 2012030094

ΘΕΜΑ

Ταξινόμηση των πολυπόδων του παχέος εντέρου, που ανιχνεύθηκαν κατά τη διάρκεια μιας κολονοσκόπησης, ως αδενωματώδεις ή υπερπλαστικοί με τη χρήση αλγορίθμων ανάλυσης εικόνας και μηχανικής μάθησης

Classification of colorectal polyps detected during standard colonoscopy as adenomatous or hyperplastic using image analysis and machine learning algorithms

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Καθηγητής **Μιχαήλ Ζερβάκης** (Επιβλέπων)
Καθηγητής **Γεώργιος Σταυρακάκης**
Δρ. **Ελευθερία Σεργάκη** (Συνεπιβλέπουσα)

Χανιά, Ιανουάριος 2020

Ευχαριστίες

Νιώθω βαθειά υποχρεωμένος, μέσα από αυτές τις σελίδες, να ευχαριστήσω όσους συνέβαλαν στο να πραγματοποιηθεί αυτή η μελέτη:

Το γιατρό γαστρεντερολόγο Κωνσταντίνο Πατίκο για τη συλλογή των δεδομένων που χρησιμοποιήθηκαν στην εργασία και φυσικά για την πολύτιμη βοήθεια που μου έδωσε στο ιατρικό κομμάτι της.

Τον επιβλέποντα καθηγητή Μιχαήλ Ζερβάκη, τον καθηγητή Γεώργιο Σταυρακάκη για την επιλογή τους να βρίσκονται στην επιτροπή μου και την συνεπιβλέπουσα δόκτωρ Ελευθερία Σεργάκη για την υπομονή της και την πολύτιμη βοήθειά της καθ' όλη τη διάρκεια της διπλωματικής μου εργασίας.

Τέλος, την οικογένεια και τους φίλους μου για τα εφόδια που μου προσέφεραν και συνεχίζουν να μου προσφέρουν, καθώς και για την ανιδιοτελή υποστήριξή τους.

Table of Contents

Περίληψη

Σήμερα, ο καρκίνος του παχέος εντέρου αποτελεί μία από τις συχνότερες αιτίες θανάτου από καρκίνο παγκοσμίως. Έρευνες έχουν δείξει ότι ο συγκεκριμένος καρκίνος είναι άρρηκτα συνδεδεμένος με τους πολύποδες του παχέος εντέρου. Η συντριπτική πλειοψηφία των γαστρεντερικών καρκινωμάτων θεωρείται ότι προέρχεται από αδενωματώδεις πολύποδες και επομένως, η έγκαιρη ανίχνευσή τους μπορεί να εμποδίσει την περαιτέρω ανάπτυξη καρκίνου. Ως εκ τούτου, ερευνώνται και αναπτύσσονται νέες μέθοδοι που προσπαθούν να ενισχύσουν το ποσοστό επιτυχούς ανίχνευσης αδενώματος (ADR). Η χρήση τεχνικών τεχνητής νοημοσύνης (AI), όπως η βαθιά μάθηση, και ειδικά τα συνελικτικά νευρωνικά δίκτυα (CNNs), αρχίζουν να εμφανίζονται σαν βοηθήματα στη γαστρεντερική ενδοσκόπηση. Τα CNNs μπορούν να αποτελέσουν σημαντικότατο υποστηρικτικό ιατρικό εργαλείο αυτόματης ιατρικής διάγνωσης των καρκινικών όγκων και των πολυπόδων του παχέος εντέρου. Για το λόγο αυτό, τέσσερα διαφορετικά CNN μοντέλα εφαρμόστηκαν στην παρούσα εργασία. Τα τρία πρώτα ασχολούνται με το πρόβλημα της ανίχνευσης πολυπόδων, ενώ το τέταρτο ασχολείται με την ταξινόμηση των πολυπόδων σε αδενώματα και υπερπλαστικούς πολύποδες. Και στα δύο προβλήματα έχουμε δυαδική ταξινόμηση. Στην πρώτη περίπτωση, οι εικόνες ταξινομούνται σε κατηγορίες "Πολύποδας" και "Υγιές" και στη δεύτερη περίπτωση σε "Αδένωμα" και "Υπερπλαστικός". Ένας συνδυασμός διαφόρων τεχνικών βελτίωσης εφαρμόστηκε στα πρώτα τρία μοντέλα για να ελεγχθεί πώς επηρεάζουν την απόδοση των CNNs. Αυτές οι μέθοδοι είναι η Augmentation, η Dropout Regularization και η τεχνική Transfer Learning. Στο μοντέλο CNN για την κατηγοριοποίηση των πολυπόδων σε αδενώματα και υπερπλαστικούς πολύποδες τα δεδομένα αφορούσαν τις τιμές των χαρακτηριστικών υφής GLCM των εικόνων. Τα δεδομένα που χρησιμοποιήθηκαν στην εργασία συλλέχθηκαν αναδρομικά από το προσωπικό αρχείο του ιατρού Κωνσταντίνου Πατίκου. Από 750 ασθενείς συνολικά συγκεντρώθηκαν 1576 εικόνες; 798 περιέχουν πολύποδες και 778 απεικονίζουν υγιές έντερο. Οι 798 εικόνες πολυπόδων χωρίζονται σε δύο κατηγορίες; 424 εικόνες με αδενοματώδεις πολύποδες και 374 με υπερπλαστικούς. Όλα τα δεδομένα προέρχονται από κοινό ενδοσκόπιο. Τα δεδομένα δεν είναι ιδιαίτερα ομοιόμορφα, καθώς οι εικόνες διαφέρουν σε μεγέθυνση, εστίαση ή χρωματισμό. Τα σύνολα δεδομένων εκπαίδευσης και επαλήθευσης για το πρόβλημα εντοπισμού πολυπόδων περιέχουν 1470 και 106 εικόνες αντίστοιχα. Τα σύνολα δεδομένων εκπαίδευσης και επαλήθευσης για το πρόβλημα ταξινόμησης των πολυπόδων αποτελούνται από 170 και 34 εικόνες αντίστοιχα. Η αποτελεσματικότητα των προτεινόμενων μοντέλων αξιολογήθηκε μετρώντας την ακρίβεια, την ευαισθησία και την ειδικότητα. Το πιο αποτελεσματικό σενάριο που ασχολήθηκε με το πρόβλημα του εντοπισμού πολυπόδων σημείωσε 92,2% accuracy, 94,4% sensitivity, 90,6% specificity, 90,9% precision, 9,4% FPR και 5,6% FNR σε δεδομένα που δεν είχε ξαναδεί. Το μόνο σενάριο που αντιμετώπισε το πρόβλημα ταξινόμησης πολυπόδων επέτυχε

85% accuracy, 88,8% sensitivity, 81,3% specificity, 84,2% specificity, 18,7% FPR και 11,1% FNR.

Λέξεις Κλειδιά

Καρκίνος παχέος εντέρου, Κοινή κολονοσκόπηση, Πολύποδες παχέος εντέρου, Αδενώματα, Υπερπλαστικοί, Τεχνητή Νοημοσύνη, Deep Learning, Συνελικτικά Νευρωνικά Δίκτυα (CNN), Ανίχνευση Πολυπόδων, Ταξινόμηση Πολυπόδων, Augmentation, Dropout, Transfer Learning, GLCM texture, Sensitivity, Specificity, False Positive Ratio (FPR), False Negative Ratio (FNR)

Abstract

Nowadays, colorectal cancer (CRC) is one of the most frequent causes of cancer fatality worldwide. Researches have shown CRC's intimate relation to colorectal polyps. The vast majority of all gastrointestinal carcinomas are considered to originate from adenomatous polyps and as a result, their early detection could prevent their transformation to cancer. Hence, new methods that are trying to enhance the adenoma detection rate (ADR) are being researched and developed. The employment of artificial intelligence (AI) techniques, like deep learning, and especially convolutional neural networks (CNNs), helps to identify cancerous tumors and colonic polyps. The CNN architecture is well-suited by design to provide beneficial solutions, including polyp detection and classification. On that account, four different CNN models have been implemented in the current thesis. The first three are dealing with the polyp detection problem, while the fourth one performs polyp's classification as ''adenomatous'' and ''hyperplastic''. In both tasks, a binary classification takes place. In the first case, image data are classified into "polyp" and "healthy" categories and in the second case into "adenomas" and "hyperplastic". A combination of various improving techniques has been applied in the first three models to see how they affect the performance of the CNNs. These methods consist of: Image Data Augmentation, the Dropout Regularization technique and the Transfer Learning technique. In the CNN model for polyp classification as adenomatous and hyperplastic polyps, the data were related to the values of the GLCM texture features of the images. The data, used in this thesis, were collected retrospectively from the extensive personal archive of doctor Konstantinos Patikos. From a total of 750 patients, 1576 images were collected; 798 contain polyps and 778 depict a healthy colon. The 798 images with polyps are separated into two categories; 424 pictures with adenomatous polyps and 374 pictures with hyperplastic polyps. All the data come from standard colonoscope, which uses white light for the

inspection of the bowel wall. The images are not very uniform, as they tend to differ in zoom, focus or coloration. The training and testing datasets for the polyp detection task contain 1470 and 106 samples respectively. The training and testing datasets for the polyp classification problem consist of 170 and 34 samples each. Performance metrics like Accuracy, Sensitivity, and Specificity were measured to evaluate the effectiveness of the proposed models. The most efficient scenario that dealt with the polyp detection task scored 92.2% accuracy, 94.4% sensitivity, 90.6% specificity, 90.9% precision, 9.4% FPR and 5.6% FNR over unseen data. The only scenario that confronted the polyp classification problem achieved 85% accuracy, 88.8% sensitivity, 81.3% specificity, 84.2% precision, 18.7% FPR and 11.1% FNR.

Keywords

Colorectal Cancer (CRC), Standard Colonoscopy, Colorectal Polyps, Gastrointestinal, Adenomas, Hyperplastic, Adenoma Detection Rate (ADR), Artificial Intelligence (AI), Deep Learning, Convolutional Neural Networks (CNNs), Polyp Detection, Polyp Classification, Augmentation, Dropout, Transfer Learning, GLCM texture, Sensitivity, Specificity, False Positive Ratio (FPR), False Negative Ratio (FNR)

1 INTRODUCTION

Colorectal cancer (CRC) is the third most frequent cancer, striking both men and women all over the world and is the second leading cause of cancer-related deaths. Colonoscopy is the best way for screening CRC. Gastrointestinal colonoscopy has provided a decrease in the occurrence and mortality of CRC through the detection and extraction of adenomatous polyps. Notwithstanding important technical advances in endoscopes over the last years, the main constraint of endoscopic examinations is operator variation. This difference depends on the operator's abilities, perceptive factors, personality qualities, experience, and philosophy. All of these constituents can be mitigated, to some extent, by strong educational efforts, but they cannot be completely eradicated. Thus, generating an Artificial Intelligence computer-based support system for the detection and classification of colorectal polyps would be an essential benefaction to reduce the variation in endoscopists' performance.

Artificial intelligence is machine intelligence that mimics human cognitive function [1]. Experimentation in AI started in the 1950s with the first applications occurring in board games, logical reasoning, and simple algebra. Attention in the field developed over the last decades due to the exponential progress in computational power and database size. [2]

Machine learning is an artificial intelligence technique in which computers use data to enhance their performance in a task without precise instruction [3]. Models of machine learning involve an application that learns to recognize and dismiss spam emails or a thermostat that learns household temperature preferences over time. Machine learning is often classified into two categories - supervised and unsupervised learning [2]. In supervised learning, a machine is trained with data that include pairs of inputs and outputs [4]. The machine learns a function to map the inputs to outputs, which can then be used towards new samples [2]. In unsupervised learning, machines are provided data inputs that are not explicitly paired to labels or outputs [4]. The machine is tasked with finding its structure and patterns from the set of objects. An example of unsupervised learning is clustering, in which a system creates clusters of similar data points from a large data set. Machine learning, and more specifically deep learning, has been widely applied in tasks such as gaming, weather, security, and media. [2]

Convolutional neural networks (CNNs) are a class of deep neural networks that are highly effective at performing image and video analyses. So, CNN models for colonoscopy could assist endoscopists in detecting and classifying polyps and performing optical diagnosis [2], [5], [6]. In order to be maximally effective, the polyp identification module should have a high sensitivity for the detection of polyps, with low rate of false positives [7].

1.1 The polyp detection problem

As already mentioned, polyps are directly related to colorectal cancer. That is why an effective model of detecting colorectal polyps is essential to be implemented. The deep learning algorithms developed in the thesis, are using supervised learning, meaning the input samples are labeled. The algorithms must be able to classify input images in two categories. These classes are determined as “Healthy”, which represents the negative samples, meaning the images that are not containing any polyps and “Polyp”, which represent the positive samples, that are the pathological pictures containing polyps. This problem constitutes the main volume of the current work, as three different convolutional neural networks have been developed for this task, as well as various ways of feeding the input images in the models to test the differences in time consummation. Also, some techniques to improve the networks’ performance have been applied, like dropout regularization, image data augmentation and transfer learning.

1.2 The polyp classification problem

The other very important problem dealt with in the current work is the classification of colorectal polyps. These polyps are classified into two types: adenomatous and hyperplastic polyps. Polyps of the first type, also referred to as adenomas, are usually cancer precursor lesions, whereas polyps of the second type are not considered to be premalignant. A definitive distinction between the two types requires polyp biopsy and histological examination of the tissue specimens [8]. Today, the international consensus for the treatment of polyposis dictates removal of all polyps, regardless of the location, the size or other characteristics, in order to prevent the possible development of cancer [9], [10], [11]. That is why a reliable system that would be capable of supporting the detection of adenomas is crucial to be developed to enhance the endoscopist’s ability to locate early stage adenomas. The method used in this work consists of the GLCM texture feature extraction technique for the data preprocessing and a convolutional neural network that is trained in this samples and performs the classification with hopeful results.

1.3 Objectives of the Diploma Thesis

The objectives of the current diploma thesis are the following:

I. The preprocessing of the input image data for the polyp detection problem in order to be fed in the convolutional neural networks. A number of different ways will be tested to check the variation in the execution time.

II. The preprocessing of the input image data for the polyp classification problem. A feature extraction and a normalization technique will be used before they are inputted in the convolutional neural network.

III. The implementation of different convolutional neural networks for the detection of the colorectal polyps and the training of these algorithms.

IV. The application of a variety of improving techniques in the convolutional neural networks and the comparison of their results to see which is the most effective.

V. The implementation of a convolutional neural network for the classification of the colorectal polyps, into adenomas and hyperplastic, the training of this algorithm and the presentation of its results.

VI. More generally speaking, the deeper learning of the tools and techniques used in the present thesis and simultaneously the familiarization with them.

1.4 Structure of the Diploma Thesis

In the 1st Chapter takes place the introduction of the polyp detection and classification problems from images taken by standard colonoscopy and the aims of the present work are set out.

In the 2nd Chapter, there is an extensive reference to colon polyps and their impact on colorectal cancer. A basic theoretical background on the colorectal polyps is analyzed. Their main categories are being presented, along with the latest techniques and methods for their detection.

In the 3rd Chapter, the basic knowledge of neural networks and deep learning are being mentioned. The definition of the neural network, its fundamental architecture and the different types of it are introduced. Furthermore, an introduction to deep neural networks and more specifically to convolutional neural networks takes place. At last, the toolset and techniques used in the thesis are being discussed. An extensive reference for Python and its libraries, Google Colab, Tensorflow, Keras, ImageJ, and others occurs.

In the 4th Chapter, the followed methodologies, the phases of work, the proposed algorithms for the detection and classification of colorectal polyps are described. In addition, an analytical reference in the data used in the thesis takes place. The way that they were collected and preprocessed and the significance of data preprocessing in deep learning is mentioned too. There is a methodical analysis of every scenario implemented in the current work.

In the 5th Chapter, the results of every method used for the polyp detection and polyp classification problems are presented. They are shown, firstly, for each scenario separately and then in comparison to each other. The metrics for the quantification of each method's efficiency are also determined, along with the way that they are calculated.

Finally, in the 6th Chapter, the results are summarized in one single table to help the deduction of conclusions. Then, they are discussed and the outcomes are presented. Lastly, ideas are suggested for the future evolution of the proposed methods.

In the Appendix AP1, some examples of the training samples are listed. Healthy images, images that contain adenomas and those who contain hyperplastic polyps.

In the Appendix AP2, some examples of the samples used for the testing phase are presented.
In the Appendix AP3, the training and validation accuracy plots and the training and validation loss plots for its scenario are listed.

2 COLON POLYPS AND THEIR IMPACT ON COLORECTAL CANCER

2.1 Introduction

Colorectal cancer (CRC), including cancer of the colon and rectum is the third most common cancer globally, with an estimated number of 1.4 million diagnoses in 2012 [12]. Incidence has traditionally been the highest in affluent Western countries, but is now rapidly increasing with economic development in many other parts of the world [13].

In contrast to other cancers, in most cases CRC develops very slowly over many years, if not decades, following the initial transformation of a normal colorectal epithelium to an adenoma. The slow progression through the adenoma–carcinoma sequence, with the possibility of detecting and removing adenomas at colonoscopy, offers great opportunities for the secondary prevention of CRCs, in addition to the opportunities for secondary prevention of deaths from CRC by detecting the cancer at an earlier, often-curable stage [14].

2.2 Colorectal Polyps

Colorectal polyp is any mass that arises from the bowel wall and protrudes into the lumen. Grossly, a polyp is classified as pedunculated or sessile depending on whether it contains a discrete stalk. Polyps occasionally cause gross rectal bleeding or, very rarely, symptoms of partial bowel obstruction. Most polyps are asymptomatic lesions detected by screening or diagnostic studies performed for other reasons [15]. Colorectal polyps are extremely common in Western countries; incidence of polyps ranges from 7 to 50%; the higher figure includes very small polyps (usually hyperplastic polyps or small adenomas) found autopsy performed in people aged >60 years [16], [17].

The main importance of polyps is their well-recognized relationship to colorectal cancer [18]. It is, now, generally accepted that most (95%) colorectal cancers arise from benign, neoplastic adenomatous polyps (adenomas). Although this adenoma–carcinoma sequence can probably never be proved directly, persuasive data exist indicating that colorectal neoplasia progresses through a continuous process from normal mucosa, to benign adenoma, to carcinoma.

Histologically, polyps are classified as neoplastic (adenomas) or nonneoplastic [19], [20]. Nonneoplastic polyps have no malignant potential and include hyperplastic

polyps, hamartomas, lymphoid aggregates, and inflammatory polyps. Neoplastic polyps or adenomas have malignant potential and are classified according to the World Health Organization as tubular, tubulovillous, or villous adenomas, depending on the presence and volume of villous tissue [21]. Tubular adenomas are composed of straight or branched tubules of dysplastic tissue; villous adenomas contain fingerlike projections of dysplastic epithelium. Approximately 70% of polyps removed at colonoscopy are adenomas [22]. From 70% to 85% of these are classified as tubular (0–25%, villous tissue), 10–25% are tubulovillous (25–75%, villous tissue), and 5% are villous adenomas (75%–100%, villous tissue). All these are presented in the following Table 2.1.

Histological Classification	Polyp Type	Malignant Potential
Non-neoplastic	Hyperplastic polyps	No
	Hamartomas	
	Lymphoid aggregates	
	Inflammatory polyps	
Neoplastic	Tubular adenomas (0–25% villous tissue)	Yes
	Tubulovillous adenomas (25–75% villous tissue)	
	Villous adenoma (75–100% villous tissue)	

Table 2.1 Classification of colorectal polyps.

Some degree of dysplasia exists in all adenomas. Most authorities recommend that dysplasia be classified as low and high-grade, because this classification reduces the problem of interobserver variation [23]. High-grade dysplasia includes the histological changes previously called “carcinoma in situ,” “intramucosal carcinoma,” or “focal carcinoma.” Abandonment of these terms is recommended because of concern for misinterpretation of the clinical significance that might lead to overtreatment, and thus they will not be used in the guidelines. Approximately 5–7% of patients with adenomas have high-grade dysplasia, and 3–5% have invasive carcinoma at the time of diagnosis. Increasing dysplasia and, presumably, malignant potential correlate with increasing adenoma size, villous component, and patient age [23]. The likelihood of invasive carcinoma also increases with increasing polyp size [20].

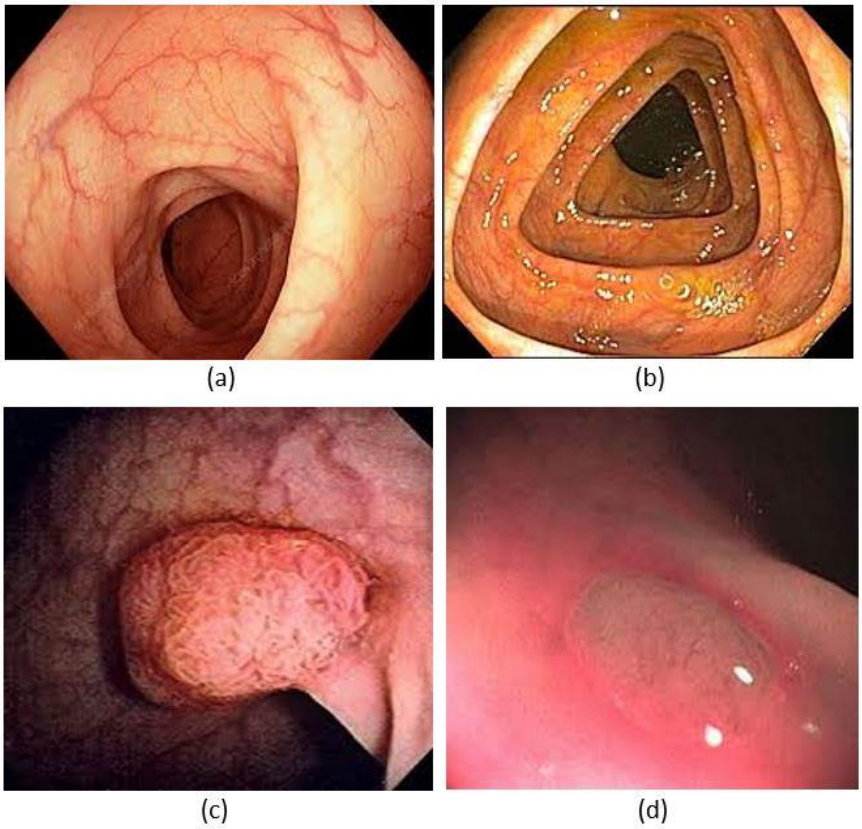


Figure 2.1 Colonoscopy images: (a) Normal descending colon, (b) Normal transverse colon, (c) Pedunculated colon adenoma, (d) Hyperplastic colon sessile polyp.

2.3 Colorectal cancer prevention

The lifetime risk of developing Colorectal Cancer (CRC) in many regions is around 5%. Approximately 45% of persons diagnosed with CRC die as a result of the disease, despite treatment [12]. Treatment modalities have largely improved over the past decade. Treatment has modestly improved disease outcome and extended survival in patients with advanced and metastatic disease. But these advancements have been accompanied by markedly increased treatment costs. As a result, modelling studies have shown that various screening strategies are cost-saving [24]. Most CRCs develop from a preclinical precursor, the adenoma. The progression from early adenoma to invasive cancer takes years [25]. The high incidence, long preclinical phase, recognizable and treatable precursor, the high cost of treatment, and the correlation of mortality with disease stage make CRC highly suitable for population screening [26]. This has been confirmed by randomized controlled trials (RCTs) that have formed the basis for international guidelines recommending CRC screening [27], [28]. Despite these recommendations, screening is currently only offered to a small proportion of the target population.

Colonoscopy is generally considered the gold standard for the detection of colorectal neoplasia. In prospective cohort studies, colonoscopy has been associated with long-

term (20–30 years) reduction in CRC mortality [29]. However, there currently exists a wide variation between endoscopists in terms of their success at detecting adenomas. Adenoma Detection Rate (ADR) (a metric defined as the proportion of colonoscopy examinations that detects 1 adenomas), is an important quality metric of colonoscopy and efforts to improve this outcome continues to be an area of active research. Performance targets of 30% ADR in males and 20% in females have been recommended as benchmarks [30].

Standard colonoscope uses white light for the inspection of bowel wall and detection of polyps and is the most widespread method for screening worldwide. Narrow band imaging (NBI), chromoendoscopy (CE) techniques and usage of high definition (HD) scopes applied in colonoscopy in order to improve ADR of endoscopists. A recent advancement in endoscopic technology is the development of push-button optical magnification, which enables a 65 magnification when the tip of the colonoscope is brought close to the mucosa (near focus) [31].

Figure 2.2 Colonoscope.

Figure 2.3 Snare polypectomy.

Colonoscopy misses 6% to 12% of adenomas 1 cm or larger, 11% of advanced adenomas, and 5% of cancers. The overall miss rate for polyps of any size was 22% [32]. The first aim of colorectal cancer screening is to detect as many polyps as possible.

Removal of polyps with polypectomy techniques is the next step which leads to the prevention of CRC. However, polypectomy is a therapeutic procedure carries risk of complications such as hemorrhage or even perforation of the bowel wall. So, it is very important, if possible, to differentiate neoplastic from non-neoplastic polyps before snare polypectomy applied because, as mentioned above, only neoplastic polyps carry risk for colorectal cancer development [18].

2.4 Latest techniques and methods

Several adjunct techniques and devices are under investigation for improving an endoscopist's ability to detect adenomas. In order to increase efficiency and decrease overall healthcare costs (post-polypectomy complications costs included), efforts have been underway to develop methods to accurately diagnose and disregard diminutive (<5 mm), non-neoplastic polyps and remove only precancerous polyps. One such method gaining significant traction is computer-aided diagnosis (CAD), which is a computer-assisted image analysis that incorporates both increased polyp identification and histopathologic differentiation without modifications to the colonoscope or the actual procedure. Furthermore, unlike alternative techniques (such as narrow-band imaging and virtual chromoendoscopy), CAD is largely operator-independent [33].

In the past two decades, several computer-aided detection (CAD) techniques have been proposed to assist endoscopists in the detection of polyps that would otherwise have been missed [34]. The ideal automatic polyp detection tool must have:

1. high sensitivity for detection of polyps
2. decreased rate of false positives
3. low latency so that polyps can be tracked and identified in near-real time

In the application of AI for colonoscopy, automatic detection and characterization of colorectal polyps has attracted the most attention compared to the differentiation of polyps as adenomatous or hyperplastic. The former aims to detect polyps, irrespective of their pathology (neoplastic polyps or hyperplastic polyps). The later then helps to visually classify the detected polyps into pathological categories. In the literature, polyp detection and classification are mostly studied using more advanced imaging such as magnifying narrow-band imaging [35].

For polyp detection, the models are trained to distinguish between polyp regions and non-polyp regions. However, for accurate polyp classification, the models are trained

to differentiate between hyperplastic versus adenomatous polyps, which requires the extraction of more granular features. Since the information on the differentiation of polyps from white-light endoscopy is insufficient, most polyp classification CAD models exploit more advanced imaging such as narrow band imaging (NBI) [35].

3 BASIC KNOWLEDGE OF NEURAL NETWORKS AND DEEP LEARNING

3.1 Introduction

As already mentioned, deep learning plays a significant role in the process of detecting and classifying colorectal polyps in recent years. It is involved in many stages of this process and it is essential to present its fundamental principles in this chapter.

3.2 What is an Artificial Neural Network?

Artificial Neural Network (ANN) is an efficient computing system, whose central theme is borrowed from the analogy of biological neural networks. ANN acquires a large collection of units that are interconnected in some pattern to allow communication between the units. These units, also referred to as nodes or neurons, are simple processors that operate in parallel. [36]

Every neuron is connected with other neuron through a connection link. Each connection link is associated with a weight that has information about the input signal. This is the most useful information for neurons to solve a particular problem, because the weight usually excites or inhibits the signal that is being communicated. Each neuron has an internal state, which is called an activation signal or activation function. Output signals, which are produced after combining the input signals and activation rule, may be sent to other units. [36]

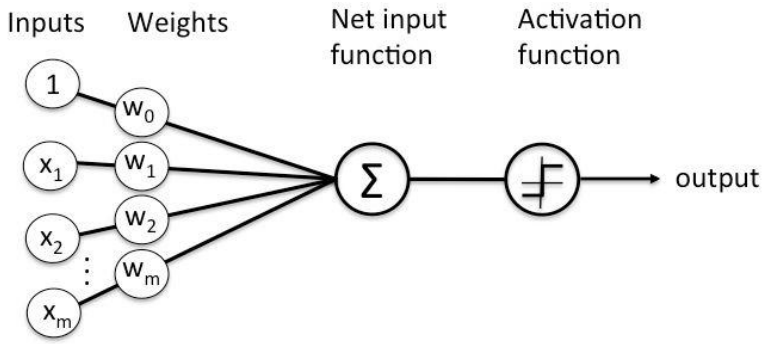


Figure 3.1 Model of a simple neuron.

3.2.1 Neural Network Architecture

From the above explanation we can conclude that a neural network is made of neurons and when we train a neural network, we want the neurons to fire whenever they learn specific patterns from the data, and we model the fire rate using an activation function. [37]

So, the basic architecture and principles of a neural network is the following:

- **Input Nodes (input layer):** No computation is done here within this layer. They just pass the information to the next layer (hidden layer most of the time). A block of nodes is also called **layer**. [37]
- **Hidden nodes (hidden layer):** In Hidden layers is where intermediate processing or computation is done. They perform computations and then transfer the weights (signals or information) from the input layer to the following layer (another hidden layer or to the output layer). It is possible to have a neural network without a hidden layer. [37]
- **Output Nodes (output layer):** Here we finally use an activation function that maps to the desired output format (e.g. softmax for classification). [37]
- **Connections and weights:** The network consist of connections, each connection transferring the output of a neuron i to the input of a neuron j. In this sense i is the predecessor of j and j is the successor of i. Each connection is assigned a weight Wij. [37]
- **Activation function:** The activation function of a node defines the output of that node given an input or set of inputs. A standard computer chip circuit can be seen as a digital network of activation functions that can be "ON" (1) or "OFF" (0), depending on input. This is similar to the behavior of the linear perceptron in neural networks. However, it is the nonlinear activation function that allows such networks to compute nontrivial problems using only a small number of nodes. In artificial neural networks this function is also called the transfer function. [37]
- **Learning rule:** The learning rule is a rule or an algorithm which modifies the parameters of the neural network, in order for a given input to the network to produce a favored output. This learning process typically amounts to modifying the weights and thresholds. [37]
- **Forward propagation:** Forward propagation is the process of moving forward through the neural network. The objective of forward propagation is to calculate the activations at each neuron for each successive hidden layer until it reaches the output. [38]
- **Backpropagation:** Backpropagation is the reverse of forward propagation. Except instead of signal, we are moving error backwards through our model. Basically, backpropagation calculates the error attributable to each neuron and that in turn calculates the partial derivatives and ultimately the gradient so that it can utilize gradient descent. [38]

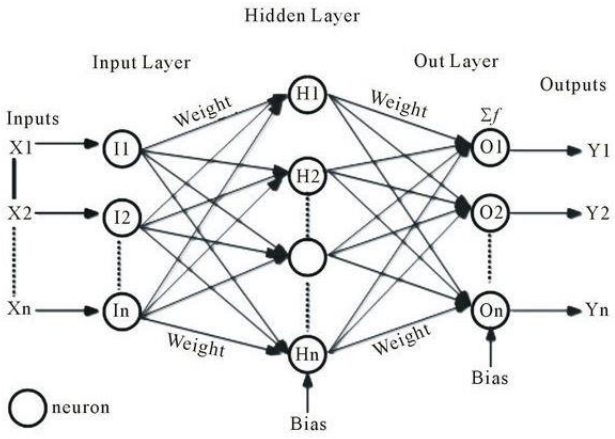


Figure 3.2 Basic neural network architecture.

3.2.2 Types of Neural Networks

There are many classes of neural networks and these classes also have sub-classes. Here we will list the most used ones:

1. **Feedforward Neural Network:** A feedforward neural network is an artificial neural network where connections between the units do not form a cycle. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network. [37]

   There are two types of feedforward neural networks:

   1.1 Single-layer Perceptron: This is the simplest feedforward neural network and does not contain any hidden layer, which means it only consists of a single layer of output nodes. This is said to be single because when we count the layers, we do not include the input layer, the reason for that is because at the input layer no computations are done, the inputs are fed directly to the outputs via a series of weights. [37]

   1.2 Multi-layer perceptron (MLP): This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way. Each neuron in one layer has directed connections to the neurons of the subsequent layer. In many applications the units of these networks apply a sigmoid function as an activation function. MLP are very more useful and one good reason is that, they are able to learn non-linear representations**.** [37]

   1.3 Convolutional Neural Network (CNN): Convolutional Neural Networks are very similar to ordinary Neural Networks. They are made up of neurons that have learnable weights and biases. In Convolutional Neural Networks, the

unit connectivity pattern is inspired by the organization of the visual cortex. Units respond to stimuli in a restricted region of space known as the receptive field. Receptive fields partially overlap, over-covering the entire visual field. Unit response can be approximated mathematically by a convolution operation. They are variations of multilayer perceptrons that use minimal preprocessing. Their wide applications are in image and video recognition, recommender systems and natural language processing. CNNs require large data to train on. [37]

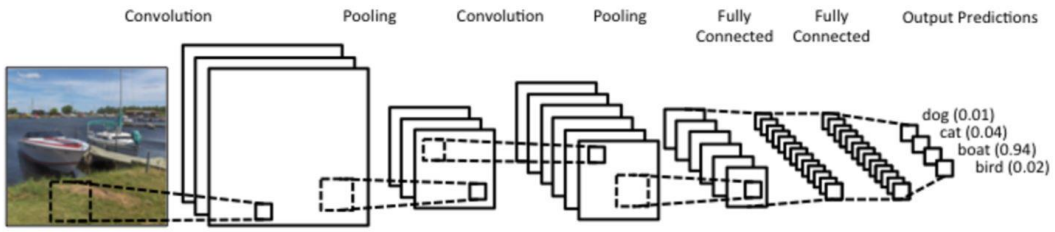


Figure 3.3 CNN for image classification.

2. **Recurrent neural networks:** In recurrent neural network (RNN), connections between units form a directed cycle (they propagate data forward, but also backwards, from later processing stages to earlier stages). This allows it to exhibit dynamic temporal behavior. Unlike feedforward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and other general sequence processors. [37]

3.2.3 Most used activation functions

The activation function takes a single number and performs a certain fixed mathematical operation on it. Here are some activations functions you will often find in practice: [37]

- ReLU
- Leaky ReLU
- Tanh
- Sigmoid

3.3 Deep Learning – Deep Neural Networks

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. In deep learning, a computer model

learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. [39]

Models are trained by using a large set of labeled data and deep-learning networks are distinguished from the more commonplace single-hidden-layer neural networks by their depth; that is, the number of node layers through which data must pass in a multistep process of pattern recognition. More than three layers (including input and output) qualifies as “deep” learning. So deep is a strictly defined term that means more than one hidden layer. [40]

More specifically, deep neural networks (DNNs) use a pipeline of many layers of processing units for transformation and feature extraction. They are based on learning features of the data in unsupervised manner (automatic feature extraction). This means higher-level features (that are found in latter processing layers) are derived from lower-level features (that are found in initial processing layers). In this way, DNNs learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts. [41] This feature hierarchy makes DNNs capable of handling very large, high-dimensional data sets with billions of parameters that pass through nonlinear functions. [40]

Because of the very large datasets that a deep neural network needs to be trained with and the great amount of computations that it performs, deep learning is in need of substantial computing power. High-performance GPUs have a parallel architecture that makes them efficient for deep learning. In combination with clusters or cloud computing, this reduces training time for a deep learning network from weeks to several hours and maybe less.

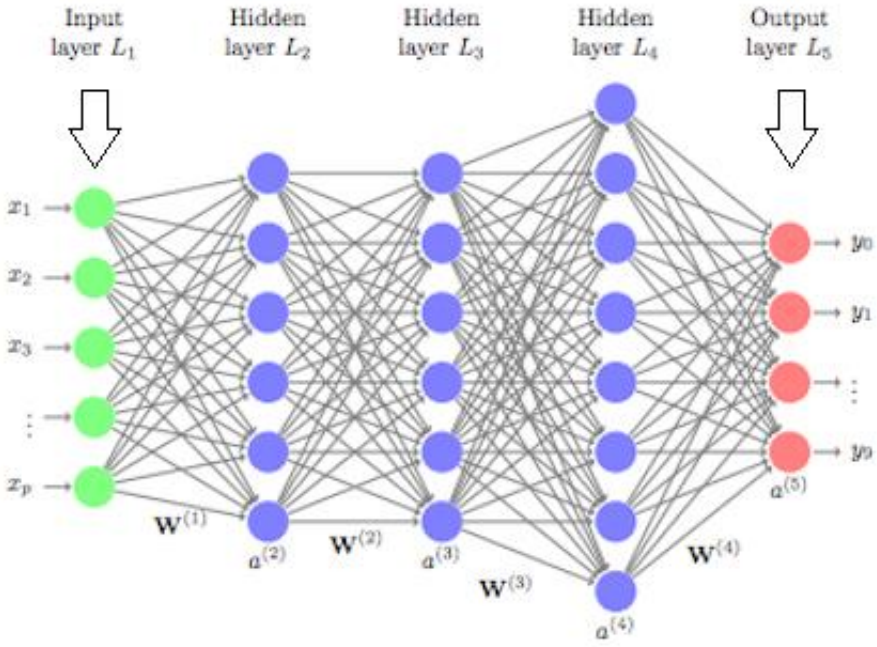


Figure 3.4 Example of a deep neural network with three hidden layers.

Maybe the most common type of deep neural networks is the convolutional neural networks (CNNs), which are being practiced in the present thesis. Their architecture and basic principles are analyzed subsequently.

3.4 Convolutional Neural Networks

In neural networks, Convolutional Neural Networks (CNNs) is one of the main categories to do images recognition or images classifications. Objects detections and faces recognition are some of the areas where CNNs are widely used. [42]

A CNN is a deep learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a CNN is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, CNNs have the ability to learn these filters/characteristics. [43]

The architecture of a CNN is analogous to that of the connectivity pattern of neurons inside the human brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area. [43]

3.4.1 Architecture of a Convolutional Neural Network

CNNs get an input image, process it and classify it under certain categories (dog, cat, polyp, healthy). Computers see an input image as an array of pixels and it depends on the image resolution. Based on the image resolution, it will see h x w x d (h = height, w = width, d = dimension). For example, an image of 6 x 6 x 3 array of a matrix of RGB (3 refers to RGB values) and an image of 3 x 3 x 1 array of a matrix of a grayscale image. [42]

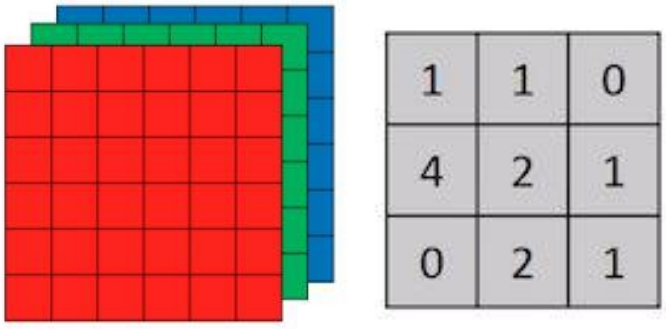


Figure 3.5 Arrays of a 6x6x3 RGB image and of a 3x3x1 grayscale image.

To train and test a CNN model, each input image will pass through a series of convolution layers with filters (Kernels), Pooling layers, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values. [42]

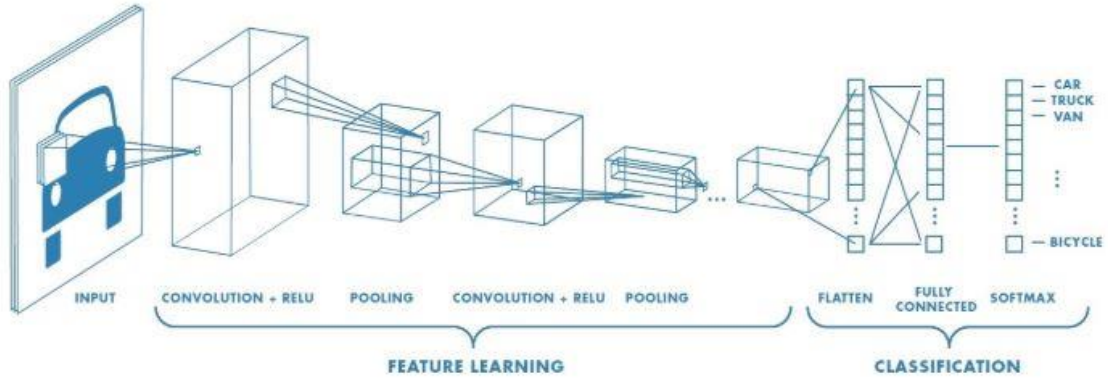


Figure 3.6 Convolutional neural network with many convolutional layers.

As mentioned above, a typical CNN consists of the:

- **Input Layer:** Here the network receives the input images in a specific color space. There are a number of such spaces like RGB, HSV, Grayscale, CMYK etc. When images reach very large dimensions, the amount of computations that take place in the network becomes very big. So, the CNN comes to reduce the images into a form which is easier to process, without missing features that are critical for getting a valid prediction.
- **Convolution Layer:** This is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel. The convolution of an image matrix with the filter, called Feature Map, gives the output. Convoluting an image with different filters results to operations such as edge detection, blur and sharpen by applying filters. [42]

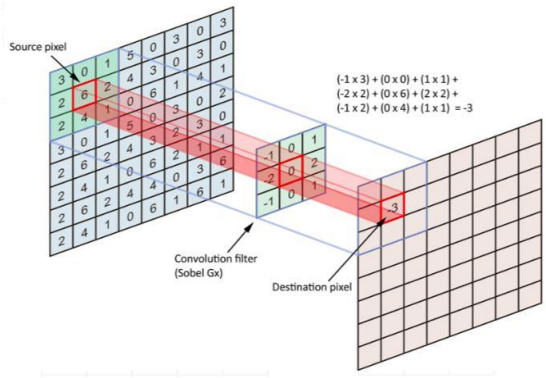


Figure 3.7 A convolution example.

The number of pixels that the filter shifts every time to perform the convolution is called Stride. For example, if the Stride is 1, then the filter moves one pixel at a time. Another important operation in this layer is Padding. If the filter does not fit in the picture, then either the picture is padded with zeros (zero-padding) or the filter ignores the part of the image that does not fit in it (valid padding).

- **Pooling Layer:** Alike to the Convolutional Layer, the Pooling layer is responsible for decreasing the spatial size of the Convolved Feature. This is to minimize the computational power required to process the data through dimensionality reduction. Moreover, it is useful for extracting dominant features which are rotational and positional invariant, consequently maintaining the process of effectively training of the model. There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the part of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the part of the image covered by the Kernel. [43]

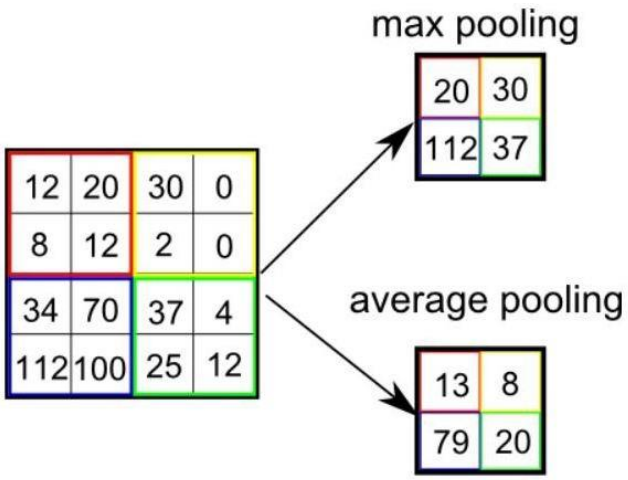


Figure 3.2 The two types of pooling.

- **Fully Connected Layer (FC Layer):** This is the last layer before the final output**.** The feature map matrix is being flattened and is converted to a column vector. Then, the flattened output is fed to a feed-forward neural network and back-propagation applied to every iteration of training. Over a series of epochs, the model is able to identify the dominating and the low-level features in pictures. [43] Lastly, there is an activation function such as softmax or sigmoid to classify the outputs to the problem's categories.

3.5 Toolset and techniques used in the thesis

A particular set of tools was used to construct the convolutional neural networks presented in the current thesis. Subsequently, these tools and techniques are reported and analyzed in detail.

3.5.1 Python

First of all, the programming language used is Python 3.6. The first environment utilized was the Anaconda, from which the Jupyter Notebook proved very useful for the early attempts. As mentioned above, CNNs are very demanding in computing power and as a result it would be very time consuming to run the algorithms in a personal computer. So, the solution found in this problem was the Google Collaboratory.

The primary Python libraries and packages used in the code are:

- **OpenCV (cv2):** OpenCV is an image processing library created by Intel and later supported by Willow Garage. Now it is maintained by Itseez. OpenCV is available on Mac, Windows, Linux. Works in C, C++, and Python. It is open-source, totally free and easy to use and install. It contains some of the most crucial functions that were used in the data preprocessing to read, resize, reshape an image, etc.
- **Numpy:** Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. [44]

  This was used to convert the images to numpy arrays and manage them.
- **OS:** The OS module in Python provides functions for interacting with the operating system. OS comes under Python’s standard utility modules. This module

provides a portable way of using operating system dependent functionality. The "os" and "os.path" modules include many functions to interact with the file system. [45]

It was very helpful while dealing with directories.

- **Matplotlib:** Matplotlib is a visualization library in Python for 2D plots of arrays. It is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. [46]

  It helped to make all the diagrams, plots and confusion matrices.

- **Pickle:** The pickle module is used for implementing binary protocols for serializing and de-serializing a Python object structure. "Pickling" is a process where a Python object hierarchy is converted into a byte stream. "Unpickling" is the inverse of the "Pickling" process where a byte stream is converted into an object hierarchy. [47]
- **Random:** This module implements pseudo-random number generators for various distributions. For integers, there is a uniform selection from a range. For sequences, there is a uniform selection of a random element, a function to generate a random permutation of a list in-place and a function for random sampling without replacement. On the real line, there are functions to compute uniform, normal (Gaussian), lognormal, negative exponential, gamma, and beta distributions. [48]

  It was used to shuffle the dataset.

- **Pandas:** In computer programming, Pandas is a software library written for Python for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. [49]
- **GC:** This module provides an interface to the optional garbage collector. It provides the ability to disable the collector, tune the collection frequency and set debugging options. It also provides access to unreachable objects that the collector found but cannot free. [50]

  It was a very useful module to clean memory garbage.

- **Seaborn:** Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. [51]

3.5.2 Google Collaboratory

Collaboratory is a research tool for machine and deep learning education and research. It is a totally free Jupyter notebook environment that needs no setup and runs

entirely in the cloud. It works with most major browsers, and is most thoroughly tested with latest versions of Chrome, Firefox and Safari. All Collaboratory notebooks are stored in Google Drive. It supports Python 2.7 and Python 3.6. The code is executed in a virtual machine dedicated to the account. Virtual machines are recycled when idle for a while, and have a maximum lifetime enforced by the system. By using the Collaboratory, the code runs in Google's GPUs, which have a much bigger computing power than a typical personal computer. As a result, a great amount of time is saved during the training of the models. [52]

3.5.3 TensorFlow

In general, TensorFlow is an open source library for fast numerical computing. It was created and is maintained by Google and released under the Apache 2.0 open source license. The API is nominally for the Python programming language, although there is access to the underlying C++ API. Unlike other numerical libraries intended for use in Deep Learning like Theano, TensorFlow was designed for use both in research and development and in production systems. It can run on single CPU systems, GPUs as well as mobile devices and large-scale distributed systems of hundreds of machines. [53]

More specifically, TensorFlow is a machine learning system that works at large scale and in heterogeneous environments. TensorFlow uses dataflow graphs to represent computation, shared state and the operations that mutate that state. It outlines the nodes of a dataflow graph across many machines in a batch and within a machine across multiple computational devices, including multicore CPUs, general-purpose GPUs, and custom-designed ASICs known as Tensor Processing Units (TPUs). This architecture gives flexibility to the application developer. TensorFlow allows developers to experiment with novel optimizations and training algorithms. TensorFlow supports various applications, but it is concentrating on training and inference on deep neural networks. [54]

3.5.4 Keras

Keras is a Python library for deep learning that can run on top of Theano or TensorFlow. It was developed to make developing deep learning models as fast and easy as possible for research and development. It runs on Python 2.7 or 3.6 and can seamlessly execute on GPUs and CPUs given the underlying frameworks. [53]

The summary of the construction of a deep learning model in Keras is the following:

- **Define the model:** Create a Sequential model and add configured layers.
- **Compile the model:** Define loss function and optimizers and call the compile() function on the model.

- **Fit the model:** Train the model on a sample of data by calling the fit() function on the model.
- **Make predictions:** Utilize the model to make predictions on new data by calling functions such as evaluate() or predict() on the model. [53]

3.5.5 Dropout Regularization

Dropout regularization is a computationally cheap way to regularize a deep neural network. Dropout works by probabilistically removing, or "*dropping out*," inputs to a layer, which may be input variables in the data sample or activations from a previous layer. It simulates a large number of networks with very different network structures and, in turn, making nodes in the network generally more robust to the inputs. Typically, a small amount of dropout can be applied after each convolutional layer, with more dropout applied to the fully connected layers near the output layer of the model. [55]

3.5.6 Image Data Augmentation

Image data augmentation is a technique that can be used to artificially enlarge the size of a training dataset by creating altered versions of images in the dataset. Training deep learning neural network models on more data can result in more efficient models. The augmentation techniques can produce modifications of the images that can enhance the ability of the fit models to generalize what they have learned to new images. Data augmentation can also operate as a regularization technique, adding noise to the training data and helping the model to learn the same features, invariant to their position in the input. Modest changes to the input photos might be useful for the problem, such as small shifts and horizontal flips. These augmentations can be defined as arguments to the ImageDataGenerator used for the training dataset. The augmentations should not be used for the test dataset, in order to evaluate the performance of the model on the unmodified images. [55]

3.5.7 Transfer Learning

Transfer learning suggests using all or parts of a model trained on a similar task. Keras provides a range of pre-trained models that can be loaded and used fully or partially via the Keras Applications API. A useful model for transfer learning is one of the VGG models, such as VGG-16 with 16 layers. The model is composed of two main parts, the feature extractor part of the model that is made up of VGG blocks, and the classifier part of the model that is made up of fully connected layers and the output layer. [55]

This model was used in the present thesis to perform transfer learning to the baseline models of each method, that are described in chapter 4.

Some other famous CNN architectures, that can be used for transfer learning are:

- **AlexNet:** AlexNet was created out of the need to improve the results of the ImageNet challenge. The idea of spatial correlation in an image frame was explored using convolutional layers and receptive fields. [56]

  The network consists of 5 convolutional layers and 3 fully connected layers. The activation used is Rectified Linear Unit (ReLU). The structural block diagram of the network can be found in the table below. [56]

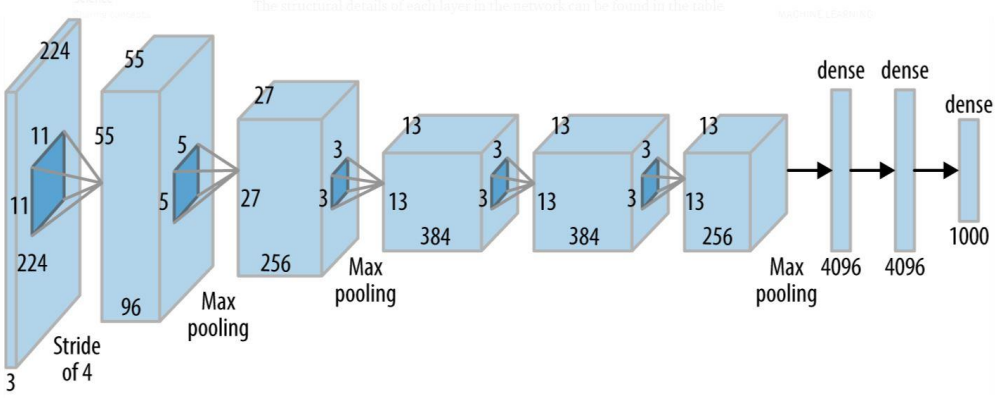


Figure 3.9 Alexnet block diagram.

  The input to the network is a batch of RGB images of size 227x227x3 and outputs a 1000x1 probability vector one corresponding to each class. Data augmentation is performed to reduce over-fitting. This data augmentation includes mirroring and cropping the pictures to increase the variation in the training data-set. The network uses an overlapped max-pooling layer after the first, second and fifth convolutional layers. Overlapped max-pooling layers are just max-pooling layers with strides less than the window size. AlexNet also addresses the over-fitting problem by using drop-out layers, where a connection is dropped during training with a probability p=0.5. Although this avoids the network from over-fitting by helping it escape from bad local minima, the number of iterations required for convergence is doubled. [56]

- **ResNet:** The network f(x) achieves an accuracy of k% on a data-set. By adding more layers to this network, g(f(x)), should have at least an accuracy of k% or in the worst case, g(.), should be an identical mapping yielding the same accuracy as that of f(x). But unfortunately, this is not what happens. Experiments have explained that the accuracy actually decreases by adding more layers to the network. The issue mentioned above happens because of the vanishing gradient problem. As the CNN becomes deeper, the derivative when back-propagating to the initial layers becomes almost insignificant in value. ResNet

approaches this problem by presenting two types of shortcut connections; the Identity shortcut and the Projection shortcut. There are multiple versions of ResNetXX architectures where 'XX' means the number of layers. The most ordinarily used ones are ResNet18, ResNet50 and ResNet101. Resnet18 has approximately 11 million trainable parameters. It consists of convolutional layers with filters of size 3x3. Only two pooling layers are used throughout the network one at the beginning and one at the end of the network. Identity connections are between every two convolutional layers. The basic building block of ResNet is a Residual block that is repeated throughout the network and is shown below: [56]

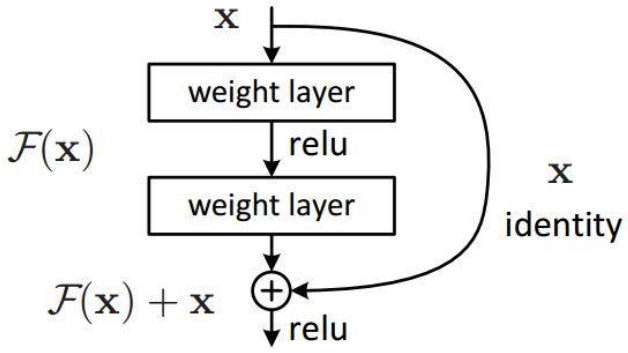


Figure 3.10 Residual block.

Rather than learning the mapping from x →F(x), the network determines the mapping from x → F(x)+G(x). When the dimension of the input x and output F(x) is equal, the function G(x) = x is an identity function. This shortcut connection is named Identity connection. The identical mapping is learned by zeroing out the weights in the intermediate layer during training, as it is more comfortable to zero out the weights than force them to one. When the dimensions of F(x) diverge from x, because of the stride length>1 in the convolutional layers in between, Projection connection is performed rather than the Identity one. The function G(x) converts the dimensions of input x to that of output F(x). [56] There are two kinds of mapping in this case:

1. Non-trainable Mapping (Padding) in which the input x is just padded with zeros to make the dimension match to that of F(x). [56]
2. Trainable Mapping (Convolutional layer) in which a 1x1 convolutional layer is utilized to map x to G(x). Across the network, the spatial dimensions are either held the same or halved. Also, the depth is either kept the same or doubled and the product of Width and Depth after each convolutional layer remains the same. So, 1x1 convolutional layers are applied to half the spatial dimension and double the depth by

using stride length of 2 and multiple of such filters respectively. The number of 1x1 convolutional layers is equal to the depth of F(x). [56]

- **Inception:** In an image classification problem, the size of the salient feature can considerably differ inside the image block. Therefore, deciding on a fixed kernel size is somewhat difficult. Larger kernels are favored for more global features that are spread over a wide area of the picture and on the other hand, smaller kernels give better results in identifying area-specific features that are scattered over the image frame. For effective identification of such variable-sized features, kernels of various sizes are needed. That is what Inception does. Rather than simply going deeper in terms of the number of layers, it goes wider. Multiple kernels of different sizes are implemented within the same layer. The Inception network architecture consists of a number of inception modules of the following structure: [56]

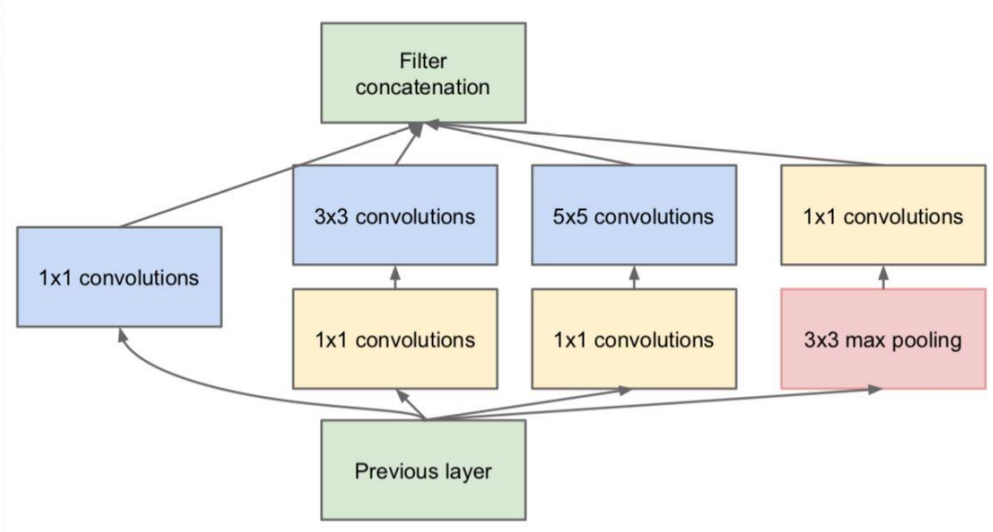


Figure 3.11 Inception module.

Every Inception module is made of four operations in parallel; a 1x1 convolutional layer, a 3x3 convolutional layer, a 5x5 convolutional layer and max pooling. The 1x1 convolutional blocks are used for depth reduction. The outcomes from the four parallel operations are next concatenated depth-wise to form the Filter Concatenation block. [56]

Inception extends the network space from which the best network is to be determined via training. Every Inception module can identify important features at different levels. Global features are obtained by the 5x5 convolutional layer, while distributed features are captured by the 3x3 convolutional layer. The max-pooling operation is effective in capturing low-level features that exist in a region. All of these features are extracted and concatenated before they are fed to the next layer. The network, through training, chooses what features hold the most values and weight respectively. If the pictures in the data-set

have a lot of global features and a few low-level features, then the trained Inception network will hold very small weights matching to the 3x3 convolutional kernel as compared to the 5x5 one. [56]

3.5.8 ImageJ

ImageJ is a public domain Java image processing program inspired by NIH Image for the Macintosh. It runs, either as an online applet or as a downloadable application, on any computer with a Java 1.4 or later virtual machine. Downloadable distributions are available for Windows, Mac OS, Mac OS X and Linux. It can display, edit, analyze, process, save and print 8-bit, 16-bit and 32-bit images. It can read many image formats including TIFF, GIF, JPEG, BMP, DICOM, FITS and "raw". It supports "stacks", a series of images that share a single window. It is multithreaded, so time-consuming operations such as image file reading can be performed in parallel with other operations. ImageJ was designed with an open architecture that provides extensibility via Java plugins. Custom acquisition, analysis and processing plugins can be developed using ImageJ's built in editor and Java compiler. User-written plugins make it possible to solve almost any image processing or analysis problem. [57]

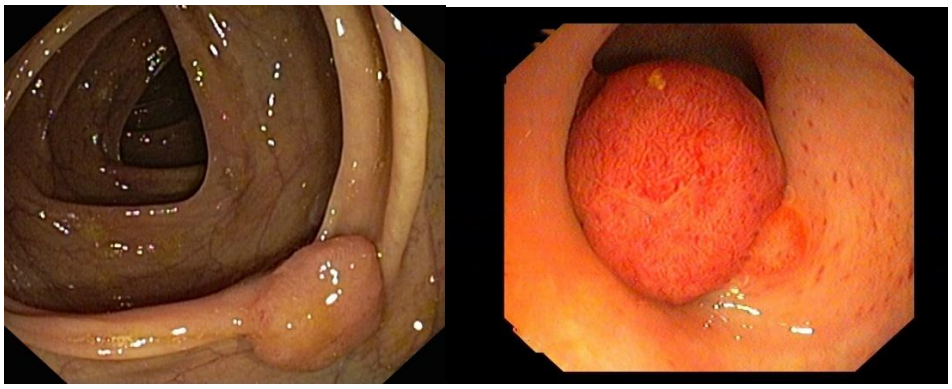
Several image processing techniques provided by the program, like sharpening, cropping and converting to grayscale format, along with the GLCM Texture plugin, that is included in it, are used in the data preprocessing for the polyp classification method proposed in the thesis.

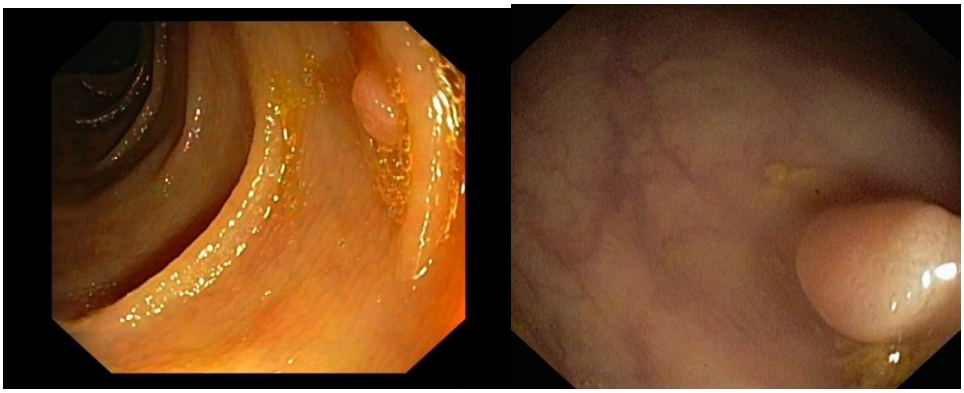
4 MATERIALS AND METHODS

4.1 Data Collection

The data, used in the thesis, were collected in a retrospective way from the personal archive of doctor Konstantinos Patikos. The period during which the images were concentrated is from November 2016 to October 2018. In the data gathered, the screening results indicated whether each individual had polyps. From a total of 750 patients, 1576 images were collected of which the 798 contain polyps and the 778 depict healthy colon. The 798 polyp images are separated into two categories; 424 picture adenomatous polyps and 374 picture hyperplastic polyps. All the data come from standard colonoscope, which uses white light for the inspection of bowel wall and detection of polyps. Every single sample apparently does not contain any name to protect the patient's privacy. The data are not very uniform in general. Not all the images have the same zoom means that some polyps are pictured very big and some very small. Additionally, the images do not have the same overall coloration due to the objective conditions under which they were taken. In order to input the data into the neural network, it was essential to preprocess them in the way explained subsequently.

Below are some examples of the data used:





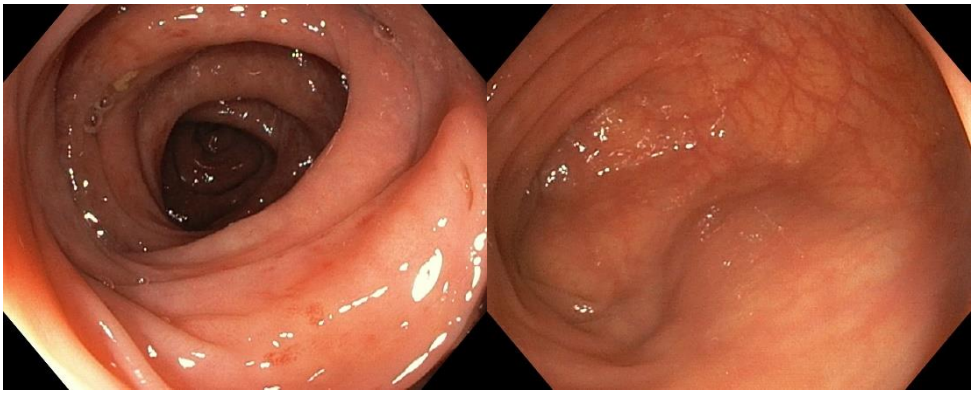


Figure 4.1 Data images: (Top left) Adenomatous polyp, (Top right) Zoomed adenomatous polyp, (Middle left) Hyperplastic polyp, (Middle right) Zoomed hyperplastic polyp, (Bottom left) Healthy colon, (Bottom right) Zoomed healthy colon.

4.2 Data Preprocessing in Deep Learning

As mentioned above, the data had to be preprocessed before they entered the neural networks. Building an effective neural network model requires, among other aspects, careful consideration of the input data format.

The most common image data input parameters are the number of images, image height, image width, number of channels, and the number of levels per pixel. Typically, there are 3 channels of data corresponding to the colors Red, Green, Blue (RGB). Pixel levels are usually [0,255]. [58]

Some of the most popular preprocessing techniques are:

- **Uniform aspect ratio:** One of the first steps is to ensure that the pictures have the same size and aspect ratio. Most of the neural network models assume a square shape input image, which means that each image needs to be checked if it is a square or not and cropped properly. While cropping, the most essential part of the picture is the center. [58]
- **Image Scaling:** When all images are square, it's time to scale each image appropriately. There is a wide variety of up-scaling and down-scaling techniques and it is usually used a library function to do this for us, which is explained subsequently. [58]
- **Mean and Standard Deviation of input data:** Sometimes it's useful to look at the 'mean image' obtained by taking the mean values for each pixel across all training examples. Examining this could give insight into some underlying structures in the pictures. The standard deviation is a measure of the amount of variation or dispersion of a set of values. Higher variance values show up whiter so that it can be understandable where the pictures vary or not. [58]

- **Normalizing image inputs:** Data normalization is an important step that guarantees that each input parameter (pixel, in the case of images) has a similar data distribution. This makes convergence faster while training the network. Data normalization is done by subtracting the mean from each pixel and then dividing the result by the standard deviation. The distribution of such data would resemble a Gaussian curve centered at zero. For image inputs it is needed the pixel values to be positive, so the normalized data are scaled in the range [0,1] or [0, 255]. [58]
- **Dimensionality reduction:** There is the option to collapse the RGB channels into a single gray-scale channel. There are often considerations to reduce other dimensions, when the neural network performance is allowed to be invariant to that dimension, or to make the training problem more tractable. [58]
- **Data augmentation:** As mentioned in the previous chapter, another common pre-processing technique involves augmenting the existing data-set with perturbed versions of the existing images. Scaling, rotations and other affine transformations are characteristic. This is done to expose the neural network to a wide variety of modifications. This makes it less likely that the neural network identifies undesired features in the data-set. [58]

Not all the above-mentioned methods were used for data preprocessing in the current thesis. The exact procedures developed are analyzed below, as well as the python libraries used to accomplish this.

All the data were uploaded in Google Drive in order to have access to them from the Google Colab. They were developed four methods in the thesis; three for the polyp detection problem and one for the polyp classification problem. The main problem encountered at this work was the classification of images to those who were healthy and those who depict a polyp. To deal with it, firstly, from the 1576 images, the 106 were separated to be used in the evaluation of the models. Those 106 images contained healthy, adenomas and hyperplastic images. The remaining 1470 pictures were split in 735 healthy and 735 polyp pictures.

4.2.1 Method 1

4.2.1.1 Data preprocessing

In this method, all the training data were stored in one directory and named by healthy.# or polyp.# for healthy and polyp images respectively. First of all, the images were put in two different lists, one for the healthy and one for the polyp images and then these two lists were concatenated in one, which contained all the training data

(both polyp and healthy). The next step was to shuffle the images in this list. Subsequently, the images were read using the cv2.imread() function and resized to 200x200 using the cv2.resize() function. The images were appended in a new list and their labels to another list. If the name of the image was healthy.#, then the label in the labels list was 1. In the opposite situation the label was 0. Then, these two lists were shuffled again. Some examples of the read and resized images are seen below:

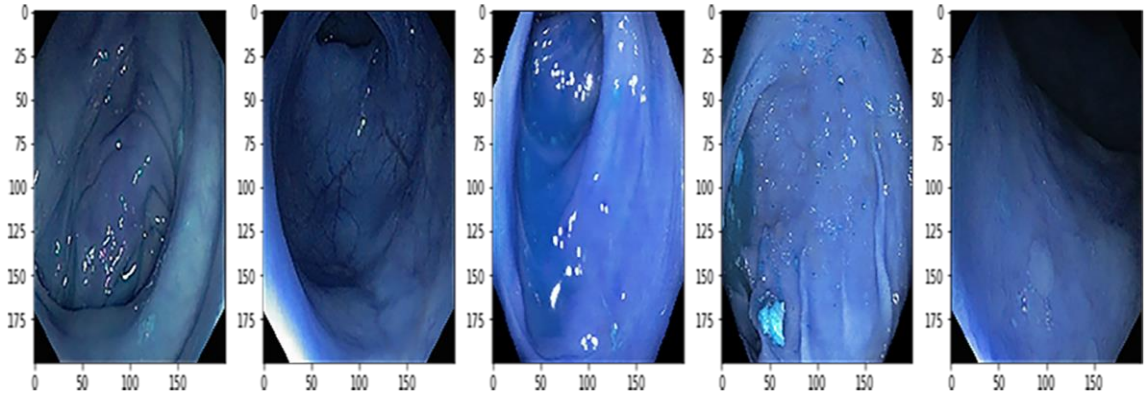


Figure 4.2 Preprocessed healthy and polyp images of method 1.

The next step was to convert the lists to numpy arrays using the np.array() function. This is happening, because the neural networks are mainly expecting to see numpy arrays as inputs. As a result, the shape of the train images array is (1470, 200, 200, 3) and the shape of the labels array is (1470,). These values represent the number of values, the height, the width and the number of channels. Consequently, the data split into train and validation (test) set using the train_test_split() function from sklearn.model_selection. In this method the test set size is the 20% of the train set.

In the data preprocessing of this procedure, image data augmentation was also used, because of the generally small provided dataset in order to prevent overfitting to some extent. Through the ImageDataGenerator of Keras, the train images were rescaled (between 0 and 1), rotated, shifted, sheared, zoomed and flipped. The test images were only rescaled between 0 and 1.

4.2.1.2 Base model architecture

The base model of this method consists of four convolutional layers, one flatten layer, one dense layer, which is typically a fully connected layer and the output layer, which is also a dense layer. All the layers have a ReLU activation function, except for the output layer, which has a sigmoid function because there are only two classes. The convolution layers have a 3x3 convolution filter and a 2x2 max pooling window. The first layer has 32 nodes, the second has 64 and the next two have 128 nodes. Finally, the dense layer before the output has 512 nodes.

In the model.compile() function, binary_crossentropy is used as a loss function, because it is a binary classification problem. The optimizer used in this model is the RMSprop with a learning rate of 0.0001.

4.2.1.3 Method scenarios

Three different scenarios ran on this model, in order to get a general picture of how the model affects the data available. The three different scenarios are:

- **Base model:** Through the model.fit() function the base model explained above trains for 64 epochs with about 36 steps per epoch. The results of the model are analyzed in the next chapter of the thesis.
- **Base model with Dropout:** The change here is that a dropout layer is added between the convolutional layers and the dense layer. This tries to regularize the model. Dropout works by probabilistically removing inputs to a layer, which may be input variables in the data sample or activations from a previous layer. In this way, overfitting is slightly avoided. The actual results are presented in the next chapter.
- **Base model with Dropout and Transfer Learning:** Here the VGG-16 model is added on top of the previous scenario to boost the performance of the base model. It is used to optimize and allow rapid progress or improved performance when modeling the problem. In reality, as seen in the results in the next chapter, it did not help as much as expected, because the task that deals with is completely different from the task in which was pretrained.

  The VGG-16 is a convolutional neural network model introduced by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. [59]

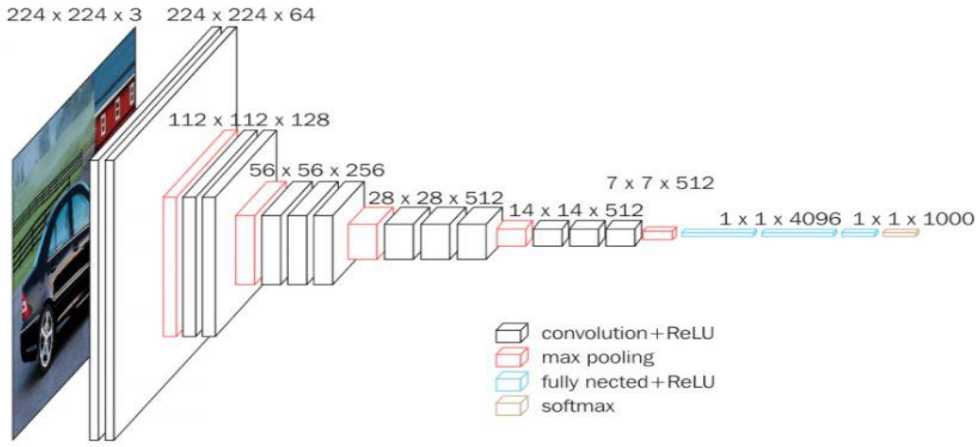


Figure 4.3 VGG-16 network architecture.

4.2.2 Method 2

4.2.2.1 Data preprocessing

In this method, all the training data were stored in one “Training” directory, which contained two subdirectories, “Healthy” and “Polyp”. They were again named by healthy.# or polyp.# for healthy and polyp images respectively. First of all, the images were read and converted to arrays using the cv2.imread() function and then converted to grayscale format using the cv2.IMREAD_GRAYSCALE command. Afterwards, they were resized to 200x200 using the cv2.resize() function. The images were appended in a new list along with their labels. If the name of the directory of the image was healthy, then the label was 0. In the opposite situation the label was 1. Then, the list was shuffled. Some examples of the read and resized images are seen below:

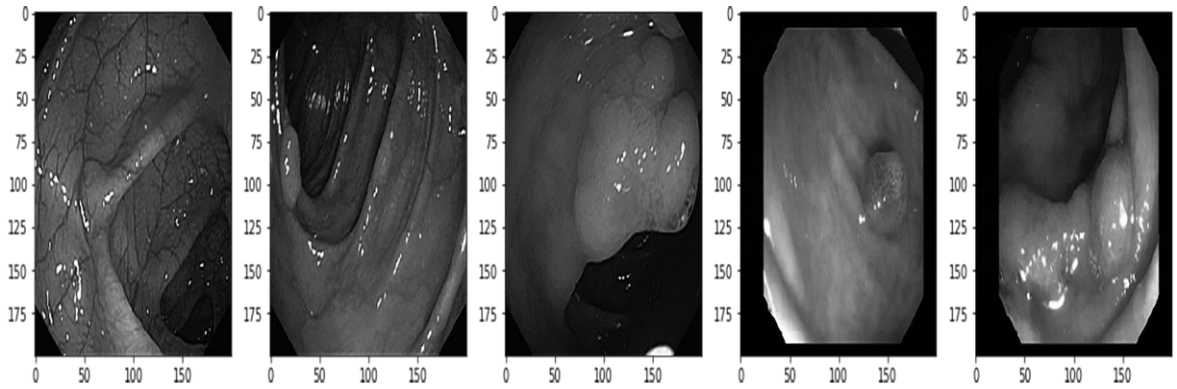


Figure 4.4 Preprocessed healthy and polyp images of method 2.

In the next step, the list X for the features and the list y for the labels are initialized and then the features and the labels are extracted and saved to X and y respectively. Next, the array is reshaped in order to make it suitable for Keras.

Finally, the features and labels of the dataset are pickled to save time, if it is to run the model in the future. This way, the preprocessing is avoided in the next time, because the pickle files can be loaded. A pickle_X file containing the features and a pickle_y file containing the labels are created. The X is normalized between 0 and 1 before entering the network, because the range of 0 to 255, where the pixel values are, is huge. So, the data are ready to be inputted in the convolutional neural network.

Image data augmentation was not initially used in the preprocessing, so there are three scenarios without it and the same three scenarios after it was applied to the data.

4.2.2.2 Base model architecture

First of all, all the necessary libraries are imported. In this model TensorFlow is used. Sequential model is selected. An API which allows to create a custom model layer by layer as it was done in the previous method. Then, Conv2D layers (hidden layers) are added with convolution window 3x3. There are 6 convolutional layers. The first and second have 32 and 64 nodes respectively and the next four have 128 nodes. Again, ReLU activation function is chosen, which is a rectified linear function. It is the most popular and versatile function. Next, the MaxPooling2D layers are using 2x2 window, find the max value ad allow that the value will pass to the next layer. The Flatten layer is used to flat the matrix. It converts 3D array layers to 1D array layers as the dense layer works as a 1D layer. Lastly, the dense layer is used with the last convolution layer because it completes the classification. In the end, a dense layer is again used with output 2, as the classification task is binary.

In this step, the model is compiled. The loss is also a kind of classification function. Binary cross-entropy is used when 2 or more labels are passed but classification probability is needed in one array. Then it distributes the probability in 1 element between 0 and 1, like if there are 2 features in training set then detection probability for 1st become 0-0.5 and for 2nd it will become 0.5-1 Next is categorical cross-entropy**,** which is used when 2 or more inputs are passed and separation probability is wanted. Then it will return a matrix of 2 elements each denoting its probability of similarity. The third one is the sparse categorical cross-entropy. This is the same as the cross-entropy but it is used when the prediction has 9 features and the test image is close to 9th label then it will show something like 9.92. Next, Adam is used as an optimizer. [60]

In the end the model is fit. Here the features and labels are passed, then the epochs and the batch size, in which the data will be processed, are defined. Epochs are the iterations, that the model runs. The validation split, splits test and train data automatically [60]. In this case validation split equals to 20%.

4.2.2.3 Method scenarios

In this method, six scenarios are implemented. In reality, are three scenarios applied with and without image data augmentation in the preprocessing of the data. The scenarios without the image augmentation are:

- **Base model:** Through the model.fit() function the base model explained above trains for 30 epochs with batch size equal to 32. The results of the model are analyzed in the next chapter of the thesis.

- **Base model with Dropout:** Here, a dropout layer is added at the end of every convolutional layer. This tries to regularize the model. Dropout works by probabilistically removing inputs to a layer, which may be input variables in the data sample or activations from a previous layer. In this way, overfitting is slightly avoided. The actual results are presented in the next chapter.
- **Base model with Dropout and Transfer Learning:** In the same way as the previous method 1, the VGG-16 model is added on top of the previous scenario to boost the performance of the base model. It tries to optimize and allow rapid progress or improved performance when modeling the problem. Unfortunately, exactly like method 1, it did not help as much as expected, because the polyp detection task that deals with is completely different from the task in which was pretrained.

After these scenarios were run, image data augmentation applied in precisely the same way as in method 1. That means rescaling (between 0 and 1), rotating, shifting, shearing, zooming and flipping the train images in order to enlarge the initially small dataset. Then the specific scenarios described above were run and the results are presented in chapter 5 of the thesis.

4.2.3 Method 3

4.2.3.1 Data preprocessing

In this method, all the training data were stored in one "Training" directory. They were again labeled by healthy.# or polyp.# for healthy and polyp images respectively. The photos should be reshaped before modeling so that all images have the same shape. This is usually a small square image. There are many techniques to achieve this, although the most common is a simple resize operation like it was done in the previous two methods, that will stretch and deform the aspect ratio of each picture and change its shape. Smaller inputs mean the model is trained faster and typically this matter dominates the selection of image size. In this case, a fixed size of 200×200 pixels are chosen. [61]

Here, the images were preprocessed and loaded progressively utilizing the Keras ImageDataGenerator class and *flow_from_directory()* API. This was slower to execute but gave the possibility to run on more machines. This API prefers data to be divided into separate train and test directories. Under each directory, there is a subdirectory for each class. For example, train/healthy/ and a train/polyp/ subdirectories were created using the makedirs() function and then the same was done for the test images. Images are then organized under the subdirectories. Additionally, it was randomly decided to hold back 10% of the images into the test dataset. This is done by fixing the

seed for the pseudorandom number generator so that it gives the same split of data each time the code is run. [61]

4.2.3.2 Base model architecture

A baseline model established a minimum model performance to which all the upcoming scenarios were compared. This is a model architecture that can be used as the basis of study and improvement. A helpful starting point is the overall architectural principles of the VGG models. This architecture includes piling convolutional layers with small 3×3 filters followed by a max-pooling layer. Collectively, these layers form a block, and these blocks can be duplicated where the number of filters in each block is increased with the depth of the network such as 32, 64, 128, 256 for the first four blocks of the model. In this particular base model, three of these blocks were used. Padding is applied to the convolutional layers to ensure that the height and width shapes of the output feature maps match the inputs of the next layers. Each layer will use the ReLU activation function and the He weight initialization, which are commonly most excellent practices. [61]

The model was fitted with stochastic gradient descent and started with a learning rate of 0.001 and a momentum of 0.9. The problem is a binary classification task, demanding the prediction of one value of either 0 or 1. So, an output layer with 1 node and a sigmoid activation was used and the model was optimized using the binary cross-entropy loss function. [61]

The *flow_from_directory()* function was used on the data generator and created one iterator for each of the train and test directories. Via the class_mode argument, it was specified that the problem is a binary classification problem and via the target_size argument the images were loaded with the size of 200×200 pixels. The batch size was set at 64. Then using the train iterator, the model was fit. The test iterator was employed as a validation dataset while training. The number of steps for the train and test iterators, which is the number of batches that will comprise one epoch, was specified as the length of each iterator and will be the total number of images in the train and test directories divided by the batch size. The model was fit for 18 epochs. [61]

The results are presented in chapter 5.

4.2.3.3 Method scenarios

After the base model was developed, three improving scenarios applied to it in order to avoid overfitting and get better results. The implemented scenarios are:

- **Dropout regularization:** A small amount of dropout was applied after each VGG block of the base model, with more dropout applied to the fully connected layers near the output layer of the model. The model was fit for 40 epochs and slightly

improved the performance of the initial model. All the results are presented in chapter 5.

- **Image Data Augmentation:** A separate ImageDataGenerator instance for the train and test dataset was created. Afterward, iterators for both the train and test sets were generated from the particular data generators. In this scenario, pictures in the training dataset were augmented with 10% random horizontal and vertical shifts and random horizontal flips that make a mirror image of a photo. Pictures in both the train and test sets had their pixel values scaled equivalently. The model was fit for 30 epochs and achieved better performance from the base model by a small margin. The results are presented in chapter 5.
- **Transfer Learning:** Again, the VGG-16 model was used to apply transfer learning to the base model. The feature extraction part of the model was adopted and then a new classifier was added to the model that is tailored to the healthy and polyp dataset. Precisely, the weights of all of the convolutional layers were held fixed during training and only the new fully connected layers were trained to learn to understand the features extracted from the model and make a binary classification. This can be achieved by loading the VGG-16 model, excluding the fully connected layers from the output-end of the model, then combining the new fully connected layers to interpret the model output and make a prediction. The classifier part of the model can be extracted automatically by setting the include_top argument to False, which also requires that the shape of the input also be defined for the model, in this case (224, 224, 3). This means that the loaded model ends at the last max-pooling layer, after which a Flatten layer and the new classifier layers are added. [61]

  The VGG-16 model was trained on the ImageNet challenge dataset. Because of this, it requires input photographs to have the shape of 224×224 pixels. For this reason, the target size used when the dataset of healthy and polyp was loaded, was 224x224. The model also demands images to be centered, meaning they have the mean pixel values from each channel as determined on the ImageNet training dataset subtracted from the input. This was achieved through the ImageDataGenerator by fixing the featurewise_center argument to True and manually defining the mean pixel values to use when centering, as the mean values of the ImageNet training dataset which are [123.68, 116.779, 103.939]. The model was fit for 10 epochs, but the performance was worse than this of the base model. The results are presented in chapter 5.

4.2.4 Method 4

This is the method proposed for classifying the polyp into adenomatous and hyperplastic. This task was completely different from the previous one, as it required a diverse approach to the data preprocessing.

4.2.4.1 Data preprocessing

The work, here, focused in 85 images of adenomatous polyps and 85 images of hyperplastic polyps. The methodology that was followed in the method was performed in ImageJ, an open source image processing program, and is described in detail subsequently. First of all, a small part of the image was cropped. This part was a small portion of the polyp inside each image. The idea meant like doing a biopsy. The next step was to sharpen the images to enhance their texture characteristics. The GLCM Texture plugin, that was used in the process for image texture feature extraction, requires grayscale images. For this reason, after the images were sharpened, they were transformed to 8-bit grayscale format.

Feature Extraction is a method of capturing the visual content of images for indexing and retrieval. Primitive or low-level image features can be either general features, such as extraction of color, texture, shape or domain-specific features. [62]

The Gray Level Co-occurrence Matrix (GLCM) method is a way of extracting second-order statistical texture features and has been used in several applications. The four features that were computed, using this method, are Angular Second Moment, Correlation, Inverse Difference Moment and Entropy and are explained below:

- Angular Second Moment is also known as Uniformity or Energy. It is the sum of squares of entries in the GLCM. It measures the image homogeneity and it is high when an image has very good homogeneity or when pixels are very similar. [62]
- Correlation measures the linear dependency of grey levels of neighboring pixels. This is often used to measure deformation, displacement, strain and optical flow, but it is widely applied in many areas of science and engineering. One very common application is for measuring the motion of an optical mouse. [62]
- Inverse Difference Moment (IDM) is the local homogeneity. It is high when local gray level is uniform and inverse GLCM is high. [62]
- Entropy shows the amount of information of the image that is needed for the image compression. Entropy measures the loss of information or message in a transmitted signal and also measures the image information. [62]

After the feature extraction was applied, all the measurements were saved in a csv file. This file had five columns; four that corresponded to the four features explained above and one that describes the type of the polyp (0 for adenomatous polyps and 1 for hyperplastic polyps). The final step before the dataset is inputted in the neural network was to normalize the data. This was done by computing the mean and the standard deviation of the dataset and then subtracting the mean from the dataset and divide the result by the standard deviation.

4.2.4.2 Neural network architecture

The neural network that was used here, was consisted of the input, the output layer and two hidden layers between them. The input layer had 10 nodes, the first hidden layer had 20, the second had 10 and lastly the output layer obviously had 2. All the layers used the ReLU activation function, except for the output that used the Soft-max function. The dataset split with 80% to be the train set and 20% to be the test set. The model was compiled using categorical_crossentropy as a loss function and the Adam optimizer. It was trained for 15 epochs with a batch size of 8. The results of the method are presented in chapter 5.

5 PRESENTATION AND COMPARISON OF METHODS RESULTS

5.1 Introduction

As mentioned above the two tasks of the present work were, firstly, to identify whether there is a polyp in the input image and further to classify the polyp in two different categories (adenomas or hyperplastic). So, in the first case, the classification took place between "Healthy" and "Polyp" images and then between "Adenomas" and "Hyperplastic" images. The methods 1, 2 and 3 are dealing with the first classification task and method 4 with the following. In the initial part of the chapter, the metrics that were used to quantify the efficiency of each method are presented. Then, the results of every method are displayed separately. Finally, those results are compared to each other in tables to help reach some conclusions.

5.2 Metrics for the quantification of method efficiency

The indicators used to quantify the effectiveness of the proposed method are sensitivity, specificity, and accuracy, which are indicators widely used in medical diagnostics. High sensitivity means a high ability to detect images with polyps. High specificity means a high ability to avoid false detection of a polyp. Accuracy is generally used to evaluate the overall performance of the proposed method. Since the algorithms are used to detect images with polyps and more specifically with cancerous polyps and send them to doctors for a specific examination, sensitivity is more important than specificity and accuracy.

Sensitivity is defined as the ability to correctly identify a true, true positive (TP) or true negative (TN) sample, in this case, the pathological images containing polyps.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \tag{4.1}$$

Specificity defines the ability to recognize and correctly identify only true negative (TN) images.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \tag{4.2}$$

Accuracy defines the possibility of true positive (TP) and true negative (TN) images being correctly identified.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \tag{4.3}$$

In the above equations:

TP is the number of positive samples (images) that are correctly classified.

FN is the number of positive samples (images) classified as negative.

TN is the number of negative samples (images) that are correctly classified.

FP is the number of negative samples (images) that are incorrectly classified as positive.

In order to measure the quantitative performance of the proposed method, the values of the following indices were calculated: FNR (False Negative Ratio), FPR (False Positive Ratio) and precision. The doctor's interest is not to diagnose a polyp image as healthy. So, it is desirable to have a minimal FNR value.

In general, the results fall into one of the following four categories: true positive (TP), true negative (TN), false positive FP or false negative, FN as described above.

The two significant true situations are:

1. TP: This is the number of pathological images, those that are containing polyps, that are correctly classified as "Polyp" images.
2. TN: This is the number of healthy images, those that are not containing polyps, that are correctly sorted as "Healthy" images.

By analogy, there are two important false situations:

1. FN: This is the number of pathological images, those that are containing polyps, that are not properly classified as "Healthy" images.
2. FP: This is the number of healthy images, those that are not containing polyps, that are mistaken for "Polyp" images.

From these four logic states, three more criteria are calculated to measure the performance. These are:

1. Precision
2. False Positive Ratio (FPR)
3. False Negative Ratio (FNR)

which are given by the following equations:

Precision defines the ability to correctly identify, true positive (TP) and true negative (TN), the polyp in the image. It is desirable to have as high a value as possible.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{4.4}$$

FPR (False Positive Ratio) is an error indicator, so it is desirable to have the lowest value possible. Defined as the probability of mistakenly identifying healthy images as pathological images, containing a polyp.

$$\text{FPR} = \frac{\text{FP}}{\text{TN+FP}} \tag{4.5}$$

FNR (False Negative Ratio) is also an error indicator, so it is desirable to have a low value too. Defined as the probability of a pathological image, containing a polyp, being mistakenly recognized as healthy.

$$\text{FNR} = \frac{\text{FN}}{\text{TP+FN}} \tag{4.6}$$

Precision quantifies how realistic the assumption of a polyp image is, in the sense that a higher value indicates an actually "true" result, and therefore more desirable. On the other hand, being an error indicator, a low accuracy value means higher values for FPR and FNR.

The last metric used to evaluate the proposed methods is the Confusion Matrix. A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix explains how the classification model is confused when it makes predictions. It gives insight, not only into the errors being made by a classifier but more importantly the types of errors that are being made. [63]

The format of a confusion matrix is the following:

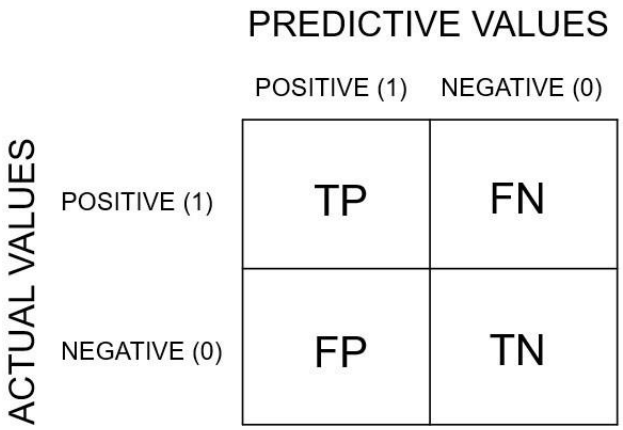


Figure 5.1 Confusion matrix template.

The definitions for the TP, TN, FP and FN were described previously in this chapter.

In order to make the analysis of the first three proposed methods as reliable as possible, 106 images were studied. From these 106 images, 53 contained polyps and 53 were depicting a healthy colon. In the case of the fourth method, which deals with the classification of polyps to adenomas or hyperplastic, 34 images were examined. From these 34 images, 18 contained adenomas and 16 included hyperplastic polyps. Examples of all the images described above are shown in the appendix AP2.

All the images, used for the prediction, were not used in the training and validation phases of the models of the proposed methods, in order to make the prediction as real as possible.

5.3 Presentation of the methods results

5.3.1 Results of Method 1

In this method, three scenarios were implemented. The first scenario is representing the baseline model of the method without any techniques applied to avoid overfitting. In the second and third scenario, dropout and transfer learning are added to the baseline model of the first scenario, respectively. From the evaluation-prediction phase of the models, the resulting outcomes are the following:

Scenario 1

After the prediction with the set of the 106 "new" images, the TP, TN, FP and FN were measured as:

TP = 47, TN = 46

FN = 6, FP = 7

Considering these values, all the metrics explained above were calculated and presented in the following Table 5.1:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 1	87,5%	88,7%	86.8%	87,1%	13.2%	11.3%

Table 5.1 Results of Scenario 1 of Method 1.

The last metric that was calculated was the confusion matrix, as described above and is presented in the following figure:

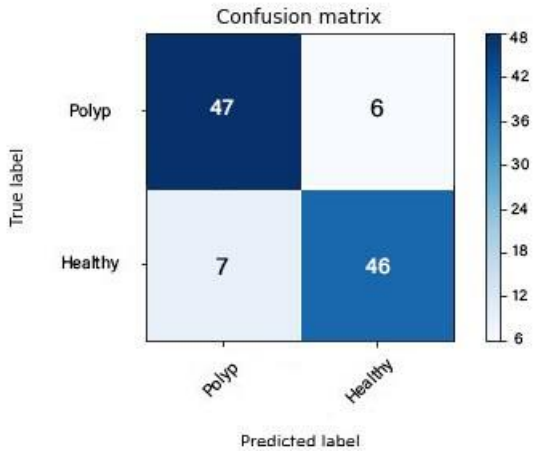


Figure 5.1 Confusion matrix of Scenario 1 of Method 1.

Scenario 2

In this scenario, dropout regularization is added. After the prediction with the set of the 106 "new" images, the TP, TN, FP and FN were measured as:

TP = 49, TN = 46

FN = 4, FP = 7

Considering these values, all the metrics explained above were calculated and presented in the following Table 5.2:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 2	89%	92,5%	86.8%	87,5%	13.2%	7.5%

Table 5.2 Results of Scenario 2 of Method 1.

The last metric that was calculated was the confusion matrix, as described above and is presented in the following figure:

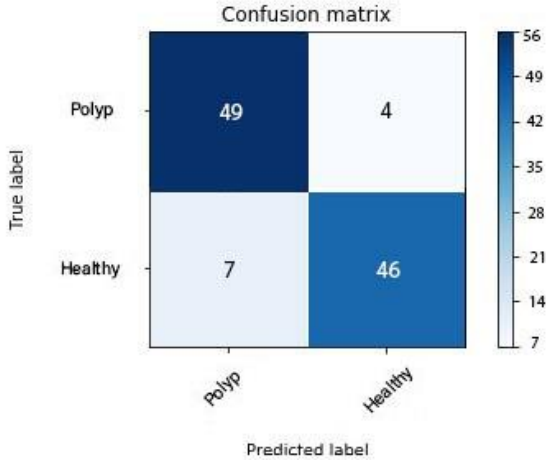


Figure 5.2 Confusion matrix of Scenario 2 of Method 1.

Scenario 3

In this scenario, transfer learning is added to the previous implementation. After the prediction with the set of the 106 "new" images, the TP, TN, FP and FN were measured as:

TP = 45, TN = 44

FN = 8, FP = 9

Considering these values, all the metrics explained above were calculated and presented in the following Table 5.3:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 3	84%	84,9%	83.1%	83,3%	16.9%	15.1%

Table 5.3 Results of Scenario 3 of Method 1.

The last metric that was calculated was the confusion matrix, as described above and is presented in the following figure:

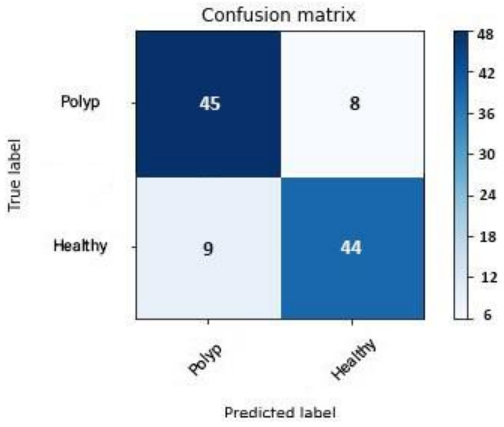


Figure 5.3 Confusion matrix of Scenario 3 of Method 1.

Finally, all the results of the three Scenarios of Method 1 are presented cumulatively in the following Table 5.4:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 1	87,5%	88,7%	86.8%	87,1%	13.2%	11.3%
Scenario 2	89%	92,5%	86.8%	87,5%	13.2%	7.5%
Scenario 3	84%	84,9%	83.1%	83,3%	16.9%	15.1%

Table 5.4 Results of Method 1.

It is understandable that in this case, the dropout regularization is helping a lot the model to achieve better performance. On the other hand, the transfer learning technique, not only does not boost its effectiveness, but it makes it worse than the initial baseline model.

5.3.2 Results of method 2

In this method, six scenarios were implemented. These scenarios are divided into two subcategories. Three scenarios without using data augmentation and three scenarios using it to avoid overfitting. In both the subgroups, the first scenario is representing the baseline model, with and without data augmentation respectively, and then in scenarios two, three, five and six, dropout regularization and transfer learning are applied to enhance the performance of the model. From the evaluation-prediction phase of the models, the resulting outcomes are the following:

Scenario 1

After the prediction with the set of the 106 "new" images, the TP, TN, FP and FN were measured as:

TP = 46, TN = 45

FN = 7, FP = 8

Considering these values, all the metrics explained above were calculated and presented in the following Table 5.5:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 1	86%	86,8%	84.9%	85,2%	15.1%	13.2%

Table 5.5 Results of Scenario 1 of Method 2.

The last metric that was calculated was the confusion matrix, as described above and is presented in the following figure:

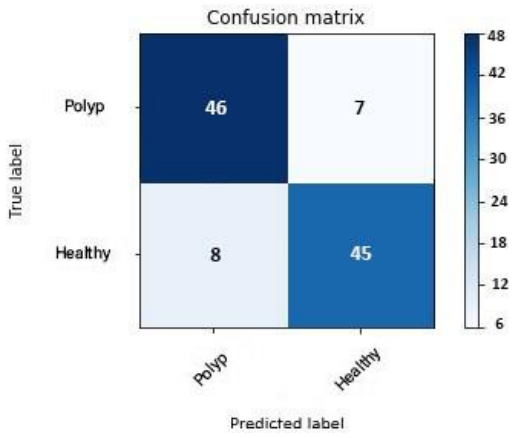


Figure 5.4 Confusion matrix of Scenario 1 of Method 2.

Scenario 2

In this scenario, dropout regularization is added to the baseline model of the previous scenario. After the prediction with the set of the 106 “new” images, the TP, TN, FP and FN were measured as:

TP = 50, TN = 46

FN = 3, FP = 7

Considering these values, all the metrics explained above were calculated and presented in the following Table 5.6:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 2	90.3%	94,4%	86.8%	87,8%	13.2%	5.6%

Table 5.6 Results of Scenario 2 of Method 2.

The last metric that was calculated was the confusion matrix, as described above and is presented in the following figure:

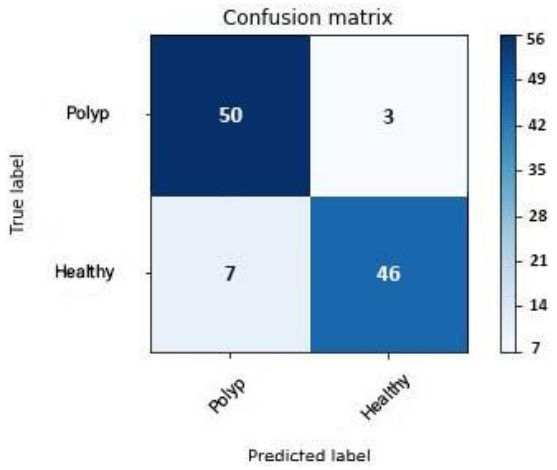


Figure 5.5 Confusion matrix of Scenario 2 of Method 2.

Scenario 3

In this scenario, transfer learning is added to the previous implementation. After the prediction with the set of the 106 “new” images, the TP, TN, FP and FN were measured as:

TP = 46, TN = 44

FN = 7, FP = 9

Considering these values, all the metrics explained above were calculated and presented in the following Table 5.7:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 3	85.1%	86,8%	83.2%	83,6%	16.9%	13.2%

Table 5.7 Results of Scenario 3 of Method 2.

The last metric that was calculated was the confusion matrix, as described above and is presented in the following figure:

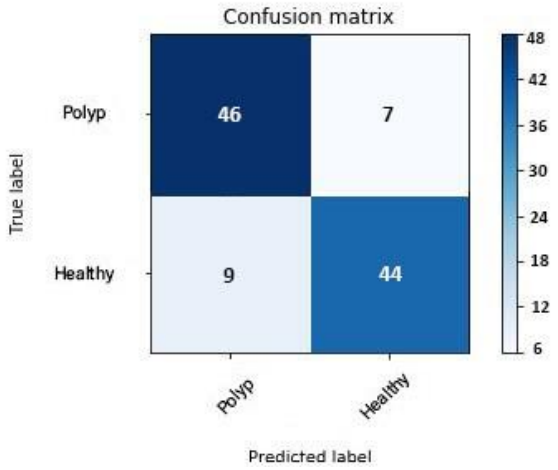


Figure 5.6 Confusion matrix of Scenario 3 of Method 2.

Scenario 4

In this scenario, data augmentation is added to the preprocessing of the input images and the baseline model of Scenario 1 is executed. After the prediction with the set of the 106 “new” images, the TP, TN, FP and FN were measured as:

TP = 48, TN = 45

FN = 5, FP = 8

Considering these values, all the metrics explained above were calculated and presented in the following Table 5.8:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 4	87.8%	90,6%	84.9%	85,7%	15.1%	9.4%

Table 5.8 Results of Scenario 4 of Method 2.

The last metric that was calculated was the confusion matrix, as described above and is presented in the following figure:

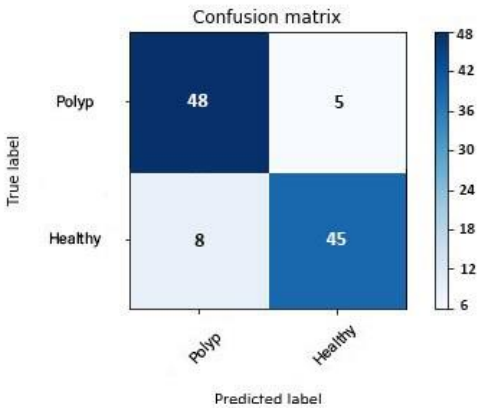


Figure 5.7 Confusion matrix of Scenario 4 of Method 2.

Scenario 5

In this scenario, data augmentation is added to the preprocessing of the input images and the baseline model with dropout regularization of Scenario 2 is executed. After the prediction with the set of the 106 "new" images, the TP, TN, FP and FN were measured as:

TP = 50, TN = 47

FN = 3, FP = 6

Considering these values, all the metrics explained above were calculated and presented in the following Table 5.9:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 5	91.8%	94,4%	88.7%	89,3%	11.3%	5.6%

Table 5.9 Results of Scenario 5 of Method 2.

The last metric that was calculated was the confusion matrix, as described above and is presented in the following figure:

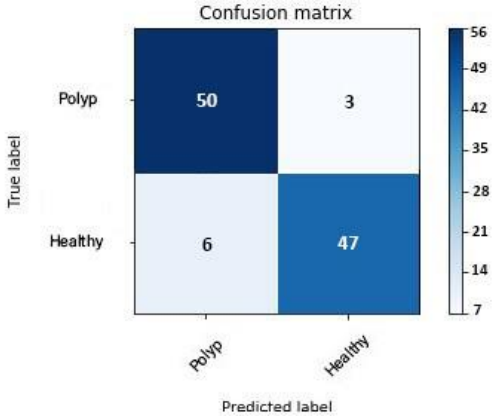


Figure 5.8 Confusion matrix of Scenario 5 of Method 2.

Scenario 6

In this scenario, data augmentation is added to the preprocessing of the input images and the baseline model with dropout regularization and transfer learning of Scenario 3 is executed. After the prediction with the set of the 106 "new" images, the TP, TN, FP and FN were measured as:

TP = 47, TN = 45

FN = 6, FP = 8

Considering these values, all the metrics explained above were calculated and presented in the following Table 5.10:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 6	86.4%	88,7%	84.9%	85,4%	15.1%	11.3%

Table 5.10 Results of Scenario 6 of Method 2.

The last metric that was calculated was the confusion matrix, as described above and is presented in the following figure:

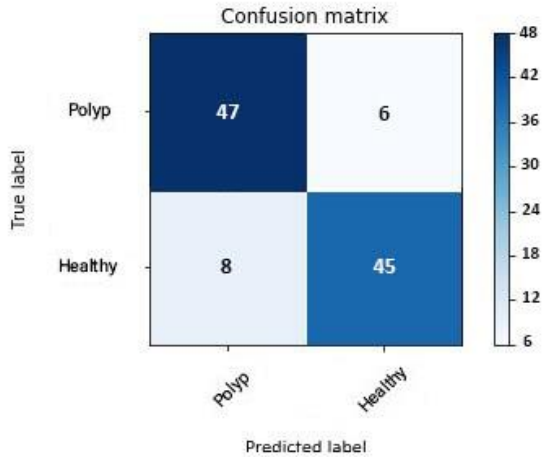


Figure 5.9 Confusion matrix of Scenario 6 of Method 2.

Finally, all the results of the six Scenarios of Method 2 are presented cumulatively in the following Table 5.11:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 1	86%	86,8%	84.9%	85,2%	15.1%	13.2%
Scenario 2	90.3%	94,4%	86.8%	87,8%	13.2%	5.6%
Scenario 3	85.1%	86,8%	83.2%	83,6%	16.9%	13.2%
Scenario 4	87.8%	90,6%	84.9%	85,7%	15.1%	9.4%
Scenario 5	91.8%	94,4%	88.7%	89,3%	11.3%	5.6%
Scenario 6	86.4%	88,7%	84.9%	85,4%	15.1%	11.3%

Table 5.11 Results of Method 2.

As it is shown from the above table, in this case, the dropout regularization is improving a lot the model's performance as it does in the previous method discussed. On the other hand, it is again clear that the transfer learning technique, not only does not boost its effectiveness, but it makes it worse than the initial baseline model like in Method 1. The data augmentation that is added in this method seems to help raise the model's efficacy to some extent.

5.3.3 Results of Method 3

In this method, four scenarios were implemented. The first scenario consists of the baseline model of the method and the following scenarios are attempts to improve the initial model. The first attempt, which the second scenario of the method, is the dropout regularization, the next is transfer learning and the last is data augmentation in the preprocessing of the input data. The improvements that were tested in the base models of the previous two methods. From the evaluation-prediction phase of the models, the resulting outcomes are the following:

Scenario 1

After the prediction with the set of the 106 “new” images, the TP, TN, FP and FN were measured as:

TP = 49, TN = 46

FN = 4, FP = 7

Considering these values, all the metrics explained above were calculated and presented in the following Table 5.12:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 1	90.1%	92,5%	86.8%	87,5%	13.3%	7.5%

Table 5.12 Results of Scenario 1 of Method 3.

The last metric that was calculated was the confusion matrix, as described above and is presented in the following figure:

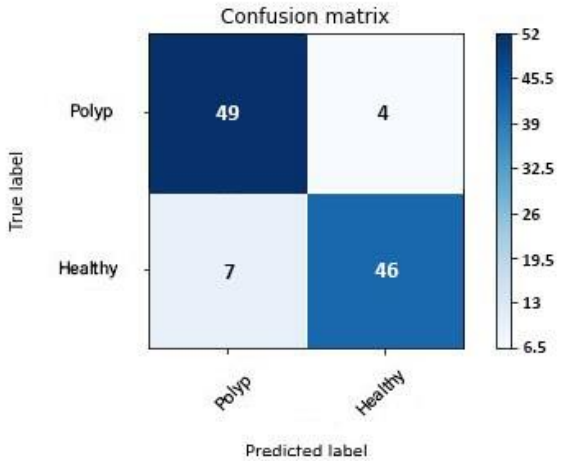


Figure 5.9 Confusion matrix of Scenario 1 of Method 3.

Scenario 2

In this scenario, dropout regularization is added to the baseline model of the previous scenario. After the prediction with the set of the 106 "new" images, the TP, TN, FP and FN were measured as:

TP = 50, TN = 48

FN = 3, FP = 5

Considering these values, all the metrics explained above were calculated and presented in the following Table 5.13:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 2	92.2%	94,4%	90.6%	90,9%	9.4%	5.6%

Table 5.13 Results of Scenario 2 of Method 3.

The last metric that was calculated was the confusion matrix, as described above and is presented in the following figure:

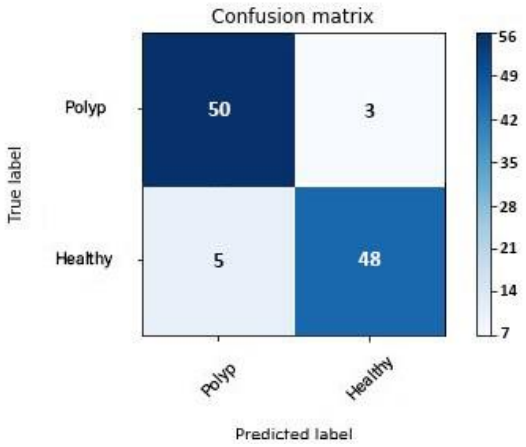


Figure 5.10 Confusion matrix of Scenario 2 of Method 3.

Scenario 3

In this scenario, transfer learning is added to the baseline model of Scenario 1. After the prediction with the set of the 106 "new" images, the TP, TN, FP and FN were measured as:

TP = 45, TN = 41

FN = 8, FP = 12

Considering these values, all the metrics explained above were calculated and presented in the following Table 5.14:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 3	81.2%	84,9%	77.4%	78,9%	22.6%	15.1%

Table 5.14 Results of Scenario 3 of Method 3.

The last metric that was calculated was the confusion matrix, as described above and is presented in the following figure:

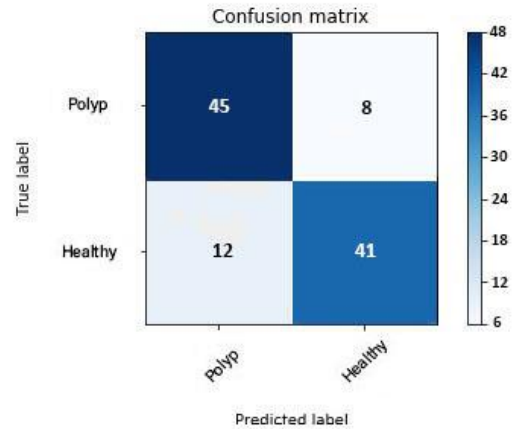


Figure 5.11 Confusion matrix of Scenario 3 of Method 3.

Scenario 4

In this scenario, data augmentation is added to the preprocessing of the input images and then the baseline model of Scenario 1 is executed. After the prediction with the set of the 106 “new” images, the TP, TN, FP and FN were measured as:

TP = 49, TN = 48

FN = 4, FP = 5

Considering these values, all the metrics explained above were calculated and presented in the following Table 5.15:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 4	91.1%	92,5%	90.5%	90,7%	9.4%	7.5%

Table 5.15 Results of Scenario 4 of Method 3.

The last metric that was calculated was the confusion matrix, as described above and is presented in the following figure:

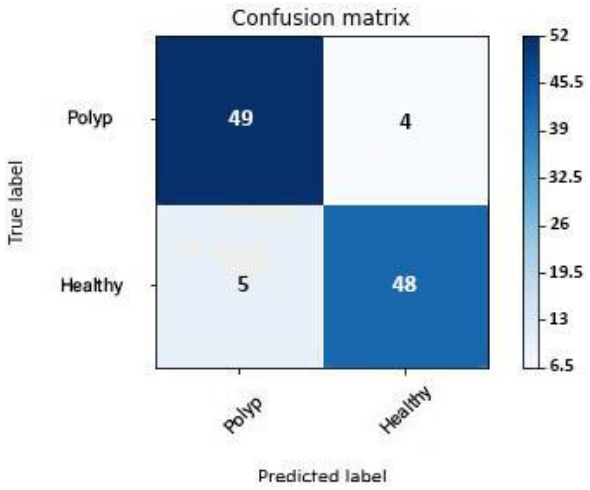


Figure 5.12 Confusion matrix of Scenario 4 of Method 3.

Finally, all the results of the six Scenarios of Method 2 are presented cumulatively in the following Table 5.16:

Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Scenario 1	90.1%	92,5%	86.8%	87,5%	13.3%	7.5%
Scenario 2	92.2%	94,4%	90.6%	90,9%	9.4%	5.6%
Scenario 3	81.2%	84,9%	77.4%	78,9%	22.6%	15.1%
Scenario 4	91.1%	92,5%	90.5%	90,7%	9.4%	7.5%

Table 5.16 Results of Method 3.

The above table shows that, in this case, the dropout regularization is improving the model's performance as it does in the previous two methods discussed. However, it is again clear that the transfer learning technique makes it worse than the initial baseline model like in Method 1 and Method 2. In fact, in this method transfer learning achieves the lowest score from all the scenarios in every other method, meaning that it is not helping at all. The data augmentation that is added in this method seems to help raise the model's efficiency to some extent, like in Method 2.

5.3.4 Results of Method 4

This is the method proposed for classifying the polyp into adenomatous and hyperplastic. There is only one scenario, here, which is the baseline model without any techniques applied to avoid overfitting or enhance the performance. It was trained, tested and evaluated only with a few images. After the prediction with the set of the 34 “new” images, the TP, TN, FP and FN were measured as:

TP = 16, TN = 13

FN = 2, FP = 3

Considering these values, all the metrics explained above were calculated and presented in the following Table 5.17:

Method	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Method 4	85%	88.8%	81.3%	84,2%	18.7%	11.1%

Table 5.17 Results of Method 4.

The last metric that was calculated was the confusion matrix, as described above and is presented in the following figure:

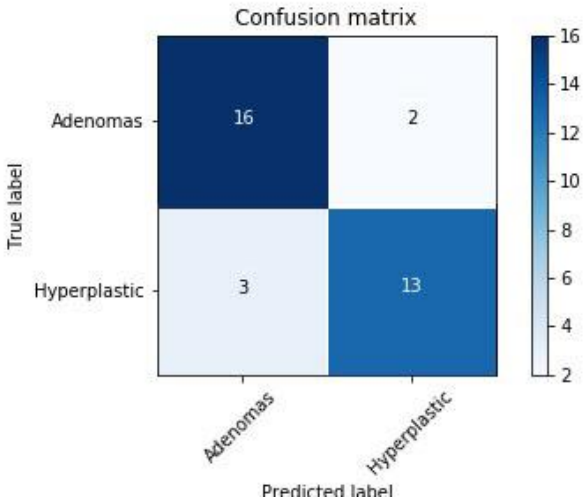


Figure 5.13 Confusion matrix of Method 4.

The plots of training and validation accuracy and training and validation loss of all the above scenarios are presented in appendix AP3.

6 CONCLUSIONS AND FUTURE WORK

6.1 Comparison of the methods results

In this section, the results of every method's scenarios are presented in the following Table 6.1. This helps to reach some conclusions about the efficiency of the proposed methods and decide which performs the best when dealing with the polyp classification task.

Methods	Scenario	Accuracy	Sensitivity	Specificity	Precision	FPR	FNR
Method 1	**Scenario 1**	87,5%	88,7%	86.8%	87,1%	13.2%	11.3%
	Scenario 2	89%	92,5%	86.8%	87,5%	13.2%	7.5%
	Scenario 3	84%	84,9%	83.1%	83,3%	16.9%	15.1%
Method 2	**Scenario 1**	86%	86,8%	84.9%	85,2%	15.1%	13.2%
	Scenario 2	90.3%	94,4%	86.8%	87,8%	13.2%	5.6%
	Scenario 3	85.1%	86,8%	83.2%	83,6%	16.9%	13.2%
	Scenario 4	87.8%	90,6%	84.9%	85,7%	15.1%	9.4%
	Scenario 5	91.8%	94,4%	88.7%	89,3%	11.3%	5.6%
	Scenario 6	86.4%	88,7%	84.9%	85,4%	15.1%	11.3%
Method 3	**Scenario 1**	90.1%	92,5%	86.8%	87,5%	13.3%	7.5%
	Scenario 2	92.2%	94,4%	90.6%	90,9%	9.4%	5.6%
	Scenario 3	81.2%	84,9%	77.4%	78,9%	22.6%	15.1%
	Scenario 4	91.1%	92,5%	90.5%	90,7%	9.4%	7.5%
Method 4		85%	88.8%	81.3%	84,2%	18.7%	11.1%

Table 6.1 Results of all Methods.

6.2 Conclusions

From the results of the previous section, it can be seen that the scenarios with the best performance over the detection of colorectal polyps are the Method's 2 Scenario 2, the Method's 2 Scenario 5 and the Method's 3 Scenario 2. As the doctor wants not to diagnose a polyp image as healthy, it is beneficial to have a minimal FNR value. The lowest FNR value is achieved by these three proposed methods. In addition, high sensitivity means a high ability to detect images with polyps, so it is essential for the proposed models to gain high sensitivity. The above-mentioned scenarios perform the highest score in sensitivity.

Furthermore, it is clear that in all the first three methods, wherever the dropout regularization was used, it seemed to improve the accuracy and more significantly the sensitivity of the baseline model. Most importantly it reduced overfitting and improved the generalization error in the neural networks, as this was the biggest problem for the proposed models in the thesis, because of the relatively small and unbalanced dataset.

Exactly like the dropout regularization, the image data augmentation in the preprocessing phase seemed to exceed the baseline models, in the scenarios that was utilized. This is because the convolutional neural networks, that are used in the present work, need as much as possible, a large training dataset with many variations in order to be trained accurately.

Unfortunately, it is obvious that the transfer learning technique underperformed in all the scenarios in which it was used. It worsened the performance of the initial model and did not provide any improvements. It did not help as much as expected, because the polyp detection task that confronts in this implementation, is completely different from the ImageNet classification task in which was pre-trained.

Concerning the polyp classification task in adenomatous and hyperplastic, the only method, that was used, seems to give hopeful results. The feature extraction, using the GLCM method, in the preprocessing of the input images, gave some very useful information about the texture differences between the two polyp categories and more specifically in entropy and angular second moment (Uniformity or Energy), which represents the image homogeneity. By focusing on these exact contrasts, a worthy classification can be achieved, that it will help the doctors in the future. The results of the method are not great, but they can be improved, as it is discussed in the next section.

6.3 Future work

The purposes of the above study were the detection and the classification of colorectal polyps detected during standard colonoscopy using deep learning algorithms and more specifically convolutional neural networks. Artificial Intelligence techniques, like deep learning, allow advanced processing of massive amounts of image data and it can potentially help clinical doctors in decision making, in problems such as detection and classification of colorectal polyps. The above-explained proposed methods can be improved in the future in some key points like:

- Expand the training dataset, because a much bigger sample set is needed to train such types of algorithms and get satisfactory results.
- Improve the quality of the training dataset, as the given images had a lot of differences in illumination, zoom, focus etc. In addition, too many similar samples should be bypassed to avoid overfitting. This makes it difficult for the neural network to classify the images in the correct way.
- Different type of preprocessing of the training dataset. This could be some feature extraction algorithms or edge detection algorithms to identify the polyp in the image, like the Canny edge detector.
- Use of some other pre-trained network for the Transfer Learning technique, such as ResNet or AlexNet and maybe use the Fine-Tuning technique to boost its performance.
- Better tune and experiment with the convolutional neural networks' parameters, like epochs, learning rate, optimizer type etc.
- Develop the algorithms to detect and classify colorectal polyps in a real time application during colonoscopy. A well-trained AI tool of this type would give a great advantage to the doctors, to avoid errors in their decisions.

Concerning, the polyp classification problem, the proposed method is only a primitive approach to the task. Much more improvements and development are needed in order to have some real results. The training and testing phases of the algorithm were performed with an obviously inadequate dataset. Hence, a much larger database is required to deal with the problem effectively. The convolutional neural network utilized in this method is a very basic one. This may lead the model to underfit, because of the high bias and the low variance. So, the development of that algorithm is essential. A deeper network topology surely can be applied, along with improving techniques like dropout regularization. The parameters of the network also can be tuned

differently and give better results. Furthermore, maybe a different normalization technique in the input data, like Min-Max normalization, can be applied and help the neural network to train better. After the preprocessing of the data and the feature extraction through the GLCM method, a feature selection algorithm, like Mutual Information Maximization (MIM) or Conditional Mutual Information Maximization (CMIM), could be performed to choose the most significant features to be used in the classification. All these improvements will affect the accuracy, sensitivity, and specificity of the model and deal with the overfitting and underfitting problems to some extent.

7 BIBLIOGRAPHY

[1] D. Poole, A. Mackworth και R. Goebel, «Computational intelligence: A logical approach.,» *New York: Oxford University Press,* 1998.

[2] M. Alagappan, J. R. G. Brown, Y. Mori και T. M. Berzin, «Artificial intelligence in gastrointestinal endoscopy: The future is almost here,» *World J Gastrointest Endosc.,* τόμ. 10(10), p. 239–249, 2018.

[3] T. Mitchell, «Machine learning.,» *New York: McGraw-Hill,* p. 414, 1997.

[4] S. Russell και P. Norvig, «Artificial intelligence: a modern approach (3rd Edition),» *Upper Saddle River: Prentice Hall,* p. 1132, 2010.

[5] Y. Mori, S. Kudo, T. Berzin, M. Misawa και K. Takeda, «Computer-aided diagnosis for colonoscopy.,» *Endoscopy.,* τόμ. 49, p. 813–819, 2017.

[6] M. Liedlgruber και A. Uhl, «Computer-aided decision support systems for endoscopy in the gastrointestinal tract: A review.,» *IEEE Rev. Biomed. Eng.,* τόμ. 4, p. 73–88, 2011.

[7] W.-L. Chao, H. Manickavasagan και S. G. Krishna, «Application of Artificial Intelligence in the Detection and Differentiation of Colon Polyps: A Technical Review for Physicians,» *Diagnostics (Basel),* τόμ. 9(3), p. 99, 2019.

[8] D. K. Iakovidis, D. E. Maroulis και S. A. Karkanis, «An intelligent system for automatic detection of gastrointestinal adenomas in video endoscopy,» *Computers in Biology and Medicine,* τόμ. 36, p. 1084–1103, 2006.

[9] S. Itzkowitz και Y. Kim, «Sleisinger & Fordtran's Gastrointestinal and Liver Disease sixth ed,» *WB Saunders Company,* τόμ. 2, 1998.

[10] C. Johnson και A. Dachman, «CT colonography: the next colon screening examination,» *Radiology,* τόμ. 216, pp. 331-341, 2000.

[11] D. Rex, R. Weddle, D. Pound, K. O'Connor, R. Hawes, R. Dittus, J. Lappas και L. Lumeng, «Flexible sigmoidoscopy plus air contrast barium enema versus colonoscopy for suspected lower gastrointestinal bleeding,» *Gastroenterology,* τόμ. 98, pp. 855-861, 1990.

[12] J. Ferlay, I. Soerjomataram, R. Dikshit, S. Eser, C. Mathers, M. Rebelo, D. Parkin, D. Forman και F. Bray, «Cancer incidence and mortality worldwide: sources, methods and major patterns in GLOBOCAN 2012,» *Interational Journal of Cancer,* p. E359–E386, 2015.

[13] H. Brenner και C. Chec, «The colorectal cancer epidemic: challenges and opportunities for primary, secondary and tertiary prevention,» *British Journal of Cancer,* τόμ. 119(7), pp. 785-792, 2018.

[14] H. Brenner, L. Altenhofen, C. Stock και M. Hoffmeister, «Natural history of colorectal adenomas: birth cohort analysis among 3.6 million participants of screening colonoscopy.,» *Cancer Epidemiol. Biomark.,* τόμ. 22, pp. 1043-1051, 2013.

[15] M. P. Colluci, H. S. Yale και J. C. Pall, «Colorectal Polyps,» *Clin Med Res,* pp. 261-262, 2003.

[16] P. Correa, «Epidemiology of polyps and cancer in the human intestine.,» *Gastroenterology,* τόμ. 77, pp. 1245-1251, 1979.

[17] A. Williams, B. Balasooriya και D. Day, «Polyps and cancer of the large bowel: A necropsy study in Liverpool.,» *Gut.,* τόμ. 23, pp. 835-842, 1982.

[18] J. Bond, «Polyp Guideline: Diagnosis, Treatment and Surveillance for Patients With Colorectal Polyps.,» *Am J Gastroenterol,* τόμ. 95, pp. 3053-3063, 2000.

[19] C. Fenoglio-Preiser και R. Hutter, «Colorectal Polyps: Pathologic diagnosis and clinical significance.,» *Ca Cancer J Clin,* τόμ. 35, pp. 322-344, 1985.

[20] C. Fenoglio και R. Pascal, «Colorectal adenomas and cancer: Pathologic relatioships.,» *Cancer,* τόμ. 50, pp. 2601-2608, 1982.

[21] B. Morson και L. Sobin, «Histological typing of intestinal tumours. International Histological Classification of Tumours,» *Journal of Clinical Pathology,* τόμ. 30(7), p. 685, 1977.

[22] F. Konishi και B. Morson, «Pathology of colorectal adenomas: A colonoscopic survey.,» *Journal of Clinic Pathology,* τόμ. 35, pp. 830-841, 1982.

[23] M. O'Brien, W. SJ., A. Zauber, L. Gottlieb, S. Sternberg, B. Diaz, G. Dickersin, S. Ewing, S. Geller και D. Kasimian, «The National Polyp Study: Patient and polyp characteristics associated with hgh-grade dysplasia in colorectal adenomas.,» *Gastroenterology,* τόμ. 98, pp. 371-379, 1990.

[24] I. Lansdorp-Vogelaar, M. van Ballegooijen, Z. AG., J. Habbema και E. Kuipers, «Effect of rising chemotherapy costs on the cost savings of colorectal cancer screening.,» *Journal of National Cancer Institute,* τόμ. 101(20), pp. 1412-1422, 2009.

[25] H. Brenner, M. Hoffmeister, C. Stegmaier, G. Brenner, L. Altenhofen και U. Haug, «Risk of progression of advanced adenomas to colorectal cancer by age and sex: estimates based on 840,149 screening colonoscopies,» *Gut.,* τόμ. 56(11), pp. 1585-1589, 2007.

[26] S. Winawer, A. Zauber, M. Ho, M. O'Brien, L. Gottlieb, S. Sternberg, J. Waye, M. Schapiro, J. Bond και J. Panish, «Prevention of colorectal cancer by colonoscopic polypectomy. The National Polyp Study Workgroup.,» *The New England journal of medicine,* τόμ. 329(27), pp. 1977-1981, 1993.

[27] The Canadian Task Force on Preventive Health Care, «Colorectal cancer screening: Recommendation statement from the Canadian Task Force on Preventive Health Care.,» *Canadian Medicine Associalion Journal,* τόμ. 165, pp. 206-208, 2001.

[28] L. v. Karsa, J. Patnick και N. Segnan, «European guidelines for quality assurance in colorectal cancer screening and diagnosis.,» *Endoscopy,* τόμ. 44(Suppl 3), pp. SE1-8, 2012.

[29] A. Zauer, S. Winawer και M. O'Brien, «Colonoscopic polypectomy and long-term prevention of colorectal-cancer deaths.,» *The New England journal of medicine,* τόμ. 366, pp. 687-696, 2012.

[30] D. Rex, P. Schoenfeld και J. Cohen, «Quality indicators for colonoscopy.,» *Gastrointestinal Endoscopy,* τόμ. 81, pp. 31-53, 2015.

[31] A. Rastogi και S. Wani, «Coloscopy,» *Gastrointestinal Endoscopy,* τόμ. 85(1), pp. 59-66, 2017.

[32] D. Heresbach, T. Barrioz, M. Lapalus, D. Coumaros, P. Bauret, P. Potier, D. Sautereau, C. Boustière, J. Grimaud, C. Barthélémy, J. Sée, I. Serraj, P. D'Halluin, B. Branger και T. Ponchon, «Miss rate for colorectal neoplastic polyps: a prospective multicenter study of back-to-back video colonoscopies.,» *Endoscopy,* τόμ. 40(4), pp. 284-290, 2008.

[33] W.-L. Chao, H. Manickavasagan και S. Krishna, «Application of Artificial Intelligence in the Detection and Differentiation of Colon Polyps: A Technical Review for Physicians.,» *Diagnostics (Basel),* τόμ. 9(3), p. 99, 2019.

[34] S. Karkanis, K. Galousi και D. Maroulis, «Classification of Endoscopic Images Based on Texture Spectrum,» σε *Proceedings of Workshop on Machine Learning in Medical Applications, Advance Course in Artificial Intelligence-ACAI99*, Chania, 1999.

[35] S. Kudo, Y. Mori, M. Misawa, K. Takeda, T. Kudo, H. Itoh, M. Oda και K. Mori, «Artificial intelligence and colonoscopy: Current status and future perspectives.,» *Digestive Endoscopy:official journal of the Japan Gastroenterological Endoscopy Society,* τόμ. 31(4), pp. 363-371, 2019.

[36] Tutorials Point, «Artificial Neural Network - Basic Concepts,» [Ηλεκτρονικό]. Available: https://www.tutorialspoint.com/artificial_neural_network/artificial_neural_network_basic_concepts.htm.

[37] D. Fumo, «A Gentle Introduction To Neural Networks Series — Part 1,» 4 August 2017. [Ηλεκτρονικό]. Available: https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc. [Πρόσβαση 18 November 2019].

[38] T. Yiu, «Understanding Neural Networks,» Towards Data Science, 2 June 2019. [Ηλεκτρονικό]. Available: https://towardsdatascience.com/understanding-neural-networks-19020b758230. [Πρόσβαση 20 November 2019].

[39] Mathworks, «What Is Deep Learning? - 3 things you need to know,» [Ηλεκτρονικό]. Available: https://www.mathworks.com/discovery/deep-learning.html. [Πρόσβαση 19 November 2019].

[40] C. Nicholson, «A Beginner's Guide to Neural Networks and Deep Learning,» [Ηλεκτρονικό]. Available: https://skymind.ai/wiki/neural-network. [Πρόσβαση 19 November 2019].

[41] M. Malik, «Basics of Neural Network,» 3 April 2018. [Ηλεκτρονικό]. Available: https://becominghuman.ai/basics-of-neural-network-bef2ba97d2cf. [Πρόσβαση 20 November 2019].

[42] Prabhu, «Understanding of Convolutional Neural Network (CNN) — Deep Learning,» 4 March 2018. [Ηλεκτρονικό]. Available: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148.

[43] S. Saha, «A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way,» Towards Data Science, 15 December 2018. [Ηλεκτρονικό]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Πρόσβαση 20 November 2019].

[44] GeeksforGeeks, «GeeksforGeeks - A computer science portal for geeks,» [Ηλεκτρονικό]. Available: https://www.geeksforgeeks.org/python-numpy/. [Πρόσβαση 20 November 2019].

[45] GeeksforGeeks, «GeeksforGeeks - A computer science portal for geeks,» [Ηλεκτρονικό]. Available: https://www.geeksforgeeks.org/os-module-python-examples/. [Πρόσβαση 20 November 2019].

[46] GeeksforGeeks, «GeeksforGeeks - A computer science portal for geeks,» [Ηλεκτρονικό]. Available: https://www.geeksforgeeks.org/python-matplotlib-pyplot-ticks/. [Πρόσβαση 20 November 2019].

[47] GeeksforGeeks, «GeeksforGeeks - A computer science portal for geeks,» [Ηλεκτρονικό]. Available: https://www.geeksforgeeks.org/pickle-python-object-serialization/. [Πρόσβαση 20 November 2019].

[48] Python Software Foundation, «Python.org,» Python Software Foundation, [Ηλεκτρονικό]. Available: https://docs.python.org/3/library/random.html. [Πρόσβαση 21 November 2019].

[49] Wikipedia, «Wikipedia,» 31 October 2019. [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Pandas_(software). [Πρόσβαση 21 November 2019].

[50] Python Software Foundation, «Python.org,» Python Software Foundation, [Ηλεκτρονικό]. Available: https://docs.python.org/3/library/gc.html. [Πρόσβαση 22 November 2019].

[51] M. Waskom, «seaborn,» [Ηλεκτρονικό]. Available: https://seaborn.pydata.org/. [Πρόσβαση 22 November 2019].

[52] Google, «Colaboratory,» Google, [Ηλεκτρονικό]. Available: https://research.google.com/colaboratory/faq.html. [Πρόσβαση 21 November 2019].

[53] J. Brownlee, «Introduction to TensorFlow,» σε *Deep Learning With Python - Develop Deep Learning Models on Theano and TensorFlow Using Keras*, p. 15.

[54] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu και X. Zheng, «TensorFlow: A System for Large-Scale Machine Learning,» σε *Proceedings of the 12th USENIX Symposium on*

Operating Systems Design and Implementation (OSDI '16), Savannah, 2016.

[55] J. Brownlee, «MachineLearningMastery,» 3 October 2019. [Ηλεκτρονικό]. Available: https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-to-classify-photos-of-dogs-and-cats/. [Πρόσβαση 21 November 2019].

[56] A. Anwar, «Towards Data Science,» 7 June 2019. [Ηλεκτρονικό]. Available: https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96. [Πρόσβαση 10 December 2019].

[57] ImageJ, «Imagej - Image Processing and Analysis in Java,» ImageJ, [Ηλεκτρονικό]. Available: https://imagej.nih.gov/ij/docs/intro.html. [Πρόσβαση 22 November 2019].

[58] N. B, «https://becominghuman.ai/,» 11 September 2017. [Ηλεκτρονικό]. Available: https://becominghuman.ai/image-data-pre-processing-for-neural-networks-498289068258. [Πρόσβαση 30 November 2019].

[59] M. u. Hassan, «Neurohive,» 20 November 2018. [Ηλεκτρονικό]. Available: https://neurohive.io/en/popular-networks/vgg16/. [Πρόσβαση 1 December 2019].

[60] py2py, «py2py,» 8 February 2019. [Ηλεκτρονικό]. Available: https://py2py.com/cnn-part-3-setting-up-google-colab-and-training-model-using-tensorflow-and-keras/. [Πρόσβαση 1 December 2019].

[61] J. Brownlee, «Machine Learning Mastery,» 17 May 2019. [Ηλεκτρονικό]. Available: https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-to-classify-photos-of-dogs-and-cats/. [Πρόσβαση 3 December 2019].

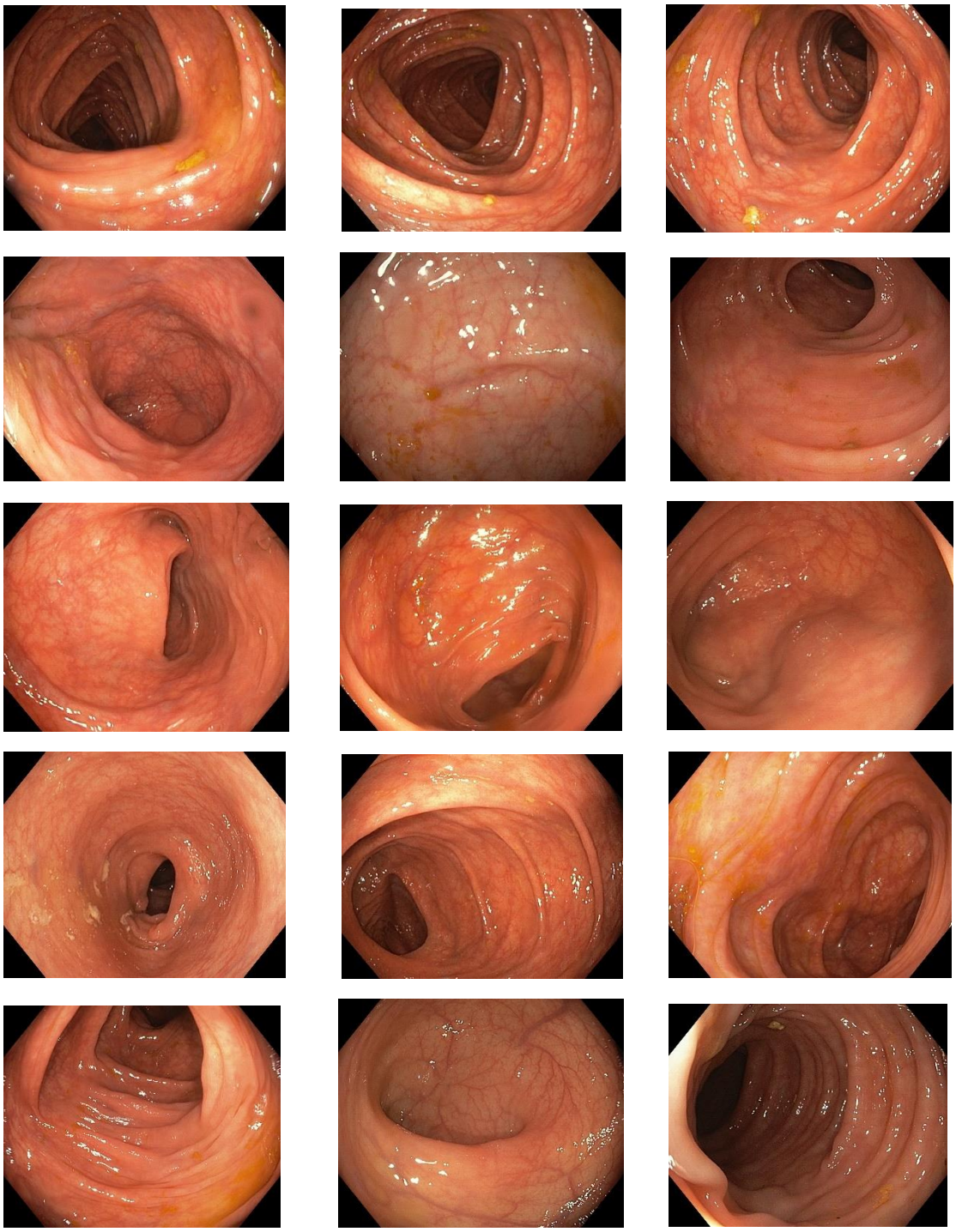
[62] P. Mohanaiah, P. Sathyanarayana και L. GuruKumar, «Image Texture Feature Extraction Using GLCM Approach,» *International Journal of Scientific and Research Publications,* τόμ. 3, αρ. 5, 2013.

[63] Geeks for Geeks, «Geeks for Geeks - A computer science portal rof geeks,» [Ηλεκτρονικό]. Available: https://www.geeksforgeeks.org/confusion-matrix-machine-learning/. [Πρόσβαση 12 December 2019].

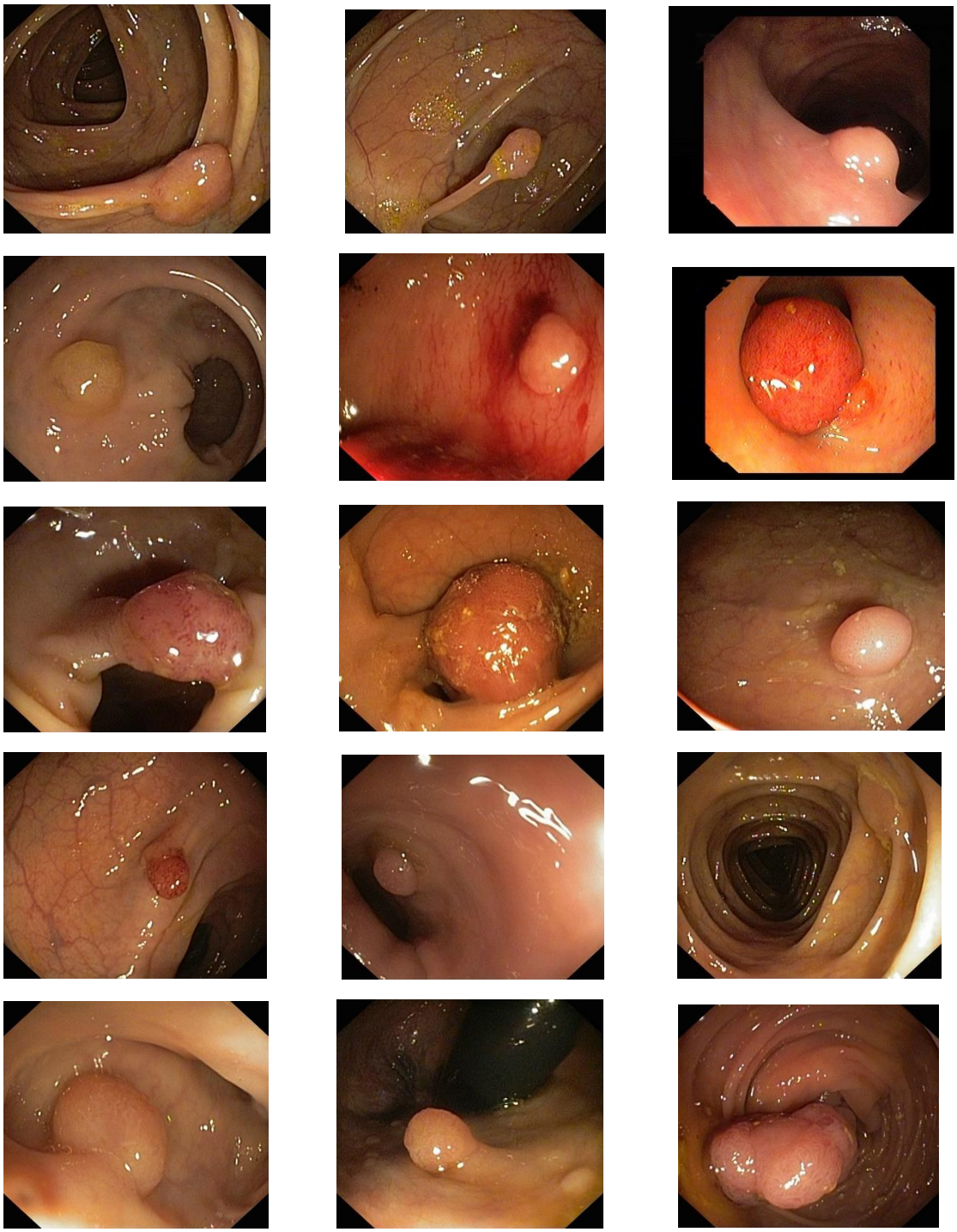
APPENDIX AP1: Training images samples

These are examples of the images used for the training phase of the polyp detection algorithms:

Healthy Images



Polyp Images (containing adenomas):

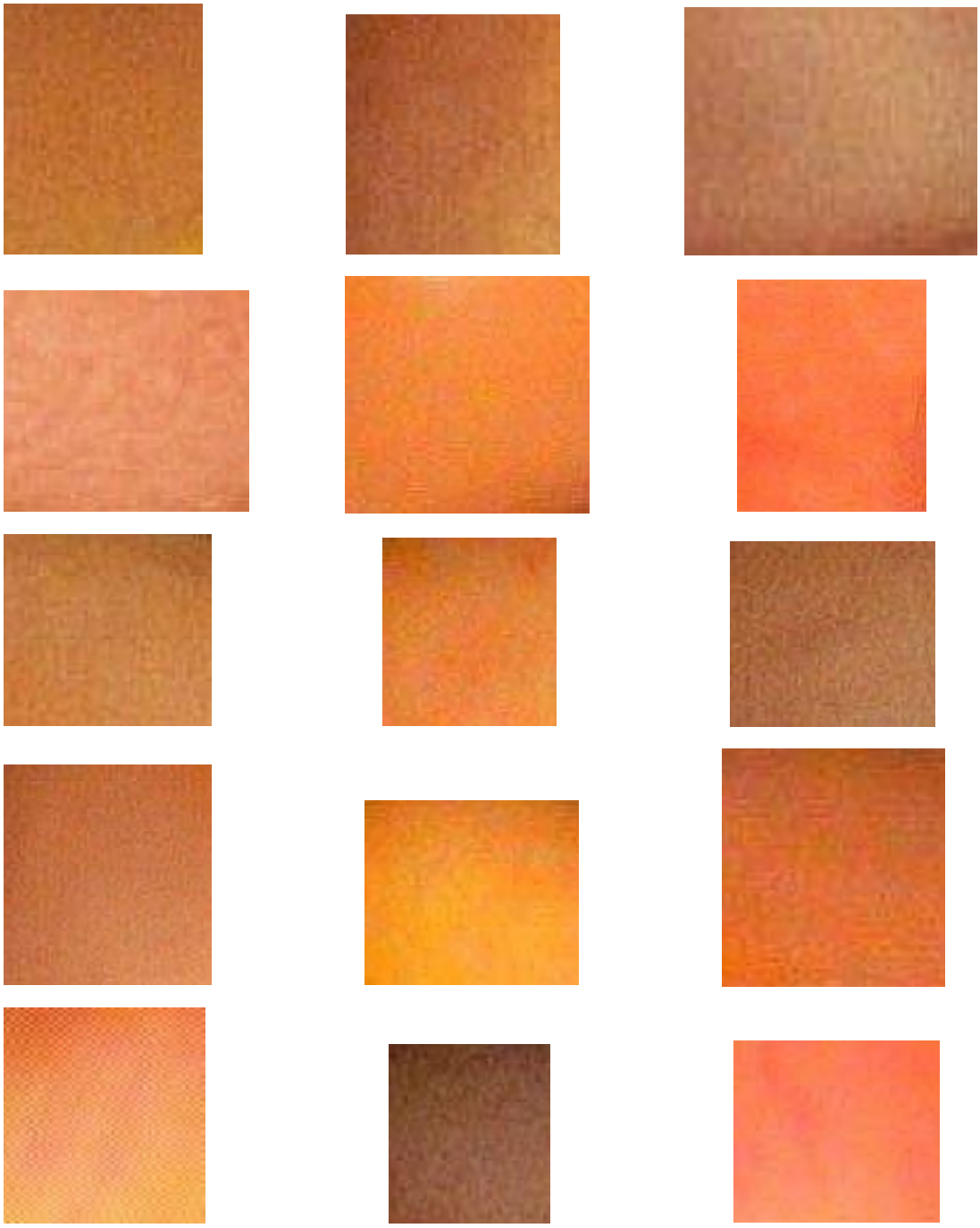


Polyp Images (containing hyperplastic):

These are examples of the images used for the training phase of the polyp classification algorithm:

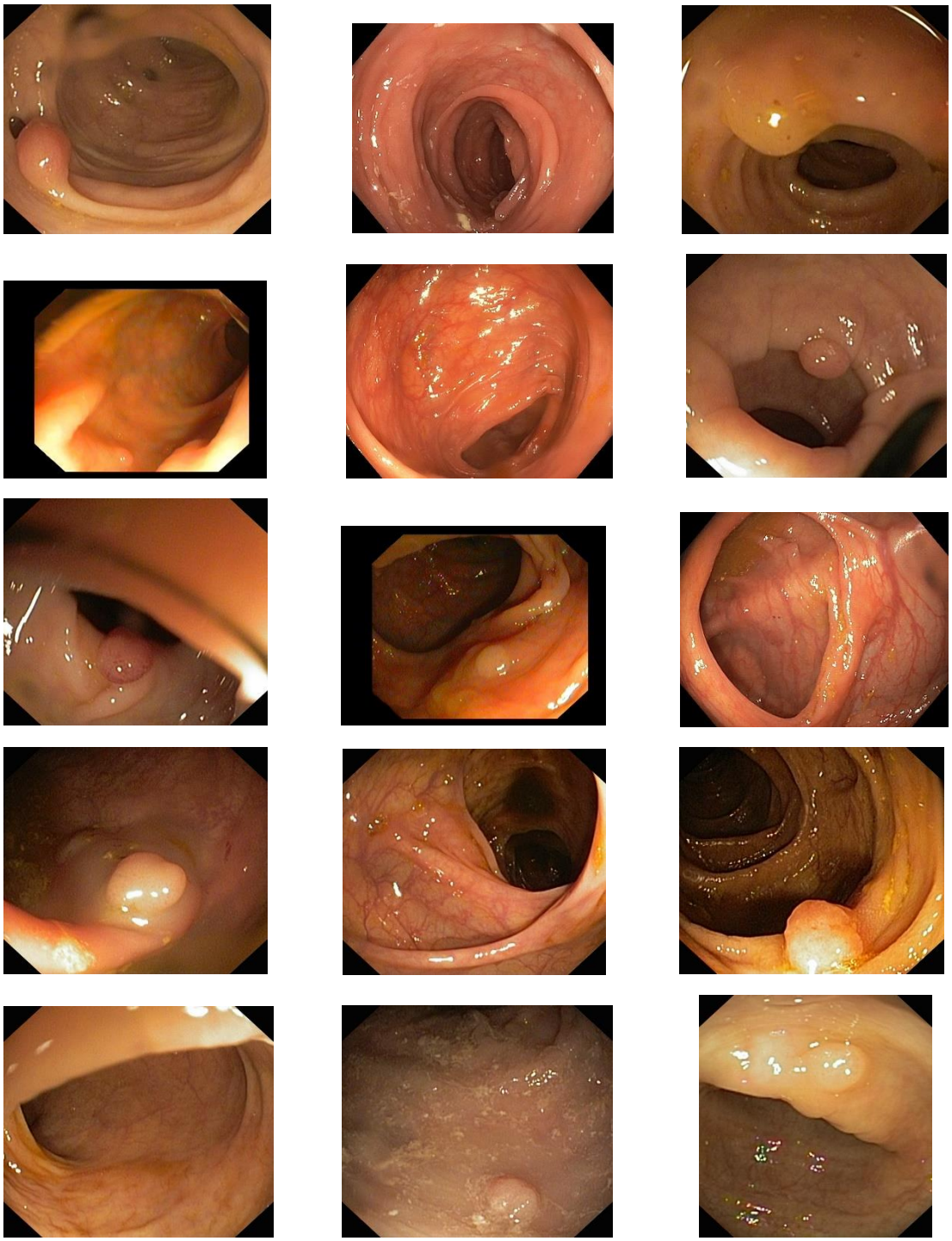
Adenomas:

Hyperplastic:



APPENDIX AP2: Testing image samples

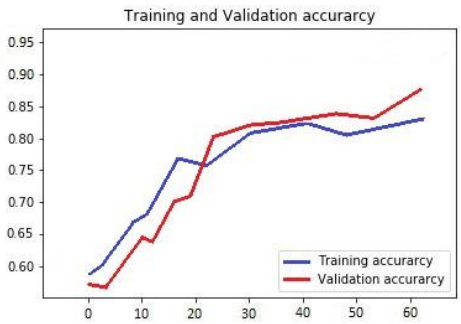
Images that were used for the testing of the implemented models for polyp detection. These images are completely "new" to the algorithms:

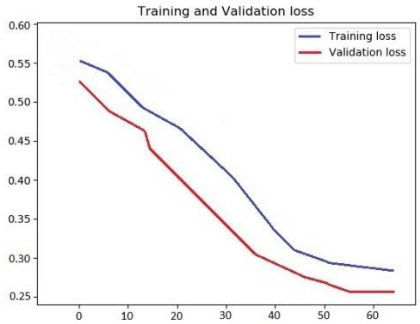


Images that were used for the testing of the implemented model for polyp classification. These images are completely “new” to the algorithms:

APPENDIX AP3: Training and validation accuracy plots - Training and validation loss plots

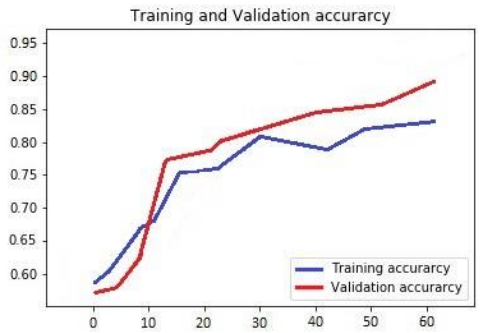
Plots for Method 1 Scenario 1:

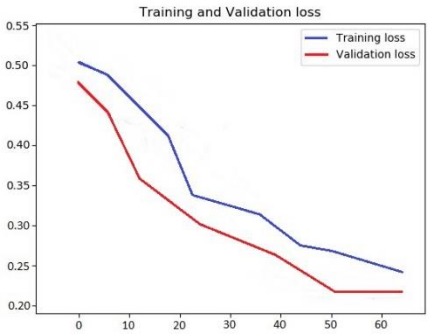




(a)Training and validation accuracy plot, (b) Training and validation loss plot.

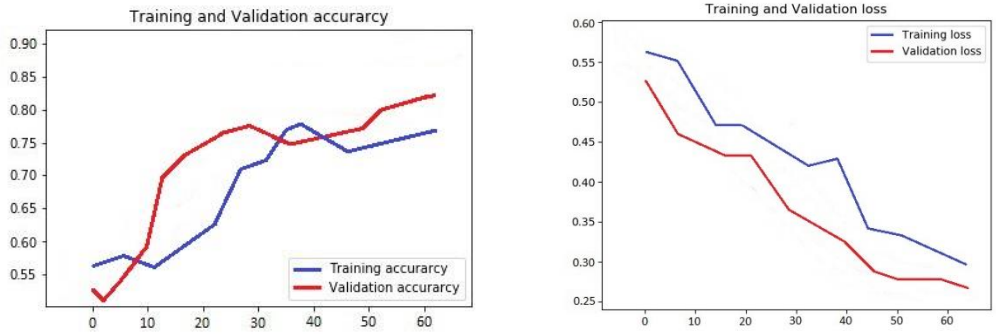
Plots for Method 1 Scenario 2:





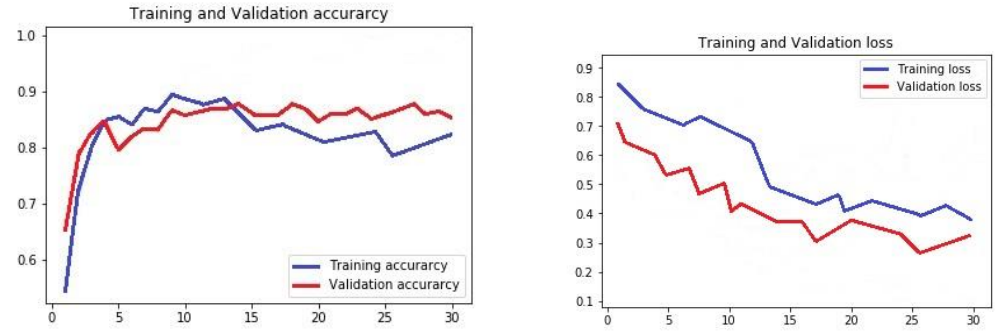
(a)Training and validation accuracy plot, (b) Training and validation loss plot.

Plots for Method 1 Scenario 3:



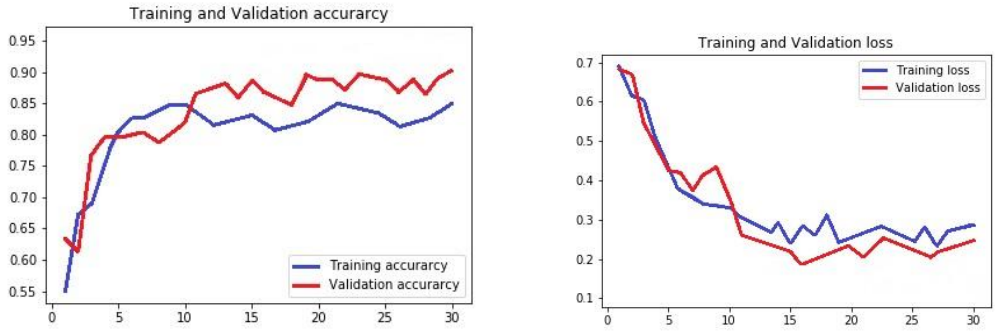
(a)Training and validation accuracy plot, (b) Training and validation loss plot.

Plots for Method 2 Scenario 1:



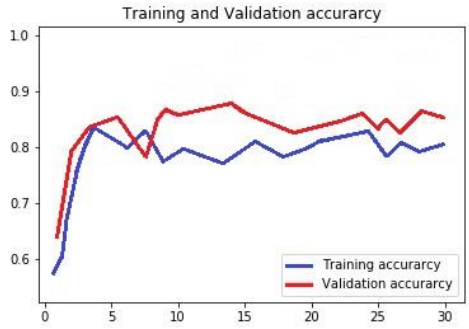
(a)Training and validation accuracy plot, (b) Training and validation loss plot.

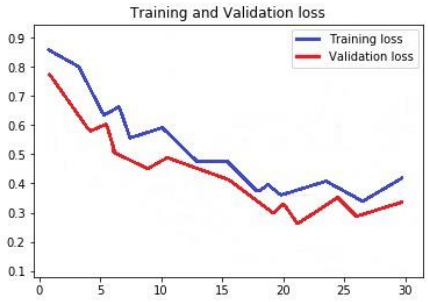
Plots for Method 2 Scenario 2:



(a)Training and validation accuracy plot, (b) Training and validation loss plot.

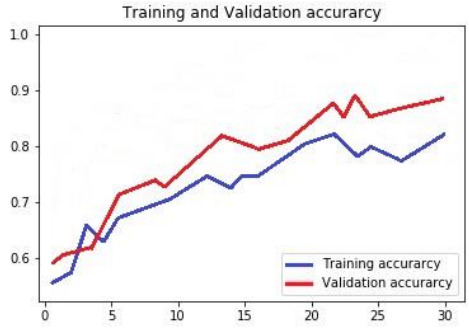
Plots for Method 2 Scenario 3:

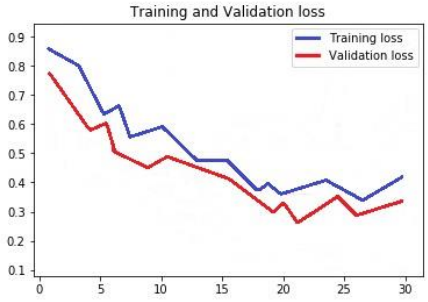




(a)Training and validation accuracy plot, (b) Training and validation loss plot.

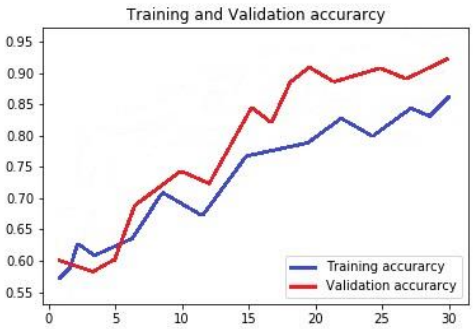
Plots for Method 2 Scenario 4:

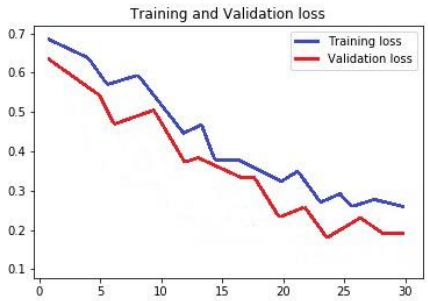




(a)Training and validation accuracy plot, (b) Training and validation loss plot.

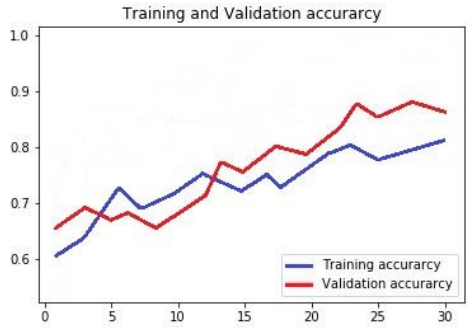
Plots for Method 2 Scenario 5:

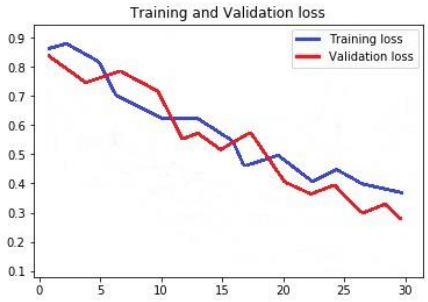




(a)Training and validation accuracy plot, (b) Training and validation loss plot.

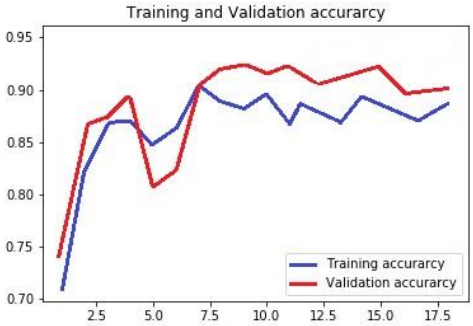
Plots for Method 2 Scenario 6:

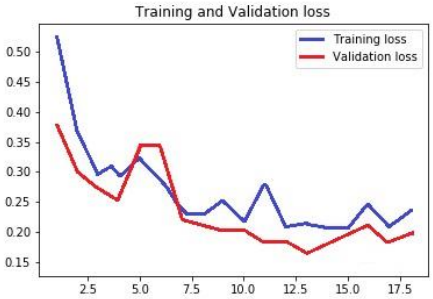




(a)Training and validation accuracy plot, (b) Training and validation loss plot.

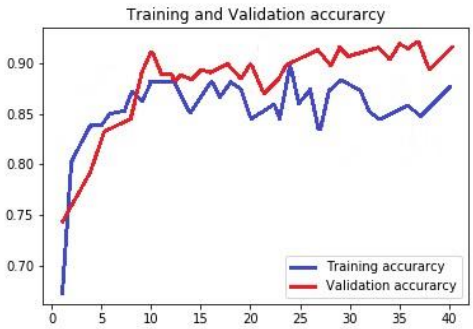
Plots for Method 3 Scenario 1:

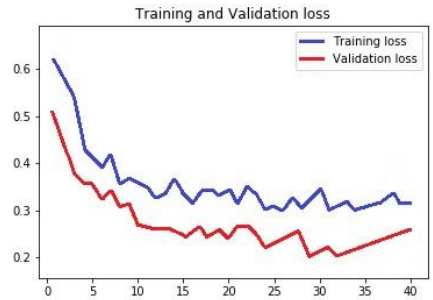




(a)Training and validation accuracy plot, (b) Training and validation loss plot.

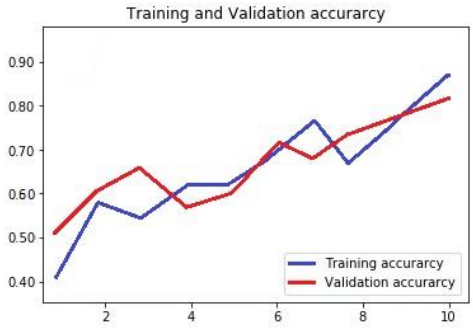
Plots for Method 3 Scenario 2:

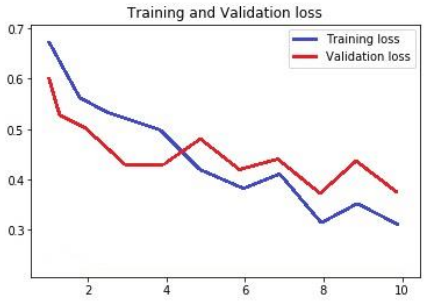




(a)Training and validation accuracy plot, (b) Training and validation loss plot.

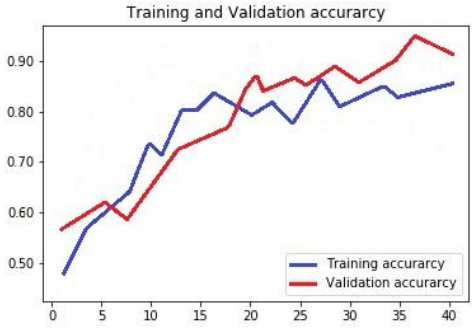
Plots for Method 3 Scenario 3:

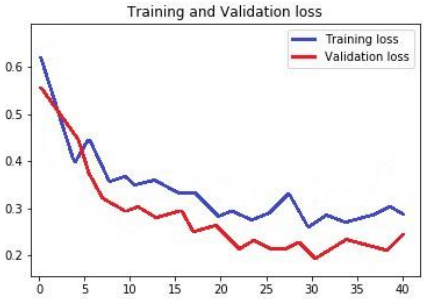




(a)Training and validation accuracy plot, (b) Training and validation loss plot.

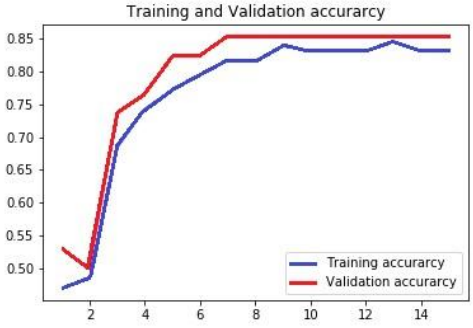
Plots for Method 3 Scenario 4:

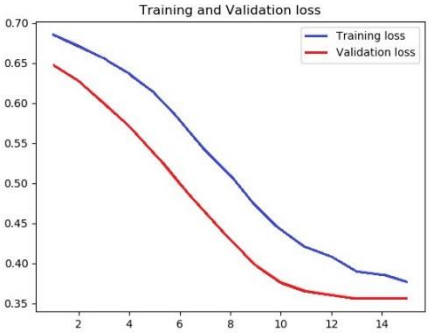




(a)Training and validation accuracy plot, (b) Training and validation loss plot.

Plots for Method 4:





(a)Training and validation accuracy plot, (b) Training and validation loss plot.