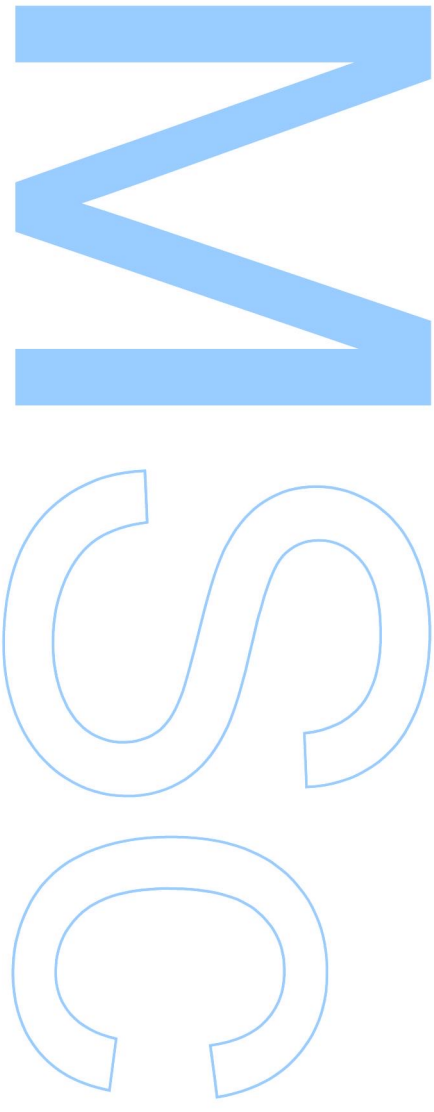
Deep Learning For Gastric Cancer Detection

Gabriel Trovão Pereira de Lima
Integrated Master's in Network and Informatic Systems Engineering
Computer Science Department
2021

Orientador
Francesco Renna, Invited Auxiliar Professor, Computer Science Department,
Faculty of Sciences of University of Porto

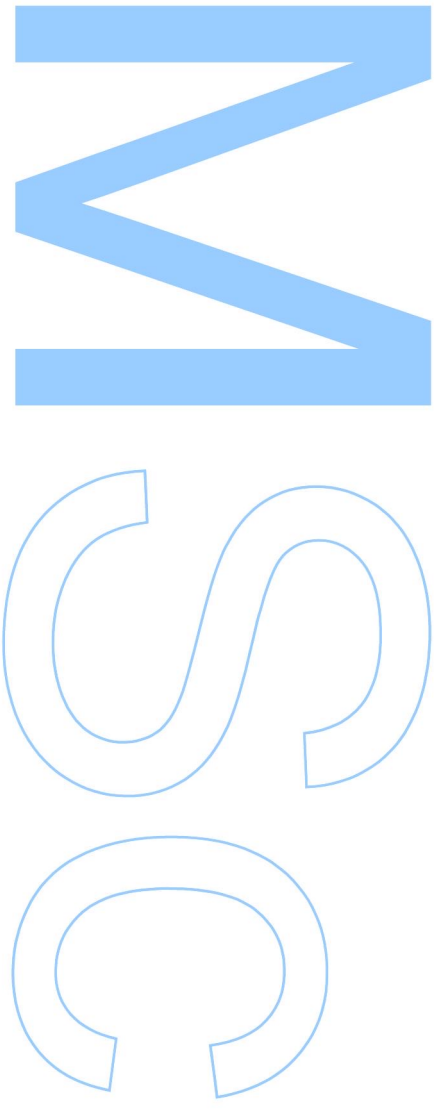
Coorientador
Miguel Coimbra, Associate Professor, Computer Science Department,
Faculty of Sciences of University of Porto



Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, _____/_____/_____



Acknowledgements

Throughout the writing of this dissertation I have received a great deal of support and assistance from many people. Without them, this work would have not been possible and for such I thank all of them.

I would first like to thank my supervisor, Dr. Francesco Renna, whose expertise and knowledge guided me through every step of my work. Your help was invaluable for me to be able to formulate the path of my research and my strategy. Your insightful feedback brought my work to a higher level and improved my reasoning. I will always be grateful for the support and encouragement that inspired me to thrive.

I would like to thank my co-supervisor, Dr. Miguel Coimbra, for the opportunity to be a part of a fantastic work group, whose mutual aid and team spirit provided for a comfortable and productive work environment. Your leadership pushed our group forward and provided us unique opportunities.

I would like to thank Joana, for her unconditional support, the love and patience in difficult times. I could not have completed my project without your counsel and your ability to rest my mind outside of my research.

I would also like to thank my family and friends for their continuous support and motivation: to my father, Artur, for the detailed medical explanations and useful conversations; to my mother, Maria José, for the strength and courage; my sister, Rita, for her useful opinions; and to my friend, Davide, for his professional and technical advice.

Abstract

Gastric cancer has been one of the most deadly types of cancer worldwide, while being the third leading cause of death by a cancer disease. Upper endoscopy is the most efficient way to detect a cancer lesion in the gastrointestinal tract, as well as the means to provide a secure and accurate diagnosis. The field of Machine Learning, and in particular, Deep Learning and Computer Vision, has had a strong impact by developing appliances used in medical and biomedical areas. Image classification through Deep Learning methodologies, more precisely, Convolutional Neural Networks has been a widely selected method to analyse images from endoscopy exams.

This study aimed to build Convolutional Neural Networks models capable of classifying upper endoscopy images, to determine the stage of infection in the development of a gastric cancer. Two different problems were covered. A first one with a smaller number of categorical classes and a lower degree of detail to perform a Macro classification. A second one had a Micro approach consisting of a larger number of classes, corresponding to each stage of infection of a gastric cancer in the Correa's cascade, excluding dysplasia.

Three public datasets were used to built the dataset that served as input for the classification tasks: Hyperkvasir, Gastrolab and Gastrointestinal Atlas. From the currently available Convolutional Neural Network architectures, four different models were chosen based on what was seen in state-of-the-art studies and their models. To test the models a 5-Fold Cross-Validation methodology was implemented. The metrics selected to explore the performance of the models were confusion matrix, Accuracy, Precision, Recall and F1-Score.

From the different Convolutional Neural Network models built, DenseNet169 achieved 0.79 and 0.72 for accuracy performance metric in the Macro and Micro approach classification tasks, respectively. The models built for this study are capable of identifying the stage of a gastric lesion in the moment of an upper endoscopy. A trade-off between the detail level of the classification and the performance of the models was registered. The higher the detail level of the classification tasks (increase on the number of classes), the worse is the performance of the models. Lastly, another tool developed were the Grad CAMs, in order to evaluate if the models classification traced back to features extracted from the region where lesions were located. A model was selected from one of the stages of the 5-Fold Cross-Validation process and an image example for each class of the Micro class problem was analysed. To the best of our knowledge, this is the first study to feature all the stages of the Correa's cascade separately, excluding dysplasia, in a classification task.

Keywords: Computer Vision; Convolutional neural network; Classification task; Gastric cancer; deep learning; upper endoscopy; Gastrointestinal Tract; Grad CAM.

Resumo

O cancro gástrico é um dos tipos de cancro mais mortal globalmente. Relativamente aos diferentes tipos de cancro, o cancro do estômago é o terceiro tipo de cancro que causa mais mortes mundialmente. A endoscopia alta é o método mais eficaz para detetar uma lesão de cancro no trato digestivo, bem como o meio mais fiável de obter um diagnóstico preciso. A área de *Machine Learning*, em particular *Deep Learning* e *Computer Vision*, tem vindo a desenvolver aplicações para uso nos campos da medicina e biomédica. Classificação de imagens usando metodologias de *Deep Learning*, concretamente *Convolutional Neural Networks*, são vistas como uma forma de analisar imagens provenientes de exames de endoscopia alta.

Este estudo teve como objetivo a construção de modelos *Convolutional Neural Networks* capazes de classificar imagens de endoscopia alta, para determinar o estado de infeções no processo de desenvolvimento de um cancro gástrico. Foram explorados dois problemas diferentes. O primeiro diz respeito a um menor grau de complexidade, com um número menor de classes categóricas e um grau de detalhe na classificação inferior. A abordagem considerada para este problema foi uma abordagem Macro. O segundo problema trata-se de uma abordagem Micro com um número maior de classes, correspondentes a cada uma das lesões que compõem a cascata de Correa relativa ao desenvolvimento de cancro gástrico, excluíndo displasia.

Foram usados três datasets de domnínio público para contruir o dataset que serviu como input para as tarefas de classificação: Hyperkvasir, Gastrolab e Gastrointestinal Atlas. Das arquiteturas *Convolutional Neural Networks* existentes atualmente, foram escolhidas quatro diferentes, tendo em conta o que foi revisto em estudos do estado-da-arte e respetivos modelos. Para testar os modelos uma metodologia de *5-Fold Cross-Validation* foi implementada. As métricas escolhidas para analisar a performance dos modelos foram matrizes de confusão, Exatidão, Sensibilidade, Precisão e F1-Score.

Dos diferentes modelos *Convolutional Neural Networks* construídos, o modelo DenseNet169 atingiu valores de 0.79 e 0.72 para a métrica exatidão, nas tarefas de classificação Macro e Micro, respetivamente. Os modelos *Convolutional Neural Networks* construídos neste estudo são capazes de identificar a fase em que a lesão no trato digestivo se encontra, aquando a realização de uma endoscopia alta. Foi descoberta uma relação inversa entre a especificidade da tarefa de classificação e a performance dos modelos. Quanto maior for o grau de especificidade das tarefas de classificação (visto num aumento do número de classes), pior é a performance dos modelos. Por fim, outra ferramenta desenvolvida foram as Grad CAMs, de forma a avaliar se as classificações feitas pelos

modelos eram devido a características extraídas da região da imagem onde se encontrava a lesão. Foi selecionado um modelo proveniente de uma das fases do processo de *5-Fold Cross-Validation* e uma imagem de cada classe do problema das Micro classes foi analisada. Tanto quanto é do nosso conhecimento, este é o primeiro estudo onde foram consideradas todas as etapas da cascata de Correa separadamente, com a exceção de displasia, numa tarefa de classificação.

Palavras-Chave: Visão Computacional; Convolutional Neural Network; Tarefa de Classificação; Cancro gástrico; Deep Learning; Endoscopia Alta; Trato Gastrointestinal; Grad CAM.

Contents

List of Tables

List of Figures

Acronyms

AC Accuracy

ADC Adenocarcinoma

AG Atrophic Gastritis

AGC Advanced Gastric Cancer

ANN Artificial Neural Network

BA Balanced Accuracy

BE Barrett's Esophagus

CAD Computer Aided Diagnosis

CAN Cancerous

CNN Convolutional Neural Network

CV Computer Vision

datagen Data Generator

DCC Departamento de Ciência de Computadores

DL Deep Learning

DN169 DenseNet169

EGC Early Gastric Cancer

FCUP Faculdade de Ciências da Universidade do Porto

FFNN Feed Forward Neural Networks

FN False Negative

FP False Positive

F1S F1-Score

GIT Gastrointestinal Tract

Grad CAM Gradient-weighted Class Activation Mapping

HE Healthy

HP Helicobacter Pylori

IM Intestinal Metaplasia

IRV2 Inception-ResnetV2

KL Divergence Kullback Leibler Divergence Loss

ML Machine Learning

ME Magnifying Endoscopy

NBI Narrow Band Imaging

NNL NasNet Large

PA Pyloric Antrum

PPV Positive Predictive Value

PRC Precancerous

PR Precision

ReLU Rectified Linear Unit

RC Recall

RMSprop Root Mean Squared Propagation

RNN Recurrent Neural Network

RN50 Resnet50

SCC squamous cell carcinomas

TP True Positive

TN True Negative

UE upper endoscopy

VGG16 VGG16 Network

5FCV 5-Fold Cross-Validation

Chapter 1

Introduction

1.1 Motivation

This study will focus on gastric cancer analysis through Deep Learning (DL) techniques and algorithms. Its purpose is to perform classification tasks, using high endoscopy images. In the matter of cancer related diseases, gastric cancer represents 7% of the world's cancers and 9% of the worldwide cancer related deaths. It is the fifth most frequent cancer type and the third leading cause of death from cancer, with approximately 950000 new cases and 783000 deaths in 2018. In Portugal, there are the highest gastric cancer mortality rates in Western Europe, making it mandatory to study and understand social and environmental factors. This will enable an earlier detection and treatment of the patients. Furthermore, the costs that treating a cancer patient entails since the diagnosis until full recovery are extremely high. It is estimated that, every year, more than 860 million euros are spent for the treatment of cancer patients, which is equivalent to almost 6% of the total health funds made available by the national health care services and the government [7, 52].

Over the years, a large number of new areas contributed to the development of very useful methods, as powerful assistants for aiding doctors to diagnose and treat diseases. Recently, the area of Machine Learning (ML), with special emphasis on DL (and Computer Vision (CV) as well), made progress mostly on technology capable of assisting medical staff. Special emphasis goes to the help provided on their diagnosis, making more accurate previsions and, consequently, more effective treatments. One of the areas in which DL and CV had a strong presence is medical imaging. Its importance increased, starting to become more applied to different healthcare situations. When dealing with a cancer patient, for a comprehensive analysis of the region under examination it is always necessary to have different kinds of information, from different sources and different diagnostic imaging techniques. Specifically, in the area of stomach cancer, endoscopic screening programs have reduced gastric mortality rates [20]. Knowing the advances on the fields of medical imaging and having the ability to obtain detailed information from the patient, CV is able to use the powerful information retrieved from medical images using techniques of feature extraction and segmenting those characteristics found. The use of feature extraction and segmentation includes activities such

as edge, corner or shape detection, used for computerised tomography. The colour, saturation and texture are also powerful tools to assist the medical staff: it is possible to obtain an abnormal spot through dermoscopy image analysis of skin's colour or texture by applying features as the ones above-mentioned. In terms of gastric cancer there also numerous possibilities for CV, the colour of the gastric mucosa for instance varies when in presence of cancer, the texture also changes due to loss of cells resulting in a transformation of the pleated texture. Afterwards, with DL this content can be used to build a Computer Aided Diagnosis (CAD) model. CAD is a system that can perform and help medical professional to make diagnoses, in particular acting as second readers. This means a medical professional will make their first attempt at diagnosing a disease in a patient and then the computer will serve as a backup to confirm that diagnosis. The main objective of CAD models is to decrease the rate of false diagnosis, by assisting physicians with a second opinion [2]. CAD models analysis are also able to detect the landmarks of the stomach and search for irregularities in it, while it is difficult to notice only by doing an endoscopy-like exam. In the end, DL and CV have become more relevant for its potential medical usages, and this is a tendency that is expected to continue to grow.

This project is held under the supervision of the Professors Francesco Renna (PhD) and Miguel Coimbra (PhD), researchers of the INESC TEC lab C-BER, on the faculty of Science of the University of Porto. The C-BER group follows a major research topic on the biomedical engineering field, concerning healthcare, disease diagnose and lesion detection.

1.2 Scope of the thesis

The scope of this Master's thesis is creating an algorithm for lesion classification on the subject of gastric and esophageal cancer based on images retrieved from high endoscopy exams. It is a system build on the computer aided diagnosis paradigm. During an endoscopy exam, the doctor often encounters situations that are hard to assess, as the malformations or problems are not clear when observed with the naked eye. It might be because the lesion is still to early to be detected and perceptible to a specialist doing an endoscopy without further tests. It is also possible the examiner does not have much experience performing this kind of exams or the lesion detected is ambiguous. Ultimately, performing diagnosis over lesions seen in the gastrointestinal tract during an upper endoscopy can be a difficult task, due to the wide variety of lesions that can happen in all the structures that are observable during the exam. These lesions have different timings and characteristics, which makes it important to perform the most accurate diagnosis possible. Only by doing so it is possible to select a proper treatment. The CAD system helps medical personal in avoiding misdiagnosis with their predictions.

Following this train of thought, the goal is to build a DL classification system. This system, under the supervised learning paradigm using labelled images from upper endoscopy exams as a source of input, can accurately predict what kind of lesion is being observed. Two different multi class classification tasks will be explored. One will include more broad concepts, leading to more comprehensive class designations. The other will be more in-depth, with a separation of the data in a

larger number of classes, where the class names are more specific lesions. It will be a CAD model efficient in assisting doctors performing diagnostics, serving as an addition to perfect their opinion. The algorithm of choice will be on the Convolutional Neural Networks (CNNs) topic.

There are several architectures that might be used for the two primary tasks. For classification tasks different architectures will be implemented and evaluated. The models built will be neural networks, precisely CNNs. From this kind of neural networks, there are several different frameworks available. Some are based on the type of connections they have like the residual neural networks, while others focus on how deeply connected the networks are such as the densely connected neural networks. The main strategy that will be used is a transfer learning strategy, which will allow for the models to have already learned at the time of the training stage, by previously performing other classification tasks. Two different classification tasks will be performed, one based on a Macro approach of the categorical classifications and other based on a Micro approach of the same classifications. The Macro approach is expected to achieve better results in the classification task, due to its general concept, which will divide the data in fewer classes. The features extracted from the images will be grouped into broader sets, resulting in an easier prediction for the images. On the other hand, the Micro approach will be more detailed, given that the number of classes increases. These scenarios will represent a trade-off between the performance and the detail in the classification task. If the classification is more detailed, with an increase of the number of lesions and less data to train and test for each class, the performance will decrease. Furthermore, one other tool considered to analyse the functioning and performance of the models will be the Gradient-weighted Class Activation Mappings (Grad CAMs). With this technique, it will be possible to understand if the models are performing the classifications based on a region of interest in the image or if their prediction was done using features that do not represent the focus of the lesions.

Chapter 2

Background

2.1 Stomach Cancer

2.1.1 Stomach Cancer - Concept

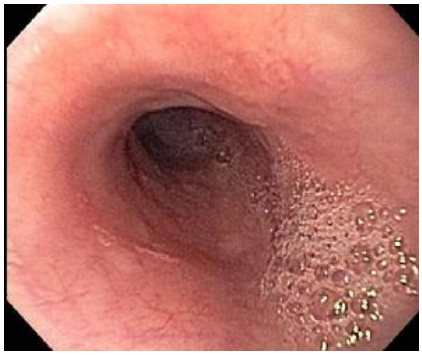
The type of cancer studied in this project is the stomach cancer, more specifically a specific area of the stomach with certain characteristics is targeted. This affected area is the one that coats the inside of the stomach, the gastric mucosa, surface that contains the glands and the gastric pits. Figures 2.1 and 2.2 show an endoscopic view of an healthy digestive tract in a routine endoscopy exam. It is possible to see the upper and lower part of the stomach, where it meets the beginning of the small intestine and also the middle section of the esophagus and its junction with the stomach. To provide context, a definition of cancer can be given: it can be referred as an abnormal cellular growth experience by a group of cells in a specific structure or organ of the human body, with possibility to spread [39]. With this fact being the reason why it can also be called malignant tumours in contrast to the benign tumours, which does not have dissemination potential. The lesions might also be removed if they are considered precancerous, to prevent the evolution to malignant. Malignant tumours can spread to lymph nodes and the bloodstream to reach different organs, in a phenomenon called metastasis [52].

The main cause for the appearance of this type of cancer is the development of a bacteria, Helicobacter Pylori (HP), in the gastric mucosa. The process by which it is possible to connect this bacteria from the moment it establishes in the mucosa until the already developed cancer is discovered is long. It takes a significant amount of time and a set of steps. Although the gastric mucosa is well protected against infections caused by bacteria, HP possesses the ability to adapt to this environment. That ability to accommodate is possible due to special characteristics that allow it to enter the mucosa and choose the best possible spatial orientation in it, which leads to a high probability of colonisation and transmission in a persistent way. The presence of HP starts by causing an infection/inflammation, targeting a normal and healthy mucosa with well-preserved gastric glands, leading to a non-atrophic gastritis [22].

(a) Middle section of the stomach, looking forward.

(b) Middle section of the stomach,looking backward.

Figure 2.1: In the second image the endoscope is "retroflexed" in a U-shape, allowing us to look back at the upper stomach, and the instrument entering the stomach from the esophagus (from [4]).



(a) Middle section of an healthy esophagus.

(b) Esophagus and the stomach lining junction.

Figure 2.2: The endoscopic view of two different parts of the healthy esophagus (from [4]).

Unless the patient receives proper treatment, it shows a tendency to be a chronic condition and evolve to an atrophic gastritis. In approximately 50% of the cases, this gastritis makes progress towards to multifocal atrophic gastritis. The name of this stage of the process is given to the fact that, considering the atrophy and degeneration of the cells, it will eventually result in the transformation of the gastric mucosa. The recondition of this structure is the first major indicator that we are in the presence of a precancerous process [64].

If the atrophic gastritis condition persists, it will move into a gastric intestinal metaplasia. In this stage, a change of the phenotype of the cells will occur. The phenotype encloses a collection of concepts like an organism's physical form and structure, its development processes, biochemical and physiological properties as well as its behaviour. The phenotypic change that happens during this phase concerns the normal epithelial cell of gastric mucosae, shifting to an intestinal phenotype. Comparing to the previous atrophic gastritis state, this is a more advanced stage as the metaplasic gastric epithelium will replace the normal epithelium, that has not suffered changes yet. The advanced state of infection/inflammation in these glands extends to the others in the gastric mucosa and, from the evolution of this metaplasic state, the next step consisting of dysplasia occurs. A dysplasia is formed when we are in the unequivocal presence of neoplastic epithelium, without evidence of tissue invasion. The state of the mucosa at this time is flat and depressed. In Figure 2.3 it is possible to

see a lesion right after it progresses and is no longer a dysplasia.

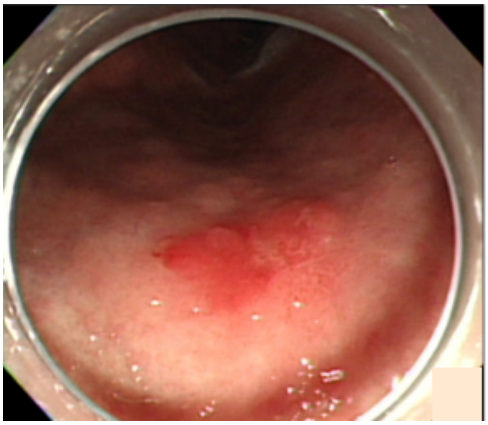


Figure 2.3: An early gastric cancer lesion at the lower body of the stomach (from [34]).

The last stage corresponds to a moment when the cancer has deeply established itself, which leads to the development of an adenocarcinoma, advanced degradation at cellular level, seen in Figure 2.4. An adenocarcinoma is an invasive type carcinoma and happens when standing before the penetration of neoplastic cells into the lower layers. There is a high probability that during this stage the neoplastic cells acquire the capability of degrading the stromal matrix surrounding the neoplastic cells. The main difference between a dysplasia and an adenocarcinoma is that the first is recognised to progress through several grades of severity according to the degree of divergence from normal. These have been classified as mild, moderate, and severe dysplasia. Severe dysplasia is almost indistinguishable from invasive adenocarcinoma, but confined within the epithelium. Furthermore, only in the invasive adenocarcinoma the neoplastic cells invade through the mucosa and penetrate tissue like the submucosa [67].

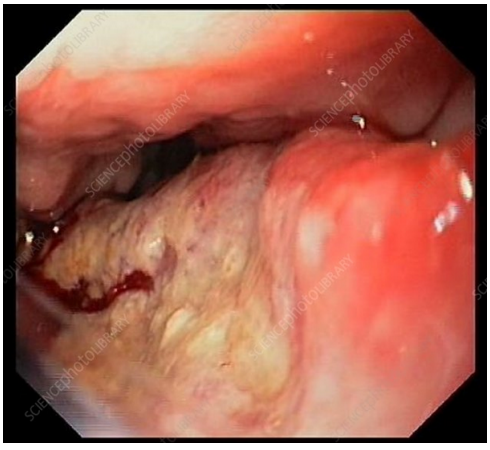


Figure 2.4: Adenocarcinoma, already proved as stomach cancer (from [65]).

The whole process described until now forms the Correa's Cascade or precancerous gastric cascade, seen on Figure 2.5 [23]. It is possible to interrupt this infection process, so it does not proceed beyond non-atrophic gastritis.

In a similar way, a different type of cancer might occur in the digestive tract, namely in the

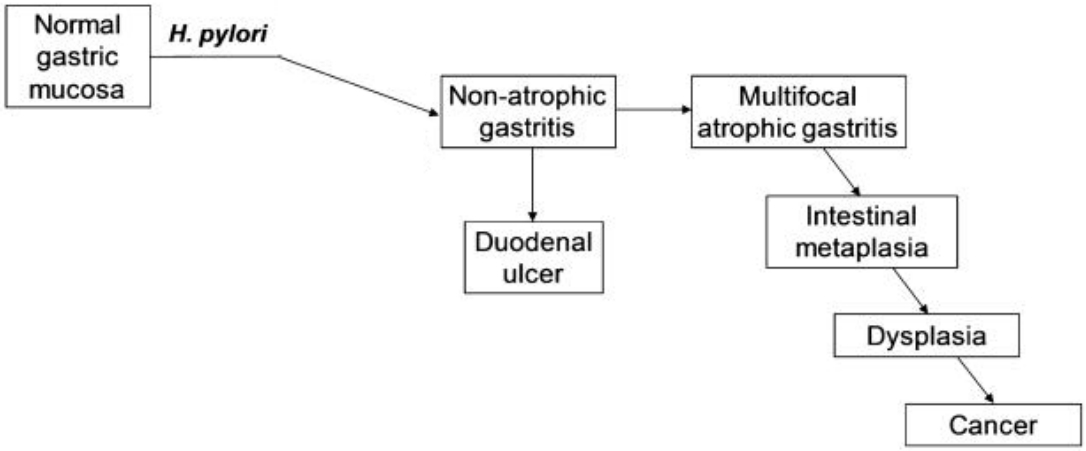


Figure 2.5: Schematic representation of the main clinical outcomes of HP infection. The right side of the scheme shows the sequential steps of the precancerous cascade (adapted from [23]).

esophagus. A healthy esophagus can be seen in the endoscopic images of the Figure 2.6. The esophagus cancer can occur in two distinguished forms: there is a chance of developing an Adenocarcinoma (ADC), in the same terms as it happens in the stomach, or a squamous cell carcinomas (SCC). This concerns sets of cells that coat the inside of the esophagus wall. These cells are epithelium ones, which means they are wide and flat in terms of shape. The main causes for the appearance of these two types of cancer are fundamentally human related factors such as smoking, alcoholic beverage, obesity and long term gastric reflux. Despite all the factors share responsibilities in what comes to causing a disease like this type of cancer, the condition that holds the biggest tendency to lead to a ADC is long term gastric reflux. On the other hand, the other factors are extremely more related to lifestyles and social conditions and, thus, intimately related to SCC [72].

The cancer symptoms in the esophagus appear later than in other structures. Usually they are only notorious when the disease is deeply rooted in the esophagic tube. By this time it is already in advanced stage. One of the first symptoms corresponds to the narrowing of this tube due to the physical presence of the cancer (dysphagia phenomenon), since this organ is narrow. Still about the differences between ADC and SCC in the esophagus, it can be said that the ADC tends to develop in the distal esophagus, in glandular cells, causing them to become intestinal cells. Such phenomenon is called Barrett's esophagus, where the substitution of the squamous epithelium can happen during the reflux esophagitis regeneration process. This scenario holds a large possibility of progressing into a displasia, which will increase the possibilities of developing an adenocarcinoma [33]. The Barrett's esophagus can be seen in Figure 2.6.

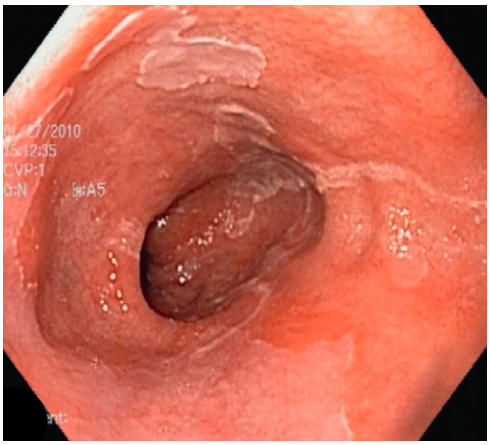


Figure 2.6: Barrett's esophagus lesion (from [53]).

2.1.2 Diagnosis (Gastroenterology)

There are several techniques to diagnose stomach cancer, which are applied under specific conditions and considering the disease's possible state of evolution in the patient. Amongst the different methods, an important one to mention is the study of HP, since the number of cases where a stomach cancer develops in the gastric mucosa with the absence of inflammation by HP is reduced. Performing a high endoscopy (also known as upper endoscopy or gastrointestinal endoscopy), followed by a biopsy is the most effective and standardised method for the diagnosis of such disease. Another example of diagnosis currently put into action is through a set of X-rays - the Upper Gastrointestinal series (Upper GI series, UGIS) - that intends to analyse the patient's upper gastrointestinal tract [50, 53].

2.1.3 Endoscopy

Going in depth on the main diagnosis technique of this Master's thesis, there are two types of endoscopy exams and its studied version is the upper endoscopy (the other is the lower endoscopy). Besides these two types of exams, an endoscopy can also be a white light endoscopy, a Narrow Band Imaging (NBI) endoscopy and chromoendoscopy. The white light endoscopy corresponds to the exam studied in this Master's thesis, images concerning the appearance of the Gastrointestinal Tract (GIT) without any change, with only a white light in the endoscopy pointing at the structures. The NBI technique is related to a type of endoscopy based on vascular and mucosal patterns that are useful to predict the histological structure of tissues [57]. It consists of using optical fibbers, which in turn allow to narrow the spectrum of the light wave. The chromoendoscopy involves the topical application of stains or pigments to improve tissue localisation, characterisation and, consequently, the diagnosis [19]. It is also possible to be a Magnifying Endoscopy (ME) exam, that provides a zoomed view of the gastric mucosa (115x optical zoom), which is comparable to stereoscopic microscopy.

Starting by clarifying the necessary materials for the exam to be performed, the endoscopy is made with a tube that is usually flexible (it can also be rigid), the insertion tube, which contains

two optical fibre cables, each one with about 5000 fibres. The cables have different functionalities. The first one is intended to illuminate the organs or the structure that is being examined, by using a light system (the previously referred white light); the second is to capture the observed image from the internal structures of the patient, by using an objective equipped at the end of the cable. This image is projected into a monitor used by the examiner to visualise the patient's digestive tract. The insertion tube is also capable of incorporating a third cable in case there is the need to use a third instrument to carry out a biopsy exam. To these components are added: a suction valve - any liquid found during the procedure must be suctioned so that every area can be properly analysed -, an inflation valve and, lastly, a valve that is able to clean the lens. These tools are inserted in the control handle, ergonomically designed to be used only with the left hand.

To carry out an exam such as an endoscopy, usually the patient is anaesthetised locally in the oral pharynx so that he can feel less discomfort or anxiety. In some cases the patient can be sedated using intravenous medication. The sedation level varies from patient to patient, it can go from light sedation administrated through intravenous medication and the above mentioned local anaesthetic (which is usually the preferred technique) to deep sedation and, in extreme cases, sedation through general anaesthesia. Nonetheless, this last level of anaesthesia is not mandatory and the number of cases where it is needed is very low, since it depends heavily on the patient's medical records. It exposes the patient to other types of risks and complications such as vomits, hyper and hypotension, arrhythmia or respiratory/cardiac arrest, in worse cases [70].

The exam itself is performed by inserting the endoscope into the patient through mouth and throat, passing in the esophagus, stomach and duodenum. The patient can be asked to put on a mouth piece, in order to keep his mouth open without a lot of effort. When passing the throat, the endoscope can be helped by the patient's deglutition movements, so that it is easier to pass. During the procedure it is possible to use the endoscopy for inflation, namely in the stomach, in order for the gastroenterologist to better examine the organ's interior. There could be different visualisations of the GIT when performing an upper endoscopy. When inside the patient's stomach, it is possible to turn the endoscope backwards making it resemble a "U" shape, in order to analyse the gastric fundus (upper part of the stomach). This is called a retroflexed view. The technique used when an upper endoscopy exam is carried out is the antegrade technique, where the device is inserted through the mouth to analyse the GIT. On the contrary, if a lower endoscopy it is a technique known as retrograde [53].

After the exam is completed, the air that had been inserted before can be sucked using the suction valve. This way the gastric folds, which is the structure where the injuries are more perceptible if they exist, are better observed. The final stage of the exam is already in the beginning of the small intestine, the duodenum, where it is possible to look for diseases like ulcers. After the exam is concluded, the endoscope is carefully removed and the patient waits until the anaesthesia effect has worn off [49, 70]. In Figure 2.7 it is possible to observe the standard procedure of an upper endoscopy.

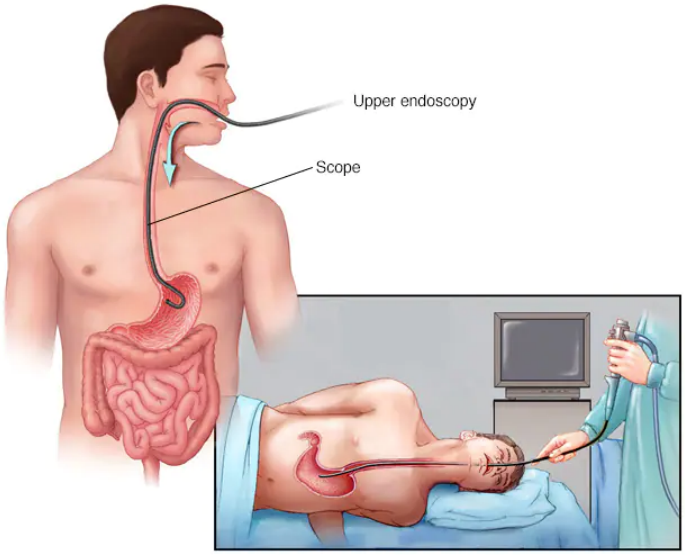


Figure 2.7: An upper endoscopy procedure, with the patient lying on its side while the examiner watches the monitor with the images from the GIT (adapted from [53]).

2.1.4 Complications that may arise from an endoscopy exam

There are complications that can come from an upper endoscopy, which are intrinsically connected to this type of exams and do not necessarily mean they are caused by medical negligence. Complications might occur due to the technical and invasive nature of this kind of exams, which can lead to light injuries/lesions - the type of injuries that require brief hospitalisation - or severe ones - in this case leading to deficiencies or liabilities. They can be associated with infections that happen under specific circumstances. One complication as an outcome of this sort of exams is haemorrhages. Bleeding as a consequence of an upper endoscopy exam is rare. The risk of such complication is low and it increases whenever the harvest of a sample of tissue for further analysis and testing - a biopsy - is part of the procedure. This kind of exams might also bring cardiopulmonary complications, e.g., blood pressure anomalies, changes in the normal heart functioning and respiratory issues (although they might happen, this kind of complications are rare) [25, 70].

Another complication that might emerge from an endoscopy is the breach/rupture of a tissue or organ. The probability of a perforation occurring during an upper endoscopy is considerably low - lower than 0.1% [25] - as well as mortality in those cases. However, there are a few factors that increase the risk of rupture in an upper endoscopy: the lack of experience from the technician, obstructions of the structures (the esophagus wall for instance), asymmetries of the esophagus, its small diameter or the inability to get the endoscopy through it. The early detection of a rupture is decisive for its treatment.

The transmission of microorganisms from patient to doctor and vice-versa, as well as the transmission coming from defective equipment are also possible complications inherent to an endoscopy exam. Although these scenarios are a way of developing infections as a consequence of an endoscopy exam, this is extremely rare. When it happens, the more frequent reason is defective equipment or

misjudgements when carrying out the standard protocol [25, 27].

An upper endoscopy might also fail in certain situations without taking place an issue like the above-mentioned. The lesions might be ambiguous, which requires more experience to detect. This detection can be made easier with the assistance of CAD model. Human factors might also occur like the physician getting distracted or feeling tiredness and, consequently, not performing a thorough exam.

2.2 Deep Learning

From the different areas capable of image analysis, one that has registered significant progress over the past few years is DL. In the specific case of UE images, Artificial Intelligence methods were developed to understand the different types of images used in these exams, from white-light endoscopy to magnifying endoscopy and chromoendoscopy. Several state of the art research papers have studied these images and its characteristics, in order to be able to optimise the process of analysing this information. From the different implemented techniques, outcomes such as the identification of regions of interest, detection of lesions along with its classification have been seen.

2.2.1 Deep Learning - Concept

DL is a field of ML, whose main target is to create learning models capable of making decisions and to do predictions/tasks. To do so, the models will be created and trained based on preexisting examples and data. The models that are used in this area are mostly deep Artificial Neural Networks (ANNs), a concept of networks that will be further detailed in the following sections. By using DL methodologies, the computer systems can learn from experience and understand the scenarios in which they are put in to perform predictions through a hierarchy of concepts. Each concept can be explained by resourcing to simpler ones. This method avoids humans having the need to specify the knowledge [31].

Coupled with DL, another ML area is CV, an area that concerns the study and interpretation of images observed by computer systems, having as main goal the decision making over real scenarios and real objects based on those sensed image. An image is formed by a process that occurs under a specific set of circumstances as the lighting conditions, the scene geometry, surface properties and the camera optics. An image is processed usually in the first stage of most CV applications. Implementing image processing is done with the goal of preprocessing the image and convert it into a more suitable form for further analysis. Examples of these actions are reducing the image's noise, exposure correction and colour balancing, increasing sharpness, or straightening the image by rotating it. From the images, a set of features is extracted, as they are the ones to be provided to the DL systems. It can be difficult to extract such features from the raw data [31, 61].

DL is used for several reasons and in different scenarios. One of the main motives why it is widely used is related to the fact that DL has the capability to perform in approximately all application

domains. Furthermore, the systems built with resource to DL are robust, they do not require precisely designed features to learn. Instead, the optimised features are learned in an automated process, bearing in mind the task to be solved. Hence, alterations such as changes in the input data will not impact the systems being built. There is also the generalisation factor, as for different types of data or applications, the same DL technique is suitable. This is commonly known as transfer learning. One last plus in favour of DL methodologies is their scalability, as multiple models built in DL have thousands of layers and can be applied at supercomputing scale [46].

2.2.2 Supervised Learning

The supervised learning paradigm is one of the DL methods for creating learning models for decision making, which is the one used for this thesis. The main characteristic of supervised learning is the availability of annotated training data. When a model is being created following the supervised learning paradigm, the training data must be labelled. An idea that a "supervisor" is instructing the learning system on the labels to associate with the training data is implemented in supervised learning. For the specific purpose of this project, which will use endoscopy images, these must have a correct classification, assigned by an experienced professional. This way the data can be used to determine the optimal parameters of the model. In supervised learning, the environments created have a collection of input data and its resultant outputs. Following this step, the DL model parameters are repeatedly updated to achieve an improved estimate for the preferred outputs. Once a positive training outcome takes place, the system earns the ability to obtain the right solutions to initially proposed queries. Among the several supervised learning techniques, there are the Recurrent Neural Networks (RNNs) and CNNs, with both belonging to the deep neural networks category [24, 29, 46].

The main advantage of this approach is the ability to collect data or generate a data output from the prior knowledge. Nonetheless, there are also disadvantages. The main disadvantage in supervised learning comes from the fact that decision boundary might be overstrained. This can happen if a training set is not diverse enough, which is the same as saying if a training set does not have samples that should be in a class [46].

2.2.3 Artificial Neural Networks (ANNs)

ANNs are made in image of the brain, imitating its structure and activity of brain neurons on a computer in a logical sense. The concept of neuron is replicated in the most proximate way to the one of a biological neural network. A neuron in an ANN has several inputs - corresponding to the dendrites in the human brain - and one output - the axon in the biological network. Each of the neurons in the networks receives multiple connections with other neurons, constantly receiving incoming signals. When the resulting sum of the signals surpasses a previously defined threshold, a response is sent as output. However, the level of complexity in an ANN is smaller than the one of biological neural network. Thus, an ANN consists of artificial neurons, or nodes, connected to each other, passing the information they received as inputs from other nodes, to the nodes they

are connected to. The ANNs interconnected computational nodes, collaborate to collectively learn from the input, with the goal of optimising its final output [59]. There are several types of neural networks in the deep learning universe. The ANNs have three types of layers, through which they are connected: the input layer, the hidden layers and the output layer [29]. The basic architecture of an ANN can be seen in Figure 2.8.

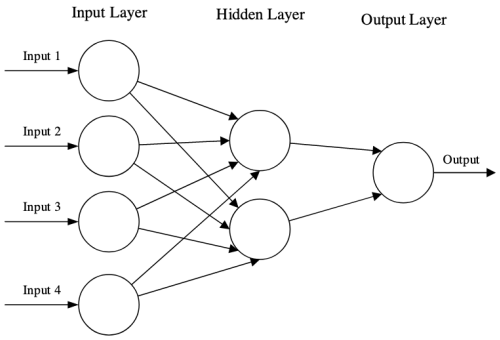


Figure 2.8: Basic Architecture of an ANN (from [41]).

The number of hidden layers might vary, according to the objective. The neurons in the network can be defined accordingly to certain individual parameters. The neurons compute a nonlinear function of the weighted sum of their inputs, as it can be seen in the equation below. The weighted sum of the inputs is performed and the output signal φ is generated. In this equation, ω is the vector of weights, υ is the vector of input signals and m is the number of inputs [3].

- $\varphi = \sum_{i=1}^{m} w_i u_i = \omega^T \upsilon$

The resulting output of the previous equation are the neurons weights associated with inputs, an activation function along with a bias and decision thresholds. Thus, the network configuration can be defined in the learning (training) step. The weights associated with the inputs are an important part for calculating the final output of the network. Initially, all the neurons in the input act only as a buffer meant to distribute the input signals to the neurons in the hidden layer. The weight in each neuron in the input layer is assigned based on its relative importance to other users [3]. In Figure 2.9 it is possible to distinguish two modules, the summation module $\sum$ and the activation module F.

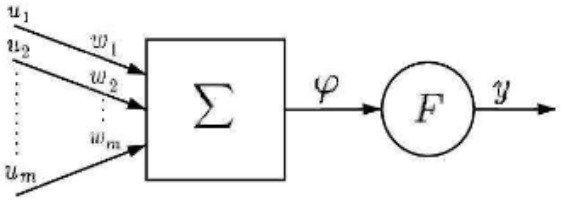


Figure 2.9: Neurons structure in an ANN (from [3]).

Afterwards, the node applies a function to the weighted sum of its inputs, the activation function. The activation function complements the role of the weights in the sense that it decides if a

neuron should be activated or not when calculating and adding a bias. Practically speaking, the activation function applies a non-linear transformation to the weighted sums, making them capable of learning and performing more complex tasks. There are several activation functions currently used and four examples can be seen in the formulas below, where x is the input to a neuron, k is a coefficient, β a given parameter and z is the value of the neurons in the output layer in the functions, respectively [3, 46].

- ReLu: $y = x^+ = max(0, x)$
- Linear: $y = k\varphi$
- Sigmoid: $y = \frac{1}{1+e^{-\beta\varphi}}$
- Softmax: $y = (F(z_i) = \frac{z_i}{\sum_j exp(z_j)})$

The Sigmoid and the Softmax activation functions are the two most used functions in the final layer of the CNN models, when standing before classification tasks. The Sigmoid activation function predicts the class membership probabilities by returning as output a value between 0 and 1 for each class. The form of this function corresponds to an "S" shape between the values of 0 and 1. This results in obtaining an output of 1 when the weighted sum of the inputs is a very large value and to obtain the output of 0 when the opposite happens (very small or negative values as the weighted sum of inputs). This activation function is more appropriate for binary classification problems, where the output can be seen as a Binomial probability distribution. On the other hand, the Softmax activation function, similarly to the Sigmoid, function predicts the class membership probabilities. However, Softmax's output is an array of probabilities, where each value is the probability of the index corresponding to each prediction. This is achieved by scaling down the values, which are afterwards converted into probabilities so that the values in the returned array sum to 1. It is a soft version of the Argmax activation function, where the output returns the a list with 1 in the index of the highest value and 0 for the other array indexes, giving full weight to the index with the highest value and no weight to the others [17].

There is also one other type of activation function which is the threshold function. In it the output signal y is 1 if φ is bigger than the constant threshold value φ_h and 0 if it is the opposite [3].

2.2.4 Convolutional Neural Networks (CNNs)

Amongst the different kinds of neural networks, the simplest type are Feed Forward Neural Networks (FFNN). FFNNs have no cycles and have all the layers above mentioned for the ANNs. The input layer requires as many nodes as coefficients of the feature vector and the output layer is set for the classification results. A FFNN shares the same architecture of the ANN, with a input-hidden-output layer strategy. One subtype of the FFNN are the CNNs, the type of neural networks in the deep learning universe known for its particular performance in the subject of image recognition [29, 45]. The CNNs and the traditional ANNs have strong similarities (they are equal in

certain aspects), considering both have a large amount of neurons meant to be optimised through a learning process. The network will still perform a score function, the weight function, in the sequence going from the input raw image vectors to the class score produced in the final output. There are a few core differences between the two of them. In the CNNs the weight operations are replaced by filters, passing on from layer to layer the data undergoes a series of successive operations, called convolutions. The formula for the convolution operation can be seen below, which can be seen as a weighted average of the function f at the moment t. In it, the weighting is given by the mirrored version of g simply shifted by amount t [45].

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{2.1}$$

An example of a 2-dimensional convolution operation is present in Figure 2.10.

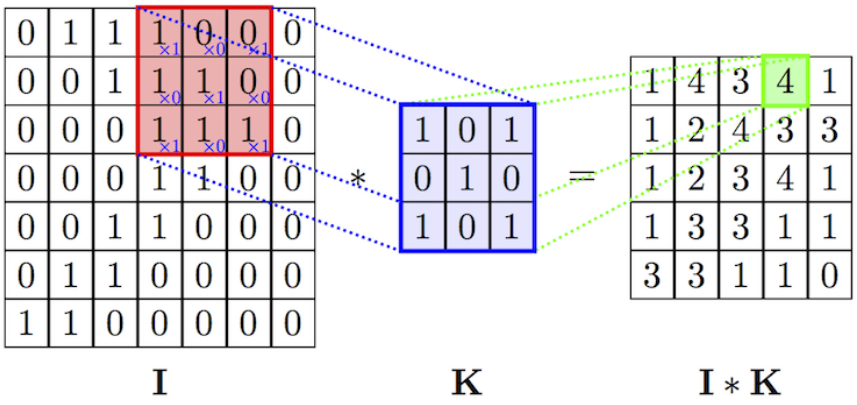


Figure 2.10: Basic operation in a CNN: a convolution operation (from [58]).

Furthermore, these filters can be directly applied to the raw data and there is no need to compute specific features from it. These factors turn the learning approach of the CNNs into an end-to-end one. The sensor retrieves the data and the observations to be classified/described, then the network automatically extracts the features and performs the classification task [29]. Another difference relies on the fact of CNNs being used mainly for pattern recognition within images purposes. This makes the network a bit different with regards to what can be encoded in its architecture. What the CNNs allow investigators to do is encode image-specific features in the architecture of the network, making it more adequate to task in which images are the main aspect. At the same time this is executed, it is possible to reduce the necessary parameters to set up the model. One more evidence that a simple and traditional ANN is not suitable for working with image detection and recognition is due to its difficulties when dealing with the computational complexity inherent to data in the form of images. The differences between ANNs and CNNs is that, in the first case, the features are chosen ad hoc by the system designer, whereas CNNs automatically learn the most effective features for the task from the training data. The solution of increasing the number of nodes and hidden layers in the ANNs to make them capable of dealing with such complexity does not work, due to not having unlimited computing power and training data.

CNN Components

A CNN is made of two separate sections, the feature extractor and the classifier. To the feature extractor are given the images as input data. In Figure 2.11 it is possible to see the two distinct sections of a CNN, the feature extractor and the classifier. After the construction of the dataset, the data is described to compute numeric or symbolic information, to be used as input for the model that is being built. This section of the network is seen as the feature extractor and it is the part of the network that will handle directly with the training data. In the feature extractor of a model there are several layers (or building blocks) that together will form the feature extractor [29]. The primary layers of this part of the network are layers such as the convolutional layers, the pooling layers, the activation function and the dropout layer [46].

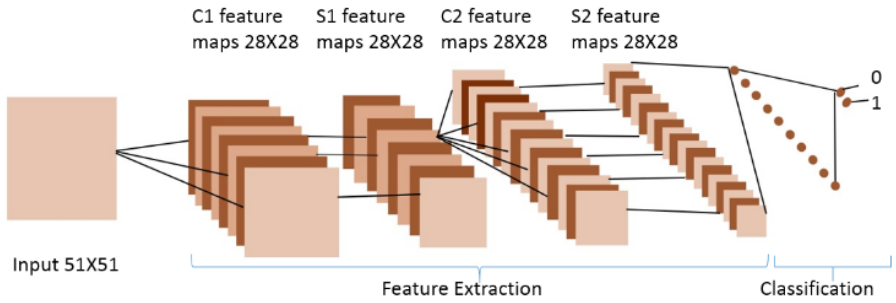


Figure 2.11: Basic components of a CNN, the feature extractor and the classifier. The first one including multiple convolution layers, max pooling layers and the activation function, while the second one includes the fully connected layers (from [42]).

The convolutional layer concern types of layers that consist of a collection of filters (also known as kernels). The input image is put together with these filters to generate the output feature map. As first step, it is necessary to proceed with the kernels definition. Each kernel is defined by a grid of discrete values, where each of these values is called the kernel weight. Furthermore, random numbers are assigned to act as the weights of the kernel at the beginning of the training process. Beside the kernel definition, the convolutional layers also include the convolutional operations, as described in Figure 2.10. This operation starts with the CNN input format being described. Afterwards, this input data is stored in the format of 2-dimensional feature maps, as the multi-channelled image is the input of the CNN. A feature map is the result of applying a convolutional network filter into the input performing the convolution operation. Given that multiple convolutions are applied to the same input, and each of these operations has a different filter, after the input is processed multiple feature maps are obtained [26]. The other two characteristics inherent to the convolutional layers are the sparse connectivity and the weight sharing. The first concept is related to the number of connections between layers, since that in CNNs only a small number of weights are available between two adjacent layers. This leads to an also small number of required weights, proportional to the memory required to store these weights (memory-effective approach). The second term comes from the fact that in a CNN allocated weights between any two neurons of consequent layers do not exist. All the weights operate with all the pixels of the matrix [46].

The pooling layer has an important task in the feature extractor of a CNN, the down-sampling of the feature maps. The down-sampling operation consists in shrinking large-size feature maps to generate smaller feature maps. This operation consists of summarising the presence of features in patches of the feature map. It is necessary due to the output feature maps being sensitive to the location of features in the input data. By applying this technique, the down-sampled feature maps become more robust to the alterations in the position of the features in the image. These changes in the position of the image features are a scenario known as *local translation invariance*. Among the several types of pooling, there are the average pooling, min/max pooling and global average pooling (GAP). The average pooling consists of summarising the average presence of a feature, while the min/max pooling summarise the least and the most activated presence of a feature, respectively. The global pooling is a different type of pooling layer and it differs from the previously mentioned pooling layers since down samples the entire feature map to a feature value, rather than down-sampling patches of the input feature map. It can be used to perform a strong summarising process of a feature's presence in an image. Furthermore, it can also be used as an alternative to using a fully connected layer to the transition from feature maps to an ouput prediction for the model. In Figure 2.12, it is possible to see three types of pooling operations [15, 46].

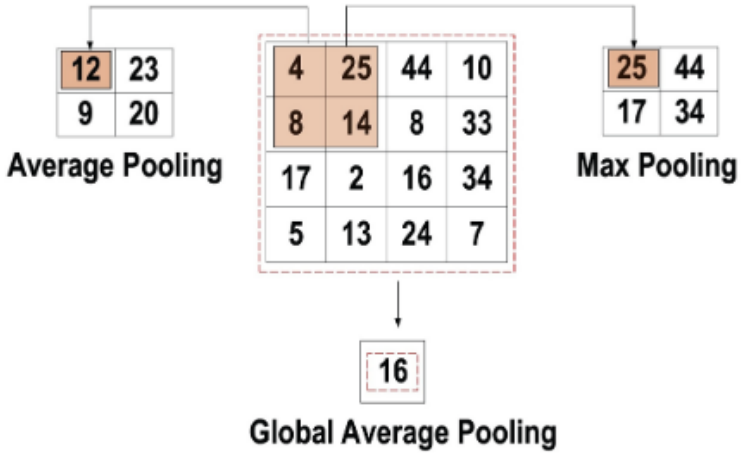


Figure 2.12: Average, max and global average pooling operations, each with its respective result for each quarter of the matrix (from [46]).

Regarding the activation function, its objective is mapping the input to output, as well as introducing non linearity into the output of a neuron. The input value is defined by computing the weighted summation of the neuron input. The activation function was previously described in depth in the section Artificial Neural Networks (ANNs).

Lastly, the dropout layer is a layer that implements a regularisation method that approximates training a large number of neural networks with different architectures in parallel. It is useful as it helps in reducing overfitting. In the training stage, random nodes in the layers are discarded, which means their output will not be proceeding to the following layer. Consequently, this operation leads to the layer to be considered and treated as a layer with a different number of nodes and connectivity when compared to the previous layer. Furthermore, by implementing this layer, it will affect the training stage, making it noisy. Ultimately, given that the outputs of a layer under a dropout layer are

randomly sub-sampled, the capacity of the network will decrease during training. Thus, there might be the need to increase the number of nodes when implementing a dropout layer. Since dropout is implemented per layer in the network, it can be implement along with the majority of layers such as dense fully connected layers or convolutional layers. It can be implemented in the input layer or any of the hidden layers, although it is not used in the output layer [10].

The other component of the model corresponds to the classifier itself. This is made entirely of fully connected layers. In this stage the observations that represent the input data will be classified. This classification will be based on what the model learned from the information extracted from the images representative of the data source used to build the model [29, 46]. The fully connected layers are the final layers of a model. They form the classifier and will be the ones determining the output predictions. In each of these layers, each neuron is connected to all the neurons in the previous layer. This is the reason why it is called a fully connected approach. The method behind this classifier is the one of a standard multiple-layer perceptron neural network (it represents a feed forward ANN). The input to the layers that make the classifier comes from the last pooling or convolutional layer in the feature extractor. This input is represented by a vector, created from the feature maps that are the result of flattening layers. The output from these classes is the final output of the CNN [46].

One last layer that has regularisation functions is the batch normalisation layer. This layer is used for standardising the input to a layer for each mini-batch (when the weights are updated). The effect of using these layers is that it will stabilise the learning process and reduce the number of training epochs necessary to train the network. The number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset. An epoch corresponds to the cycle of the entire training dataset, which happens when the training dataset is passed to the network forward and then backward once. The number of epochs depends on the diversity of the dataset [11, 62]. A batch normalisation layer can be implemented during the training stage by calculating the mean and standard deviation of each input variable to a layer per mini-batch. Consequently, these statistics can be used to carry out the standardisation. After the training stage, these mean and standard deviation of inputs for the layer can be set as mean values observed over the training set. This operation can be applied to input variables for the first hidden layer or to the activations from a hidden layer [12].

Training stage of a CNN

In this stage, the weights of the neurons are updated as the neural network is trained to classify for the goal. Considering that the learning paradigm underlying is the supervised learning, it is necessary to have a set of correctly labelled training data for this stage. In this set, as mentioned earlier, it is necessary to have the correct classification/answer to every data entry. This leads to the cost function, since the goal is to create a model given a set of observations that can achieve an optimal solution. This function will assess how close the model's solution is to the optimal one. Hence, it must be minimised [29]. It might be also necessary to perform data augmentation on the dataset, if the amount of data is small when considering the dimension and complexity of the tasks that are

mean to be solved. Furthermore, data augmentation can also be used to optimise the model, by decreasing the duration of the training stage. Data augmentation is a technique for applying to the images random operations, like normalisation operations. It is a tool to artificially expand the size of a training set. To do so, it creates modified versions of images in the dataset [16].

The learning process is as follows: it starts by loading the input (it is usually a multidimensional vector) to the input layer; then, the input layer will distribute it to the hidden layers; afterwards, the hidden layers will assess the information and make decisions considering the previous layer and weigh-in if a stochastic change in itself will improve or worsen the result/final output; the last layer (the output layer) produces a final class score and a result based upon it, this layer also contains loss functions linked with every class [41]. This is called a feed forward propagation algorithm. The loss function is a function of the parameters of the classifier that is used to optimise such parameters. It is used in the last layer of the model to calculate the predicted error generated by the model, throughout the training samples. This error is representative of the difference between the real and the predicted output. The loss function can be used to estimate the loss of the model so that the weights can be updated to reduce the loss on the next evaluation. The loss function (or cost function) varies according to whether its a regression or a classification problem. The choice of this function must match the framing of the specific predictive modelling problem, such as classification or regression [13, 55].

Regarding classification losses, two of the most common measures are the Hinge Loss (or Multi Class SVM Loss) and the Cross Entropy Loss (also known as Negative Log Likehood). The Hinge Loss concerns a margin in the classification. It is based on the fact that the score of the correct classification should be greater than the sum of all the incorrect categories by a large margin. Considering this, the Hinge Loss is used for maximum-margin classification (their application is mostly in Support Vector Machines) [55]. The formula of the Hinge Loss is shown below:

$$HL = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} - 1) \tag{2.2}$$

The Cross Entropy Loss is the most common measure for classification problems. The cross-entropy loss increases as the predicted probability diverges from the real label [55]. The formula involves logarithmic values and can be seen below:

$$CEL = -(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \tag{2.3}$$

One last loss function is the Kullback Leibler Divergence Loss (KL Divergence). This function measures how different one probability distribution is from the baseline distribution. If the result of the KL Divergence is equal to 0, then it is possible to assume that the distributions are identical. Practically speaking, the KL Divergence is similar to the Cross Entropy loss. This loss function measures the amount of information is lost, in case the probability distribution is used to approximate the wanted target probability distribution. The KL Divergence is used mostly in models which are

meant to learn to approximate more complex functions than multi-class classification. One example of a problem where it can be used is the case of an autoencoder, which objective is to learn a dense feature representation under the model built to reconstruct the original input. The formula for the KL Divergence can be seen below [13, 14].

$$D_{KL}(p(x)||q(x)) = \sum_{\chi \in X} p(\chi) \log \frac{p(\chi)}{q(\chi)} \tag{2.4}$$

During the training phase it is important to care for the amount and quality of the data that are being used to train the network. Otherwise, over and underfitting phenomena might occur. Overfitting happens when a network assesses the training data too well. A network might overfit if it is trained with a huge or too reduced amount of data and with a large number of parameters. In this case, given the data and parameters of the network, it will only perform for the cases that were covered in the training data. The model will not be able to perform in unseen cases. The most common solution to avoid overfitting-related issues is to stop the training of the model. This can be done by using a slow learning rate or to use small random initial values. On the other hand, we might also face the concept of underfitting, which happens when the model is not trained enough. In other words, the model could not acquire enough knowledge from the data to perform classifications of equal images to the ones used for training, resulting in a poor performance of the system [41]. In Figure 2.13, the under and overfitting phenomena are observable as well as the standard fitting.

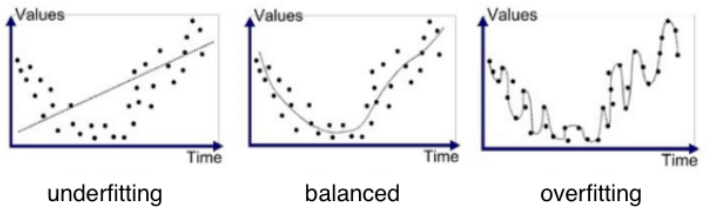


Figure 2.13: Underfitting and overfitting phenomena (from [5]).

Once the model is fully built and the activation and loss functions are chosen, an optimisation algorithm can be used to determine a function that minimises the cost function. One widely used is the stochastic gradient descent. A gradient descent consists on finding the values of the parameters (coefficients) of a function allowing to minimise the cost function. This is a method that is best used when the optimal parameters can not be calculated analytically, hence an optimisation algorithm is needed. The process of a gradient descent starts by setting the values for the coefficient(s) of the function (they can be close to zero, although usually they are not set to zero). On the next step, the coefficients are evaluated while being passed to the function and later calculating the cost. Subsequently, the derivative of the cost is calculated and the process is repeated until the cost of the coefficients is zero or close to zero. This process can be slow on large data sets, since it demands a prediction for every instance in the training set. Due to this fact, the stochastic variant of the algorithm is used, as in the process of this version the update of the coefficients is performed for each training instance (and not at the end of the batch of instances). The batch size corresponds

to the hyperparameter that sets the number of samples for which predictions will be made before the internal model parameters are updated. A mini-batch size corresponds to a specific size of the batch where it is bigger than 1 but smaller than the size of the training set [11]. The first stage in this procedure is meant for the randomisation of the order of the training data set. The updates on the coefficients are done as for the standard variant, except the cost is not summed over all training patterns, alternately it is calculated for one training pattern [8]. The stochastic gradient descent can be seen as way to perform learning while using the gradient.

To efficiently compute the gradients, an algorithm is used. It is called back propagation algorithm and it allows the cost information to flow backward through the network to compute the gradient. The back propagation algorithm can numerically evaluate an analytical expression, which can be computationally expensive. With the back propagation algorithm is possible to compute the gradient $\nabla x f(x, y)$ for an arbitrary function f, where x is a set of variables whose derivatives are needed, and y is an additional set of variables representative of inputs to the function, but whose derivatives are not required [31].

Examples of CNN architectures

In this section, we will provide a description of commonly seen CNN architectures used to solve DL classification problems. Multiple frameworks for architectures will be described, starting from simpler older ones and going into deeper and more recent architectures. Different aspects of each architecture will be covered such as the type of connections used, the size of their building blocks, how many layers the models have and the performance in known dataset challenges.

- VGG16 Network (VGG16)

The VGG architecture is a classic CNN framework that intends to investigate how the depth of a network affects the Accuracy (AC) in classification tasks of large scale datasets. Hence, the depth of this network was a topic widely covered by the authors in K. Simonyan and Andrew Zisserman [63]. Parameters of the architecture were fixed and, at the same time, the depth of the network was increased by adding more convolutional layers. While doing so, the authors tried to maintain it as stable as possible. This increase of the depth of the networks was possible due to the use of small convolutional filters with dimensions $3x3$. One other aspect that characterises the VGG framework is its simplicity, given that beside the already mentioned convolutional filters, the remaining layers are only pooling layers and a fully connected layer. There are 5 max-pooling layers that carry out spatial pooling after several (but not all of) the convolutional layers. The set of convolutional layers is followed by 3 fully connected layers, with the first two having a different number of channels compared to the last, where the number of channels is equal to the number of classes (1000 in the case of the Imagenet dataset). The final layer is a softmax layer and all the hidden layers use ReLU as activation function. The number of channels in the convolutional layers starts with 64, which is small number of channels and, therefore, a small width. Afterwards, this number is increased by a factor of 2 after each max-pooling layer until it reaches 512. The VGG16 is, therefore, a version of this architecture

where there are 16 layers, and 13 of them correspond to convolutional layers and the last three are fully connected layers, corresponding to its classifier, followed by a softmax activation function [63]. A schematic representation of this architecture can be seen in Figure 2.14.

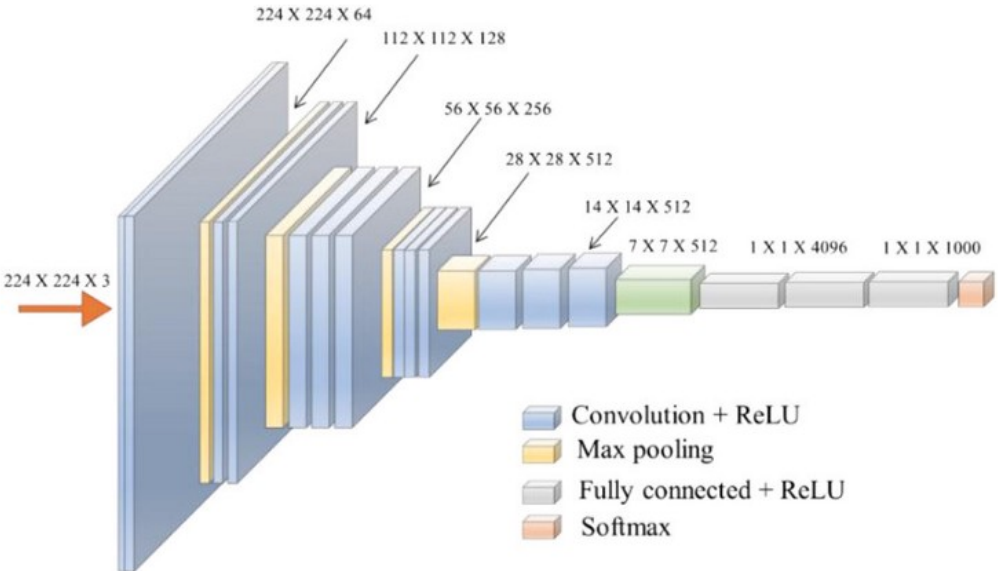


Figure 2.14: VGG16 architecture with the sequence and channel width of each layer (from [54]).

Even though these networks are deep, the number of weights is not greater than the number of weights of a more shallow network, with larger convolutional layers widths (number of channels) and receptive fields [63].

- Resnet50 (RN50)

This architecture derives from the residual learning framework and its purpose is to facilitate the training of deep networks. The layers in these architectures are reformulated, turning them in residual learning functions with reference to the layers input. Learning better networks can be a problem due to the lack of convergence, caused by the vanishing/exploding gradients. Vanishing gradient occurs when the backpropagation algorithm advances backwards from the output layer to the input layer, leading the gradient to decrease. The gradient approaches 0, which leads to the initial weights remaining unchanged. On the opposite, exploding gradient happens when the gradient continuously increases as the backpropagation algorithm progresses. In this case, the weights will be largely updated, causing the gradient descent to diverge [56].

A solution to these problems consists in the introduction of a deep residual learning framework, based on the idea that it is easier to optimise the residual mapping than to optimise the original, without reference mapping. In Kaiming He et al. [35] it is also introduced the concept of shortcut connections, which correspond to connections where one or more layers are skipped. In this case particularly, these connections are meant to perform identity mapping and its outputs are added to the outputs of the stacked layers. While doing so, these connections do not interfere with the computational complexity.

The networks designed in Kaiming He et al. [35] are composed by several blocks of convolutional

layers like what is seen in Figure 2.15. For these blocks, in each pair of 3x3 filters is added a shortcut connection. The RN50 consists of 50 layers with a 7x7 64-layer as first convolutional layer, followed by the several 64, 128, 256 and 512 convolutional blocks and ending in an average pooling layer, followed by the output layer.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 2.15: Resnet architectures developed for the dataset Imagenet with the building blocks shown in the brackets. In the 50-layer column it is possible to observe the architecture of the RN50 and its building blocks (from [35]).

The building block of the Resnet50/101/152 described by the authors follows a bottleneck design. Such is implemented in order to reduce the training time. In the three Resnets referred above, for each residual function a stack of 3 layers is used unlike for the Resnet18/34. These three layers are the 1x1, 3x3 and 1x1 convolutions as seen in the highlighted area in the Figure 2.15. The 1x1 convolutions intend to resize the images dimensions. The first convolution decreases the image and the second one restores its original size. This way, the 3x3 convolution has a bottleneck with smaller input/output dimensions. In the Figure 2.16 it can be seen the difference in terms of blocks from the Resnet18/34 to the Resnet50/101/152, with the introduction of the third convolution.

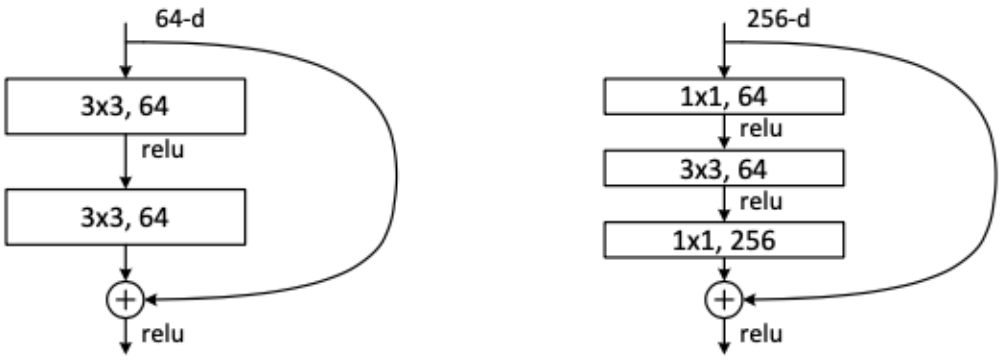


Figure 2.16: Resnet18/34 and Resnet50/101/152 (bottleneck) building blocks, the latter one with an additional convolution (from [35]).

Ultimately, very deep residual networks are easier to optimise, when compared to other simple architectures with only a large number of stacked layers. They show higher training error when the

depth increases. And, lastly, the residual networks have gains in the AC performance metric from the increased depth, having achieved better results than other networks.

- Inception-ResnetV2 (IRV2)

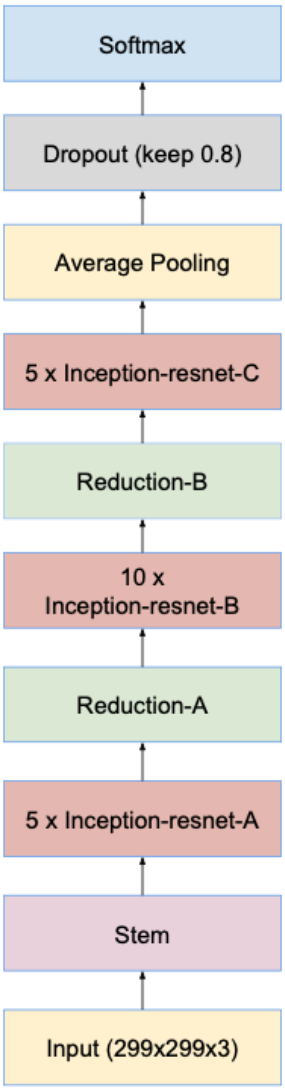
The Inception-Resnet architecture represents a new approach to the already known Inception architecture with the introduction of residual connections. The purpose of adding such connections is to accelerate the training of the Inception networks. In some cases, the performance of models built on this new type of architectures improved when compared to Inception only based models. This is related to the fact that Inception architecture belongs in a group of very deep convolutional networks. Added to this fact, residual connections are considered to be important in the training of deep architectures. The reason behind this idea is related to the replacement of the filter concatenation stage in the Inception architecture with the newly introduced residual connections. By doing so, this architecture is able to maintain its computational efficiency, while obtaining the advantages from the residual approach. These motives are the reason why it makes sense to create an Inception architecture with residual connections between layers.

The Inception architecture is highly tunable, which gives the possibility to perform several changes to the number of filters in the layers, without affecting the quality of the fully trained network. The residual version of the Inception network is different from the original, specifically in the blocks of the network. In the Inception-ResNet, each Inception block is followed by a filter-expansion layer (this represents a 1×1 convolution without activation). This is used in order to scale up the dimensionality of the filter bank before the addition to match the depth of the input. In turn, it will compensate for the dimensionality reduction that results from the Inception block. In Figure 2.17a, it is possible to see the large scale structure (the blocks) of the IRV2 and in Figure 2.17b there is the sequence of layers, including the input block of the IRV2.

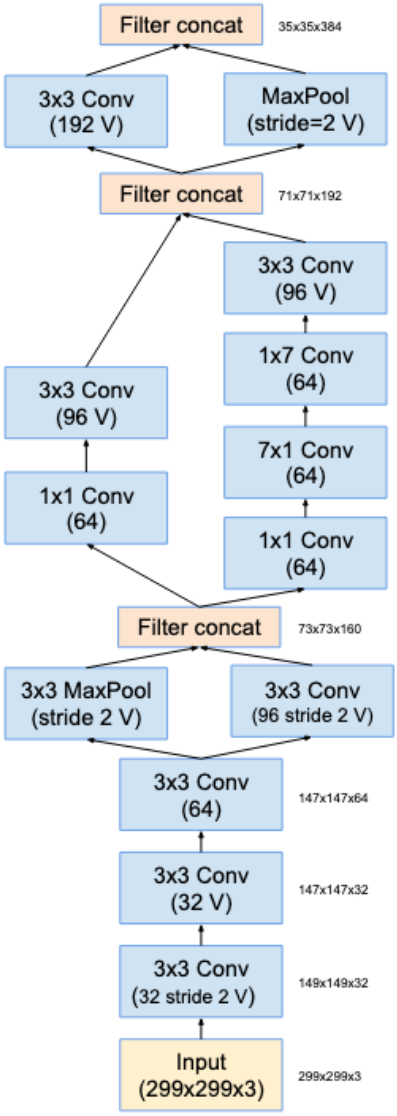
One other difference between the residual and the non-Residual Inception variants is that for Inception-Resnet, the batch-normalisation is used only on top of the traditional layers (not on top of the summations). This is due to the fact that by omitting the batch-normalisation in some of the layers, the memory expenditure decreases. This makes it possible to increase the overall number of Inception blocks substantially. However, with more computational power, such trade-off should not be necessary.

Apart from the batch-normalisation layers, the number of filters for the Inception-ResNet architectures must also be controlled. The creators of this architecture found that if the number of filters exceeds 1000, the residual variants would start to exhibit instabilities and the network would not progress in the training stage. This would lead for the last layer before the average pooling layer to start to generate only zeros after a few tens of thousands of iterations. Such situation could not be avoided, even if the learning rate was lower or with extra batch-normalisation layers in this layer [69].

Lastly, scaling down the residual connections before adding them to the previous layer activation stabilises the training. The IRV2 is a hybrid version of the Inception architecture with improved performance. The introduction of residual connections leads to improvements in the training speed for this architecture [69].



(a) The blocks with its respective layers from the Inception-ResNetV1 and IRV2 networks.



(b) Input part of the pure versions of the IRV2 and InceptionV4 networks, which includes the stem.

Figure 2.17: The blocks that constitute the Inception-ResNet (versions 1 and 2) and the input phase for the models InceptionV4 and IRV2 (from [69]).

- DenseNet169 (DN169)

The DN169 is a network that comes from a class of CNNs, the Dense Convolutional Neural Network. The goal of the DenseNet architecture is to create deeper, more accurate and efficient in the training stage networks. The main difference when compared to the previously CNN is that, while these past CNNs share the characteristic of creating shorter paths from the early layers to the latter layers, the DenseNet does not implement this idea. Instead, it connects all the layers directly to each other, as long as their feature-map sizes match. This allows to maximise the information flow between the layers in the network. In addition, the layers obtain more input coming from the previous layers and they pass on their feature-maps to the subsequent layers. Such action favours

the preservation of the feed-forward network nature. Opposing to what is done in the ResNet, the DenseNet combines the features in the network by concatenating them (while the ResNet does it by summing before passing them to the layers). As a result, a certain layer will have a number of inputs equal to the number of layers that precede it. As an example, when in the 10^{th} layer, this layer will have 10 inputs. These inputs are the feature-maps from the previous convolutional blocks. The feature-maps from the layer are then passed on to the remaining subsequent layers in the network (if a network has N layers then when in the nth it will pass its feature-maps to the following $N - n$ layers). As an outcome from this methodology the number of connections in a network with N layers is $\frac{N(N+1)}{2}$, as opposed to the regular architectures where it is N. The term "dense" derives from this greater number of connections. The representation of how this architecture and its feature maps function can be seen in Figure 2.18.

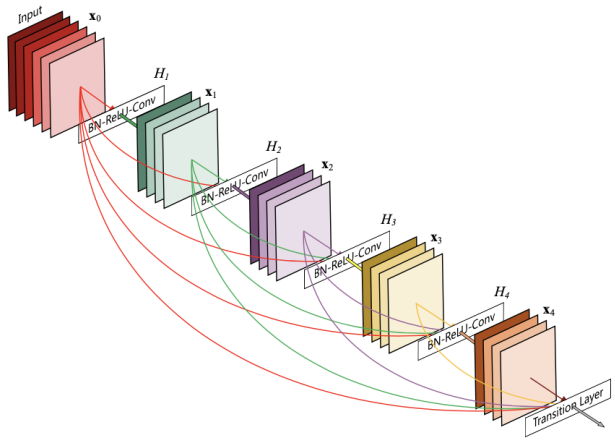


Figure 2.18: Practical perspective of an example of a DenseNet with 5 blocks, where each layer takes all the feature-maps from the previous layers as input (from [38]).

In the DenseNet architecture there is a distinct difference between the information that is added to the network and the information that is preserved in the network. The DenseNet layers are often narrow, with 12 filters per layer. Consequently, only a small set of feature-maps are employed and contribute to the "knowledge" of the network. The features that were already in the network when this scenario happens stay unchanged. The decision/classification is then performed by the final classifier based on the collection of feature-maps from the network. One considerable advantage from the DenseNet architecture is the improved flow of information and gradients in the network. This leads to an easier training stage, given that every layer has access to the gradients from the loss function, which results in an inherent deep supervision. This is a major contributor to the better training of deeper network architectures. One more characteristic about the DenseNets is related to its ability to take advantage of the network by implementing feature reuse. Through such, it is possible to create models that are easier to train and efficient in terms of parameters. The concatenation of the several feature-maps learned throughout the different layers leads to a higher variation in the input of the following layers, as well as an improvement in efficiency. When compared to another architecture used for this Master's thesis, the InceptionResNet (that also implement feature concatenation from different layers), the DenseNet is simpler and more efficient [38]. In Figure 2.19 it is possible to see

the standard layers of the DN169 architecture.

Layers	Output Size	DenseNet 169
Convolution	112×112	7×7 conv, stride 2
Pooling	56×56	3×3 max pool, stride 2
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv
	28×28	2×2 average pool, stride 2
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv
	14×14	2×2 average pool, stride 2
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Transition Layer (3)	14×14	1×1 conv
	7×7	2×2 average pool, stride 2
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Classification Layer	1×1	7×7 global average pool
	1000	1000D fully-connected, softmax

Figure 2.19: Blocks and the layers that form the DN169 (from [1]).

Finally, the DN169 has several advantages that encourage its use. Just like the RN50 it reduces the vanishing gradient problem, strengthens the feature propagation, implements feature reuse and a decrease on the number of parameters. The different connectivity pattern introduced, the direct connections from any layer to all subsequent layers, improves the flow of information throughout the network [38].

- NasNet Large (NNL)

This architecture was created in 2018 by Google Brain authors (Barret Zoph et al. [73]) with the goal of creating an architecture that could learn directly on the dataset selected, regardless of the dataset's size. In order to do so, the network would be trained in a small dataset and, afterwards, this block would be transferred to a larger dataset. The essential idea behind the NNL is that it is inspired in the Neural Architecture Search (NAS) framework. This framework uses a reinforcement learning search method in order to optimise the architecture configurations. Applying the NAS framework directly in a large dataset, such as Imagenet, is an expensive task computationally speaking. The concept of transferability is achieved by creating a *search space*. This is built so that the complexity of the architecture is independent of the depth of the network and the size of the input images. In this *search space* there will be convolutional networks with the same convolutional layer, but with different weights inherent to each one. Therefore, the goal is to find the best layers and its structure.

The NASNet is adaptable in a way that it allows to create several versions of this architecture, by varying the number of convolutional cells and number of filters in the convolutional cells. This way, different versions of the NASNet architecture with different computational demands can be created.

The first approach to a problem where the NAS framework is used is to attempt to search for an

architecture building block (cell or layer) on a small dataset and then transfer the block to a larger dataset. In a more detailed way, the process is described as follows: a controller RNN examines child networks with different architectures; afterwards, these child networks are trained to convergence to obtain accurate results in a validation set; these results are used to update the initial controller leading it to generate better architectures over time [73]. A scheme of how the NAS framework operates can be seen in Figure 2.20. In this figure, the controller predicts architecture A from a search space with probability p; afterwards, a child network with the chosen architecture A trains to convergence, while achieving AC R; in the last step, the gradient of p is scaled by R to update the RNN controler.

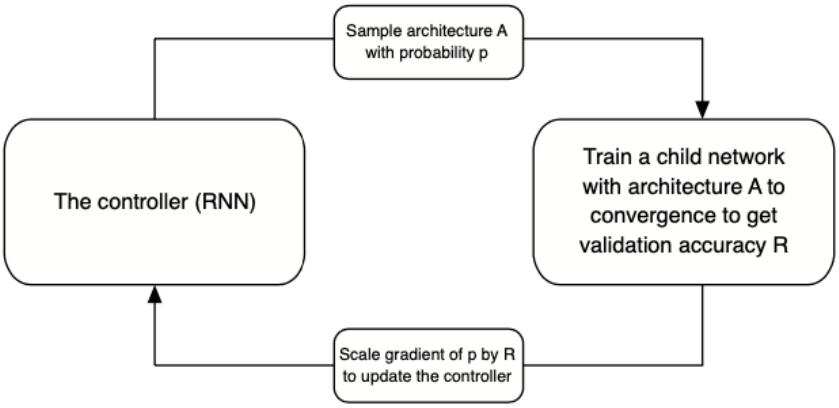


Figure 2.20: Scheme illustrating the standard framework for NASNet (from [73]).

In the paper [73], the investigators started by choosing a cell (or the best convolutional layer) in the small dataset (CIFAR-10 in this case), then applied the object selected to the Imagenet, along with more copies of this object on top of the first object. In this methodology, the overall architectures of the CNNs are manually predetermined. They are defined by the convolutional cells which are repeated several times, where each convolutional cell has the same architecture, while having different weights. To find these convolutional cells a reinforcement learning search method is used. To build scalable architectures to receive images of different sizes, two specific types of cells are required in order to receive as input the feature maps: Normal cells and Reduction cells. The Normal cells are convolutional cells that return a feature map of the same dimensions. On the other hand, Reduction cells correspond to convolutional cells that return a feature map with its height and width being reduced by a factor of two. The Normal Cells and the Reduction cells can share the same architecture. However, the authors found that having different layers and cell blocks has more advantages for the network. In the end, what varies in each convolutional network chosen by the NASNet framework is the structure of the Normal and Reduction Cells, searched by the controller. The structure of these cells can be found in the previously referred *search space*, where each cell receives as input 2 hidden states, h_i and h_{i-1}. These states are the outputs of the two cells in the previous layer or the input image. The rest of the convolutional cell is afterwards predicted by the controller RNN, using the initial hidden states. The predictions carried out by the controller for each convolutional cell are grouped into blocks of size B, where each block has 5 prediction steps made by 5 softmax classifiers. The 5 steps are described below.

- Steps 1 and 2 - Choose one hidden state from the set of h_i, h_{i-1} and hidden states from previous layers. Subsequently, choose another hidden state from the same set of possible choices.
- Steps 3 and 4 - An operation to perform on the first hidden state is selected and afterwards another one to perform on the second hidden state.
- A method to combine the output of steps 3 and 4 is selected to create an hidden state.

In Barret Zoph et al. [73], an adequate size for B was thought to be 5. Thus, the RNN controller repeats the previous 5 prediction steps 5 times, which corresponds to the B blocks in a convolutional cell. Some of the operations that can be selected for the steps 3 and 4 are: 3x3 average pooling, 5x5 max pooling, 1x1 convolution, 3x3 max pooling, 1x7 then 7x1 convolution and 3x3 convolution.

Chapter 3

State of the Art

3.1 DL methodologies for lesion detection in stomach cancer

Currently, researchers have already performed academic studies whose main target was to use artificial intelligence and deep learning techniques to detect the appearance of stomach cancers.

3.1.1 Toshiaki Hirasawa et al. 2018

Toshiaki Hirasawa et al. [37] combined artificial intelligence with Convolutional Neural Networks (CNNs) to create a software capable of diagnose gastric cancer through endoscopic images. The training data consisted of 13584 endoscopic images of gastric cancer. The test data had only 2296 images, gathered from 69 patients with 77 gastric lesions. This data was applied to the constructed model, in order for it to perform the predictions. Their CNN-based system was built according to a Single Shot MultiBox Detector architecture [47]. This type of architecture achieved in the past positive results in terms of precision over object detection tasks on standard datasets. A detection task returns a bounding box when in the presence of a lesion, instead of simply providing a label for the image. This bounding box indicates the regions of the images where the lesions are present. This architecture is referred at as single shot since the tasks of object localisation and classification are done in a single forward pass of the network; MultiBox is the name of the technique for bounding box regression first described by Wei Liu et al. [47]; lastly, the Detector is self-explanatory as the network is an object detector that also classifies those detected objects.

An explanation on what Single Shot MultiBox Detector (SSD) is becomes necessary, in order to understand how the model was built. This method is based on a feed-forward convolutional neural network, producing a finite defined number collection of bounding boxes and scores for the presence of an instance belonging to each object class in those boxes. The first layers of the network are built considering a basic architecture used for high quality image classification (also known as the base network). This base network corresponds to VGG16 network, described in Chapter 2. It was used as the base network in their research because it shows a strong performance in high quality image

classification tasks while, at the same time, discarding the fully connected layers. A set of auxiliary convolutional layers were used, rather than the fully connected layers, due to the fact that it enables to extract features at multiple scales and diminish the input data for each subsequent layer [28, 71].

The SSD method discretizes the output space of the bounding boxes (boxes that delimit the region in the image that is going to be analysed) into a set of default boxes that consist of different aspect ratios and scales per feature map location. When the model is about to do a classification, it generates scores for the presence of each feature per default box. Afterwards, it takes this information into consideration and applies corrections to the box that offered a better result, so that it can match more accurately the object shape. Therefore, SSD are easier to train and to integrate into systems that must have a detection component [47, 48].

The images were obtained from two hospitals and two clinics, from screening or preoperative exams from daily routine exams, performed with standard endoscopes. The training dataset consisted of ground truth labels, which were obtained via histological analysis. Additionally, the lesions in the images were marked by a specialist on gastric cancer, using rectangular frames. In the data meant for testing 62 cases had only one lesion, 6 cases had two and 1 case had three lesions [37].

In the test stage when detecting a gastric cancer lesion, the CNN issued an output with the position of the lesion in the stomach and the name of the disease. The options for the disease were "early gastric cancer" and "advanced gastric cancer". The first lesion relates to an early stage of the cancer infection, described by the authors has having subtle morphological changes. On the other hand, the advanced gastric cancer lesion concerns a notorious lesion, with considerable morphological changes and generally a poor prognosis for the patient. At the same time, the authors created rules to perform the tests, considering that some of the images present in the test dataset were relative to the same gastric lesion. When the CNN detected only one stomach lesion in several images which had the same stomach lesion (but were not detected as that one in particular), it was still defined as a correct prediction. Additionally, when the lines delimiting the gastric cancer were unclear and the CNN detected only a partial gastric cancer lesion, it was also considered a correct prediction. Consequently, the model achieved an overall Recall (RC) of 0.92 and a Positive Predictive Value (PPV) of 0.31. 70 lesions in a total of 71 with a diameter greater than or equal to 6mm were detected by the CNN, resulting in a score of 0.99. All the missed lesions were considered difficult to identify and distinguish from gastritis even for experienced endoscopists, as they were superficially depressed and differentiated-type intramucosal cancers.

On the other hand, 0.69 of the lesions diagnosed were benign. The misclassifications happened due to the similarities between the kind of lesions that were expected to be found and other lesions, e.g., gastritis with redness, atrophy, and intestinal metaplasia. Despite the fact that these are stages of the development of a gastric cancer according to the Correa's cascade, for this study in particular they were not included in the gastric cancer section.

Despite the relatively good results obtained, this study has several limitations. The data used came strictly from high-quality endoscopic images both for the training and test datasets. Also, the scenarios were only of gastric cancer cases, which represents a smaller part in the universe of

endoscopic exams (as mentioned earlier there also scenarios of gastritis and intestinal metaplasia). One last limitation is related to the lesions considered to be well classified when they were not, not being histologically proven, hence hidden lesions may be included in these records.

3.1.2 Bum-Joo Cho et al. 2019

In another study, Bum-Joo Cho et al. [20] explored the classification of stomach neoplasms through endoscopy images using different convolutional neural networks architectures. The data used in this project were endoscopic white-light images of pathologically confirmed gastric lesions. The dataset had a total of 5017 images from 1269 patients. The test set had 812 images from the original number concerning 212 patients. Similarly to the previous study, the authors defined criteria to evaluate the quality of the images, e.g., if the images were out of focus/blurred/shadowing or if they did not have any pathology.

The five categories in which the data would be classified were: advanced gastric cancer (AGC), early gastric cancer (EGC), low grade dysplasia (LGD), high grade dysplasia (HGD) and non-neoplasm (NN). The last category included benign ulcers, erosions, intestinal metaplasia or any form of gastritis. The classifications could later be included in two different sets of categories. They could be classified as cancer (AGC and EGC) and non-cancer (LGD, HGD and NN). The other set was the neoplasm category and it was divided in neoplasm (AGC, EGC, LGD and HGD) versus non-neoplasm (NN). When creating the test and training datasets, the images were divided by using random sampling and it was based on the patients and not on the images. Multiple images could belong to the same patient as some images were taken with different angles, directions or distances. For each patient in the test set there would be 5 patients in the training set (1:5 ratio) in each category of gastric lesions. Consequently, lesions belonging to the same category in one patient were assigned together either into the training group or the test group. Nonetheless, if a patient had at the same time lesions of different categories, the lesions could belong to different datasets because lesions of different categories were randomised independently. The test dataset was not balanced in order to place an analogous number of lesions in each category.

Afterwards, the authors used the training dataset built by them to fine-tune the CNNs that would classify the gastric lesions. These CNNs were pretrained in the ImageNet dataset. Subsequently, the test data was used to evaluate the performance of the built models. Only the model with the best performance was evaluated in the next step of the study, where a comparison was made between it and the endoscopist's performance. This third dataset was used to approve the established models and compare it with the performance of three endoscopists. These endoscopists classified the dataset without knowing the final diagnosis. There were 200 images from 200 patients. This dataset served as a prospect validation one.

Regarding the CNN models, three models were built: Inception-v4, Resnet-152 and IRV2. Once the models were built with the training dataset and their performance evaluated with the test set and the prospect validation set, the primary outcome measures were to classify the models performance

for the above-mentioned paradigms (gastric cancer and non-cancer followed by gastric neoplasm and non-neoplasm). The statistical measures used to study the model's performance were the area under the curve (AUC)[1], the RC, specificity, the PPV and the negative predictive value (NPV). The continuous variables were expressed as the mean (and standard deviation). The categorical variables were expressed as a percentage with 0.95 confidence interval (CI).

For the test stage, firstly the authors evaluated the 5 category evaluation performance. The IRV2 had the best performance with 0.85 of AC. Regarding the per-category AUC of the models it had its highest for lesions AGC and lowest for lesions HGD. The per-category RC was highest for lesions with non-neoplasm and lowest for lesions with HGD.

In the first part of the binary classification task (focused on whether the gastric lesion was cancer or not) the IRV2 showed the best performance with an AUC of 0.88. The AC, RC and specificity in the classification of the gastric cancer were 0.82, 0.76 and 0.85, respectively. On the other hand, the IRV2 also had better results in the classification of gastric neoplasms with an AUC of 0.93. The AC, RC and specificity were 0.86, 0.84 and 0.87, respectively.

The last dataset, the prospective validation dataset, had images of 74 cancers against 126 non-cancers and 130 neoplasms against 70 non-neoplasms. When classifying the validation dataset into 5 categories, the endoscopist with the best performance had an AC of 0.88, while the IRV2 achieved 0.76 for AC (these calculations are the weighted average of each category). Thus, the endoscopist had a better performance. About the per-category performance, the group of endoscopists showed overall the highest result when diagnosing the AGC (AC varying between 0.98 and 0.99) and their second highest diagnostic performance was in the diagnosis of NN (AC varying between 0.86 and 0.90). Yet, regarding LGD and HGD, their performance was lower with an AC range between 0.80 and 0.86, which also happened with IRV2. Onwards to the binary classification performance of the endoscopists, in the task of determining if the gastric lesion was cancer or not, the best performance had an AC of 0.98, a distinct value from 0.76 of the IRV2. Nonetheless, the model and the remaining endoscopists achieved results with no statistically significant difference. In the final classification problem, the gastric neoplasms, the examiner with the best performance achieved an AC of 0.97, significantly better than the 0.74 of the IRV2. In this case, there was no statistical difference between the endoscopist with the worst performance and the model. All the presented values were obtained with a CI of 0.95.

The described project stated the value of high performance models when analysing gastric lesions categorised into certain types. The authors concluded that the models created were capable of assisting professional medical staff when conducting endoscopy screening procedures by predicting the histology of ambiguous lesions and accomplishing diagnostic and treatment scenarios. The authors also stated that automated classification and diagnosis of ambiguous lesions with such models can lead to the reduction of unnecessary biopsies, surgeries and surgery-related complications.

This study held complications in terms of data quantity since the binary classification of the

[1] A performance measurement for classification problem at various thresholds settings, informative of how much the model is capable of distinguishing between classes.

performance of the model by prospective validation was lower than the endoscopy with the best performance. Bearing in mind that the number of neoplasms was low when compared to the number of non-neoplasms lesions in the training dataset, the performance could be improved if the training dataset had a greater number of images and more balanced data (in this study the authors did not do class-balancing). Beside, in retrospective, studies such as this one there is always a tendency for some selection bias. This is due to the fact that some images were taken from an older endoscopy device with low brightness and/or resolution, when compared to a recent one.

3.1.3 V. V. Khryashchev et al. 2019

A different approach consisted on the evaluation of images from Magnifying Endoscopy (ME) and Narrow Band Imaging (NBI) and it was explored by V. V. Khryashchev et al. [43]. In this article the authors designed and implemented an algorithm for pathology detection of gastric lesions in endoscopy images through convolutional neural networks. The dataset type is different from the one focused on the present thesis and covered in the previous literature. Despite this fact, the author's DL approach and logic are similar, which makes this paper one to be considered.

ME and NBI are considered high-tech methods of endoscopic diagnostics and, consequently, increase the level of diagnosis of a pathology in areas seen in endoscopy images, as seen in Chapter 2. After geometric transformations were applied, the dataset used for this paper consisted of 1293 images (357 cancer alterations images). The pathological changes concerning the gastric mucosa searched for in the images are related to microvascular patterns and the microstructure pattern. This kind of changes will be observable at the edge of the lesions, having the name of demarcation line. Due to the polygons abnormal and complex shape in the images, the demarcation line formed rectangles.

The gastric mucosa has different components with different structures along with a complex microarchitecture, which demands for a assessment of both the microvascular pattern and the microstructure pattern of the epithelium. Knowing this, the authors defined three classes for the algorithm's object detection: normal mucosa (I), non-cancerous pathology (II) and cancer (III). The model developed was also based on the SSD-VGG16. Initial layers were based on a short version of the VGG-16 architecture. Also, the VGG16 shortened neural network trained on the Imagenet was used to initialise part of the weights of the SSD-VGG16. The remaining layer's weights were initialised using Xavier initialisation [18] and as an optimisation algorithm a stochastic gradient descent was used. Afterwards, the model was trained with 1193 training images.

The model was tested with 100 images (24 belonging to class III, 76 images of the remaining classes). The quality of the model was measured with the average precision (AP - averaging of the AC values for different threshold values) and the mean Average Precision (mAP - main measure of the operation quality of object detectors). The AP values were 0.83 regarding the cancer class and 0.92 for the non-cancer class, the value for the mAP was 0.88. This paper maintains a solid logic and a well constructed model, the SSD-VGG16 with the VGG16 neural network trained on Imagenet. However, one reasonable problem, as stated by the authors, is related to the size of the dataset, which

should be larger. The results can be improved if more data are used to train and test the model.

3.1.4 Takumi Itoh et al. 2018

Another method is present in a study performed by Takumi Itoh et al. [40]. This paper stands out as its premise is different from the remaining literature due to the fact of studying Helicobater Pylori-related infections. The purpose of the article was to create a system, a CNN, capable of detecting and diagnosing an early infection caused by the presence of HP. The input also came from upper endoscopy images of patients.

The endoscopy method used was the standard white-light one and the HP-cause infection was diagnosed following the presence of redness and swelling of gastric mucosa. The images that formed the dataset were from 139 patients. 46% of the patients were considered HP positive by having greater than or equal to 10 U/mL on a serum HP IgG antibody levels (the ones with less than or equal to 3 U/mL on HP IgG antibody were considered negative). The training set images were obtained from the lesser curvature of the stomach, and after angle transformations (45°, 90° and 180°) with the purpose of data augmentation, reached 596 images. The number of test images was 30 due to the fact that the initial number of images was 179 and 149 of them were alter transformed to obtain the 596 training images.

The model built was a CNN with resource to the GoogLeNet Deep CNN for standard object recognition [68]. This CNN was a 22-layer network and the transfer learning process was based on fine-tunning, which in turn was based on a pre-trained network. The transfer learning process was the method adopted for the learning of the training data. A stochastic gradient descent was also used to optimise the network. The method analysed each image by focusing on the centre of the image and updating the resolution. The results of the model were compared to the serum HP IgG antibody levels tests the patients took. The performance metrics chosen to evaluate the results were the RC, the specificity and the area under the receiver operating characteristic (ROC) curve (AUC). Considering the intervals of classification for positive and negative the RC was 0.87, the specificity 0.87 and the AUC was 0.96.

The model showed positive insights for the stated problem. Unlike what happens when doctors perform an endoscopy exam followed by a biopsy to a patient, the model is not affected by the site at which the biopsy specimen (tissue sample) is harvested. The authors also selected images of the selected curvature of the stomach which meant a higher sense characteristic of this area and the diagnosis of HP infection was simplified. On the other hand, the authors excluded patients that had a procedure to eradicate HP and patients with IgG antibody levels between 3U/mL and 9U/mL. This is a considerable interval, however generally these values are in a boundary region indicative of infection and the data on this paper did not have cases where values belonging to this interval occurred. Hence, if this kind of information was included in the dataset it would increase the performance of the diagnosis and the coverage of the paper.

3.1.5 Qi He et al. 2020

One other type of classification problem is one related to anatomical landmarks detection, also using images from upper endoscopy exams. Qi He et al. [36] performed a 12 class classification task, using upper endoscopy images. This study intended to build several models capable of classifying images from a dataset with 3704 images from the GIT, coming from upper endoscopy exams. Despite differing from the scope of this Master's thesis, a task such as the one in this article involves requisites similar to the ones where the main goal is the classification of images containing lesions from the GIT in upper endoscopy images. The images that would be used as input and to be classified in this study were previously classified by experienced medical professionals, according to its location in the esophagus, stomach and small intestine. The images were further selected, due to the noise coming from external factors such as food residue or simply images where the region of interest and the main display had been blocked. Data augmentation operations were also performed in the data, as cropping and resizing the images were the main pre-processing tasks before being able to train the model. From the final dataset, the 3704 images were divided in classes such as lower body, antrum and duodenal bulb. These classes concerned an antegrade view. However, there were also classes with images from the retroflexed view of the endoscope like fundus and middle-upper body. There was one last class named as unqualified, corresponding to the ones where the models did not achieve any classification.

The CNN architectures chosen by the authors to perform the classification tasks were: RN50, Inception-v3, VGG11-bn, VGG16-bn and DenseNet-121. The authors did fine-tunning of the models and also performed transfer learning to build them. All of these models were pretrained on ImageNet dataset. To fine tune the models, the last fully connected layer of the CNNs was replaced by a fully connected layer block where the number of layers was equal to the number of existent classes. Afterwards, a mini-batch training was applied and the training loss minimised with multi-class cross entropy loss.

In order to select the best out of the chosen architectures, the 5 models were built. All the models were built with a batch normalisation layer, except for the Densenet-121. The performance metric used to analyse the models behaviour was AC. Two different problems were addressed: one where the class "unqualified" was one of the classes used and one where this class was not present. The CNN models without this class achieved better performances than the ones that include it. This is due to the fact that this class added ambiguity to the training of the models, given that it contains images that in part resemble other images. The VGG11-bn achieved the highest value of 0.94 for dataset 2 and the Densenet-121 was the best model in two datasets, with an AC score of 0.91 and 0.88, making it the best model.

3.1.6 Chathurika Gamage et al. 2019

A last paper noteworthy is the Chathurika Gamage et al. [30]. This paper objective was to classify anomalies and landmarks in upper endoscopy images. To do so, the authors built different CNN

models with resource to transfer learning on the Imagenet dataset. Afterwards, the authors selected the three models with the best performance and created an ensemble model, using only the feature extractors from the above mentioned models. The authors also approached a technique to reduce the processing time and memory consumption when constructing the models. Nonetheless, while doing so, the AC of the systems built was monitored in order to prevent it from decreasing. From the set of CNNs currently available, the authors used the architectures DenseNet, ResNet, VGG16, Xception, InceptionV3 and IRV2. The first three architectures were the ones used in the ensemble, given that from the 3 model ensemble testing, the ensemble featuring the first three architectures (with the variations DenseNet-201, ResNet-18 and VGG16) was the one to achieve better results.

The dataset the authors used for the defined classification tasks was the Kvasir dataset. This data was representative of 8 classes. It consists of 8000 images, which were afterwards split in a ration of 80:20, where 80% concerned training and the remaining data was used for test. These classes were divided in anatomical landmarks and pathological findings. The first set of classes concerned the classes z-lines, pylorus and cecum; the second regarding the lesions had classes such as esophagitis, polyps, and ulcerative colitis. Additionally, the dataset the authors used also had 2 classes concerning specifically polyps. Each class of the final dataset had 1000 images. The authors used 5-Fold Cross-Validation (5FCV) in the training set in order to tune the hyper-parameters. In the next stage, the test set was used to provide an estimation to the model selected from the previous stage.

As part of the preprocessing of the data, the images were down-sampled, due to their different resolution, into 224x224 pixels. Following this step, to the previously pre-trained CNN feature extractors a global average pooling layer was added and then the feature vectors were obtained. After this action, the resulting feature vectors were added to obtain a final feature vector. Later, the resulting feature vector was passed to an independent classifier to obtain the predicted class labels. This independent classifier corresponded to a single ANN hidden layer with 128 hidden units with ReLU. This ANN was trained using a mini-batch stochastic gradient descent algorithm, which size was 64 for training and test data sets. Additionally, the dropout value was defined to be 0.8 and a batch normalisation layer was used, so that the hidden input layers of the network could be normalised. For the classifier, a Softmax activation function and a categorical cross-entropy loss function were used and the learning rate was 0.001. The result for the AC performance metric of this ensemble was 0.97. However, performance metrics such as Recall, Precision and F1-Score were also calculated, having achieved the same result as AC.

3.2 State of the art outcomes

Regarding the researched literature in this Master's thesis, conclusions can be drawn about what is currently being used as DL techniques when it comes to investigating gastric cancer and images from the GIT. It is possible to identify common patterns among all the research. First of all, there is a tendency for the authors to choose CNNs as they are the best solution for object detection and image classification. Despite using the same artificial neural networks, the author's choice for architecture

varies. The ones whose main goal is to do object detection in their work tend to select the SSD architecture. The models in the articles covered achieved positive results. The architecture that achieved the best result in terms of AC was an ensemble of models, consisting of the architectures DenseNet-201, ResNet-18 and VGG16. A standard CNN with SSD and VGG16 architecture also achieved positive results. However, the results do not illustrate the complexity of the model or of the problem trying to be solved. In this paper, the authors built a successful detection and classification model but the classification tasks were just binary ones. Consequently, due to the simplicity of the tasks and the approach to the problem, even though the reasoning is solid and well-explained, the model itself is not strong and robust.

On the opposite, the model with the worst results seems to be a valid choice for a model solving classification tasks. This is due to the fact that the classification tasks the developed model was meant to solve were extensive and with multiple classes. They divided the classification into two subdivisions, the cancer versus non-cancer and the neoplasm versus non-neoplasm. Each of these subcategories had different classes to which the different observations could belong to. Despite not having a detection task, the classification tasks in this paper were considerably more difficult to perform than the analogous on the other articles. Furthermore, they constructed different models, more precisely three models: Inception-v4, Resnet-152, and IRV2. This represents one more strong motive regarding the question of why this study was valuable, even though it had the worst performance results. Also, the performance of the models was tested on an supplementary dataset as well as the doctors ability, which made possible for a comparison between the two sets of results. In the other articles the datasets in general were also marked by doctors before the construction of the model started. Even so, in no other article this stage of comparing the results with medical personal was so emphasised.

Another fact that is ultimately considered by almost every author is to pretrain the networks. Five out of the six articles discussed pretrained networks. The papers by Bum-Joo Cho et al. [20], by V. V. Khryashchev et al. [43], by Qi He et al. [36] and by Chathurika Gamage et al. [30] pretrained their models with the ImageNet dataset. Takumi Itoh et al. [40] presented a model where the object recognition task was done by a pretuned GoogLeNet CNN. Hirasawa et al. [37] in their work did not pretrain the network, however they had a major advantage in terms of training and testing data quantity with over 12000 images and 2000 images, respectively. It is important to point out that the model with the more complex and extensive task (Bum-Joo Cho et al. [20]) still obtained very good results and it pretrained the network. To this extent, pretraining the model in future works is a strong possibility, with chances of improving the performance results. Excluding pretraining and prospective validation datasets, all the authors worked with hospitals, institutes and clinics to create the dataset used to train and test the model.

It was also meaningful to analyse studies where the goal was to build models capable of classifying landmarks from images obtained in upper endoscopy exams. Papers such as Qi He et al. [36] and Chathurika Gamage et al. [30] built models using state-of-the-art architectures like Densenet and Inception and achieved positive results. One of the main advantages of the first paper was the wide variety of areas of the GIT depicted in its dataset. One other important aspect is the number of classes concerning the classification problem. This was a multi-class classification problem divided in

twelve different classes. Eleven of these classes were related to a specific part of the GIT, while the remaining one was more generic. Chathurika Gamage et al. [30] also performed anatomical landmark classification with the additional task of also classifying lesions. The results achieved were positive and the approach was one which had not yet been studied in the previous articles, an ensemble model. This ensemble model was made of three known architectures from the previous papers: DenseNet-201, ResNet-18 and VGG-16. Several different performance metrics were analysed, first to choose the CNNs that would integrate the ensemble and then to evaluate the ensemble itself. Hence, an ensemble can be a good approach when dealing with classification tasks such as this ones. Once again, this was a multi-class classification problem with multiple classes, 8 in particular, which, like Qi He et al. [36], is one of the main advantages of the study.

The papers shared some problems. A common flaw to the projects that intended to do object detection was that the authors almost never evaluated its performance. Beside, most of the papers (all but Hirasawa et al. [37]) had an insufficient amount of data as pointed by the authors. In their work it was mentioned that they did not try different sizes of datasets, saying that more training images could increase the performance and lead to a better diagnostic ability, but the association of the number of training images and the CNN AC was not measured. In general, the more complex the detection and classification tasks are, the greater the volume of data is required. There is a trade-off between the detail inherent to the detection and classification tasks and the performance. A brief description of the results for each paper can be seen in the Table 5.4 below.

Table 3.1: Table containing the summarised results of the papers covered. The task type concerns classification (C) tasks and detection tasks (D). The performance is measured in the metric inside parenthesis and it can be: AC, RC and Mean Averaged Precision.

Project	Dataset Size		Task type	Type of model built	Performance
	Training	Test			
Hirasawa	12584	2296	D,C	VGG16-SSD	0.92(A)
Bum-Joo Cho	4205	812	C	Inception-Resnet V2	0.82(A)
Khryashchev	1193	100	D	VGG16-SSD	0.88(mAP)
Itoh	596	30	C	GoogleLeNet	0.87(R)
Qi He	2960	730	C	DenseNet-121	0.91(A)
Chathurika Gamage	6400	1600	C	Ensemble	0.97(A)

Chapter 4

Methodology

In this chapter, we will cover the gathering of the data to build the dataset. Afterwards, the implementation of all the models, as well as the selected architectures of the models, accordingly to the state of the art insights will be described. Lastly, the performance metrics and criteria on which the results are analysed will be addressed.

4.1 Dataset Construction

4.1.1 Definition of the classes

Before the selection of the images and videos that would be part of the dataset, it was first necessary to establish the number of classes and define which ones the models should be able to predict in both problems (Micro and Macro classification). This decision was made based on the information seen in Chapter 1. According to what was seen regarding the stages of the development of a gastric cancer and the Correa's cascade, the classes chosen for the Macro class problem were:

- Healthy (HE) - class containing healthy structures, with no lesions and that can be fully reused for the Micro class problem.
- Precancerous (PRC) - class containing an intermediate stage of infection, where the images will later be divided into the classes Atrophic Gastritis (AG), Intestinal Metaplasia (IM) and Barrett's Esophagus (BE).
- Cancerous (CAN) - class containing the final stages of infection, where a cancer lesion is already present, where the images will later be divided into the classes Advanced Gastric Cancer (AGC) and Early Gastric Cancer (EGC).

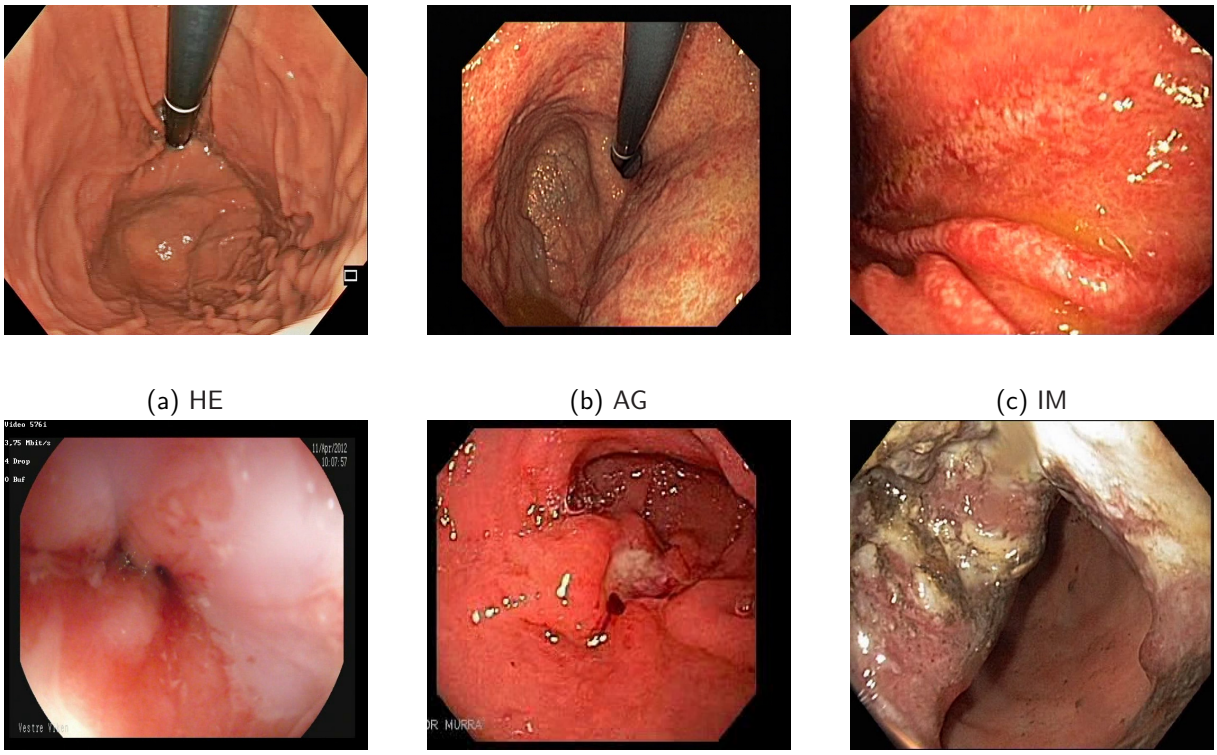
Afterwards, these classes were then separated (apart from HE) in more extensive sets of classes that would go into further detail clinically speaking. Initially, there was one more class designated by

"Other" and its purpose was to have images from other lesions of the gastrointestinal tract like polyps and ulcers. However, due to the complexity of this class and its diversity in terms of the image's features, this class was discarded. Nonetheless, the images for this class were still retrieved from the chosen datasets. Regarding the Micro Class problem the following classes were created.

- HE - exactly the same as the HE class from the Macro Class problem, with the exact same images. These images correspond to normal structures, without an evidence of any lesion of inflammation.
- AG - new class containing images with a specific lesion from PRC. This lesion is the first lesion to appear during the process of developing a gastric cancer. It can be described as a chronic inflammation of the gastric mucosa, which results in a transformation of this structure, due to atrophy of the cells.
- IM - new class containing images with a specific lesion from PRC. This lesion happens when the AG lesion persists and evolves. Broadly speaking, the advanced stage of degeneration and atrophy of the cells causes for them to change their nature and turn into intestinal-type cells.
- BE - new class containing images with a specific lesion from PRC. This lesion happens in the intersection between the esophagus and the stomach and it is characterised by the development of an IM in this area. Once such lesion takes place in this area, it is possible to say with certainty that there is a BE occurring.
- EGC - new class containing images with a specific lesion from CAN. By this time, the lesion is already in a cancer stage, completely established in the structures where it began. Degradation at cellular level starts to occur.
- AGC - containing images with a specific lesion from CAN. This is the final stage of the lesion, where there is advanced cellular degradation. The lesion extends to other structures like the submucosa.

In Figure 4.1 it is possible to see an example of each lesion in images belonging to the built dataset.

After the classes were defined, the number of images was chosen. Given the difficulty of these problems and the complexity inherent to build a model capable of predicting unseen data from up to six different classes, a fair amount of images for each class was considered to be 1000. This number was considered to be enough for the models to be capable of performing an effective training, followed by an adequate validation stage and, lastly, the test stage. This number of images was defined bearing in mind the classes from the Micro problem, as they are more specific and larger in number when compared to its analogous. Hence, the objective once the data retrieval was finished was to have around 6000 images in the dataset, 1000 per class.



(a) HE (b) AG (c) IM

(d) BE (e) EGC (f) AGC

Figure 4.1: Images containing each lesion present in the built dataset for the classification tasks.

4.1.2 Public datasets

From a list of possible datasets from which data could be collected, the first one analysed was the Hyperkvasir dataset. The Hyperkvasir dataset is the largest publicly released GIT image dataset [6]. The data belonging to this dataset comes from real gastrointestinal examinations (upper endoscopy), while they are being performed by experienced gastroenterologists. The images and videos in Hyperkvasir come from routine clinical examinations from a Norwegian hospital and were collected between 2008 and 2016. The devices used to collect the data in this dataset were standard endoscopy equipment built by Olympus and Pentax. The data are fully anonymous, given that all file names were renamed to randomly generated file names.

A part of these images is latter labelled by these doctors. The images for the dataset of this Master's thesis have categorical labels, indicating the class to which an image belongs, without further spatial information regarding the region where the lesion can be found within the image. The dataset contains 110079 images and 374 videos of anatomical landmarks and normal/pathological findings. These numbers represent approximately a million images and video frames existent in the dataset.

When considered from a technical point of view, this dataset is of upmost quality. Experiments were made by the creators of the dataset to prove its quality. The goal of these was to provide baseline metrics and to give insights on the statistical properties of the dataset. Amongst the different tasks for which the dataset was designated, classification tasks were also covered. Such tests were performed

using state-of-art methods to evaluate how such would perform on the labelled section of Hyperkvasir. Some examples of architectures used for this goal were the RN50 and the DenseNet161 [6]. The creators evaluated the performance of their models in this dataset using performance metrics like Precision (PR), RC and F1-Score (F1S). They achieved for these performance metrics values like 0.63 for PR [6].

From the Hyperkvasir dataset 1348 images and videos were extracted to be added to the dataset of this Master's thesis. These were all labelled images and videos and they belonged to the classes HE, BE and AGC.

The next dataset to be used was the Gastrolab dataset[1]. This dataset is similar to the previously described Hyperkvasir. It offers a range of images and videos in terms of pathological findings that the Hyperkvasir did not possess. From this dataset, the first images from AG and IM were collected. In total 311 images were collected from this dataset and divided among all the classes except for BE and AGC, with HE being the class with the most images collected from this dataset.

The last dataset from which images were retrieved in order to form the dataset for this study was the Gastrointestinal Atlas[2]. This is a central american dataset made by doctors and medical students from around the world. Their goal is to share their experience regarding endoscopy exams, through the upload of video clips. This data covers nearly all areas of gastrointestinal pathology detectable when performing an endoscopy.

When compared to the Hyperkvasir dataset, it can be said the video frames (images taken from the videos) coming from the Gastrointestinal Atlas do not show the same technical quality. Despite the high quality videos, the fact that all of images from this dataset come from videos is the exact reason why the frames are not of high quality. They are conditioned by a series of factors like if the camera from the endoscopy was focused and the light is pointing directly at the lesion in the moment of capturing a frame. These represent outdated scenarios in the images from the Hyperkvasir dataset, which have already gone through a selection and segmentation phase. Hence, the images from Gastrointestinal Atlas are of less quality, leading to less features and details per image.

From the Gastrointestinal Atlas dataset 5024 images were extracted. For the class HE only a few were retrieved, bearing in mind that with the images gathered from the two previous datasets, the objective of 1000 images was almost complete at this moment. However, 210 images and videos were extracted from this dataset for this class. For certain classes, like AG the maximum number of elements from this data bank was extracted, as up until this moment there were only 19 images in the dataset corresponding to this class. Consequently, for AG 684 images and videos were extracted from Gastrointestinal Atlas. This scenario was also seen for the classes IM and EGC. These were the classes that needed a large number of images in order to achieve in the end the desired number of elements per class in the dataset. The remaining classes already had the expected number of images coming from the other datasets, mainly from Hyperkvasir. A summary of the number of images retrieved from each dataset and the total number of images per class can be seen in the Table 4.1.

[1] https://www.sciencephoto.com/contributor/gas+h9b
[2] https://www.gastrointestinalatlas.com/english/english.html

Table 4.1: Table containing the numbers of images organised by class, showing the numbers that were extracted from each dataset and the total number of images that were collected in the end for the classes, as well as the total number of images in the final built dataset.

Class	Hyperkvasir	Gastrolab	Gastrointestinal Atlas	Total of images
	Images and videos	Images and videos	Videos	
HE	764	193	210	1167
AG	0	19	684	703
IM	0	13	512	525
BE	324	0	742	1066
EGC	0	0	900	900
AGC	260	20	750	1030
	1348	245	3798	5391

Ultimately, the Gastrointestinal Atlas dataset was the one with the strongest representation in the final dataset with a total of 3798 images, considerably more than the second dataset, Hperkvasir (1348). There were advantages and disadvantages when analysing the outcomes of the dataset. Some classes had a positive representation with over 1000 images (HE even achieved over 1100). On the other hand, there were underrepresented classes like AG but mainly IM, which had the lowest number of images in the whole dataset. Adding to this downside, almost 80% of the images from the dataset came from the videos in Gastrointestinal Atlas dataset. As mentioned earlier, this could have an effect on the results, given that it could reduce the quality of the images.

Frames were extracted from videos with a sampling rate of 1 frame per second. In general, an average of 30 frames were collected from each video. Nonetheless, this technique of extracting images from videos for the dataset has its drawbacks. If the movement of the camera of the endoscope is slow and mainly focused on a specific area, it will lead to a reduction in the diversity of the frames extracted. One other aspect influencing the quality of the frames is related to the focus of the camera. It can happen during the movement of the endoscope for the camera not to be focused on a specific lesion or structure, resulting in a blurry image. In datasets that consist only of images and not video frames, it could be the images are of low quality (for instance, the Gastrolab dataset). These are the main reasons why the images could have a downgrade in terms of quality, as mentioned earlier.

4.2 Implementation of the classification models

The architectures chosen for the models to be developed were the DN169 [38], IRV2 [69], NNL [73] and RN50 [35]. These architectures were selected following previously specified criteria. The first one is related to what was considered by authors of papers/studies in Chapter 3 as a good choice performance wise in classification tasks. Following this principle, the first models chosen were RN50, IRV2 and DN169. These correspond to architectures which had been tried and tested successfully in gastric lesion classification tasks, such as the ones in this Master's thesis. Several authors seen in the referred chapter used it as main architectures in their work with positive results. Although, in the case of DN169, it was not the exact same version of the architecture, but the DenseNet121

and DenseNet201 were studied. The last architectures was chosen due to its innovative character. This architecture was the NNL. From what was seen in Chapter 3, this architecture had not been chosen before to perform tasks directly related to the classification of GIT images. However, this is a relatively recent architecture, with its first appearance dating to the beginning of 2018.

To implement all of the architectures, the Python library, *Tensorflow*[3] was used, alongside with a software present in this library, the *Keras* [21] library. The images were loaded to the models following their division in the class directories from each subset. At this point, a rescale was applied to the input images. This operation is carried out in order to place the images at the same level, this way the models will look at the images and treat them equally. This method is often a standard measure in preprocessing images, in order to normalise the pixel values (set the pixel values between 0 and 1) and to be able to use a standardised learning rate.

Furthermore, the *batch size*, which is meant to determine the number of samples that are to be propagated throughout the network, was also defined. The number set for the *batch size* in this case was 32. This means that the models will measure how many train samples exist and, using the *batch size*, they will calculate how many samples will be used in each epoch. For this thesis, 30 epochs were used for each model. It is important to try to maintain a balance when setting the *batch size*. If it is lower than the number of samples then it will require less memory, as the network will be trained using less samples. In turn, this will make the networks train faster due to the weights of the network being updated after each propagation.

Beside *batch size*, there is another parameter noteworthy. This parameter corresponds to the input dimensions of the different networks considered in the thesis. It is a tuple of integers with the form *(height,width)* and the value set was (224,224). This is the size to which all images found in the different stages of the process will be resized.

Once the parameters of the models were established, the pretrained networks were loaded with the weights from the ImageNet dataset. This process is known as transfer learning. It is known by its ability to build accurate models, while optimising the training and testing time. Using this method, it is not necessary to start the learning process from the beginning, as it uses patterns which have already been learned when solving a different problem. It is useful as it allows to take advantage from other learning processes. In this case, transfer learning directly concerns the use of the different chosen architectures previously trained on a large dataset, the ImageNet dataset. This training was a process similar to what is carried out in the present thesis, i.e., these architectures were used to solve classification tasks in the ImageNet dataset. When creating the models no early stopping was used to stop the models before the 30 epochs were completed. Beside, one other important parameter when setting the models was their optimiser as well as the learning rate. The optimisers are algorithms used to change the weights and the learning rate attributes of the network. They allow to reduce the loss registered during the training stage which, in turn, will provide better results. The algorithm used for the built models was the Root Mean Squared Propagation (RMSprop). This algorithm main purpose is to tackle the highly decreasing learning rates, by using an adaptive learning rate. A good

[3] https://www.tensorflow.org/

rho value to use for this algorithm is 0.9 and the default learning rate, which is the one used for the models built, is 0.001. One other possible choice was the gradient descent optimiser, but contrary to it, the RMSprop restricts the oscillations vertically. Consequently, it allows to increase the learning rate and, in turn, a faster convergence. The main difference between these two optimisers relies on the way the gradients are calculated. A different choice could be the Adam optimiser, which is an algorithm for gradient-based optimisation of stochastic objective functions. It computes individual adaptive learning rates for several parameters, by using estimates from the first two moments of the gradient. This algorithm is similar to RMSprop, as it tries to implement some of its features, like the non-stationary settings. However, there a few core differences between the Adam algorithm and RMSprop. While Adam updates are estimated directly using a running average of first and second moment of the gradient, RMSprop updates its parameters by using a momentum on the rescaled gradient [44].

4.2.1 Considered deep neural network architectures

The chosen architectures have different attributes and strengths, along with advantages and disadvantages. All share the ability to perform in classification tasks, achieving positive results. One of the architectures chosen was the RN50. As mentioned in Chapter 2, this architecture derives from the residual learning framework, with the main goal of facilitating the training of deep networks. From the existent residual networks, three examples were considered: the RN50, Resnet101 and Resnet152. These examples only differ in terms of depth of the network, as the Resnet101 has more layers than RN50 and Resnet152 has more layers than Resnet101. However, despite the fact that the Resnet101 and 152 are significantly more deep, the 152 layered one still has lower complexity than the VGG16 (also seen in Chapter 2) for instance. From these examples the Resnet50 (RN50) was chosen.

The following architecture selected was the IRV2. This architecture, along with the RN50, was widely used in the papers researched in Chapter 3. The authors from these articles obtained positive results in classification tasks when using these architectures. Hence, they were also used in this Master's thesis for a similar classification task with images of the same nature, but with an increase of the number of classes. Along with the IRV2, more architectures following the Inception framework were considered. The Inception-V4 seen in Christian Szegedy et al. [69] was one of the studied networks to understand if this was an option for the present classification tasks. However, the Inception-V4 is very similar to the IRV2, as the raw cost of both networks is the same. Furthermore, the step time of the Inception-V4 is considerably lower than the IRV2, due to the increased number of layers of the first network compared to the second one. With the introduction of the residual connections, as described in Christian Szegedy et al. [69], the training of the networks became faster. At the same time, their performance was better than the networks without residual connections. Hence, the architecture chosen for the present classification tasks was the IRV2.

Another considered architecture was the DN169 a deeper, more accurate and efficient in the training stage network. This architecture first appeared in 2016 and it introduced the concept of direct connections between any two layers with the same feature map size. As seen in Huang et al. [38], it

has reduced optimisation difficulties and scales without any issue to hundreds of layers. Furthermore, this architecture was tested in different and challenging classification tasks, throughout four different datasets (CIFAR-10, CIFAR-100, SVHN, and ImageNet). The densely connected networks also allow for feature reuse throughout the network, leading to learning more compact and accurate models. In this case, such fact becomes an even greater advantage, due to the relatively small dimensions of the dataset and, consequently, smaller amount of features obtained from the images. One last advantage achieved through the use of this architecture is related to the the fact that, despite improving in terms of accuracy with growing number of parameters, it requires less parameters and less computation (the same as less training) to achieve state-of-the-art performances.

The last architecture to be chosen was the NNL. This architecture was first described in June 2018, thus being a recent architecture. It was also tested in CIFAR-10 and Imagenet, similar to the DN169. The key point of this architecture is its transferability, due to the search space created when building the model [73]. Despite having the downside of being less effective in smaller datasets, such as the one in this Master's thesis, the NNL still achieves state-of-the-art results. This architecture is flexible in a sense that it can be scaled in terms of computational costs and parameters to adapt to different types of problems. Additionally, with the search space feature of this architecture it is possible to obtain a model capable of performing in small datasets.

4.3 Gradient-weighted Class Activation Mappings (Grad CAMs)

Another tool implemented and considered to be helpful when analysing the models was the Grad CAMs [60]. This tool increases interpretability of the models and helps to decide which one's classifications are closer to the real scenario. The Grad CAMs are images which allow to visualise the class activation maps of the networks, making it possible to understand from which part of the image the model considered a certain class to be present. Through the Grad CAMs in an image of the test set, it is possible to notice which features are activated by the model when analysing it.

For this section, one image from each class was selected. The images chosen for each class were images where the selected model performed a correct classification regarding the lesion present in the image. Beside, it should be possible to observe clearly that the features that led to the classification of the model were the ones that corresponded to the lesion (or absence of it, in the case of HE). Only images from the Micro problem classes were selected, given that the lesions seen in these images would be more specific in terms of classification and, therefore, would be more detailed.

Similarly to the model architectures, the Grad CAMs implementation was done using open-source code from *Keras* [21]. To implement a Grad CAM the first step was to load one of the previously built models, more specifically, the DN169 model from the second iteration. The final model was considered after the training performed in the dataset built and new predictions were made. The Grad CAMs can be applied to every type of CNN architectures and deeper CNN models are more likely to capture a high-level construct. One other important aspect of the CNNs is that, since they naturally retrain spatial information lost in the fully connected layers, the last convolutional layer has

the best trade-off between high-level semantics and detailed spatial information.

The images that would serve as input to the Grad CAM needed to be resized and reshaped. This was done in order to turn them into the same size as the images received by the models when in the 5FCV process and to normalise the pixels values, respectively. Grad CAMs use the gradient information that enters the last convolutional layer of the CNN to understand each neuron and its importance, resulting in a decision of interest regarding the image.

To obtain a localisation map of height v and width u with the locations of a class c discriminated it is first necessary to compute the gradient of the prediction score for the class c. This gradient Y_c is calculated before the softmax activation function and concerning the feature maps A^k of a convolutional layer (i.e. $\frac{\partial y^c}{\partial A^k}$). These backward gradients afterwards go through a global average pooling operation to obtain the neuron importance weights, α_k^c. It represents a partial linearisation of the deep network downstream from A, and obtains the importance of feature map k for a target class c. These weights are obtained using the equation (4.1), which represents the global average pooling of the score gradient with respect to the elements of the map A^k, where Z is the total dimension of the map [60].

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{4.1}$$

After calculating the weights for this class, a weighted combination of the forward activation maps is carried out, immediately followed by a ReLU activation function.

$$L_{Grad-CAM}^c = ReLU(\sum_k \alpha_k^c A^k) \tag{4.2}$$

Where the summation represents the linear combination and the ReLU is applied to the linear combination of feature maps, given that the features that have a positive influence on the class of interest are the only ones that matter. The features that have a positive influence in this class are the ones whose intensity should be increased in order to increase the gradient of the score for the class, Y_c.

The next stage was the construction of a heatmap that would later be merged with the original image. The heatmap built is of the same size as the convolutional feature maps. The ReLU activation function is applied to the linear combination, given that only the features that have a positive influence on the class matter. If the ReLU was not applied, the class activation map would be highlighting more than just the desired class c, which would lead to a low localisation performance. The last step was the projection of the heatmap in the input image, which produced the final Grad CAM. A colour map was selected to better understand the colour scheme portrayed in the Grad CAM [51, 60].

4.4 5FCV Strategy

To test the models with the available data, the chosen strategy was using a 5FCV Algorithm. K-fold cross-validation represents a technique used to evaluate the predictive capability of models in classification tasks. It is widely used across the machine learning areas, as it is straightforward and easy to understand as well as easy to implement. The primary idea is to divide the data for a certain problem in three sets: the training, validation and test sets. For this thesis, the dataset is divided into five equal parts. The proportions for each set were defined to be three parts for the training set and one for the validation and test sets.

There were restrictions upon the moment of dividing the data into the subsets. The first one was that it was mandatory to have all the video frames coming from the same video in the same subset. This is the same as saying data from one patient (correspondent to one video) should be in the same subset. Otherwise, when performing predictions for a lesion (a patient) they had already seen in the training stage, the models would have all the necessary features to perform a correct prediction. This would not be the result of a learning process, but memorising the features of that specific lesion. Hence, the solution for this problem was to try to divide the frames into the subsets, keeping in mind their source specifically in the case of videos. This was carried out while trying to maintain the number of images per class in each subset balanced.

To accomplish these requirements, an algorithm was developed. This method started by dividing by 5 the total number of images per class in the existing classes of the dataset. This should be the number of images per class in each of the five subsets, as it represents the most balanced and fair way of dividing the data. Due to the issue involving the video frames, it was not possible to achieve such numbers. Consequently, the result of the division of the number of images per class by 5 was used as the maximum number of images for each class subset. Following this step, the videos were analysed by their number of frames in descending order and bearing in mind the maximum number of images each subset could have. If the number of video frames from a video when being analysed was smaller than the number of slots available in that subset, then the video could be placed in that subset. Otherwise, it would be put on hold and analysed again for the following subset. At some point, there were too few slots available, so that none of the remaining videos could fit, as they would have too many frames. In this case, the remaining position in that class subset would be filled with images. These represent images directly extracted from the datasets and not from videos. Below it is possible to see the algorithm represented by a flowchart in Figure 4.2.

This process goes on until the last video frames and images are placed in the remaining subset. In the course of this process, the number of images that got placed in each subset was carefully watched, so that it did not become unbalanced comparing to the other subsets. Considering the descending order, in case there was a video with a greater number of frames than the maximum number of frames in that class subsets, then those video frames would be the only images in the last corresponding class subset of the 5FCV subsets. The number of images per class per subset can be seen in the Table 4.2. In the Table 4.3 it is possible to observe the number of images and the number of video frames in each class when building the dataset. Each image and each full video represent 1

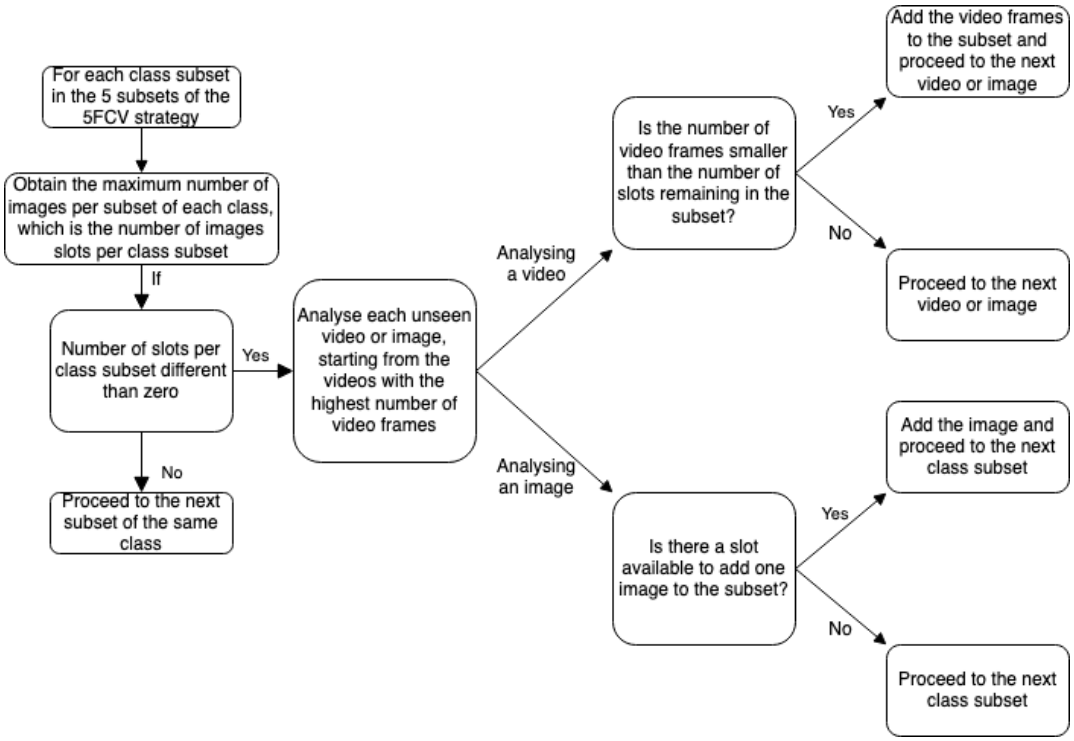


Figure 4.2: Algorithm to divide the video-frames and images per subset.

patient.

Table 4.2: Table containing the numbers of images per class subset in each of the subsets of the 5FCV process.

Class / Subsets	HE	AG	IM	EGC	AGC	BE	Total
1	234	136	105	180	206	214	1075
2	234	151	90	180	194	213	1062
3	233	120	90	180	210	213	1046
4	233	150	120	120	210	213	1046
5	233	146	120	240	210	213	1162

Table 4.3: Table containing the number of images coming from videos (video frames) and images in each class subset in the 5FCV process. Inside parenthesis in the 'video' column there are the number of patients/videos from where the frames came from.

Subset	HE		AG		IM		EGC		AGC		BE	
	I	V	I	V	I	V	I	V	I	V	I	V
1	24	210(7)	19	117(3)	15	90(1)	-	180(2)	6	200(2)	14	200(2)
2	114	120(2)	1	150(2)	-	90(2)	-	180(2)	14	180(2)	3	210(3)
3	233	-	-	120(4)	-	90(2)	-	180(3)	-	210(4)	3	210(5)
4	233	-	-	150(5)	-	120(3)	-	120(3)	-	210(5)	3	210(7)
5	233	-	-	146(5)	-	120(2)	-	240(1)	1	209(7)	71	142(5)

After the subsets were populated using the above described algorithm, the pretrained networks were loaded with the weights from the Imagenet dataset, as mentioned in Section 4.2. The instance

of each model was loaded with an input shape previously defined, in order to set an equal size for all the images that would serve as input for the models. While loading the pretrained networks, their fully connected layers (the classifier part) were discarded.

Afterwards, new layers were added to the models to build the classifier itself. The same layers were added to all considered models. The number of layers added to the models was chosen in order to stop the models from becoming too deep.

This classifier was made almost entirely of fully connected layers. It started with the layer that includes the pretrained model. In it the classifier was *frozen* by setting it as a non-trainable layer. Recalling the transfer learning concept, the CNN models for classification tasks can be divided in two parts: the feature extractor and the classifier. The feature extractor is meant to obtain the features from the images that are given as input to the models, as they will be used to train the models. The classifier's goal is to later take the output of the feature extractor and classify the images in the test set with one of the classes. This first layer objective is to discard the classifier the pretrained architecture had and use only the feature extractor, already tuned during the pretraining. It is expected that the new classifier can perform classifications using the new classes, from the dataset built for the 2 problems at hand. In the end, there must be only one classifier and it should be trained solely in the built dataset. Therefore, the first layer removes the fully connected layers coming from the classifier in the pretrained architecture (new ones will be later added) and keeps the pretrained architecture convolutional layers from the feature extractor.

The following layers were the *Flatten* and *Batch Normalisation* layers. Regarding the *Flatten* layer, its purpose is to reorganise the images, by rearranging the shape of the images. To do so, it adds an extra channel dimension and changes the output shape. The *Batch Normalisation* layer is the layer responsible for the normalisation of the inputs. It applies a transformation which maintains the average output close to 0 and the standard deviation close to 1. This layer works differently in the training and inference (test) stages. During the training, the layer normalises its output using the average and standard deviation from each input variable in the the input batch. During the test stage, the *Batch Normalisation* layer uses the already calculated mean and standard deviation of every batch seen during the training. The model performs an average of these two values of each input batch. Consequently, this layer will normalise its inputs during the test stage after having been trained on similar data in the training stage [12].

The next layers to be added were the fully connected *Dense* layers. These layers were interleaved with dropout layers with ReLU activation, except for the final pair of layers. The final two layers were one more *Batch Normalisation* layer and a fully connected *Dense* layer with a *softmax* activation. This activation function takes the output N values (being N the number of classes in the classification task) and it normalises the outputs. As outputs of the models, the values are weighted sums, while after applying *softmax* they are converted into probabilities that sum to one. An array is formed with the output of this activation function, where each value is interpreted as the probability of membership for each class.

The models were later compiled, saved and the predictions for the test set carried out. Below in

Figure 4.3 it is possible to see a schematic representation of the layer sequence of the classifier built.

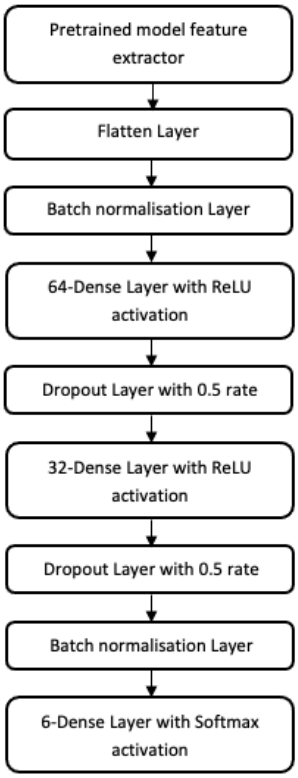


Figure 4.3: Layers that make the classifier for every model built.

4.5 Performance Metrics

The performance metrics were selected according to what was covered in Chapter 3. The metrics used by investigators from the papers and studies seen in this chapter were analysed and chosen, keeping in mind the subject and problems that would be seen in this Master's thesis. The performance metrics in multi-class classification problems can be very useful in determining which of the classes were correctly predicted and how the models behaved in the different stages of the 5FCV process. Hence, the performance metrics chosen were: AC, F1S, PR, RC and the Confusion Matrix. The datasets for the problems seen in this work are balanced, so it was not necessary to take into consideration any weights from classes when calculating the performance metrics.

Confusion Matrix

A Confusion Matrix is a cross table that shows the number of occurrences between two variables: the real classification and the predicted classification. In this study, the rows display the real classification and the columns correspond to the predicted classification. The classes are organised in the same order both in the rows and in the columns. Thus, the correctly classified elements are

located in the main diagonal of the matrix. Ideally, the confusion matrix should consist of only zeros, except for the main diagonal where there should be the number of elements of each class [32]. From the confusion matrix it is possible to extract all the values regarding each class, inserted in the following categories:

- True Positive (TP) - elements considered to be positive and are in fact positive, in other words, classified as coming from the class to which they truly belong to.
- True Negative (TN) - elements the model predicted as not coming from a certain class, when they actually do not belong to that specific class.
- False Positive (FP) - elements predicted as coming for a certain class but that are actually negative, which means they are classified as coming from this class, when in reality they do not come from it.
- False Negative (FN) - elements of the class being analysed at the moment predicted as coming from a different one.

An example of a confusion matrix created during the 5FCV process can be seen on the Figure 4.4.

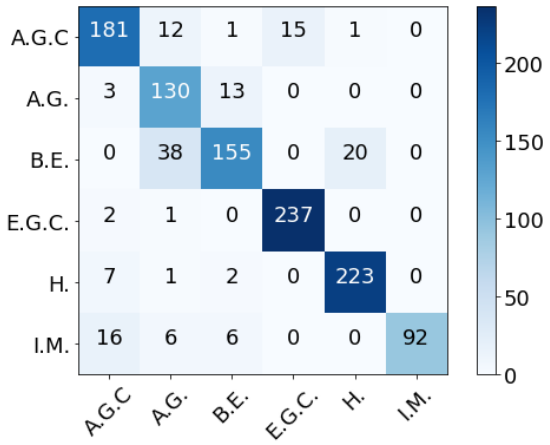


Figure 4.4: Example of a confusion matrix for the classes considered in the Micro class problem. As specified, the rows correspond to the correct values, while the columns correspond to the predicted values.

Accuracy (AC)

The AC performance metric is based on the number of TP and TN found in the confusion matrix. It intends to measure how many predictions from the model were correct throughout all the classes. In other words, when one selects a random sample from the data and the model performs a prediction for it, the AC provides the probability that the model is correct. It is calculated using all the values from the confusion matrix: TP, TN, FP and FN. It is calculated using the following formula:

- *Accuracy*: $\mathrm{AC} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FN}+\mathrm{FP}}$

In the numerator are the units correctly classified by the models, which, practically speaking, correspond to the values in the main diagonal. On the other hand, in the denominator there are also the elements outside the main diagonal, that have been incorrectly classified by the models. AC provides an overall measure of how the model is performing correct predictions in the dataset. It is a metric suited for analysing individuals, instead of an entire class for instance [32].

Precision (PR)

The PR performance metric evaluates the performance of a model through the predictions performed for an whole class, by keeping in mind the correct and wrong predictions. In other words, this is a metric that measures the proportion of images the model classifies as from a certain class and are, in fact, from such class (how reliable the model is when performing a prediction for a given class) [32]. The formula for the PR performance metric is as follows:

- *Precision*: $\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$

Recall (RC)

The RC performance metric corresponds to the metric evaluating the model's performance in the context of the real class. This means it evaluates all the predictions that were made for a specific class, both right (TP) and wrong (FN), and calculates how accurate the model was in this case [32]. The RC formula can be seen below:

- *Recall*: $\mathrm{RC} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$

F1-Score (F1S)

The F1S performance metric is based on the RC and PR metrics. It can be seen as a weighted average from these two metrics, given the way of how it is calculated. It ranges from 0 (worst value) to 1 (best value). The F1S can be compared to an harmonic mean, as both PR and RC have equal contributions to calculate this metric. Beside, the harmonic mean is a way of finding the trade-off between the other two metrics. The formula for the F1S involving the PR and RC values is:

- *F1-Score*: $\mathrm{F1S} = 2 \cdot \frac{\mathrm{PR}\cdot\mathrm{RC}}{\mathrm{PR}+\mathrm{RC}}$

Previous studies shown that it gives larger weights to smaller classes, while rewarding models that have similar PR and RC values. Furthermore, PR and RC take values in the range [0,1], if one of them is close to 0, the F1S drops.

To perform the PR, RC and F1S in a multi-class classification problem such as the ones covered in this thesis, it is necessary to calculate the Macro version of these performance metrics. This method calculates the performance metrics of the model by doing the average of all PR, RC and F1S values previously calculated for each class. For each class the above mentioned formulas are applied, obtaining an array of 6 values for each performance metric (3 arrays). The final values are then achieved by doing the average over each of the three sets of values. In turn, this leads to all the classes having the same weights for the average calculations, in order to avoid the distinction between highly and poorly populated classes [32].

Chapter 5

Results and Discussion

Among the different known methods to evaluate the performance, graphs presenting the evolution of validation accuracy and loss throughout the epochs, performance metrics such as AC, PR, RC and F1S were used, as discussed in Chapter 4. All these tools were implemented with the Python library *Scikit-Learn* and *Keras*.

The results are separated in two different stages: results for the Macro Class models and, afterwards, results for the Micro Class models. Following the same premise of Chapter 4, where the line of thought was identical for both problems, the process applied to the Micro Classes problem was adapted to make it possible to do the same for the Macro Classes problem. Despite the fact that all the chosen metrics were used for both problems, the results section divides itself and the Macro Classes results are presented alongside with its conclusions followed by the Micro Classes results and its conclusions. Summing up, in this chapter, the analysis carried out over the two distinct problems will first look into the problems expected results in general, followed by each problem with the corresponding 5 iterations of the 5FCV and a few observations and insights on why the results were satisfying or with a margin for improvement. Lastly, for each problem the number of patients in each test subset of the 5 iterations will be analysed to try to establish a relation with the final results.

A comparison between the results of the two approaches and the way the models behave when dealing with the two different problems was also carried out. At an initial stage, conclusions can be drawn, taking into consideration that, despite having different target objectives - which are 3 class classification and 6 class classification for Macro and Micro, respectively - the goal is still in both cases to perform a multi-class classification problem. However, the difficulty level in the Micro Class problem is considerably higher than in the Macro Class problem. This is due to the fact that in the first one, the models attempt to predict a correct classification out of six possible choices, while the second one does the same task but with three possible outcomes for the images.

The models for the 6 class problem require more training with more elements, which does not happen since the size of the dataset is not large enough to guarantee perfect training conditions for the models. In the end, for more classes forming the Micro classes problem, there is less data available. On the opposite way, the number of subjects for two of the classes (Precancerous and

Cancerous) in the Macro Classes problem grows in quantity. As explained in Chapter 4, the images from AG, IM and BE classes were put together to form the PRC class and the images from EGC and AGC classes were tied to form the CAN class. As an outcome, these classes contained 2590 images for the PRC one and 1929 for the CAN one. The number of images for the class HE was the same for both problems, but given the fact that out of 1167 images, most of them - 764 images - were taken from the Hyperkvasir dataset. In turn, it led to good scores in terms of predictions. Accordingly to the stated, the expected results were for the models regarding the Macro Class problem to outperform in every aspect the models in the Micro Class problem. This would be better seen in better margins in AC, RC, F1S and PR performance metrics values.

Furthermore, beside the fact that the Macro results should be better than the Micro, it would also be plausible for the same trends to happen on both problems. For instance, if the models in the Macro class problem consider the images from one class like HE as a different type of lesion belonging to other class, it should be PRC. This is related to the fact that PRC has images from AG, the class that could share the most features with HE, as it is the first stage of infection.

5.1 Macro Class Problems Results

The results of the Macro Class Problem were developed following a logical order. Firstly, each iteration of the 5FCV was investigated and subsequently the results were merged into a final set of results. For each iteration, we present a confusion matrix for each model considered and the values of the chosen performance metrics - AC, RC, F1S and PR - one set of confusion matrix and 5 performance metrics for each model built. In this section, the broad perspective of the models and its metrics/confusion matrices will be analysed, followed by a more detailed analysis on each iteration of the 5FCV along with conclusions over why the results obtained were such.

After all the models were trained, an attempt to predict the correct classifications for the data of each validation set was carried out. This is the same as saying that for the images in the validation subset of the correspondent iteration, the model was tested to evaluate its behaviour and performance when predicting the classifications. As a result, a summary confusion matrix was also built for these data and for each model. In general, the patterns shown throughout the 5 iterations and, consequently, in the final matrices were also noticed.

5.1.1 Global Results

The results for this section of the problem were expected to be better than those obtained for the case of Micro Classes, given the lower difficulty of the prediction tasks and the size (number of images) of each class when compared to the Micro class problem. The challenge was to create models capable of distinguishing images that appeared to have HE, PRC or CAN lesions. For the classes PRC and CAN the number of images was bigger than for the Micro class problem. This was due to the merge of the classes AG, BE and IM for the PRC class and EGC and AGC for CAN, as

described in Chapter 4. Despite not having a larger number of images for this problem, most of the images from HE come from the Hyperkvasir dataset. Hence, considering the class stays the same for both problems, the patterns and results observed for this problem will also happen in the Micro class problem.

A summary of the built models was created after all the models were trained and tested. Each metric was evaluated and for each model an average of the 5 values obtained throughout all the iterations was calculated. In the end, a table for each model was created with all average values for the chosen performance metrics (AC, RC, F1S and PR). Those tables were latter merged into one. The class HE, for this problem, has the lowest number of images. However, the difference is not significant taking into account the current problem (around 1200 images for HE and 2000 and 2500 for CAN and PRC, respectively). Given this fact, the performance metrics can be carried out considering the dataset was balanced and classes do not need to be weighted.

The averaged results of the 4 built models can be seen in the Table 5.1. The models achieved satisfying results, with a not so considerable difference between the best and the second best models. The model DN169 had the best results along with IRV2, having had the same results in some of the performance metrics. The PR performance metric for these two models achieved the highest value (0.81). When comparing these two models, it can be said they are nearly identical in terms of performance metrics results. The difference between the two relies specifically on the number of FN and TP for the classes individually. IRV2 had better RC values for PRC than DN169 but this difference is faint. They had the same number of FN for this class, while the DN169 had a higher number of TP for PRC. Consequently, IRV2 had a higher number of CAN FN. Nonetheless, these two models had an overall similar performance with all PR and RC values for all classes standing between 0.75 and 0.81.

Despite not being as close as seen in the Table 5.1, the validation results for these two models were still similar with a tight margin. The highest number of FN in each class came from the same classes as it did in the test set. The difference between the two scenarios was the overall number of FP and FN, which was higher than in the test set. Due to this fact, the main drawback in this stage were the numbers surrounding the HE, which got more confused both with PRC and CAN. The previous patterns relating PRC and CAN not only were seen as they were intensified.

The remaining models, RN50 and NNL, achieved results similar to the previous ones. These models showed the lowest RC values for the CAN class. While the previously discussed models misjudged CAN lesions predicting them as PRC, RN50 and NNL also performed predictions thinking these lesions belonged to HE. Therefore, generally speaking these two latter models did not carry out a training stage as effective as their peers, leading into difficulties in the test stage and causing misunderstandings in the classification of CAN. One class in which RN50 stood out was HE, as it obtained the highest RC values (but still very close to the rest). This model was the one out of the 4 models that made less previsions as if the images had CAN lesions, when they in fact were HE structures. This is a remarkable achievement considering that for images with so many differences in theory there should be a small degree of confusion (or even no confusion at all) with these 2 classes.

As for the validation overall results for the latter two models, these were once again slightly worse than the ones regarding the test set. Still, CAN was the exception as in general the models performed better in the validation stage than in the test stage. In the end, they achieved a higher number of TP for this class. The pattern related to the HE class seen for DN169 and IRV2 still occurred. Lastly, the gap between the two models discussed and DN169/IRV2 increased in the validation stage. At the same time, some trends were kept, like the RN50 ability to predict HE images better than the remaining models.

These conclusions regarding the individual performance metrics values for each class were taken from the analysis of the summary confusion matrices of the 4 models, reported in Figure 5.1.

The results for all the models in general were as expected with less difference between models, as it will be seen in the next section. The models had close results, and in some cases they were even the same. This scenario was the expected one considering the problem's difficulty was lower due to the decrease on the number of classes. This head the models into a better training phase and, thus, better performance. In addition, 2 out of the 3 classes had more data available for all the stages of the process.

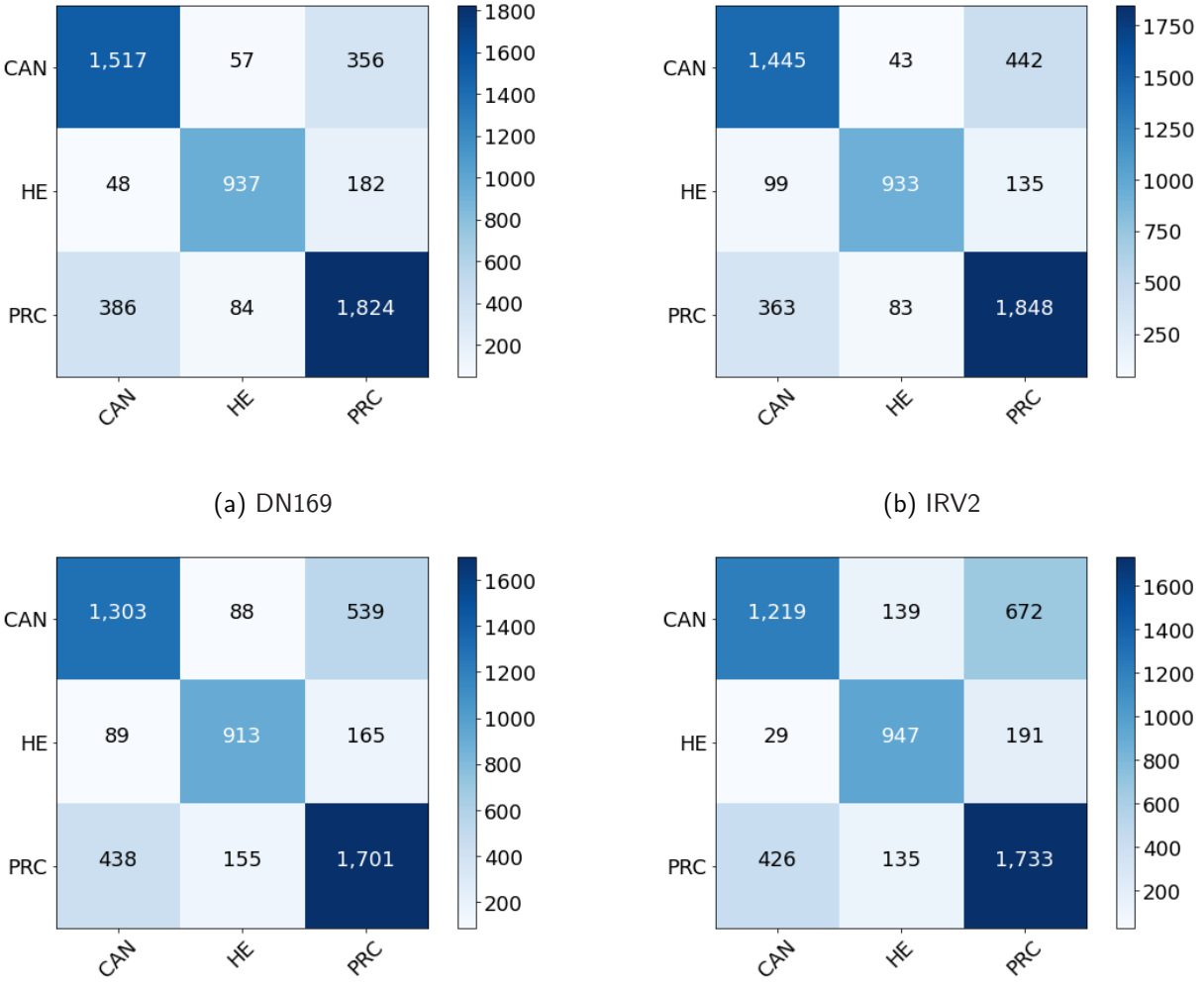
Table 5.1: Results on mean ± standard deviation for all model's performance metrics in the Macro class problem.

Model	AC	RC	F1S	PR
DenseNet169	**0.79 ± 0.03**	**0.79 ± 0.03**	**0.79 ± 0.03**	**0.81 ± 0.03**
InceptionResnetV2	0.78 ± 0.03	0.78 ± 0.03	0.78 ± 0.03	0.81 ± 0.03
NasNet Large	0.72 ± 0.03	0.73 ± 0.03	0.73 ± 0.03	0.75 ± 0.03
Resnet50	0.72 ± 0.03	0.73 ± 0.03	0.73 ± 0.03	0.75 ± 0.03

5.1.2 Subset analysis in Macro classes

In each iteration of the 5FCV as a result of a different test subset, different images were evaluated, leading to several possible outcomes and conclusions. By analysing each iteration through a detailed process, it is possible to gather certain information about the data, the number of patients or the quality of the dataset from which the images come from. In this section, possible causes and correlations between the data and the results were also investigated. Conclusions were drawn regarding confusion between two or more classes in the same model across the 5FCV process iterations. The decision to claim that there was confusion between two classes was based on whether the classes showed issues in more than one model and the models that showed confusion had the best results performance wise. It would not be plausible to decide two classes get mixed with each other if such confusion only happens in the model with worst performance results.

The number of patients in each subset was one of the variables analysed, as reported in the Table 5.2. This analysis of the number of patients both in the test and validation subsets was carried out considering all the subsets from the 5FCV process were used once as test/validation subset. This analysis started with the assumption that each video and each single image on the



(a) DN169

(b) IRV2

(c) NNL

(d) RN50

Figure 5.1: Summary confusion matrices for all models in Macro Class problem.

dataset corresponded to a single patient. One disadvantage from this reasoning was not being able to determine if different videos indeed belonged to the same patient. Some of the data banks from where images were retrieved, like Hyperkvasir and Gastrolab, had no patient distinction, making it impossible to judge if a certain image was related to the same individual. The Gastrointestinal Atlas dataset, despite having a light description of the patient and clinical case, also had different videos for different stages of the infection in the same patient. Practically speaking, these images can be seen as from a different patient in light of the studied problems in this master's thesis, since they will probably have a different classification from the model.

It is well known in classification problems that the diversity of the dataset is of massive importance for the quality of the data and, therefore, for the performance of the models. The premise is that the more diversity the images have, the better representation a DL algorithm can learn from the data. With this intuition, the expected trends regarding this analysis were for the models to have had success when the number of patients in the subsets was higher and to have poor results when a

insufficient number of patients was registered. The Table 5.2 shows the number of patients per test subset in each iteration.

Table 5.2: Table containing the number of patients per subset for each class for the Macro class problem.

Patient Class / Subset	Healthy	Precancerous	Cancerous
1	31	54	10
2	118	11	18
3	233	14	7
4	233	18	8
5	233	83	9

All the values mentioned for the average results of the performance metrics for all the models can be seen in Table 5.3. For the first and second iteration, all the models built had very positive performances in the classification tasks. In both of these iterations there was a tendency to wrongly classify a small number of CAN as PRC images as seen in the tables below.

Table 5.3: Performance metrics results for every model in each iteration of the Macro class problem.

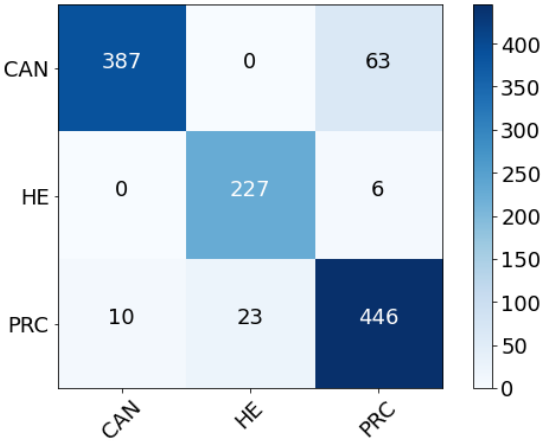
DN169				
Iterations	AC	RC	F1S	PR
1	0.91	0.92	0.92	0.92
2	0.96	0.97	0.97	0.97
3	0.53	0.55	0.55	0.55
4	0.77	0.71	0.72	0.83
5	0.79	0.81	0.80	0.80

IRV2				
Iterations	AC	RC	F1S	PR
1	0.88	0.90	0.89	0.89
2	0.87	0.88	0.88	0.88
3	0.66	0.65	0.66	0.67
4	0.69	0.65	0.67	0.76
5	0.82	0.83	0.82	0.85

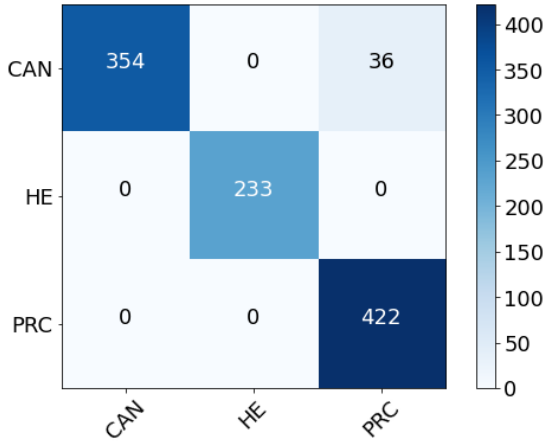
NNL				
Iterations	AC	RC	F1S	PR
1	0.85	0.87	0.87	0.87
2	0.83	0.86	0.85	0.84
3	0.53	0.54	0.54	0.55
4	0.70	0.65	0.67	0.77
5	0.70	0.73	0.71	0.71

RN50				
Iterations	AC	RC	F1S	PR
1	0.82	0.84	0.84	0.84
2	0.84	0.86	0.86	0.87
3	0.51	0.55	0.52	0.52
4	0.69	0.65	0.67	0.76
5	0.74	0.76	0.75	0.74

The confusion matrices in Figure 5.2 show the results for the best model DN169 (despite the small margin when compared to the other models). This model achieved AC values of 0.91 and 0.96 in the first and second iterations, respectively. The confusion matrices in the Figure 5.3 shows the performance metrics values for the RN50 that obtained the lower classifications, even though these do not represent poor results in terms of performance (0.82 for the first and 0.84 for the second iteration for AC values).

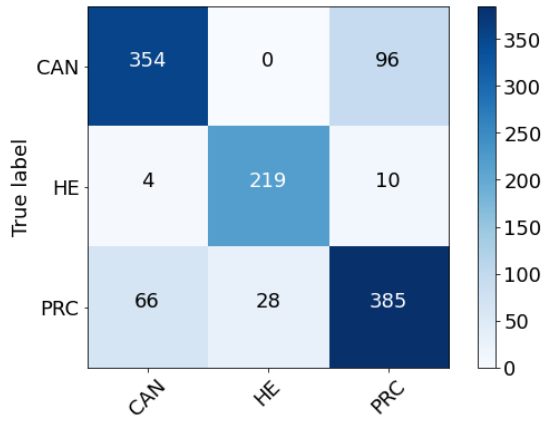


(a) First confusion matrix resultant from the first DN169 model predictions

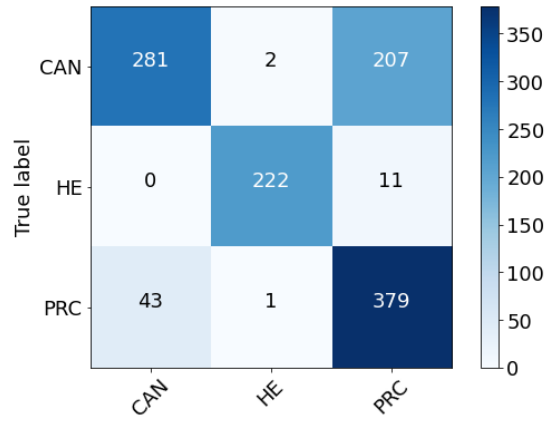


(b) Second confusion matrix resultant from the first DN169 model predictions

Figure 5.2: Confusion matrices from the first and second iterations from the DN169 models.



(a) First confusion matrix resultant from the first RN50 model predictions



(b) Second confusion matrix resultant from the first RN50 model predictions

Figure 5.3: Confusion matrices from the first and second iterations from the RN50 models.

These confusion matrices show that the classes PRC and CAN revealed themselves as harder to predict for the RN50 model than for the DN169. This was confirmed by the RC values which were 0.86 and 0.93 for CAN and PRC respectively in the DN169 model, as opposed to the RN50 where it achieved values of 0.79 and 0.80. On the contrary, both of these models unveil themselves as the ones with lowest PR values for HE out of the 4 models in the first iteration. The total number of FP was 23 and 28 for DN169 and RN50, respectively. All of these FP come from the class PRC.

Another observation common to both iterations between two models is the similar RC values for the models RN50 and NNL for the class CAN. In both iterations, these models obtained the lowest RC values for this class when compared to their peers. The number of FP was higher for this class in these two models and it was due to mostly wrong CAN predictions. This might be happening due to lesions that are already in an advanced state but can not yet be considered cancer lesions. An

example could be an IM lesion that has still not progressed into an EGC, but its appearance can already reveal certain signs an EGC lesion would present.

Beside what has already been stated, the models IRV2 and NNL are also very similar in the first iteration. Like what was seen for RN50 and DN169, the FP numbers in HE are close (15 for NNL and 14 for IRV2) and the number of TP for the class PRC was equal (393). In the second iteration, once again these models showed an approximate number of FP for CAN.

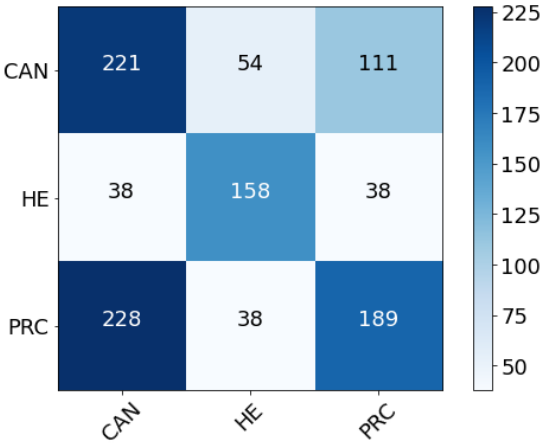
Generally speaking, the models achieved very reliable results for the first two iterations. DN169 clearly stood out and the remaining three models were nearly identical to each other. Their performance metrics values were consistently close and they also shared the same flaws and issues. For IRV2 those errors were not so visible for HE and PRC, while RN50 was the best PR wise for CAN.

The third and fourth iteration provided worse performance at all levels when compared to the previous ones. In both iterations there was a tendency for the models to interpret CAN images as PRC images and vice versa. The subset used as test data for the third iteration was the subset 1. The CAN subset inside of it is a good example of low diversity, due to the fact that its images came from a small number of large videos logically leading to an also high number of frames. In a universe of 386 images, 380 came specifically from merely 4 patients. If the images from the large videos start to get poorly classified, then the model might eventually predict wrong classifications for rest of the images. Once again, this could have happened due to the test subset for CAN being mostly built with images from a small number of large videos. Moreover, this behaviour could be also related to the structure of the video. When in a upper endoscopy, the doctor sometimes traverses the entire digestive tract until the endoscopy device reaches structures such as the pyloric antrum where it can be located the lesion. To do so, it crosses several structures that could not be as damaged as the primary focus of the lesion and, hence, have an appearance leading into a PRC classification by the model. Yet, the frames coming from the same video are all tied to the classification of the video, which is CAN.

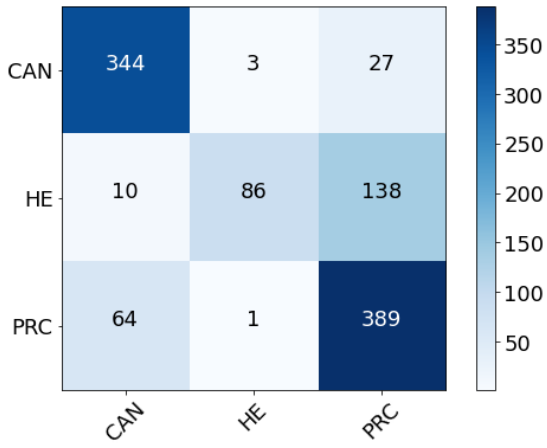
The class HE in the third iteration was confused with the two other classes in the same proportion. However, in the fourth iteration, this confusion was essentially involving PRC. As mentioned in the beginning of the section, if there were to be confusion between HE and other class, this should be PRC due to the proximity in terms of features from the images of specific lesions.

This scenario can be analysed observing the confusion matrices from the DN169 model for the third iteration and fourth iteration, in Figure 5.4. In this particular case, the images that were classified as having PRC lesions when they were in fact HE, mostly came from videos taken from the GastroIntestinal Atlas dataset. Out of 139 FN images in these situation, 120 came from 4 videos obtained from the GastroIntestinal Atlas dataset. The resulting images do not show the same quality as the ones coming from the Hyperkvasir dataset, for instance. Thus, there is a higher probability that the models have issues when classifying these images.

There could be another cause for the drop performance-wise of the models in the fourth iteration when referring to the class PRC and it is related to the diversity in the training stage. For this iteration the subsets used for training were the subsets 3,4 and 5. It is plausible to say that, in terms



(a) Confusion matrix from the DN169 model in the third iteration



(b) Confusion matrix from the DN169 model in the fourth iteration

Figure 5.4: Confusion matrices from the third and fourth iterations from the DN169 models.

of patient's diversity for PRC, these subsets do not show a significant level of diversity. They have a reduced number of patients when considering the subsets 3 and 4, as seen in Table 5.2.

This might seem false if the number of patients in subset 5 is considered. However, from the 83 patients present in the subset, 76 belong to BE. Hence, there are only 7 patients correspondent to the classes AG and IM. Furthermore, BE is a class whose images tend to be misinterpreted as cancer in the early phase of the lesion and, mostly, with IM. Such fact narrows down the model's training ability and, consequently, it will lead into less capable models when presented upon new situations.

The fifth and final iteration had an uniqueness separating it from the rest of the subsets. It was the fact that IRV2 was the best model out of all 4 models. Performance metrics like the F1S and AC achieved values of 0.82, the highest in this iteration and the PR metric obtained 0.85 (also the highest). The models recovered their positive performance in predicting images coming from HE that had not been seen in iterations 3 and 4. The number of FN was considerably low, since the maximum achieved was 8 in the NNL.

The classes PRC and CAN are still misclassified with each other. RN50 and NNL had the most problems with these two classes, as the first model mentioned had 84 FN for PRC and the second obtained 147 FN for CAN.

5.2 Micro Class Problems Results

Similarly for the case of Macro Classes, the Micro Class Problem results were analysed following the use of the 4 metrics above mentioned. Each iteration had the 4 already known confusion matrices, one for each model. The number of patients stays the same overall, bearing in mind that the classes PRC and CAN were divided into IM, AG, BE and EGC and AGC, respectively. From this division,

the only class in which there were fewer images than its peers its IM as it previously belonged to PRC class. Despite having a large number of images, only a few in PRC came from IM, considering the bigger blocks were from AG and BE. The smaller number of images per class in this problem as a result of the breakdown of two classes from the Macro class problem would be intimately related to the results in the Micro class problem. Being the number of patients intrinsically connected with the diversity of the data and bearing in mind the number of patients was divided in unequal proportions, the diversity dropped. In turn, this made an impact in the training and testing of the models, as it will be seen later in this section.

The complexity of this problem is higher than the previous one, so the results were expected to decrease accordingly. The same patterns registered in the previous problem regarding iterations and confusions can also be transposed to the current problem with small differences. The results of the validation stage supported the above mentioned scenarios for each model. As for the Macro classes problem after the same procedure was carried out regarding the validation data in each iteration, having built a confusion matrix for each model and, consequently, an analysis on the performance metrics RC and PR.

5.2.1 Global results

In light of what was previous stated, the results were summed up in the Table 5.4. As shown in this table, the DN169 was the best model on average in all the metrics. Despite having all performance metrics values higher than its peers with significant margin, there were times when the second best model, IRV2, was as good as the DN169. This happened for instance in the HE class, where IRV2 and DN169 achieved 0.81 and 0.80 for RC metric, respectively. Nonetheless, DN169 still stands out as the best model as a whole and had better results with margins of 0.08 in AC, 0.10 in RC and 0.09 in F1S and PR. The worst model generally speaking was the RN50 being outperformed in every metric by the other models. Only in the AC values, it came closer to NNL with 0.57 $\pm$ 0.06, but still with reasonable margin of distance. Regarding IRV2 and NNL, these two models are very similar overall in their metrics.

Table 5.4: Results on average for all model's performance metrics in the Micro class problem.

Model	Accuracy	Recall	F1-Score	Precision
DenseNet169	**0.72 $\pm$ 0.06**	**0.73 $\pm$ 0.08**	**0.71 $\pm$ 0.06**	**0.73 $\pm$ 0.06**
Inception-ResnetV2	0.65 $\pm$ 0.06	0.64 $\pm$ 0.08	0.63 $\pm$ 0.06	0.65 $\pm$ 0.06
NasNet Large	0.60 $\pm$ 0.06	0.60 $\pm$ 0.08	0.58 $\pm$ 0.06	0.60 $\pm$ 0.06
Resnet50	0.57 $\pm$ 0.06	0.56 $\pm$ 0.08	0.54 $\pm$ 0.06	0.56 $\pm$ 0.06

As seen in Table 5.4, the DN169 is the most consistent model and achieves very positive performance results, making it the best classifier in the two problems. One possible explanation for such scenario might be connected with what was seen in Chapter 4, precisely in Figure 2.18. The improved conditions to perform the training stage, due to its flow of information and gradients throughout the network can be a decisive factor. Given that each layer has direct access to the

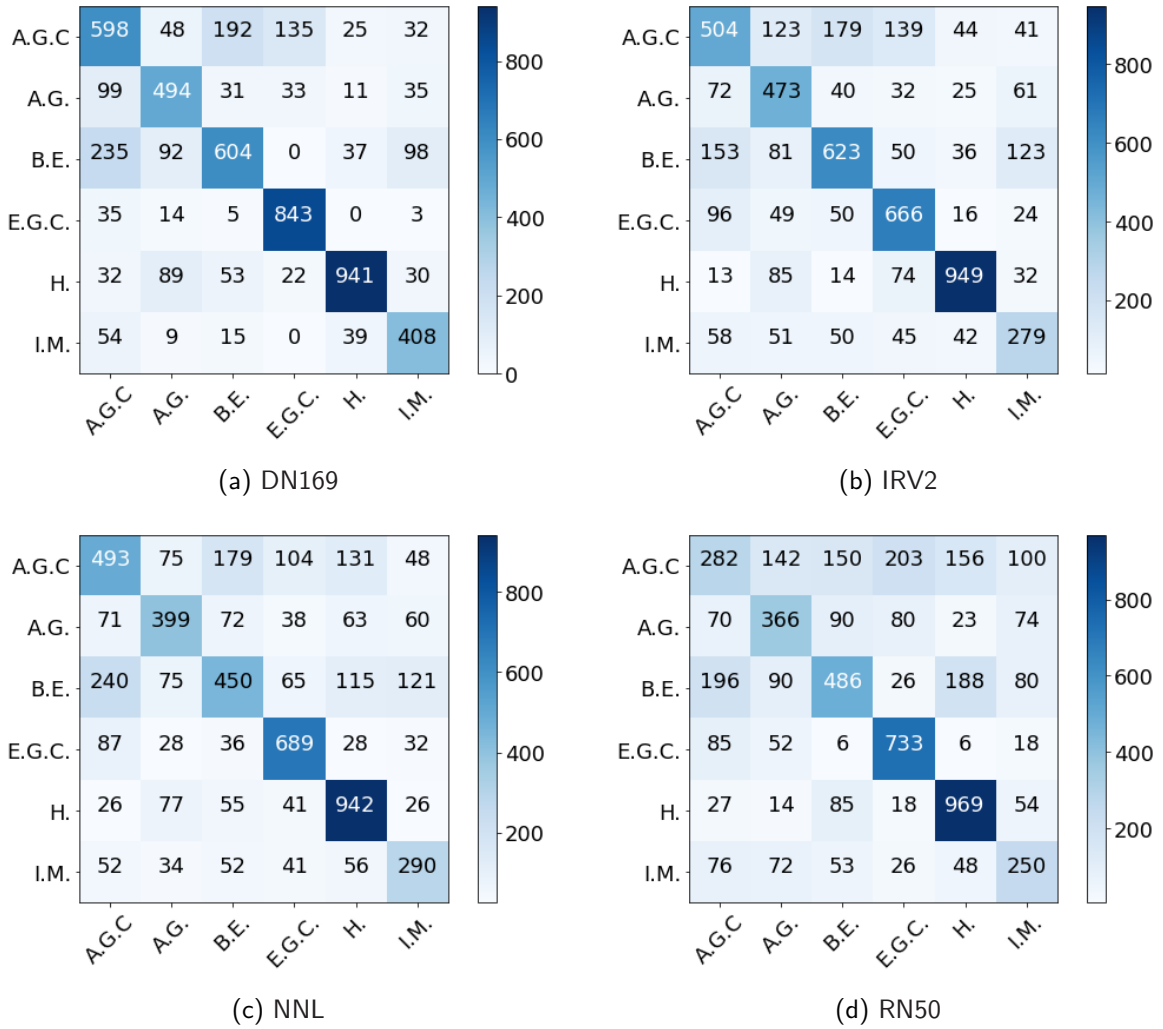
gradients from the loss function and to the original input data, the training of deeper architectures becomes easier. Furthermore, one other important detail is the regularising effect that derives from the dense connections, which reduces the overfitting in tasks with smaller training subsets. All these factors put together might be the reason why the DN169 is the best classifier.

The final confusion matrices of each model were built by summing up the value of each cell from the confusion matrices of each iteration, following what was done for the Macro classes. DN169 model has the best performance in 4 out of 6 classes. The class where DN169 had the best performance was EGC with 0.94 for the RC performance metric, while its worst class was BE with a performance of 0.57. DN169 struggled to predict AGC images, mostly predicting these images as if they were from BE and EGC. Another class where this model did not achieve positive results was BE, given that it classified a reasonable number of images as AGC instead of this class. The classes where the DN169 model did not have the better performance when compared to the other model's confusion matrices were the classes BE and HE, since IRV2 and RN50 were the ones who topped the number of predictions in these classes, respectively. The confusion matrices coming from the validation set demonstrated the exact same patterns and scenarios described above for the test stage. The results were not as good as in the test set, since the number of TP in general was slightly lower. Despite this fact, DN169 still showed the best results and the same problems in terms of confusion between the classes.

Two models with very similar confusion matrices were IRV2 and NNL. In the majority of the classes, these two models had very close RC values, with their biggest difference being in between the class BE. The similarity between these two models was one of the most distinct patterns seen in the validation set and it was best seen in the RC and PR performance metrics values and in the number of the TP of the confusion matrices. Another topic related to AGC, both of these models had complications when trying to predict its images, misclassifying them with BE. On the other hand, their best class, in terms of RC values, was HE. Another reason that shows why these two models share the same behaviour is supported by the fact that neither of the models is superior to the other in all classes. NNL is better than IRV2 in the classes IM and AGC, while for the rest of the classes the opposite happened.

RN50 had the worst performance in terms of averaged performance metrics values. This last model failed to predict more than half of the existing number of images for three classes, half of the total number of classes. Those were AGC, BE and IM. All of these recorded PR values below or equal to 0.50. The majority of the wrong classifications when carrying out predictions for the above mentioned classes was mostly due to the confusion between themselves.

However, RN50 was also the model that had the better score out of all models for RC in HE, by obtaining 0.83. Another observation drawn of its matrix is that RN50 outperformed both NNL and IRV2 in two different classes apart from HE. In EGC, RN50 had a better score than both NNL model and IRV2; and in BE this model was superior to NNL. In the confusion matrices below (Figure 5.5), it can be seen the confusion matrices obtained for the test stage in the end of all iterations.



(a) DN169 (b) IRV2

(c) NNL (d) RN50

Figure 5.5: Summary confusion matrices for all models in Micro class problem.

5.2.2 Subset analysis in Micro classes

Following the line of thought implemented in the Macro Class problem, the first iteration was the one where the models achieved better performance scores. Along with it, the second iteration closely followed the results obtained in the first iteration, being the most similar iteration to the first one. Both accomplished satisfactory results given the context of the problem. When addressing the first iteration it can be said that the first DN169 model was the most successful one, having achieved 0.88 for the AC metric. Its remaining metrics were also positive with 0.88 for PR and 0.87 for both RC and F1S. The performance metrics values for the second DN169 model varied between 0.83 and 0.86. These models had a reliable performance from a PR point of view since that, in general, from all the predictions the model made, the majority of them were correct. RC was never lower than 0.70 on both models, which led to an average value of 0.87 on the models from the two iterations. Thus, in general the performance metrics values dropped around 0.02/0.03 from the first to the second model. Also, for the iterations in the Micro class problem, the number of patients in each subset was analysed as seen in the Table 5.5.

Table 5.5: Number of patients per subset for each class for the Micro class problem.

Subset \ Patient Class	HE	AG	IM	BE	EGC	AGC
1	31	22	16	16	2	8
2	118	3	2	6	2	16
3	233	4	2	8	3	4
4	233	5	3	10	3	5
5	233	5	2	76	1	8

The result of overall metrics for DN169 was due to the almost perfect scores when predicting images corresponding to the classes AG, HE and EGC. The latter two achieved RC above 0.96, making them the better predicted classes. On the other hand, BE was not well learned by the model during the training stage, which led to the lowest RC value regarding the DN169 first iteration. The remaining classes achieved RC values around 0.75 and 0.90.

All the classes except AG obtained relatively good PR values, with all of them being above 0.85. The average value for PR was 0.86. From the set of the 6 predicted classes, in terms of RC IM was the hardest to predict for every model. Although, PR wise it always obtained one of the best scores and it was even equal to 1 for DN169. All this information regarding the first iteration can be observed through the analysis of the confusion matrix below in Figure 5.6, the best confusion matrix across all 5FCV iterations and in the Table 5.6.

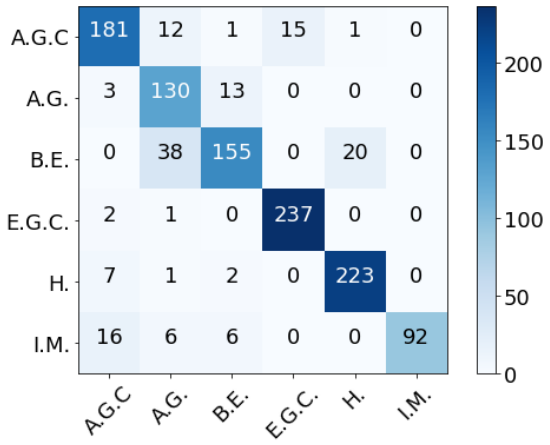


Figure 5.6: DN169's Confusion Matrix for the first iteration.

Table 5.6: Results for the first iteration in the Micro class problem for the DN169

Metric \ Class	AGC	AG	BE	EGC	HE	IM
RC	0.86	0.89	0.73	0.99	0.96	0.77
PR	0.87	0.69	0.88	0.94	0.91	1.00
F1S	0.86	0.78	0.79	0.96	0.94	0.87

In the remaining models, a drop in terms of performance values was recorded on models from

both iterations, transversal to all performance metrics. The RN50 was the model with the lowest values, having achieved 0.63 of AC in the second iteration. These models fell behind DN169 having achieved performance metric values around 0.53 and 0.75, with IRV2 being the best out of the three. This model obtained very positive RC values in most of the classes, however some could not follow the results held by DN169. While the latter one kept the same records for all classes, IRV2 could only maintain the high values for classes such as HE and AG with the downgrade of others like IM, culminating in unstable numbers.

The NNL and IRV2 were very similar between each other. Their performance metrics values differed in about 0.02/0.03 with a slight advantage for NNL. Such fact is deeply related to their confusion matrices as they shared a great number of similarities. There were classes with almost identical number of TP predicted by the models, for instance IM in the second iteration models which had 42 and 45 correct predictions for IRV2 and NNL, respectively. Also, in the first iteration the classes where the models practically behaved the same manner were HE and BE. The number of TP for the IRV2 in these classes was 164 and 226 and for the NNL model was 161 and 229, respectively.

It can be said that NNL has better performance than IRV2 by virtue of the performance metrics values. Nonetheless, this slight advantage can be better seen and explained when analysing in detail the confusion matrices for the models. As for the majority of the classes, the NNL advantage is minimum and there are classes where the models behaviour is close. In turn, if one only looks at the performance metrics this can be deceiving, considering for instance AC is calculated by looking at the confusion matrix as a whole and the other performance metrics values are an average of each class value.

As previously said, RN50 was the worst model for the specified stages of the 5FCV process. This trend persists in the following phases. The performance metric values achieved modest numbers within a range of 0.65 to 0.69 in the first iteration and 0.58 to 0.63 in the second one. There were some classes that had very positive results concerning PR and RC values. In the first iteration HE and EGC classes got for the RC metric 0.99 and 0.88, respectively. Yet, this model on both iterations also had low values, which ended up influencing the results negatively. Some classes like IM achieved 0.37 (first iteration) and 0.21 (second iteration) as results for RC measure.

According to what was mentioned before, IM was a class that caused problems to the models. Apart from the classifications carried out by DN169, the models had several difficulties classifying images belonging to it. Often images with such lesions got confused with AG images, returning such classification for them. These two stages of infection are consecutive in the development process of gastric cancer. Their physical appearance can be similar depending on whether the lesion is in advanced state and almost progressing into the next one. This could be the reason why the model shows difficulties when performing predictions and misjudging one class with the other. In Figures 5.7a and 5.7b two distinct lesions where the model NNL outputted the wrong predictions for an IM image (classified as AG) and an IM image correctly classified can be observed.

In general for all the models in the above mentioned iterations, IM images led to confusion in the models, with this class being mistaken as AG and BE classes, considering the model predicted many

(a) IM image classified as AG by NNL

(b) IM image classified correctly by NNL

Figure 5.7: Two examples of IM images with similar features where NNL made two different classifications.

images from IM as these two instead of the correct classification.

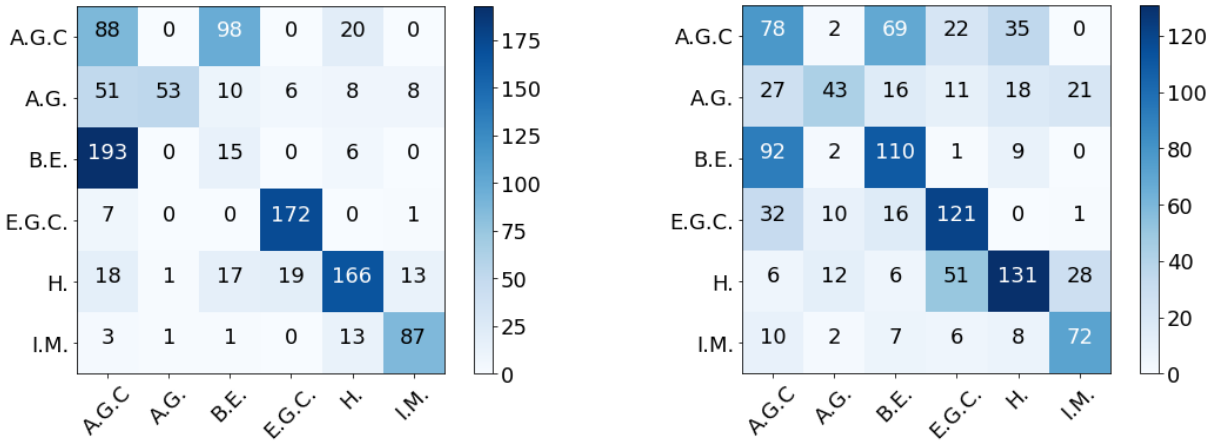
Images belonging to this class might also be misclassified as BE images. The timing in the process of developing a cancer lesion is the same for both of them. Regardless of taking place in different structures of the digestive tract (stomach and esophagus for AG/IM and BE respectively), as explained in Chapter 2, in order to say there is a BE lesion in the esophagus there has to be an IM first. The phenomenon of having a lesion in the esophagus and it develops into an IM lesion is by definition what BE is. These two lesions are the ones that happen immediately before the lesion progresses into a dysplasia (and consequently, into an EGC). This fact could be a strong argument when trying to separate and distinguish both lesions. Another reason why IM is so problematic might be related with the fact that a great number of images that make this class are from the Gastrointestinal Atlas dataset, as already seen in other moments, the quality of the images from this dataset is lower compared to images from the HyperKvasir dataset. Thus, it is reasonable and justifiable the confusion between these two classes by the models.

For the third iteration all the models drop significantly in terms of performance in the same proportions as in the Macro class models. What had been until this moment the best model, the DN169, could only get 0.54 for average accuracy while IRV2 that had been closely following it obtained 0.52. The confusion matrices are the best tool to understand what happened in the third iteration. Through their analysis it is clear that in this iteration the only classes which performance was valuable were EGC, HE and IM. The remaining classes generally speaking across all four confusion matrices struggled to maintain the highest number of predictions as their TP predictions. AGC and BE classes were a major source of confusion for all the models, with the models performing a great number of predictions as AGC, when they were in fact BE. An example can be shown by DN169, where it predicted 193 lesions out of 214 as AGC when they were BE. There is one possible reason why the models might have misclassified these images so harshly. That reason is related to the fact that, as explained in Chapter 4, the division of the videos in images was made with the 1 frame per second paradigm. While most images came from videos, the movement of the endoscope in the videos was continuous along the digestive tube. It was not always focusing the centre of the

lesion or the location where there were unequivocal evidence of the presence of AGC. Hence, in the borders/boundaries of the lesion there could be features which resembled other types of lesion like the one which got the model confused, BE. A movement such as the one described and capable of reflecting such scenarios can be seen in Figure 5.8.

Figure 5.8: Image sequence showing the continuous movement of the endoscope along the GIT and the alterations it might provoke in the images.

In the confusion matrices below (Figure 5.9) it is possible to see the results of the prediction attempts of the models DN169 and IRV2 for the third iteration.



(a) DN169 (b) IRV2

Figure 5.9: Confusion matrices for the models DN169 and IRV2 in the third iteration.

Apart from the confusion matrices presented, the ones left shown the same type of problems. Another issue involving AGC was with the HE class. The test subset for the third iteration was subset 1. There were 206 images in this subset related to AGC coming from 2 different videos. In each video, 100 frames were extracted following the previously referred paradigm, meaning they were long videos. To this class, 6 more images were added concerning images (without video frames). Consequently, the number of patients in this subset was small, which led to a lower degree of diversity in terms of lesions. For the models to correctly classify the images in a subset such as this, and bearing in mind the images come from long videos and low diversity lesions, a specific set of features should have been learned by the model. Hence, if the model did not learn the kind of features needed, it is more likely to classify all images poorly, if a few were classified as such. This is the same scenario seen in the Macro class problem following an identical result.

Apart from the motive mentioned before for BE, this correlation with the distribution of the number of images and its source could also be a reason why the prediction results were so unsatisfactory. For this test subset, the subset regarding BE also had 2 videos with 100 frames extracted in each one with 14 more images. Once more, the number of patients is reduced compared to the test subset in the first iteration (which had 76), the same insights towards diversity and, in the end, if the first images are misjudged then all are.

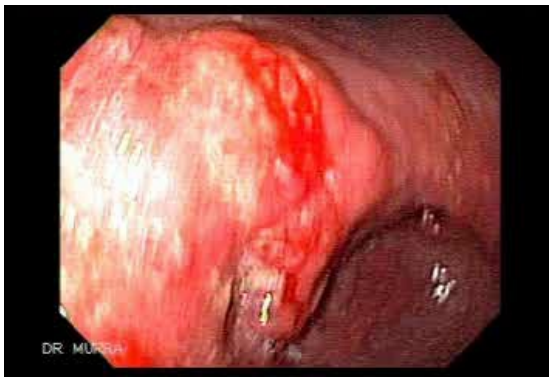
After all the models were trained and tested, it became clear the models from the iterations 4 and 5 could also be grouped, as they share conclusions and characteristics. First of all, on both iterations there is the tendency for the performance to drop in all models when compared to the first and second iterations, but not a drop as significant as in the third iteration. Another important factor is that IRV2 is the only model that behaves differently across the two iterations. In the fourth iteration this model and NNL are similar. As for the fifth iteration, its performance is essentially identical to the one from DN169.

The fourth iteration recalled the first and second ones. In this iteration the DN169 had a better performance than its peers, both in terms of confusion matrix and performance metrics wise. As seen in the Table 5.7, DN169 obtained performance metrics values between 0.67 and 0.70, representing a considerable margin (a difference around 0.13/0.14) to the other models. This scenario is just like what was seen in the first and second iteration, where the remaining models also had a difference interval of this magnitude. Regarding the confusion matrices the difference from one to the others was significant. All the TP values predicted by the DN169 were higher than the TP numbers obtained by the rest of the models in all classes except the classes HE and AG, where DN169 had the worst result. Once again, taking into account that the metrics are calculated based on the average of the metric value for each class, this led to a RC far superior for DN169 than the ones shown by its peers. The better predicted class by this model was IM, with a RC of 0.90. Although its worst class, HE had one of the best PR value, which means that, even though the model made a small number of predictions for HE, the great majority of them were right. As seen in its confusion matrix, the model made 94 predictions for HE and hit 86 of them. The reason for the low RC value for this class is the confusion with AG, which can be explained by the number of patients (3 patients) in the test subset and the origin/quality of the videos from where the classified images were extracted (GastroIntestinal Atlas).

The other three models in this iteration had very similar results and confusion matrices. All of it were between 0.50 and 0.60. To these three models the major source of confusion was the class HE, having achieved their worst RC and PR values when classifying these classes. Once more, a lot of images with no lesion got mistaken for AG, like what was seen for the DN169 in this iteration.

Additionally, another error in this iteration is related to AGC since, along with EGC, it was a source of confusion for the models. When a patient presents one of these two lesions, it can be said that he is already in a cancer level and the infection is already in the final stages of the Correa's cascade. Following this line of thought, the two are very similar, hence the model struggled in deciding which should be the correct classification for an image. When referring to this particular iteration, all models misclassified generally speaking in their previsions AGC for EGC and vice versa, some

more than others. In Figures 5.10a and 5.10b it can be observed the same lesion classified as AGC and EGC by the model IRV2. These two images are from the same video and there are similarities between the two lesions, for instance the blood. However, in Figure 5.10a there is the light from the endoscope directly over the injury, what may have cause for the model to extract different features resulting in a different classification.



(a) AGC image classified as EGC by IRV2

(b) AGC image classified correctly by IRV2

Figure 5.10: Two examples of AGC images with similar features where IRV2 made two different classifications.

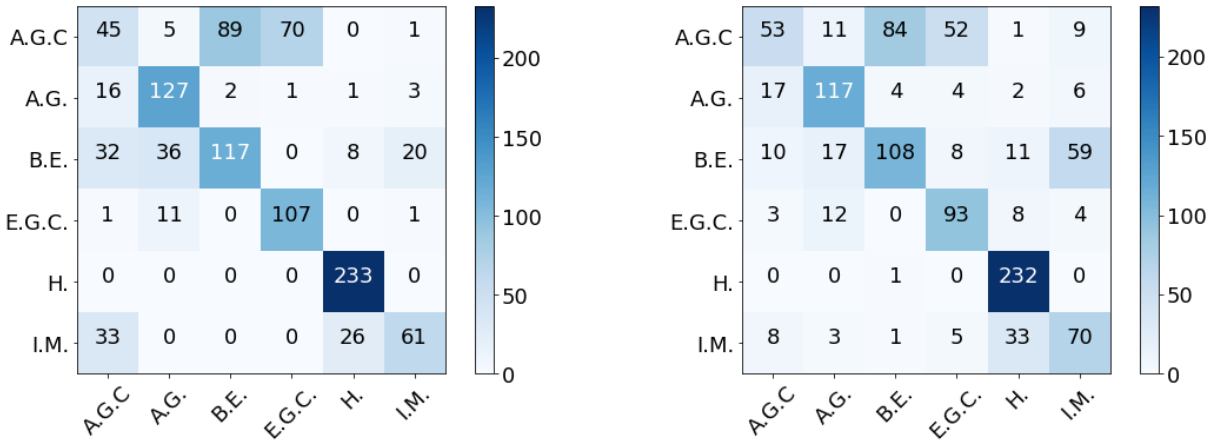
(a) EGC image classified as AGC by IRV2

(b) EGC image classified correctly by IRV2

Figure 5.11: Two examples of EGC images with similar features where IRV2 made two different classifications.

The fifth iteration recaptures the trends observed in the fourth iteration, but with the standout for the IRV2 model, considering it is the only one that shifts its behaviour. The models DN169 and IRV2 are nearly equivalent, with performance metrics values with a maximum difference of 0.03. A stronger evidence of the affinity between the models is the comparison between their confusion matrices, as seen in the Figure 5.12.

Beside the two discussed models, NNL and RN50 held similar results and confusion matrices for both iterations. They have poor/good results in the same classes, as proven by the performance metrics. In the fifth iteration, both achieved positive RC values for the classes HE (0.97 for both) and decent results for AG (around 0.65).



(a) DN169 (b) IRV2

Figure 5.12: Confusion matrices for the models DN169 and IRV2 in the fifth iteration.

Two classes that standout in this iteration by their confusion with each other are IM and BE. A tendency that has already occurred in the past iterations and that got explained previously by the definition of BE and IM and what the lesions represent in the context of the development of a gastric/esophagus cancer.

There were also wrong predictions when classifying the classes AGC and BE. The class AGC was not well learned by the models, given that the training for this class specifically had a small degree of diversity with a small number of images when compared to the number of frames extracted from videos. From 626 images that were used to train this class, 619 came from videos. Considering images represent a bigger factor of diversity than video frames, the number of frames coming from videos did not favour the model's training for this class.

Table 5.7: Performance metrics results for every model in each iteration of the Micro class problem.

DenseNet169				
Iterations	AC	RC	F1S	PR
1	0.84	0.83	0.83	0.86
2	0.81	0.82	0.79	0.79
3	0.46	0.50	0.45	0.47
4	0.71	0.73	0.71	0.74
5	0.64	0.63	0.61	0.63

InceptionResnetV2				
Iterations	AC	RC	F1S	PR
1	0.75	0.71	0.71	0.73
2	0.70	0.67	0.67	0.68
3	0.47	0.49	0.45	0.44
4	0.52	0.54	0.52	0.57
5	0.63	0.62	0.60	0.61

NasNet Large				
Iterations	AC	RC	F1S	PR
1	0.72	0.69	0.69	0.72
2	0.67	0.64	0.64	0.66
3	0.42	0.45	0.41	0.39
4	0.56	0.57	0.56	0.57
5	0.59	0.57	0.55	0.57

Resnet50				
Iterations	AC	RC	F1S	PR
1	0.65	0.62	0.61	0.63
2	0.57	0.54	0.53	0.59
3	0.40	0.42	0.40	0.44
4	0.51	0.51	0.50	0.53
5	0.50	0.48	0.44	0.46

As mentioned throughout the subsets analysis, the patient analysis represents an enormous factor

in the training and overall performance of the models. The conclusions it might deliver are not always linear, for instance in a way that more patients does not always necessarily mean better predictions. If the training set has a great number of patients associated with a great number of images and videos, it might not learn each feature properly given that there are many. In turn, if the test set is more specific when speaking in terms of diversity, this could represent a drawback for the model's predictions.

Another scenario can happen when the training stage does not have a sufficient number of patients. When there are only a certain number of frames extracted from one video and a few more images which do not come from videos, unless the test data are of identical lesions (same circumstances) as the lesions in the training set, the models might become very restricted in terms of capability of predicting different kinds of images from the same lesion. In the dataset for this master's thesis, this situation happened for IM in the subset 1. In it, there were 90 video frames coming from one video and 15 images, which provides a total of 16 patients. Given the dataset in general, it can be said that this is a good number of patients, due to the number of images. However, apart from DN169 that had reasonable results, all the other models failed to predict their IM test images. This can possibly be due to the small diversity from the video that is the majority of the training set for this class.

On the contrary, if a subset has a small amount of videos, which goes by saying a small amount of patients, the diversity is inferior. The features learned by the model with these patient's lesions will be small, as they represent specific lesions from each patient and possibly with not many characteristics in common between each other. Hence, the model's training will not be adequate to classify a test subset with a high degree of diversity, with several different patients and, consequently, different lesions.

Nonetheless, the videos can be very detailed when describing a certain lesion and its circumstances. This can lead to the model being able to learn thoroughly the features of that specific lesion. This leads to one third scenario where the model learns the features of certain lesions in a detailed way by means of an appropriate number of videos. This way the model will not overfit the data and will be capable of predicting several lesions instead of only a reduced number.

The iteration that had the best results was the one where the subset 5 was the test subset, corresponding to the first iteration. This was also the subset with the highest number of patients in all the dataset. Around 70% of the patients in this subset were from HE class (233 out of the 327) and, considering that this class had 233 images, maximum diversity was achieved (all the HE images were images which did not come from videos). This condition was likewise seen in subsets 3 and 4 (that were used in the second and fifth iterations, respectively). Bearing in mind that all these subsets had maximum diversity concerning healthy patients, the results for all three were very positive (RC values were always above 0.92 for all models). In each of the training stage for these iterations, there was a subset from HE with maximum number of patients (233) leading to a high degree of diversity. This could prove the initial premise saying that the higher the number of patients the better the results would be.

However, if all the subsets in the training set showed such numbers, there could an extreme degree

of diversity. The data regarding this class would be so diverse, the model could not learn each feature properly. This was the case where the training set was built with the subsets 3, 4 and 5, each with 233 patients for the test subset. The models were not able to learn from an exaggerated number of patients (when compared to the number of patients in the test subset) and, consequently, too much diversity.

Essentially, the diversity in the test subset working against the models can be briefly explained. When in the training stage, a model might have a limited number of patients from which it will learn the features. Contrarily, in the test subset it might have to perform predictions for data consisting of a large number of patients. For this specific task the model might not have learned enough features from the training data to classify sparse data. Therefore, it will have a poor performance in the test set. The full information regarding the number of patients can be seen in the Table 5.5.

The drop in terms of performance in the third iteration can be due to the image/frames organisation of the dataset. The images (frames) that came from larger videos were the first ones to populate the initial subsets (subsets 1 and 2) and the images and smaller videos were distributed by the last subsets (subsets 4 and 5). Such action caused the last subsets to have a great advantage when compared to the first subsets as they would have a lot more diversity, which would reflect positively on all stages of the 5FCV process.

Some trends that were described in this section were also present in the Macro class problem. This can be an indicator of the consistency in the training phases of the algorithm, which led to similar conclusions in two problems of different complexity. One preeminent conclusion from both problems is that the models tend to misjudge classes corresponding to an intermediate stage of lesion (PRC) between each other, like IM with AG or AG with BE. One clear example of this situation is present in the iterations 1 and 2 with the confusion between AG and HE or AG and BE. Another aspect which was mentioned earlier contributing to this difference was that some images are of higher quality due to the datasets from which they come from when compared to video frames and other images from other datasets. One clear example of this situation was the class HE as the majority of its images came from the Hyperkvasir dataset and the remaining images came from videos from the Gastrolab and Gastrointestinal Atlas datasets with lower quality.

5.3 Grad CAMs

After analysing the performance of the models, the next step was implementing the Grad CAMs. As mentioned in Chapter 4, these would serve as a tool to try to analyse the patterns the models obtained from the images of each class. The goal by using this technique was to evaluate the images that the model predicted correctly and images the model failed to perform the correct prediction. The Grad CAMs concern predictions carried out by the models DN169 and IRV2 from the second iteration, as they are the models with the better results when addressing performance metrics. For each class, a set of images was analysed to understand which areas in the images the models captured as coming from a certain class. The gradient in the images is meant to represent the models perception in an

image of a certain class. In practical terms, this means that if an area is filled with a strong green colour (near 1 in the gradient) than the model considered such area to have a strong presence of the Grad CAM class. However, if the area is filled with an intense blue colour, the model considers this area as weak regarding the present of the class analysed in the Grad CAM.

The first class to be analysed was the class HE. For this class, it would be expected that the models had no issues detecting the structures without a lesion. Considering these structures from the HE images represent the majority of the image, the models had several points in the image from where features related to HE could be found. These structures could possibly be found in two different situations the first one would be in images or video frames solely healthy, which means they did not have any lesion. The other scenario for this class would be for healthy structures to be seen before and after lesions in video frames with one of the possible lesions. In this case, however, despite practically speaking the video frames concern healthy structures, their correct classification is the video's classification. Hence, the model would correctly analyse the structures as healthy, thus performing an HE prediction, but this would not be the true classification. In the Figure 5.13 it can be seen the original HE image and the corresponding Grad CAM by the model DN169 in the second iteration. Along with these two images, it is also present in this figure the Gradient that serves as measure for all the Grad CAMs. This Gradient corresponds to the scale in which a model considers the specified class to be in a certain area of the image. If the area is covered in Green it means the model considered that in such area the specified class was present. On the contrary, if the area was blue the model did not find any traces of the specified class in it.

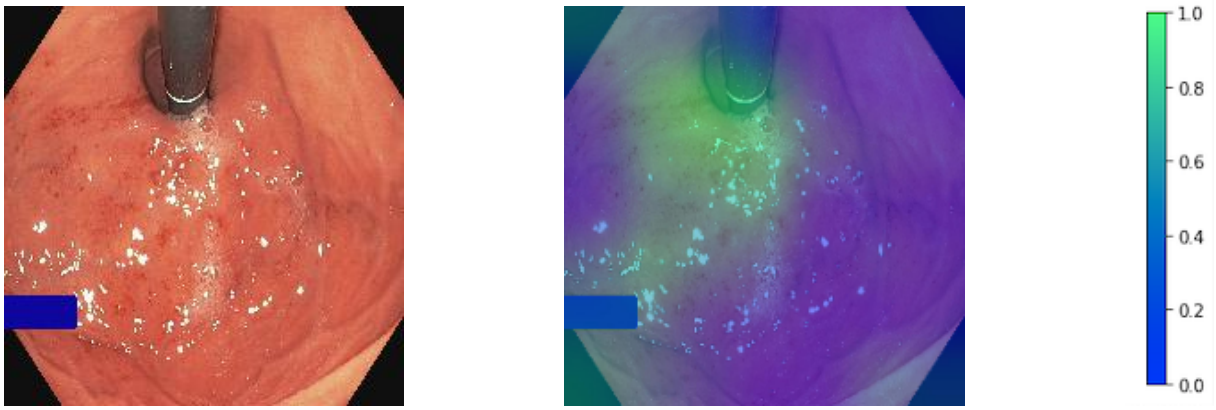
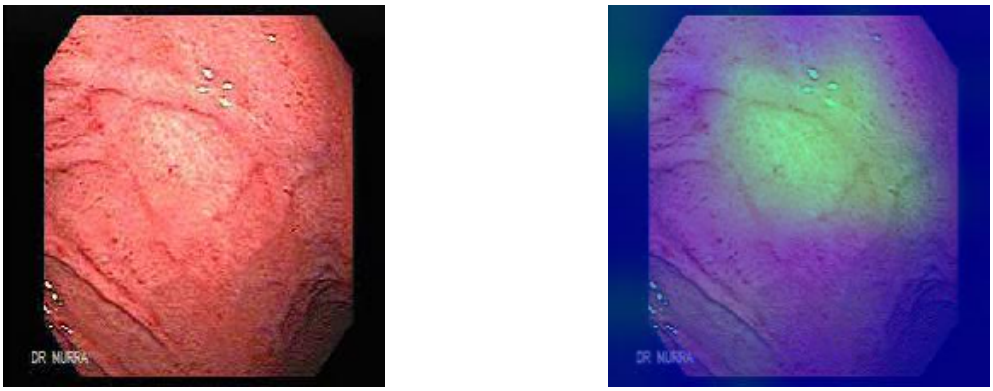


Figure 5.13: Original HE image and the Grad CAM from the prevision DN169 performed regarding HE in this image from the DN169 model in the second iteration with the respective Grad CAM gradient on the right.

The following class to be analysed was AG. In the images from this class in theory there could be a lot of HE structures, due to AG being the first stage of infection in the process of developing a gastric cancer, as stated in Chapter 2. There is also the possibility that, likewise what was mentioned for HE class, the beginning or the surrounding areas of a more advanced lesion like EGC or AGC look similar to a AG lesion. In Figure 5.14, it can be seen in the original image structures which have started to lose the pleated texture, characteristic from the AG stage of infection in the centre of the image. Consequently, as the prevision of the model is correct, the area with the more and stronger

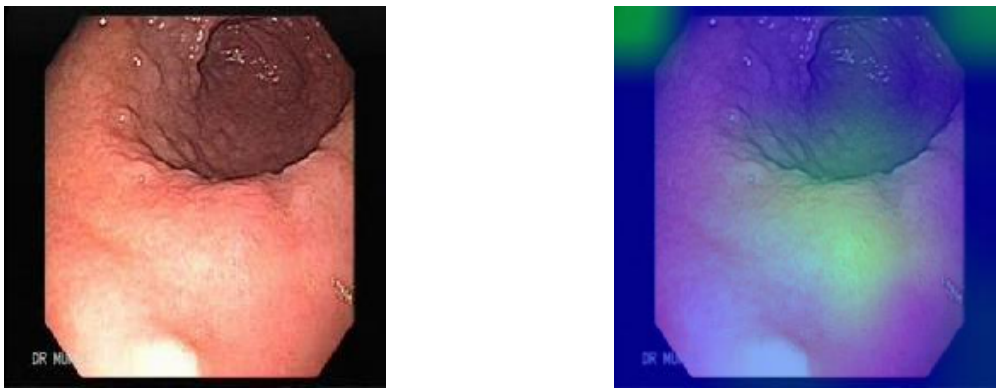
incidence of green should also be in centre of the image, as evidenced in the Grad CAM image.



(a) Original AG image. (b) DN169 Grad CAM image.

Figure 5.14: Original AG image and the Grad CAM from the prevision DN169 performed regarding AG in this image from the DN169 model in the second iteration.

Afterwards, the class investigated was IM. The only model for which this class achieved positive results in the averaged performance metrics in the second iteration was the DN169. The remaining models misclassified these images mostly as BE but also as AG. Such confusion was already seen and explored in the previous sections as well as their motives. According to what was covered in the Chapter 2, after AG, an IM lesion should aggravate the state of infection by causing for the pleated texture to disappear completely in the primary zone of infection, along with the change of the stomach's structure where the lesion lies, possibly resulting in the structures to become flat. This scenario can be seen in Figure 5.15.



(a) Original IM image. (b) DN169 Grad CAM image.

Figure 5.15: Original IM image and the Grad CAM from the prevision DN169 performed regarding IM in this image from the DN169 model in the second iteration.

For the class BE, the selected image to demonstrate the prediction performed by DN169 should be one where it is clear that the structures are slightly different from the ones in the other Grad CAMs. This is due to the fact that this type of lesion takes place in the esophagus. This class had errors in several of the different models built, throughout the five iterations. In the second iteration, DN169 misclassified it mostly as IM due to what was explained before in the Chapter 2 and previous sections of the current chapter. The example for a correct prevision of this model for a BE lesion can be seen

on Figure 5.16. The lesion in the referred image occurs throughout the esophagus and in the section between the esophagus and the stomach, the lower esophagus.

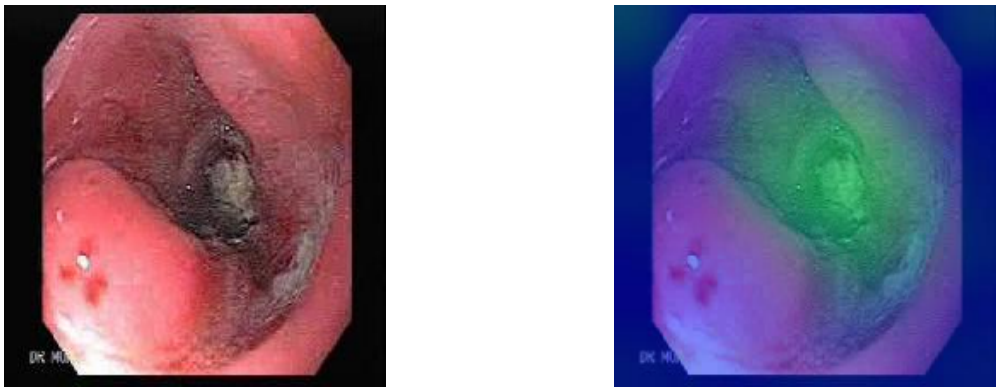
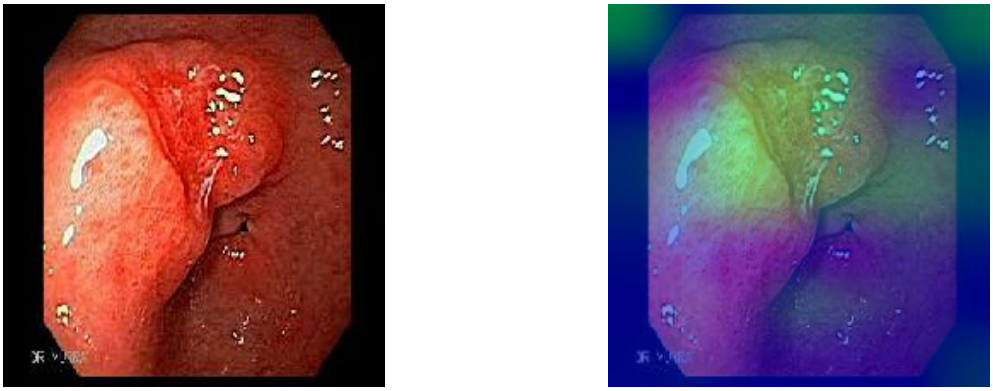


Figure 5.16: Original BE image and the Grad CAM from the prevision DN169 performed regarding BE in this image from the DN169 model in the second iteration.

The class that followed was EGC. This was the class along with HE that was better learned by all the models in general. Even in the iterations with worst average performance metrics results, this class still had positive results from all the models. There were a few images where injuries from this class were confused as being AGC injuries, specifically in the second iteration. From the results, is is possible to conclude the models did not struggle in finding structures with EGC lesions. This could lead to a clear and determined classification from the models, by selecting a specific area where the lesion occurs, reflecting it later in the Grad CAM. One clear example where this scenario happens can be seen in Figure 5.17.

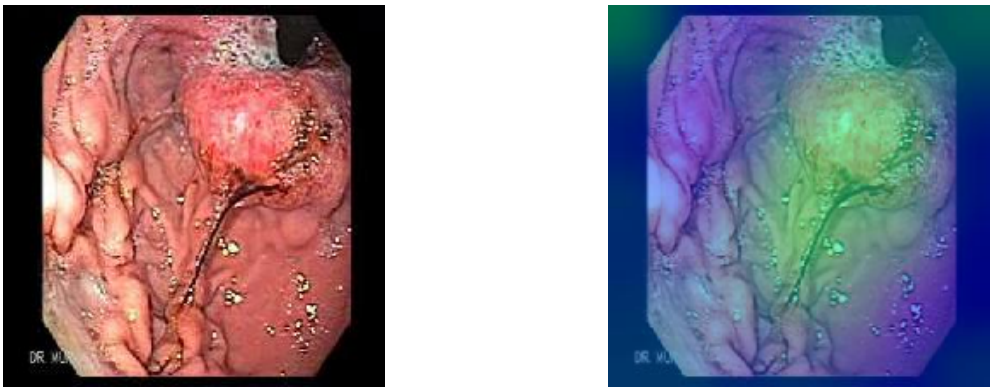


(a) Original EGC image. (b) DN169 Grad CAM image.

Figure 5.17: Original EGC image and the Grad CAM from the prevision DN169 performed regarding EGC in this image from the DN169 model in the second iteration.

The last class was AGC. In a similar situation to what was seen for HE, the Grad CAM for this class could be scattered all over the image. However, unlike HE class, instead of having no lesion, the structures have almost every area covered by an AGC lesion. The centre of the lesions in this class should be easy to identify by the model, with the images having a lot of features from which it is perceptible an AGC lesion. However, in the surrounding areas to the centre of the lesion there could be some features that resembled other features from other classes. Like what was said previously for

the Grad CAMs of the classes AG and IM, the first features to appear in images with classification AGC could appear to be mild and characteristic from other types of lesions. Hence, the Grad CAM in an AGC image might be coloured with green in the centre of the lesion. And, as the area being analysed distances itself from the centre, the green colour should also start to fade. An example from a Grad CAM regarding a AGC lesion can be seen on Figure 5.18.



(a) Original AGC image. (b) DN169 Grad CAM image.

Figure 5.18: Original AGC image and the Grad CAM from the prevision DN169 performed regarding AGC in this image from the DN169 model in the second iteration.

Chapter 6

Conclusions

The present Master's thesis aimed to solve two different classification tasks by using Machine Learning methods, more precisely Deep Learning algorithms. The subject to be analysed was medical imaging, specifically images from upper endoscopy exams, containing lesions of the Gastrointestinal Tract. These lesions correspond to different stages of the development of gastric cancer in the Correa's Cascade. In the field of Deep Learning, the use of Convolutional Neural Network to solve this type of problems has been increasing and is of today a reliable tool to be applied in such problems. The two classification problems to be solved were of different complexity, one was a multi-class classification problem involving 3 classes (Macro) and the second one involving 6 classes (Micro). For this study, different Convolutional Neural Network models were created. They were implemented by resourcing to the concept of transfer learning, while using the *Keras* library from the software *Tensorflow*. To build the networks that would perform the classification tasks, the chosen architectures were DenseNet169, Inception-ResnetV2, NasNet Large and Resnet50. The strategy used to build the models was a 5-Fold Cross-Validation strategy, where each subset of the data had different class subsets, with an approximately equal number of images per class subset.

The models built using the transfer learning concept were pretrained in the Imagenet dataset. After each test stage, the performance metrics Accuracy, Recall, F1-Score and Precision were measured, along with confusion matrices for each model. From all the metrics used, in the end an average of each value was calculated and a summation confusion matrix for each model was built.

There were positive results when distinguishing images without lesions from the ones with a lesion. There was also a trend where the models achieved positive results in separating mild lesions from severe lesions, specifically in the Macro Classes, the Precancerous from the Cancerous lesions. On the other hand, the models struggled with images which lesions resembled other lesions from a different class. The predominant situations where these examples were observed relate to the classes Early Gastric Cancer and Advanced Gastric Cancer, but also Intestinal Metaplasia, Barrett's Esophagus and Atrophic Gastritis. Such examples happened in its vast majority following the Micro Classes problem. For the first scenario there might be a confusion due to the similarity between these two lesions, where sometimes professional examiners also have difficulties in distinguishing them. The second scenario might be related to the fact that all of the lesions happen in similar structures. This concerns

primarily structures which have not yet been severely transformed, due to the existence of a lesion.

One other outcome from the analysis carried out in the Micro Class problem is the possibility of the model misclassifying a lesion, due to the surrounding area of it. Whenever a model is analysing an image containing an advanced lesion, the areas immediately before or after the centre of the lesion might not appear so severe. As seen in Chapter 5, the model might consider such area to be in a less advanced state of injury. Following these circumstances, the model could predict a video frame where the endoscope focused specifically in one of the areas referred. Consequently, it will predict such area as having a lesion in a previous stage to the actual stage of infection.

Other conclusion from this study is related to the quality of the images and video frames. This is a factor that should be taken into consideration when building the dataset. An endoscopy can be a difficult exam to collect images from. The continuous movement of the camera in the endoscope might lead to video frames where the image is not focused, introducing noise to the dataset.

In terms of network architectures, the ones that achieved better results in terms of averaged performance metrics were the DenseNet169 and the Inception-ResnetV2. As mentioned earlier, in Chapter 5, the DenseNet169 is better prepared when it comes to dealing with overfitting phenomena. Keeping in mind the dimensions of the dataset and the redundancy coming from the video frames, this aspect of the DenseNet169 is an advantage when compared to the other models created. The Inception-ResnetV2, as seen in Chapter 4, represents a deep network architecture and an upgrade when compared to the Resnet50. This upgrade regarding the residual network derives from the fact that Inception-ResnetV2 maintains the computational efficiency coming from the Inception architecture coupled with the higher training speed gained from the use of the residual connections, from the residual network architecture. Differently, the NasNet Large did not achieve similar results to the DenseNet169 and Inception-ResnetV2, as it represents an architecture best suited for datasets of large dimensions, which is not the case in this Master's thesis. Hence, the averaged performance metrics results regarding this architecture, despite falling short when compared to the previous two mentioned, can improve if more images are added to each class of the dataset.

6.1 Future Work

As future work, a plausible idea would be to increase the size of the dataset, by gathering more data. By increasing first the number of images per class with more lesions, the models would have a better training stage. This would happen mostly due to the higher diversity of lesions. As mentioned earlier in Chapter 5, a subset with the majority of images coming from very different lesions could lead to poor classifications (as a result of learning too many features with a small amount of data). However, if for a large amount of features there is an equally large amount of data (images), the previous scenario would not be a possibility. Another possible scenario would be reinstating the class "Others". If more data could be collected regarding other type of lesions like what was done initially, this class could achieve a similar number of images to its peers. An even more advanced step would be to extend the analysis over the images of the Gastrointestinal Tract. This would

consist of separating the images from the class "Others" into the specific lesions that make this end. By separating, classes such as "Polyps" and "Ulcers" would be formed. This modification would change the scope of the classification tasks. Instead of being images containing lesions that concern specifically the development of a gastric cancer, it would be images containing lesions capable of occurring in the Gastrointestinal Tract. Along with the new classes consisting of images with new lesions, new structures could also be analysed. The entire esophageal tube could be analysed, instead of simply the transition between esophagus and stomach - structure where the Barrett's Esophagus occurs. One other structure in the Gastrointestinal Tract examined during an endoscopy is the duodenum. This part of the Gastrointestinal Tract can also have several different lesions associated with it. Nonetheless, images containing regular structures (without a lesion) of it can also be added to the dataset, in the Healthy class.

One other solution as future work might be the usage of a new and more recent architecture. The High-Resolution Network (HR.Net) is a recent model, that first appeared in 2020. Unlike other Convolutional Neural Network models, such as the Resnet50, the HR.Net maintains high resolution representations of the input data throughout the whole process of training the network [66]. Its two main features are its capability to connect the high-to-low resolution convolution streams in parallel and repeatedly exchange the information across resolutions. These are different characteristics from the ones the previously used models have. They lead to the architecture's principal advantage of having semantically richer and spatially more precise representation of the data. Thus, this architecture is a valid option to be used in problems such as the ones covered in this Master's thesis [66].

Besides adding a new type of task and a new architecture, it is also possible to make an attempt to improve the already built classification models. The possible method of doing so is by creating an ensemble of models with the three models that have the better results, in terms of averaged performance metrics. This solution would, therefore, include the DenseNet169, the Inception-ResnetV2 and the NasNet Large models. In this strategy the three models would be combined to solve the classification tasks. These models would be trained and tested using the same data. One advantage from a set of models such as this one when compared to a standalone simple model is that the ensemble can reduce the variance of the predictions. A simple model has increased flexibility and can scale in proportion to the size of the dataset. This factor leads to the high variance values and it can be reduced by the ensemble. A traditional way of combining models to form an ensemble is the *committee of networks* [9]. For this technique, all the models have the same configuration, but different training weights. Each model is later used to make a prediction and the final prediction would be calculated as an average of the predictions. However, this method is computationally expensive in terms of training the models. Furthermore, the more models are added to the set of models, the more the performance decreases [9].

One other possibility that could be considered as future work is to change the task type. Instead of being just a classification task (divided in two different problems), there could be an object detection task. Apart from building a model capable of performing predictions for what lesion is seen in an image, a new model could detect where the lesion is located. Such task could be performed using a VGG16-SSD model like the one built by Toshiaki Hirasawa et al. [37], one of the papers seen in

Chapter 3. After identifying the lesion, the model would proceed to place a bounding box surrounding the area of the lesion. Afterwards, to understand if the performance of the model was a positive one, a metric such as the area of intersection, along with Precision and Recall could be used.

Bibliography

[1] Khalid Alafandy, Hicham Omara, Mohamed LAZAAR, and Mohammed Al Achhab. Investment of classic deep cnns and svm for classifying remote sensing images. *Advances in Science Technology and Engineering Systems Journal*, 05:652–659, 10 2020. doi:10.25046/aj050580.

[2] Noura AlHinai et al. *Biomedical Signal Processing and Artificial Intelligence in Healthcare*. Academic Press, 2020. ISBN: 978-0-12-818946-7. doi:10.1016/C2018-0-04775-1.

[3] Michal Antkowiak. Artificial neural networks vs . support vector machines for skin diseases recognition. 2006.

[4] Digestive Health Associates. Upper endoscopy (egd) faqs. Online, accessed in 2021-09-14.

[5] Anup Bhande. What is underfitting and overfitting in machine learning and how to deal with it. Online, accessed in 2021-01-13, March 2008.

[6] H. Borgli et al. Hyperkvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy. *Scientific Data 7*, 42, August 2020. ISSN: 1541-4612. doi:https://doi.org/10.1038/s41597-020-00622-y.

[7] Freddie Bray et al. Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *A cancer journal for clinicians*, 68:394–424, November 2018. doi:10.3322/caac.21492.

[8] Jason Brownlee. Gradient descent for machine learning. March 2016, accessed in 2021-09-15.

[9] Jason Brownlee. Ensemble learning methods for deep learning neural networks. December 2018, accessed in 2021-08-22.

[10] Jason Brownlee. A gentle introduction to dropout for regularizing deep neural networks. December 2018, accessed in 2021-09-17.

[11] Jason Brownlee. Difference between a batch and an epoch in a neural network. July 2018, accessed in 2021-09-24.

[12] Jason Brownlee. A gentle introduction to batch normalization for deep neural networks. January 2019, accessed in 2021-08-22.

[13] Jason Brownlee. How to choose loss functions when training deep learning neural networks. January 2019, accessed in 2021-09-02.

[14] Jason Brownlee. How to calculate the kl divergence for machine learning. October 2019, accessed in 2021-09-04.

[15] Jason Brownlee. A gentle introduction to pooling layers for convolutional neural networks. April 2019, accessed in 2021-09-04.

[16] Jason Brownlee. How to configure image data augmentation in keras. April 2019, accessed in 2021-09-17.

[17] Jason Brownlee. Softmax activation function with python. October 2020, accessed in 2021-09-20.

[18] Jason Brownlee. Weight initialization for deep learning neural networks. February 2021, accessed in 2021-09-16.

[19] Marcia Irene Canto. Chromoendoscopy. Online, acessed in 2021-01-08.

[20] Bum-Joo Cho et al. Automated classification of gastric neoplasms in endoscopic images using a convolutional neural network. *Endoscopy*, 51:1121–1129, December 2019. doi:10.1055/a-0981-6133.

[21] François Chollet. Keras. Online, 2015, accessed in 2021-09-21.

[22] Pelayo Correa and JeanMarie Houghton. Carcinogenesis of helicobacter pylori. *American Gastroenterological Association*, 133:659–672, August 2007. doi:10.1053/j.gastro.2007.06.026.

[23] Pelayo Correa and M Blanca Piazuelo. The gastric precancerous cascade. *Journal of Digestive Diseases*, 1:2–9, January 2012. doi:10.1111/j.1751-2980.2011.00550.x.

[24] Delany S.J. Cunningham P., Cord M. *Supervised Learning*. Springer, 2008. ISBN: 978-3-540-75171-7. doi:https://doi.org/10.1007/978-3-540-75171-7_2.

[25] Sociedade Portuguesa da Endoscopia Digestiva. Endoscopia digestiva alta. Online, accessed in 2021-01-03.

[26] Arden Dertat. Applied deep learning - part 4: Convolutional neural networks. Online, November 2017, accessed in 2021-09-15.

[27] American Society for GastroIntestinal Endoscopy. Adverse events of upper gi endoscopy. Online, accessed in 2021-08-29.

[28] Eddie Forson. Understanding ssd multibox — real-time object detection in deep learning. Online, November 2017, accessed in 2021-01-04.

[29] Francesco Renna. Introduction to Deep Learning and Convolutional Neural Networks. Presentation of deep learning and cnn, Faculty of Sciences, Department of Computer Science, 2019.

[30] Chathurika Gamage, Isuru Wijesinghe, Charith Chitraranjan, and Indika Perera. Gi-net: Anomalies classification in gastrointestinal tract through endoscopic imagery with deep learning. In *2019 Moratuwa Engineering Research Conference (MERCon)*, pages 66–71, 2019. doi:10.1109/MERCon.2019.8818929.

[31] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning (Adaptive Computation and Machine Learning series)*. The MIT Press, nov 2016. ISBN: 978-0262035613.

[32] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview. *ArXiv*, abs/2008.05756, 2020.

[33] Rodger C. Haggitt. Barrett's esophagus, dysplasia, and adenocarcinoma. *Human Pathology*, 25 (10):982 – 993, 1994. ISSN: 0046-8177. doi:https://doi.org/10.1016/0046-8177(94)90057-4.

[34] Chung Min Han et al. Genetic profile analysis of a patient with metachronous gastric cancer with a family history of gastrointestinal cancers. *The Korean Journal of Helicobacter and Upper Gastrointestinal Research*, 17:218–223, December 2017. doi:10.7704/kjhugr.2017.17.4.218.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[36] Q He et al. Deep learning-based anatomical site classification for upper gastrointestinal endoscopy. *International Journal of Computer Assisted Radiology and Surgery*, 15:1085—-1094, May 2020. doi:https://doi.org/10.1007/s11548-020-02148-5.

[37] Toshiaki Hirasawa et al. Application of artificial intelligence using a convolutional neural network for detecting gastric cancer in endoscopic images. *Gastric Cancer*, 21:653–660, January 2018. doi:10.1007/s10120-018-0793-2.

[38] Gao Huang et al. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[39] National Cancer Institute. What is cancer? - understanding cancer. Online, accessed in 2020-12-29.

[40] Takumi Itoh et al. Deep learning analyzes helicobacter pylori infection by upper gastrointestinal endoscopy images. *Endoscopy International Open*, pages 139–144, February 2018. doi:10.1055/s-0043-120830.

[41] Ryan Nash Keiron O'Shea. An Introduction to Convolutional Neural Networks. OEM Specification and Technical Information 6470-000-01, , 2015.

[42] Mina Khoshdeli, Richard Cong, and Bahram Parvin. Detection of nuclei in he stained sections using convolutional neural networks. 02 2017.

[43] V. V. Khryashchev et al. Deep learning for gastric pathology detection in endoscopic images. In *Proceedings of the 2019 3rd International Conference on Graphics and Signal*

Processing, page 90–94. Association for Computing Machinery, 2019. ISBN: 9781450371469. doi:10.1145/3338472.3338492.

[44] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[45] Sunil kumar Jangir. Beginner's guide for convolutional neural network (cnn / convnets). Online, December 2018, accessed in 2021-01-18.

[46] Alzubaidi L. et al. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 08, mar 2021. doi:https://doi.org/10.1186/s40537-021-00444-8.

[47] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, page 21–37, 2016. ISSN: 1611-3349. doi:10.1007/978-3-319-46448-0_2.

[48] Jianxiang Ma, Anlong Ming, Zilong Huang, Xinggang Wang, and Yu Zhou. Object-level proposals. pages 4931–4939, 10 2017. doi:10.1109/ICCV.2017.527.

[49] John Hopkins Medicine. What is an upper gi endoscopy? Online, accessed in 2020-12-26, .

[50] John Hopkins Medicine. Upper gastrointestinal series. Online, accessed in 2021-01-03, .

[51] Divyanshu Mishra. Demystifying convolutional neural networks using gradcam. Online, October 2018, accessed in 2021-09-21.

[52] Samantha Morais et al. Trends in gastric cancer mortality and in the prevalence of helicobacter pylori infection in portugal. *European Journal of Cancer Prevention*, 25:275–281, July 2016. doi:10.1097/CEJ.0000000000000183.

[53] Laurent Palazzo, Anne Marie Lennon, and Ian Penman. *Gastrointestinal Endoscopy in Practice*. Churchill Livingstone, 2011. ISBN: 978-0-7020-3128-1.

[54] Vigneashwara Pandiyan. *Modelling and in-process monitoring of abrasive belt grinding process*. PhD thesis, 03 2019.

[55] Ravindra Parmar. Common loss functions in machine learning. Online, September 2018, accessed in 2021-01-14.

[56] Kurtis Pykes. The vanishing/exploding gradient problem in deep neural networks. Online, May 2020, accessed in 2021-08-08.

[57] Barbeiro S. et al. Narrow-band imaging: Clinical application in gastrointestinal endoscopy. *GE - Portuguese Journal of Gastroenterology*, pages 40–53, 2019. doi:https://doi.org/10.1159/000487470.

[58] Ihab S. Mohamed. *Detection and Tracking of Pallets using a Laser Rangefinder and Machine Learning Techniques*. PhD thesis, 09 2017. doi:10.13140/RG.2.2.30795.69926.

[59] Saed Sayad. Artificial neural network. Online, accessed in 2021-09-14.

[60] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[61] Linda Shapiro and George Stockman. *Computer Vision*. Pearson, March 2000. ISBN: 0130307963.

[62] SAGAR SHARMA. Epoch vs batch size vs iterations. Online, September 2017, accessed in 2021-09-23.

[63] K. Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

[64] Pentti Sipponen and Heidi-Ingrid Maaroos. Chronic gastritis. *Scandinavian Journal of. Gastroenterology*, 50:657–667, April 2015. doi:10.3109/00365521.2015.1019918.

[65] Science Source. Stomach cancer biopsy, endoscope view. Online, accessed in 2021-01-08.

[66] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5686–5696, 2019. doi:10.1109/CVPR.2019.00584.

[67] Jae Kyu Sung. Diagnosis and management of gastric dysplasia. *Korean Journal of International Meidicine*, 31:201–209, March 2016. doi:10.3904/kjim.2016.021.

[68] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. doi:10.1109/CVPR.2015.7298594.

[69] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4278–4284. AAAI Press, 2017.

[70] John K Triantafillidis et al. Sedation in gastrointestinal endoscopy: Current issues. *World Journal of Gastroenterology*, 19:463–481, January 2013. doi:10.3748/wjg.v19.i4.463.

[71] Muneeb ul Hassan. Vgg16 – convolutional network for classification and detection. Online, November 2018, accessed in 2021-12-11.

[72] Yuwei Zhang. Epidemiology of esophageal cancer. *World Journal of Gastroenterology*, 19:34, September 2013. ISSN: 5598–5606. doi:10.3748/wjg.v19.i34.5598.

[73] Barret Zoph et al. Learning transferable architectures for scalable image recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8697–8710, 2018. doi:10.1109/CVPR.2018.00907.