Universidade de Aveiro
2021

Inês Filipa Fernandes Videira Lopes

Redes Neurais Convolucionais para deteção de *landmarks* gástricas

Deep Convolutional Neural Network for gastric landmarks detection

Universidade de Aveiro
2021

Inês Filipa Fernandes Videira Lopes

Redes Neurais Convolucionais para deteção de *landmarks* gástricas

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Biomédica, realizada sob a orientação científica do Professor Doutor Francesco Renna, Professor Auxiliar Convidado do Departamento de Ciências de Computadores da Faculdade de Ciências da Universidade do Porto e do Professor Doutor Augusto Silva, Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri

presidente	Prof. Doutor Manuel Pedro Graça professor auxiliar em Regime Laboral da Universidade de Aveiro
arguente	Doutor Hélder Pinto de Oliveira professor auxiliar convidado da Faculdade de Ciências da Universidade do Porto
orientador	Prof. Doutor Augusto Silva professor associado da Universidade de Aveiro

agradecimentos

Gostaria de endereçar em primeiro lugar, a minha gratidão, ao meu co-orientador Prof. Dr. Francesco Renna, ao Prof. Dr. Miguel Coimbra e ao Prof. Dr. Augusto Silva que foram as pessoas com uma contribuição mais direta na realização deste projeto, além da motivação e excelente orientação que recebi, agradeço a confiança que me foi depositada.

Por fim, uma palavra de agradecimento especial aos meus pais, irmão, namorado e amigos pelo apoio e encorajamento durante este ano letivo.

palavras-chave

Inteligência artificial, rede neural convolucional, *landmarks* gastrointestinais, esofagogastroduodenoscopia

resumo

O cancro gástrico é o quinto cancro mais incidente no mundo e quando diagnosticado numa fase avançada a taxa de sobrevivência é de apenas 5%-25%. Assim, é essencial que este cancro seja detetado numa fase precoce. No entanto, os médicos especializados neste diagnóstico nem sempre são capazes de uma boa performance de deteção durante o exame de diagnóstico, a esofagogastroduodenoscopia (EGD). As lesões precoces nas paredes do sistema digestivo são quase impercetíveis e confundíveis com a mucosa do estômago, sendo difíceis de detetar. Por outro lado, os médicos correm o risco de não cobrirem todas as áreas do estômago durante o diagnóstico, podendo estas áreas ter lesões.

A introdução da inteligência artificial neste método de diagnóstico poderá ajudar a detetar o cancro gástrico numa fase mais precoce. A implementação de um sistema capaz de fazer a monitorização de todas as áreas do sistema digestivo durante a EGD seria uma solução de forma a prevenir o diagnóstico de cancro gástrico em estados avançados. Este trabalho tem como foco o estudo da monitorização de *landmarks* gastrointestinais (GI) superiores, que são zonas anatómicas do sistema digestivo mais propícias ao surgimento de lesões e que permitem fazer um melhor controlo das áreas esquecidas durante a EGD.

O uso de redes neurais convolucionais (CNNs) na monitorização de *landmarks* GI tem sido grande alvo de estudo pela comunidade científica, por serem redes com uma boa capacidade de extração *features* que melhor caraterizam as imagens da EGD.

O objetivo deste trabalho consistiu em testar novos algoritmos automáticos baseados em CNNs capazes de detetar *landmarks* GI superiores para evitar a presença áreas não cobertas durante a EGD, aumentando a qualidade deste exame.

Este trabalho difere de outros estudos porque foram usadas classes de *landmarks* GI superiores mais próximas do ambiente real da EGD. Dentro de cada classe incluímos imagens com patologias e de tecido saudável da respetiva zona anatómica, ao contrário dos demais estudos. Nos estudos apresentados no estado de arte apenas foram consideradas classes de *landmarks* com tecido saudável em tarefas de deteção de *landmarks* GI.

Testámos algumas arquiteturas pré-treinadas como a ResNet-50, a DenseNet-121 e a VGG-16. Para cada arquitetura pré-treinada, testámos algumas variáveis: o uso de *class weights* (CW), o uso das camadas *batch normalization* e *dropout*, e o uso de *data augmentation*. A arquitetura CW ResNet-50 atingiu uma *accuracy* de 71,79% e um coeficiente de correlação de Mathews (MCC) de 65,06%.

Nos estudos apresentados no estado de arte, apenas foram estudados sistemas de *supervised learning* para classificação de imagens EGD enquanto, que no nosso trabalho, foram também testados sistemas de *unsupervised learning* para aumentar o desempenho da classificação. Em particular, arquiteturas *autoencoder* convolucionais para extração de *features* de imagens GI sem *labels.* Assim, concatenámos os *outputs* das arquiteturas *autoencoder* convolucionais com a arquitetura CW ResNet-50 e alcançamos uma *accuracy* de 72,45% e um MCC de 65,08%.

keywords

Artificial intelligence, convolutional neural network, gastrointestinal landmarks, esophagogastroduodenoscopy

abstract

Gastric cancer is the fifth most incident cancer in the world and, when diagnosed at an advanced stage, its survival rate is only 5%-25%, providing that it is essential that the cancer is detected at an early stage. However, physicians specialized in this diagnosis have difficulties in detecting early lesions during a diagnostic examination, esophagogastroduodenoscopy (EGD). Early lesions on the walls of the digestive system are imperceptible and confounded with the stomach mucosa, being difficult to detect. On the other hand, physicians run the risk of not covering all areas of the stomach during diagnosis, especially areas that may have lesions. The introduction of artificial intelligence into this diagnostic method may help to detect gastric cancer at an earlier stage. The implementation of a system capable of monitoring all areas of the digestive system during EGD would be a solution to prevent the diagnosis of gastric cancer in advanced states. This work focuses on the study of upper gastrointestinal (GI) landmarks monitoring, which are anatomical areas of the digestive system more conducive to the appearance of lesions and that allow better control of the missed areas during EGD exam.

The use of convolutional neural networks (CNNs) in GI landmarks monitoring has been a great target of study by the scientific community, with such networks having a good capacity to extract features that better characterize EGD images.

The aim of this work consisted in testing new automatic algorithms, specifically CNN-based systems able to detect upper GI landmarks to avoid the presence of blind spots during EGD to increase the quality of endoscopic exams.

In contrast with related works in the literature, in this work we used upper GI landmarks images closer to real-world environments. In particular, images for each anatomical landmark class include both examples affected by pathologies and healthy tissue.

We tested some pre-trained architectures as the ResNet-50, DenseNet-121, and VGG-16. For each pre-trained architecture, we tested different learning approaches, including the use of class weights (CW), the use of batch normalization and dropout layers, and the use of data augmentation to train the network. The CW ResNet-50 achieved an accuracy of 71.79% and a Mathews Correlation Coefficient (MCC) of 65.06%.

In current state-of-art studies, only supervised learning approaches were used to classify EGD images. On the other hand, in our work, we tested the use of unsupervised learning to increase classification performance. In particular, convolutional autoencoder architectures to extract representative features from unlabeled GI images and concatenated their outputs withs with the CW ResNet-50 architecture. We achieved an accuracy of 72.45% and an MCC of 65.08%.

CONTENT

ABBREVIATIONS

ACC Accuracy

AI Artificial Intelligence

BA Balanced accuracy

Bn Batch normalization

CAD Computer Aided Decision

CAMs Class activation maps

CNN Convolutional neural network

CW Class weight

DA Data augmentation

DL Deep Learning

DRL Deep reinforcement learning

EGC Early gastric cancer

EGD Esophagogastroduodenoscopy

ESGE European Society of Gastrointestinal Endoscopy

FN False negative

FP False positive

Fps Frames per second

F1 F1-score

GI Gastrointestinal

GIST Gastrointestinal Stromal Tumors

ILSVRC ImageNet Large Scale Visual Recognition Challenge

LMT Logistic Model Tree

MAPGI Modular Adaptive Preprocessing for Gastrointestinal tract Images

MCC Matthew’s Correlation Coefficient

ML Machine Learning

MLP Multilayer Perceptron

M-NBI Magnification endoscopy with narrow-band imaging

MSE Mean-squared error

NN Neural network

RC Recall

ReLU Rectified Linear Unit

RF Random Forest

RNN Recurrent neural network

CE Convolutional encoder

SGD Stochastic Gradient Descent

SPED *Sociedade Portuguesa de Endoscopia Digestiva*

SSS Systematic screening protocol for the stomach

VGG Visual Geometry Group

TN True negative

TP True positive

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

1.1. Motivation

According to the World Health Organization, in 2020, stomach cancer was the fifth most incident diagnosed cancer and the fourth leading cause of cancer deaths worldwide in both sexes and all ages [1]. In 2020, approximately 768 793 deaths of people with stomach cancer and 1 089 103 cases of stomach cancer were registered worldwide. It is estimated to have an increase of 66.3% of incident cases and an increase of 70.5% of deaths from 2020 to 2040 [1]. In 2020, stomach cancer was the fifth cancer with more cases in Portugal and at ages between 60 and 79 years (most common age of gastric cancer) the incidence rate was 56.5% and the mortality rate was 41.3% [1]. The incidence was about four times higher than the European average [2].

Worldwide, the survival rate of advanced stages of gastric cancer is only 5% - 25%. However, for early stages, the survival rate reaches 90%, meaning that to improve patient survival it is crucial to detect gastric cancer early [3] [4].

Most gastric cancer cases, precisely 70%, originate in developing countries, including Eastern Asia, Central and Eastern Europe, and South America. This translates into a fatality-to-case ratio of 63% and 68% in developed countries, and 75% and 81% in developing countries between men and women, respectively [5]. This shows the impact of modern medical technology.

The Esophagogastroduodenoscopy (EGD) is a diagnostic exam performed in real-time that aims to detect gastric cancer, which is performed by an endoscopist. The endoscopist is responsible to detect gastric cancer on the walls of the digestive system, preferably, at an early stage. However, there are several studies that show that endoscopists have difficulties in detecting early gastric cancer (EGC) [6] [7]. The EGC lesions are only subtle changes, that is why they are difficult to detect. In addition, endoscopists are affected by their subjective state, possibly leading them to miss important parts of the digestive system that could be crucial in EGC diagnosis. During the EGD, it is important to avoid blind spots, in order to map the entire digestive system.

The key to better gastric cancer management is to find strategies that help to detect cancer lesions in early stages, in order to increase the survival rate. It is crucial to empower health systems with better tools. In recent years, the use of artificial intelligence (AI) has been increasing to help EGC detection. The use of AI for computer-assisted EGD can potentially improve the EGD quality in order to have a complete examination of the digestive system.

1.2. Objective

Nowadays, a fundamental discussion of the endoscopy community is about the standard protocol creation that ensures the EGD exam quality. The digestive system mapping, one of the topics highly discussed in the execution of the protocols, is an EGD quality indicator of great importance. The creation of digestive system photodocumentation during the exam is essential for endoscopists to guarantee that all areas of the upper gastrointestinal (GI) system are examined.

Deep Learning (DL) has been widely studied for EGD improvement. Some preliminary works have demonstrated that Convolutional Neural Networks (CNNs) have the potential to recognize upper GI landmarks, which is the focus of this work.

The aim of this work consists in test new automatic algorithms, specifically CNN-based systems able to detect the presence of blind spots during EGD to increase the quality of endoscopic exams.

1.3. Thesis organization

Accordingly, this work has six chapters, the present chapter explains in a succinct way the motivation and the objective of this work. The second chapter aims to explain the main concepts for a better understanding of this work. This chapter explains the gastric cancer and respective diagnostic exam (EGD) and shows the system digestive anatomy for better comprehension of the gastric cancer and EGD exam. The main gaps and challenges of the EGD exam will be explained in this chapter. Still in this chapter, we will be introducing the main and essential DL concepts. In the third chapter, the state-of-art associated with the work objective will be presented. In particular, DL studies focusing on the improvement of EGD will be presented, with a special focus on the monitoring of upper GI landmarks studies.

In chapter four, we will explain in more detail the novel methodology that will be pursued during the thesis work. This chapter will explain the design of our experiments, specifically the used datasets and the details of our architectures. In chapter five will describe the experimental methodology, the performance metrics, and the results of our architectures. Finally, in the last chapter, we will explain the main conclusions of our work.

CHAPTER 2 BACKGROUND

2.1. Gastric Cancer

Gastric cancer is the fifth most incident cancer in the world and is more common in people older than 55, in particular between 60 years old and 70 years old [1] [8]. The researcher's community defends that the gastric cancer origin is in digestive system lesions that progressed to gastric cancer.

The upper GI anatomy is shown in Figure 2.1. The stomach is formed by mucosa, submucosa, muscle layers and serosa. The mucosa consists of epithelium, lamina propria and muscularis mucosa. The muscle layers consist of circular, longitudinal, and oblique muscles. The serosa is the membrane that lines the outside of the stomach [9].

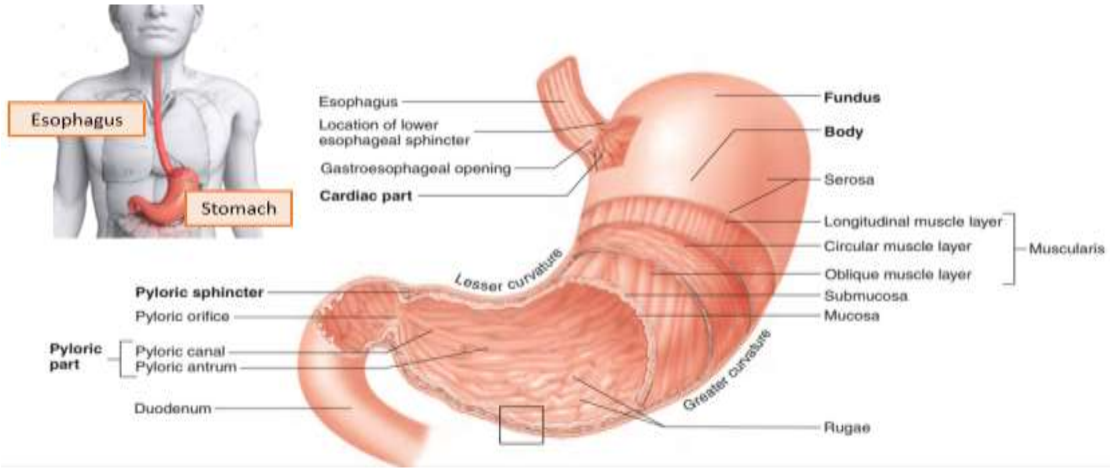


Figure 2. 1 *Upper GI anatomy. (Adapted from* [10]*)*

There are many types of gastric cancer, such as, adenocarcinoma, GIST (Gastrointestinal Stromal Tumors), leiomyomas, neuroendocrine tumors, or lymphomas. The most common is adenocarcinoma, corresponding to 90% of gastric cancers, precisely [2] [9]. Gastric cancer is developed in tissues that revest the stomach. The adenocarcinoma cancer starts to develop in the inner layer of the stomach, the mucosa [9]. The GIST tumors are rare tumors, which are believed to come from Cajal interstitial cells. These cells are in the stomach wall. Gastric lymphomas cancer is from immune cells of the stomach. The neuroendocrine tumors are from nerve or endocrine cells of the stomach [9].

Until now, it is not known why stomach cancer occurs. However, there are various variants/factors of risk that could help to develop gastric cancer. *Helicobacter pylori* infection is the main and, at the same time, one of the most treatable factors of risk for stomach cancer [9]. *Helicobacter pylori* infection affects over 2 billion people worldwide, however, only less than 1% of the people infected by this bacterium develop gastric adenocarcinoma [2] [5] [11]. *Helicobacter pylori* bacterium is transmitted by feces and saliva, the people most affected live with poor socioeconomic conditions. This bacterium could reside in the stomach and cause chronic inflammation or gastric ulcers. This infection could progress to cancer when bacteria persists in the stomach. However, this infection, before progressing to cancer, will pass to various precancerous stages, which can be detected and treated, preventing the cancer's appearance. The treatments of bacteria consist of taking antibiotics.

The salt has been associated with increased incidence and mortality rates of gastric cancer. The presence of salt helps the *H. pylori* infection to occur and damages the mucosa of the stomach that contributes directly to the development of stomach cancer. The excessive consumption of foods with nitrates can increase the risk of developing stomach cancer as well. Smoking and drinking alcohol are also associated with an increase in gastric cancer rates. The rate of stomach cancer in smokers is two times higher. On the other hand, fruit, vegetables and physical activity are observed to be protective [9] [5]. Genetic inheritance is a factor which increases the risk of developing gastric cancer, responsible for the appearance of 8-10% of gastric cancer [2]. The hereditary mutations may increase the risk of developing stomach cancer.

In the initial phase, most stomach cancers do not cause any symptoms. Therefore, there are many times that stomach cancer is not suspected, which leads to detecting the stomach cancer in an advanced stage. The symptoms could be: - abdominal discomfort or pain; - feeling of fullness after eating a light meal; - heartburn, indigestion and belching; - nausea and/or vomiting, especially if this includes blood; - accumulation of fluids in the abdomen; - lack of appetite; - weight loss [9].

2.2. Esophagogastroduodenoscopy (EGD)

The Esophagogastroduodenoscopy (EGD) is the pivotal procedure in the diagnosis of gastric cancer. The EGD consists in introducing an endoscope, that is a thin, flexible, and illuminated tube. The EGDs are real-time examinations. The endoscope is introduced through the throat to the stomach, thus allowing to observe the lining of the esophagus, the stomach and the first part of the small intestine (upper GI areas). If the EGD operator detects abnormal areas, biopsies can be performed using instruments inserted through the endoscope. Then, biopsies are examined by a specialist [9].

Before starting the exam, it is necessary to prepare the patients. In an ideal preparation [12], the patient should drink a mixture of water with mucolytic and defoaming agents for minimizing time and effort to remove mucus and froth from the mucosal surface during the procedure. One the other hand, to inhibit peristalsis in order to detect subtle mucosal changes, an antiperistaltic is administered to the patient [12]. In the days before the exam, the patient should have restrictions on eating. The patient should not eat solid foods in the last 6 to 8 hours before the exam. It is possible to perform the EGD with anesthesia/sedation, which reduces the discomfort associated with exam [13] [14].

When the endoscopist inserts the scope, he starts to see if the patient shows risk factors in the mucosa surface, such as gastritis associated with *H. pylori* bacterium, gastric atrophy, or intestinal metaplasia. Gastritis, gastric atrophy, and intestinal metaplasia are precancerous lesions that when untreated, could develop cancer. So, if the patient does not have any of these risk factors/lesions, it means that lesions suspicious for gastric cancer are less [12]. During EGD, it is recommended to pay special attention to areas associated with an increased cancer risk or regions where pathologies are frequently missed, such as upper esophagus, gastroesophageal junction, 3 o'clock segment in Barrett esophagus (right esophageal hemisphere) and lower esophagus [15]. Magnified EGD helps to find possible risk factors. Advanced imaging techniques can help to detect subtle changes in the mucosa, such as the use of acetic acid, Lugol's iodine, and narrow-band imaging. However, the EGC lesions are difficult to recognize for endoscopists because the mucosa often shows only subtle changes [4] [16] [15].

Narrow-band imaging systems (M-NBI) are widely used in order to evaluate gastric lesions. Magnification endoscopy with M-NBI is an advanced technology that uses enhanced contrast between vessels and mucosal surfaces to illustrate abnormal vascellum morphologies. However, this system has some disadvantages, such as a lack of an objective evaluation metric for the diagnosis and the difficulty in mastering this diagnosis technique. The M-NBI is an optical digital imaging technique that applies reflection of dual narrow-band wavelengths, which are 415 nm (blue) and 540 nm (green) [17]. The standard image of endoscopy is white light, it is a good standard as a starting point. However, with electronic image enhancement, the M-NBI could detect more information [15]. In Japan, enhanced imaging techniques are usually employed to improve diagnostic accuracy [15].

M-NBI is useful to characterize known lesions at a close view, however, at a distant view, it is not useful for EGC detecting. However, the liked color imaging (LCI) is a pre-processing and a post-processing technology that has sufficient brightness to illuminate a wide lumen, detecting EGC as orange-red [18].

2.2.1. EGD photodocumentation

The EGD is an optimal diagnostic exam to detect gastric cancer and associated lesions, which is essential in clinical routine to preserve the life quality of patients. However, this diagnostic exam has some gaps that are important to address in order to save more patients. The EGD quality varies significantly because of cognitive and technical factors. There are discrepancies between endoscopists' performance, which varies according to their knowledge and experience. In addition, early lesions are difficult to detect because they are unrecognizable superficial lesions. For these reasons, the endoscopists must be well trained and armed with adequate knowledge to have a good performance during the EGD [4].

In Japan, there is a systematic training program focused on the detection of subtle mucosal changes [15]. Japan is a world leader in high quality EGD diagnostic and the clinical routine [15] [6].

In Western countries, there are many cases that gastric cancer is diagnosed when it is already at an advanced stage [6]. These late detections can be explained by cancers that may be missed at the time of endoscopy. A study performed in Western countries showed that in patients that realized EGD in an antecedent year, EGC lesions were not detected in 7.2% of cases that had gastric or esophageal cancer within the next year. Among these, 73% of cases arose from endoscopists errors. The study showed that one of the main causes of the non-detection of EGC lesions is due the endoscopists not detecting the lesions, more precisely 27% of all wrong diagnosis [6].

There is a significant fraction of patients who only know that they have gastric cancer when it is already in an advanced stage because the endoscopists may miss some relevant areas during EGD. During the EGD, it is important to avoid blind spots, to map the entire upper GI system. In order to avoid blind spots, different procedures have been proposed by different countries. There are some protocols, but some of these protocols are not practicable in a clinical environment, depending on the development of the country and available materials in each hospital. There is not a standardized protocol to map the entire GI system that is followed worldwide. However, it is crucial to develop reliable and feasible guidelines, where endoscopists performance measurements are registered, identifying services and individual endoscopists with lower levels of performance. The photodocumentation of all normal anatomical landmarks and abnormal findings is an indicator of a complete examination that should be included in performance measurements [19].

Several important endoscopic societies have launched many protocols. In 2001, the European Society of Gastrointestinal Endoscopy (ESGE) presented the first EGD photodocumentation proposal, which includes 8 upper GI landmarks (2 esophageal, 4 gastric and 2 duodenal). In the esophagus, the landmarks were the proximal esophagus (taken 20 cm from the incisor) and the z-line (taken 2 cm above). The gastric landmarks were the cardia and fundus on retroflexed view, the body (taken from the upper part of the lesser curvature), the angulus on partial retroflexion and the antrum. The duodenal landmarks were the duodenal bulb and the second part of the duodenum (taken near the ampulla) [20].

It was concluded that the endoscopists that take more than four pictures and take longer time to execute EGD have a better performance than endoscopists with taking less pictures in shorter executing time. This suggests that collecting more pictures and taking longer times during EGD may improve the detection of EGC. From this point of view, in 2013, the Systematic Screening Protocol for the Stomach (SSS) was proposed by the Japanese Society of Gastroenterological Cancer Screening and consists of to take 22 endoscopic stomach photos, which are to take of 4 or 3 quadrant views. The proposed SSS is shown in Figure 2.2 [12] [15] [21]. The minimum total procedure time recommendation is 8 minutes: 2 minutes to clean the walls, 4 minutes to gastric examination (according to the SSS protocol) and 2 minutes to esophageal examination [15].

In 2016, the ESGE reformulated its guidelines and added to the 8 landmarks 2 more landmarks, 1 esophageal and 1 gastric. The added esophageal landmark was the distal esophagus and the added gastric landmark was the body on greater curvature [20].

In Portugal, the ESGE guidelines are followed, meaning that a total of 10 photos are taken during the EGD exam (see Figure 2.3). The EGD landmarks followed in Portugal are: proximal esophagus; distal esophagus; z-line and diaphragm indentation; cardia and fundus on retroflexed view; body (including lesser curvature); body on retroflexed view; angulus on partial retroflexion; antrum; duodenal bulb; second part of the duodenum, including the ampulla [20].

According to the *Sociedade Portuguesa de Endoscopia Digestiva* (SPED), in order to validate the EGD quality, it is important to take photos during the exam corresponding to some anatomical locations (landmarks) and all abnormal walls of the digestive system [22]. In addition, the exam duration is also an EGD quality indicator, which should be at least 7 minutes. In fact, endoscopists who take more than 7 minutes can detect more lesions [23].

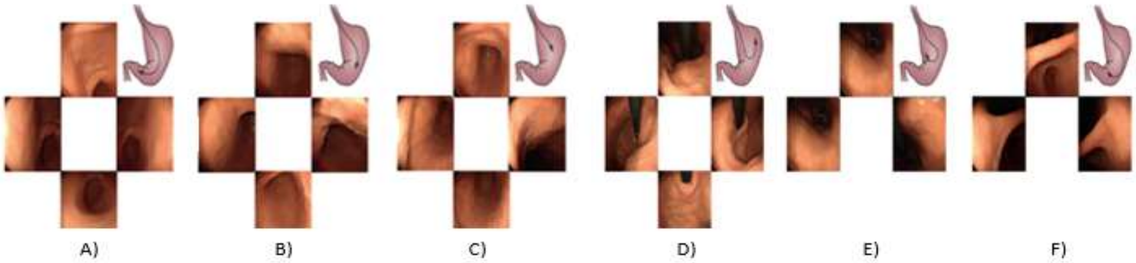


Figure 2. 2 *Endoscopic photos proposed by SSS protocol. A), B) and C) represent the photos in the antegrade view (view in the normal direction, along the digestive system – direction arrows). D), E) and F) represent the photos in the retroflex view (view in opposite direction of the system digestive – direction arrows). A), B), C) and D) Photos taken in 4 quadrants (anterior wall, greater curvature, lesser curvature, posterior wall). E) and F) Photos taken in 3 quadrants (anterior wall, lesser curvature and posterior wall). A) Photos in antrum. B) Photos in the lower body. C) Photos in the middle-upper body. D) Photos in fundus-cardia. E) Photos in the middle-upper body. F) Photos in incisura. (Adapted from* [15]*).*

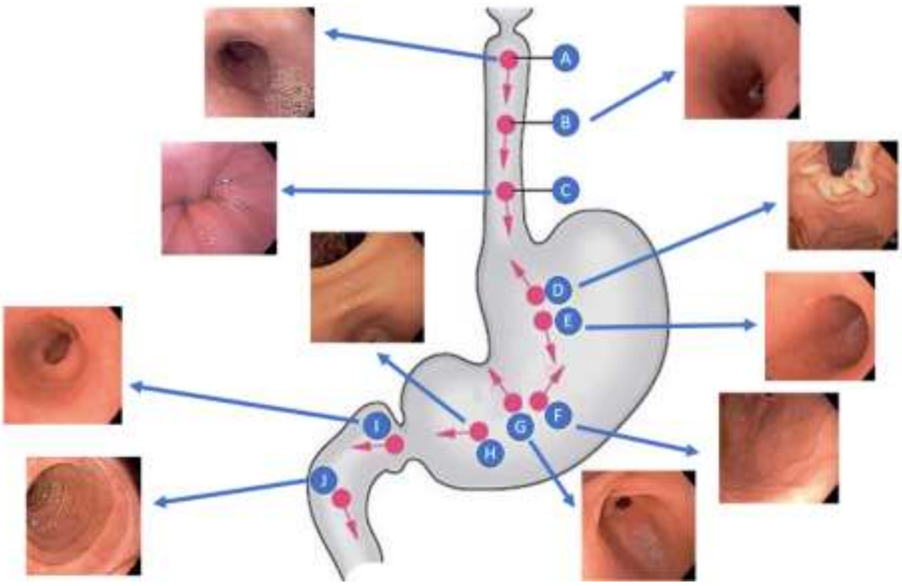


Figure 2. 3 *Suggested anatomical landmarks by Departments of gastroenterology of Hospital Egas Moniz, Lisbon and Instituto Português de Oncologia, Porto. A) Proximal esophagus; B) Distal esophagus; C) Z-line and diaphragm indentation; D) Fundus-cardia on retroflexed view; E) Body (including lesser curvature); F) Body on retroflexed view; G) Angulus on partial retroflexion; H) Antrum; I) Duodenal bulb; J) Second part of the duodenum, including the ampulla. (Adapted from* [20]*)*

The photodocumentation during the EGD exam is crucial to observe the mucosa extension and the anatomical landmarks are crucial to not have missed areas during the EGD exam. There are discrepancies among endoscopists in their performance in detecting EGC. The endoscopists must follow a specific training to recognize early cancerous lesions [4] [16] [15]. In the near future, it is necessary to create standardized protocols and develop learning systems practicable by worldwide endoscopists to reduce deaths due EGC detection errors [4] [12] [16].

In the last decades, the endoscopic technology has undergone remarkable advances in favor of detecting gastric cancer in early stages. High-quality endoscopy delivers better health outcomes and has seen an explosion of interest to apply AI in endoscopy with the objective to mitigate the endoscopists' skill variations and to improve the EGD. Recent studies successfully used AI in the field of endoscopy [4] that have the objective to improve exam quality through the detection of upper GI landmarks to make better monitoring, avoiding missed anatomical zones. The medical community came to the conclusion that computer-assisted diagnosis will help and support a medical decision preventing possible errors and improving the health quality.

2.3. Deep Learning

In recent years, the scientific community has found good results with AI applications in clinical cancer research. AI systems process data to support meaningful decision-making [24] and can be used to process and analyze multifactor data from multiple patients and thus predict the cancer appearance.

The concept of AI first emerged in 1956, whose objective was to build machines that execute tasks with the same logical reasoning as the human [25]. Since then, AI has been the subject of several investigations, namely Machine Learning (ML) and Deep Learning (DL). ML and DL were first proposed in the 1980s and 2010s, respectively [25]. DL is a subfield of ML which attempts to learn high-level abstractions in data by utilizing hierarchical architectures [26]. To produce an accurate model, DL models usually require a massive amount of training data [25] [26] and powerful hardware for training, preferably with GPUs [25].

Supervised learning is a specific area of ML, where the majority of DL solutions operate, which consists in learning how to make decisions and to perform specific tasks through learning functions from the observation of labeled examples. In recent years, DL has been widely used in image processing tasks, such as, classification, segmentation, and reconstruction [26]. The classification consists in producing a label given an image and is very useful in biomedical applications. Image classification can be the core of computer-aided decision (CAD) systems which help the human operator in taking decisions, reducing the number of unnecessary exams and the number of missed detections. The objective of segmentation is partitioning an image in multiple segments (sets of pixels), it is similar with classification, but the label is assigned to each pixel of the image. The reconstruction is a regression problem and has the objective to recover a full image of interest from partial measurement. Through reconstruction, it is also possible to increase the resolution of an acquired image and reduce the impact of reconstruction artifacts.

DL refers to a class of learning algorithms, which are based on a specific kind of classifiers, that are neural networks (NNs). The DL network is formed by multiple layers of neurons. The architecture of a NN is shown in Figure 2.4. In Figure 2.4, the leftmost layer is called the input layer and the respective neurons are called input neurons. The rightmost is called the output layer with the respective output neurons (in this case, it is only one output neuron). The middle layers are called hidden layers (in this case, there are two hidden layers) [27]. One neuron can perform a simple decision, whereas multiple connected one of them can make more complex decisions.

The neurons of multi layers have individual parameters, such as weights (associated with inputs), activation functions and decision thresholds. The NN of the kind shown in Figure 2.4 is also called multilayer perceptrons (MLPs), in which perceptrons are neurons with the respective learning parameters [28]. Each perceptron receives one or more inputs, which then are weighted sums to produce an output. The sum is passed through a non-linear function, called activation function. The Rectified Linear Unit (ReLU) is an activation function that consists in rectifying the positive values and removing negative parts (see Figure 2.5) [29]. In Figure 2.5 also shows the sigmoid activation function, it is used for 2-label classification tasks. When $x < -10$, the output of the sigmoid function is close to null and when $x > 10$, the output is near 1 [29]. On other hand, the softmax activation function is used for multi-classification tasks (see Figure 2.5). The output of the softmax function is the probability of the object to belong to each possible outcome and the target class will be the class with the highest probability. The probability range is 0 to 1 and the sum of all probabilities is equal to 1 [30] [31].

The weights and decisions thresholds/bias of the connections are adapted during the learning/training phase. The learning phase consists of training the NN to classify what we want based on a training dataset for which we know the correct class. The objective of the training phase is getting an optimal solution, which is measured through the loss function. This function quantifies how well the model finds the optimal weights and thresholds, and when such weights and thresholds are obtained, the loss function should be the lowest possible. The loss function evaluates the performance of our model through the calculation of error of the prediction between the actual class and the predicted value [24] [26] [32].

The process of minimizing the loss function is performed using the backpropagation algorithm. The NNs are initialized with aleatory weights and are adjusted with backpropagation based on error rate obtained in the previous iteration. The artificial neurons predict the outputs these outputs are used together with the true labels of the training data to compute the loss function. Next, the loss function gradient is calculated through the chain rule, which uses the actual network parameters. Then, the parameters (weights and thresholds) are updated to reduce the error and prepared for the next output prediction. The network learns by repeating this process, until the error is reduced to an acceptable level, and the model adequately interprets the input data. The model learns with consecutive forward and backward iterations. The forward stage is the current output predict/current weights and bias in each layer [24] [26] [32].

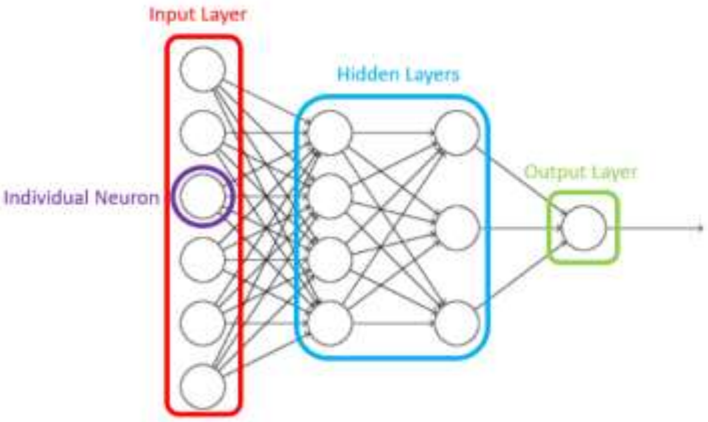


Figure 2. 4 *Architecture of neural network (NN). (Adapted from* [27]*)*

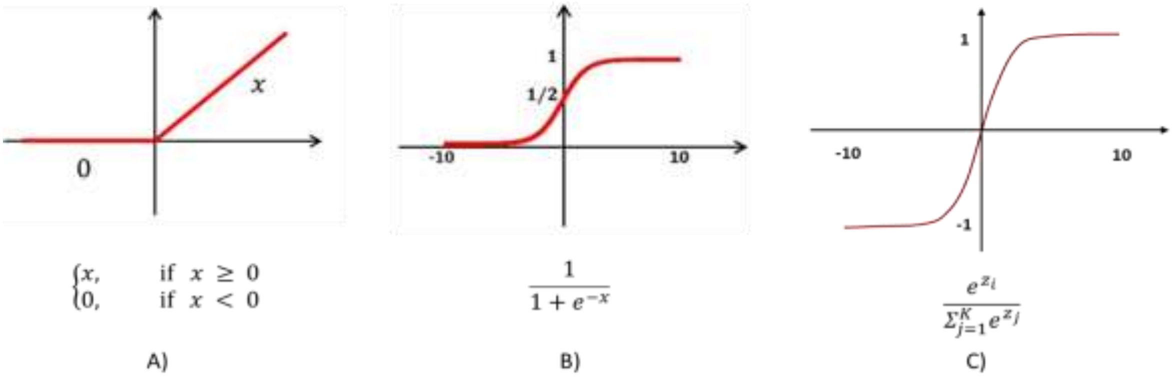


Figure 2. 5 *Activation functions. A) Rectified Linear Unit (ReLU) activation function with respective equation. B) Sigmoid activation function with respective sigmoid curve and equation. C) Softmax activation function with respective curve and equation, where z_i are the values of the input vector, the summation represents the normalization term, and the K represents the number of classes* [31]*. This activation function converts a vector of values to a probability distribution. (Adapted from* [29]*)*

The backpropagation represents the algorithm used to compute the gradient of the loss function with respect to the network parameters using the chain rule, Figure 2.6. The gradient descent is influenced by learning rate, which is a tuning parameter that determines the step size at each iteration. The learning rate influences the convergence of the algorithm. When the learning rate is higher, we have the risk of overshooting the minima. One the other hand, when the learning rate is smaller, learning rates will consume more time to reach the lowest point [27] [33].

The gradient descent uses all the samples of the training set to do a single update for a parameter in an iteration. However, when the training set has a big size, the gradient descendent algorithm is not efficient, making the convergence of the algorithm slow. Through Stochastic Gradient Descent (SGD), the process could be accelerated since this optimization method uses a randomly subset of training samples (mini-batch) from the training set to do an update for a parameter in an iteration [27].

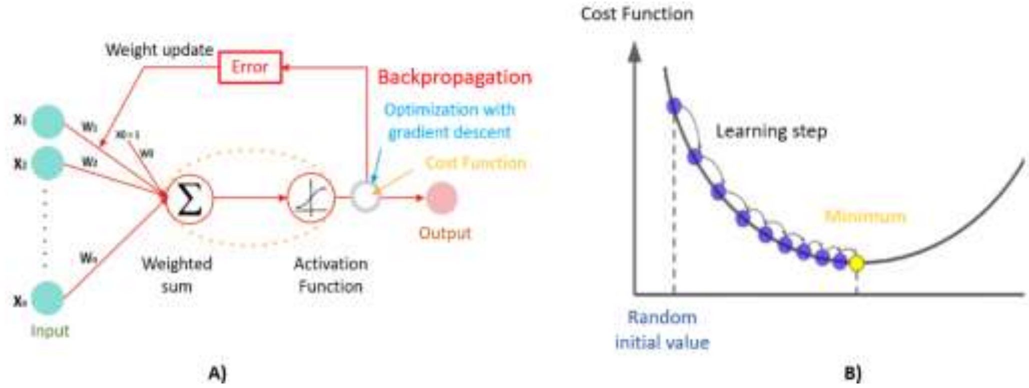


Figure 2. 6 *A) Representative scheme of backpropagation. B) Gradient descent. (Adapted from* [34] [35]*)*

Sometimes, the NNs have a poor performance that may be due to underfitting or overfitting the data. Overfitting happens when the model learns the detail and the noise of training data and then, when a new data is used in the model, the impact of this noise or random fluctuations is felt. The model learns wrongly the noise of training data, and this has a negative impact on performance of the model. One the other hand, underfitting happens when the model cannot capture the data behavior and consequently the model is not able to correctly fit the desired classification/regression function. Generically, overfitting has a good performance in training data, but a poor generalization to new data, while underfitting has a poor performance on the training data and in generalization of the other data [36].

NNs have good performances in classifying features extracted from raw data. However, to directly classify images with NNs, it is necessary to convert the bidimensional image into a unidimensional vector before training the model. So, the parameter number increases, and the model becomes inefficient. CNNs overcome these limitations by extracting automatically relevant features from the images using convolutional operations. In CNNs, the exhaustive weight multiplications are replaced by convolution filters to reduce the number of parameters. In addition, the optimal features/filters of CNNs are learnt automatically during the training stage. In this sense, CNNs require a much lower pre-processing compared with NNs, NNs filters are hand-engineered and trained [37].

2.4. Convolutional Neural Networks

The Convolutional Neural Networks (CNNs) is one of the DL methods with the greatest potential. CNN models allow us to reduce the number of parameters to learn, they are more efficient than dense multiplication. CNNs are specially thought to extract features from images without losing important content [26] [38].

CNNs have three main types of neural layers, which are convolutional layers, pooling layers, and fully connected layers. The schematic CNN architecture is shown in Figure 2.7.

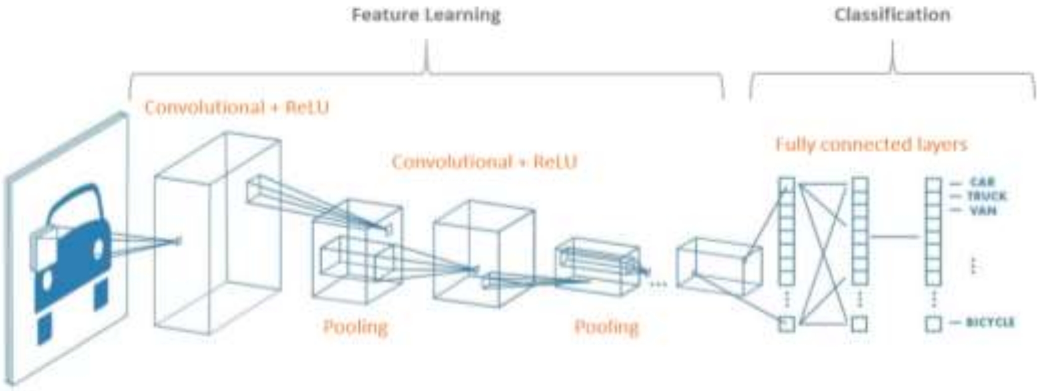


Figure 2. 7 *CNN architecture diagram. (Adapted from* [39]*)*

Convolutional layers utilize various kernels to convolve the whole image, generating various feature maps. The kernel filter is a matrix that moves over the image to extract the features from the image through dot product between sub-regions of the image and the kernel matrix. In Figure 2.8 is represented the convolutional operation between a sub-region of the input image and a kernel filter. In a feature map, the weight sharing mechanism reduces the number of the parameters, being an advantage by reducing memory requirements. Furthermore, the convolutional layers are equivariant to translation due to the weight sharing mechanism, which means that the convolutional networks are able to generalize the object in different locations to be useful for classification [26] [40] [41].

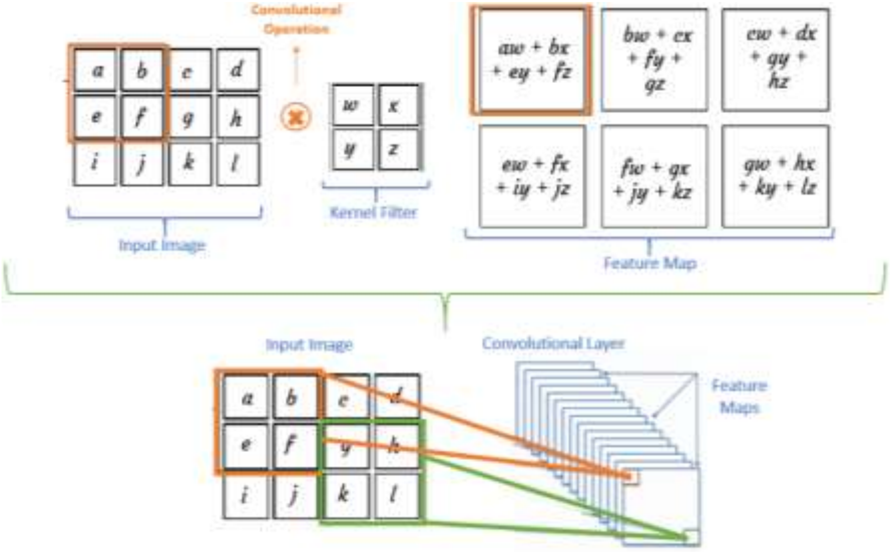


Figure 2. 8 *Diagram of the convolutional operation between input image and kernel filter and representation of many feature maps (convolutional layer).*

Pooling layers are usually interleaved between convolutional layers. Pooling layers reduce the number of parameters of the network and feature maps dimension. In addition, pooling layers only extract dominant features (rotational and positional invariant features). Max pooling and average pooling are the most used strategies to reduce dimensions of the feature maps to increase the computational efficiency (see Figure 2.9). Max pooling returns the maximum value from the sub-region of the image covered by the kernel filter, while the average pooling returns the average value from the sub-region of the image covered by the kernel filter [39]. Max polling is also a noise suppressant [39].

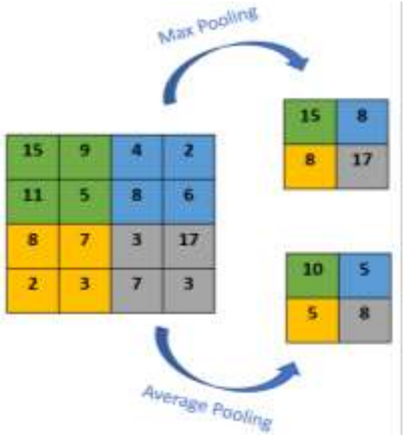


Figure 2. 9 *Diagram representing max pooling and average pooling.*

Fully connected layers follow the last pooling layer, and there are numerous fully connected layers converting 2D feature maps to 1D feature vector, as shown in Figure 2.7. These layers require a large computational effort because the fully connected layers operate as the MLPs, thus requiring more parameters.

2.4.1. CNN architectures

In this section, we will describe the CNN models used for gastric landmark detection, which were usually pre-trained with the ImageNet repository. In ImageNet Large Scale Visual Recognition Challenge (ILSVRC), CNNs models have achieved top accuracy scores and have been extensively used by researcher's community [26]. The ILSVRC is an annual computer vision competition, which began in 2010 and is a benchmark for object detection and classification of computer vision models [24]. The ILSVRC uses the ImageNet repository, which is a large image database designed to use in visual object recognition software research, which has 14 million labelled images in 1 000 classes [24] [26] [42] [43].

2.4.1.1. VGG architecture

The VGG was created by the Visual Geometry Group (VGG) from Oxford University. There are several VGG architectures, depending on the layers number and consequently the parameters number. The first VGG architecture was VGG-11, which had an error rate of 10.4% with ImageNet repository. Then, to improve VGG-11 performance batch normalization is used, creating the VGG-11-bn [44]. The batch normalization allows to make the model more stable and faster through normalization of the input of network layers [45]. However, more VGG architectures were emerging with better performances, namely the VGG-16.

In the 2014 ILSVRC challenge, the VGG-16 model won the 1st place to detect objects within an image of 200 classes and won the 2nd place to classify images with 1 000 categories [46]. VGG-16 is one of the architectures of ILSVRC top competitors [43]. VGG-16 is a simple and uniform convolutional neural network architecture [42]. This model is often adopted by the scientific community because the pre-trained weights are freely available online, allowing the fine-tuning of this powerful model on new tasks [24]. However, VGG-16 consists of 138 million parameters [42], which can be challenging to train and is a model that occupies a large space in disk (about 528 MB to train ImageNet weights) [46].

VGG-16 consists of 16 convolutional layers [42] (see Figure 2.10). The input is an RGB image with a fixed side of 224×224 pixel. The image is passed through blocks of convolutional layers, where some convolutional layers are followed by max pooling layers. Three fully connected layers follow a stack of convolutional layers, which have different depths in different architectures, according to the new task. In Figure 2.10, the last fully connected layer has 1 000 channels (one for each class of the ILSVRC classification), in this case. All hidden layers have ReLU activation function [43] [47]. To the end, VGG-16 has a softmax layer. The softmax layers assign probabilities to each class and have the same number of nodes as the output layer [48].

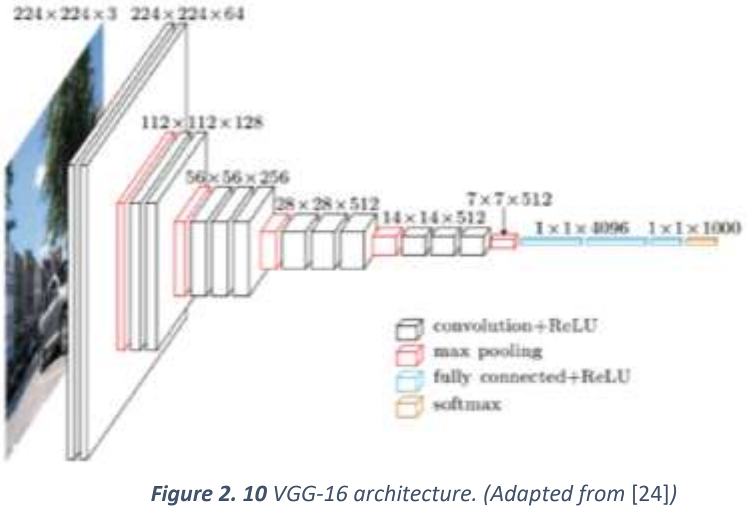


***Figure 2. 10** VGG-16 architecture. (Adapted from* [24]*)*

2.4.1.2. GoogLeNet architecture

The GoogLeNet architecture won the 1st place to classify images in the 2014 ILSVRC challenge [42]. GoogLeNet architecture has 22 layers (27, including the pooling layers) [49]. This architecture uses 9 inception modules (see Figure 2.11 and Figure 2.12) [50]. The idea of inception modules is to allow a more efficient computation and solve overfitting problems through a dimensionality reduction [51]. Instead of stacking multiple operations, the inception modules that characterize the GoogLeNet architecture operate at the same level, thus improving the performance. The modules consist in operating parallelly with multiple kernel filters sizes, which involves a convolution on input with not one, but three different sizes of filters (1×1, 3×3, 5×5) along with max pooling. Then, the outputs of each filter are concatenated and sent to the next layer. To improve this process, GoogLeNet modules have an extra 1×1 convolutional before the 3×3 and 5×5 convolutions and after the max pooling, which is called inception module with dimension reductions (see Figure 2.12) [51]. GoogLeNet uses global average pooling at the end of the last inception module [49].

The strength of this network is the computational efficiency and practicality. This model runs in devices with low-memory [52]. The independent building blocks used for the construction of the network are about 100, but with parallel operations, the model has only 22 layers deep (counting only the layers with parameters) [52].

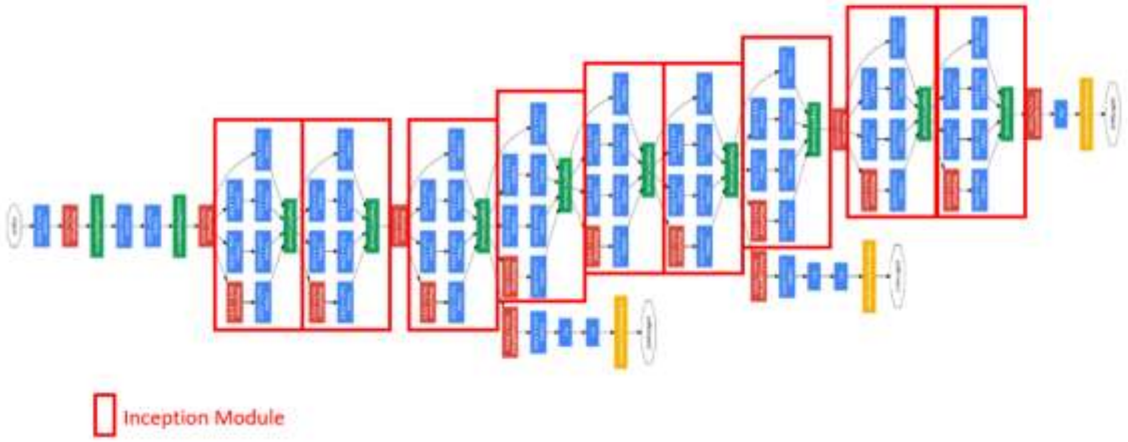


***Figure 2. 11** GoogLeNet architecture. (Adapted from* [52]*)*

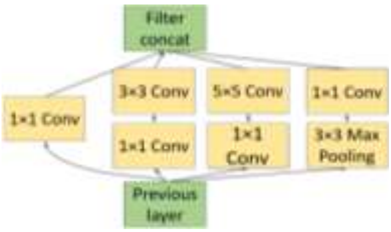


***Figure 2. 12** Inception module. (Adapted from* [52]*)*

GoogLeNet is also known by Inception-v1 and there are also more three versions (Inception-v2, Inception-v3 and Inception-v4). The Inception-v2 characterized by the introduction of batch normalization. Then, factoring convolutions were added creating the Inception-v3. The advantage of factoring convolutions is that of reducing the number of parameters and, therefore, overfitting without decreasing the network efficacy [53] [54]. For example, the 5×5 convolution in inception module of Inception-v1 architecture was replaced by two 3×3 convolutions, 5×5 convolution has 25 parameters while 3×3+3×3 have 18 parameters, which means that the number of parameters was reduced 28%. The Inception-v3 architecture uses three different inception modules (Module A, Module B and Module C), where the number of parameters was reduced with convolution factorizations [54]. In Figure 2.13 is shown the Module A, Module B and Module C. In Figure 2.14 is shown the Inception-v3 architecture.

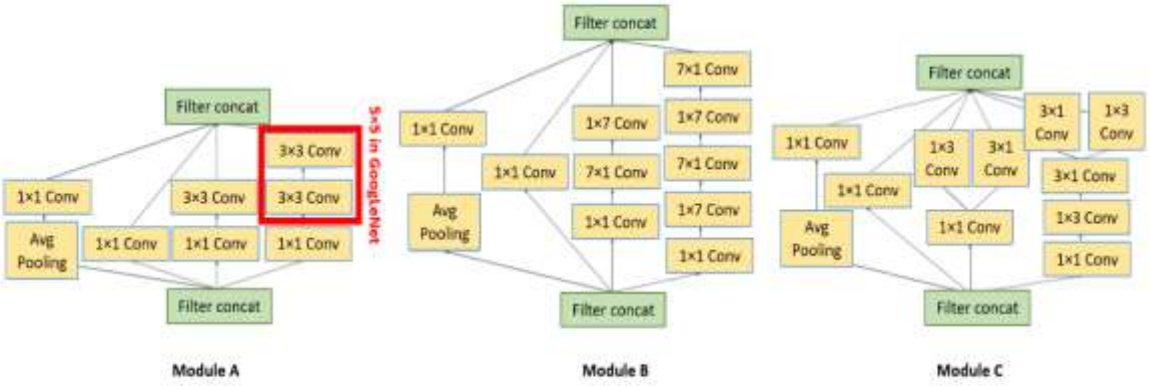


Figure 2. 13 *Inception modules (Module A, Module B, Module C).*

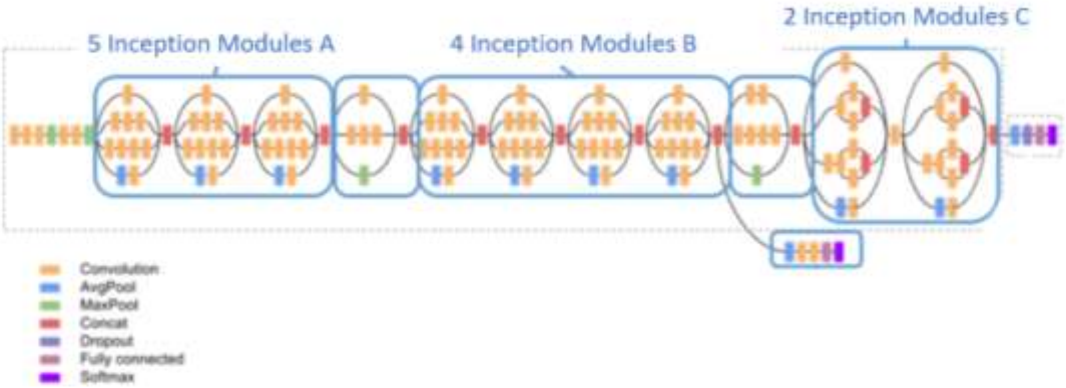


Figure 2. 14 *Inception-v3 architecture. (Adapted from* [54]*)*

The Inception-v4 is more uniform and has more inception modules than Inception-v3. The Inception-v4 also uses the techniques from Inception-v1 to Inception-v3, such as the batch normalization. The Inception-v4 also has the Module A, Module B and Module C, and has another module called stem [53]. The Inception-v4 is shown in Figure 2.15.

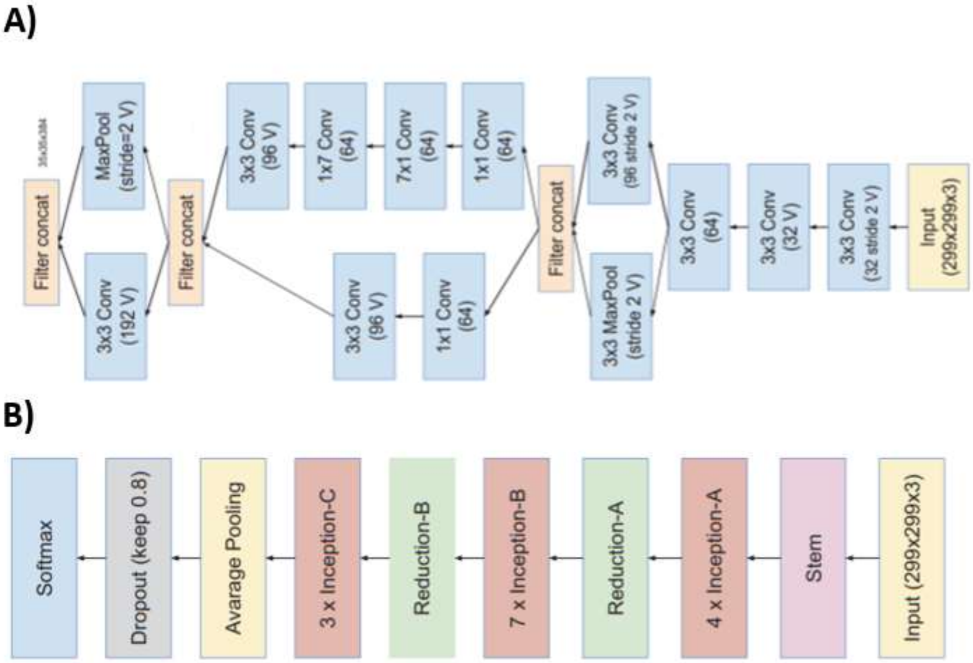


Figure 2. 15 *A) Inception-v4 stem block. B) Inception-v4 architecture* [55].

2.4.1.3. ResNet architecture

ResNet is a very powerful network that won 1st place in the ILSVR 2015 classification task [56]. There are many types of ResNet architectures, depending on the number of layers. ResNet-50 has 50 layers and is the one of the most used ResNet models.

Since 2013, the DL community started to build deeper networks because they were able to achieve high robustness and better performances. However, it was observed that when more layers were added to the network, the accuracy value saturated or decreased abruptly. The residual block is used in the ResNet network to improve the accuracy with increased number of layers [57]. Residual blocks consist in connecting the output layer of one layer with the input of an earlier layer (skip connections). The idea is that the network learns the residual mapping instead of learning the maps of underlying layers. In Figure 2.16, the residual block is shown, where the model instead of learning a direct mapping, uses the difference between a mapping applied to x, $M(x)$, and the original input, x. The residual function is $F(x) = M(x) - x$ [58].

ResNet-50 has four stages, which are shown in Figure 2.17. The first layers of the ResNet-50 are convolutional layers and max pooling layers, with kernel sizes 7×7 and 3×3, respectively. Each stage has several residual blocks, and each block has three convolutional layers. As we progress from one stage to another, the channel width is doubled and the size of the input is reduced to half [57] [59].

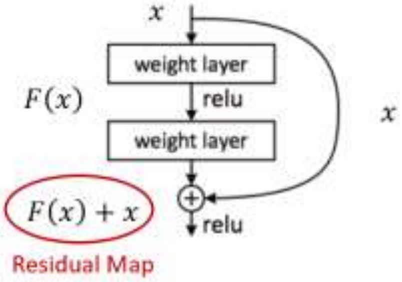


Figure 2. 16 *Residual block. (Adapted from* [58]*)*

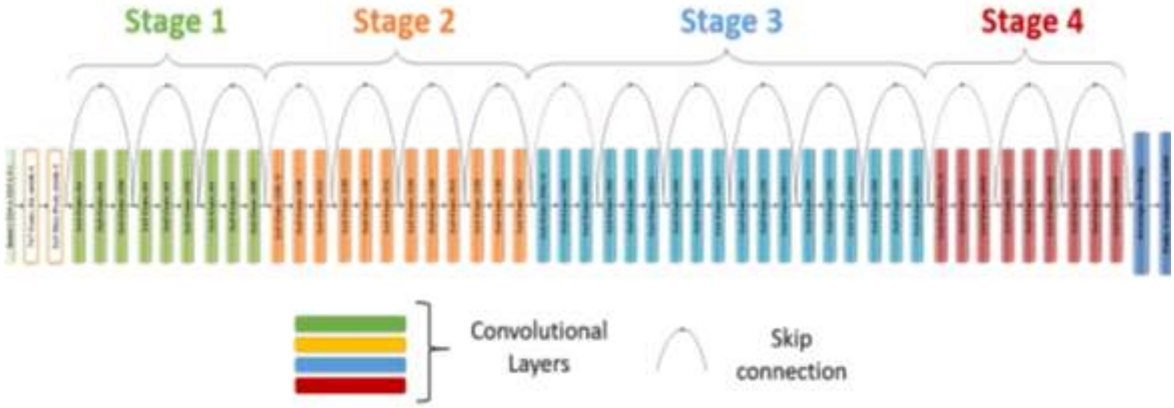


Figure 2. 17 *ResNet-50 architecture* [57].

2.4.1.4. DenseNet architecture

The DenseNet (Densely Connected Convolutional Network) architecture was proposed in 2017 at the Conference on Computer Vision and Pattern Recognition [60].

The DenseNet architecture can achieve good accuracy with fewer parameters compared with the ResNet architecture. The DenseNet architecture consists of a set of dense blocks, where each layer receives inputs from all preceding layers and passes its own feature map to all subsequent layers. These connections make the network thinner and more compact, reducing the number of channels and increasing the computational efficiency [61]. There are several DenseNet architectures, which vary according to the number of layers in dense blocks [62]. In Figure 2.18 is shown a dense block and the DenseNet-121 architecture.

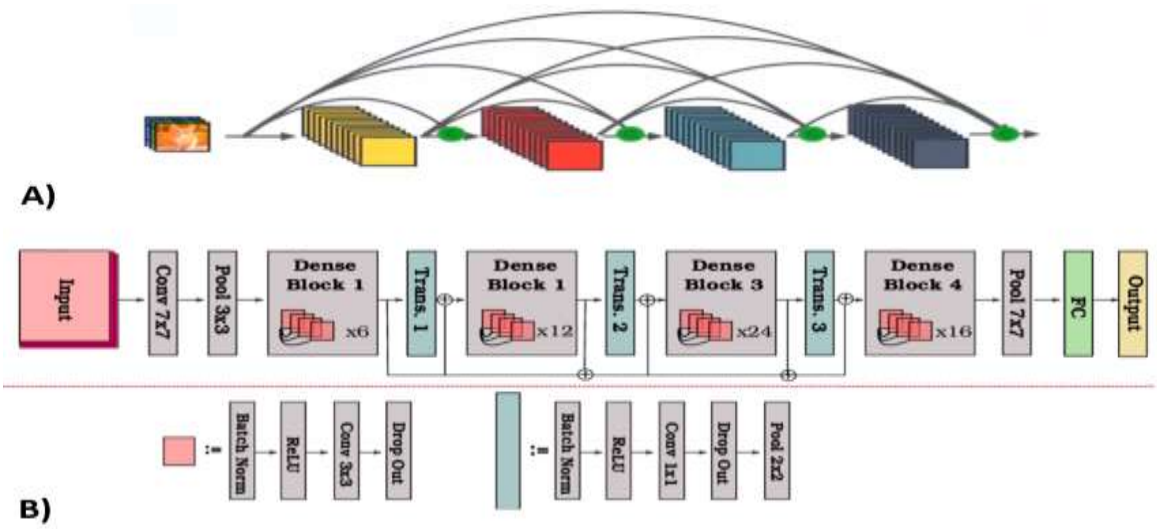


***Figure 2. 18** A) DenseNet block. B) DenseNet-121 architecture. (Adapted from* [61] [63]*)*

2.4.1.5. NASNet architecture

The NASNet architecture was developed by Zoph et al. in 2017 [64]. The NASNet architecture is constituted by normal cells and reduction cells (see Figure 2.19 B)). Normal cells are convolutional cells that return a feature map with the same dimensions, while reduction cells return a feature map with reduced height and width by a factor of two. The normal and reduction cells have the controller Recurrent Neural Network (RNN). In Figure 2.19 A) is shown the construction of blocks in a convolutional cell, where the controller RNN recursively predicts the rest of the convolutional cell structure, given two initial states (in this case state H_1 and H_2). In each block, the controller selects a pair of hidden states and respective operations to combine the hidden states [64].

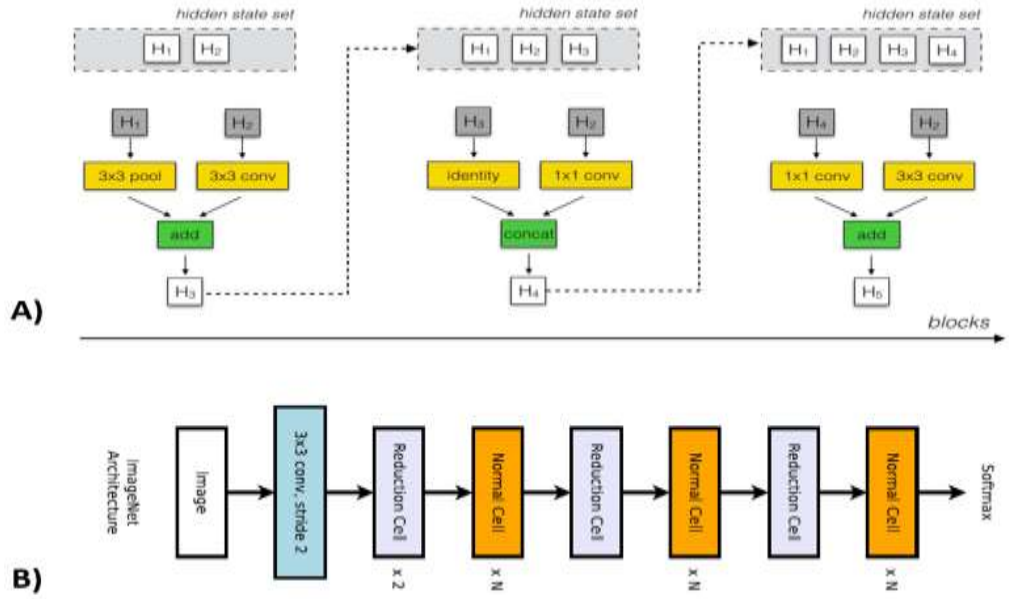


***Figure 2. 19** A) Construction of blocks in a convolutional cell. B) NASNet architecture. (Adapted from* [64]*)*

2.4.2. Transfer Learning

CNN models can be trained on one task, and then fine-tuned on another task, which is a technique called transfer learning [24] [65]. Usually, the network is trained with a freely available large data repository, for example the ImageNet. Then, the network is fine-tuned on a specific task with a new repository [24]. Changing the structure of the fully connected layers has been increasingly usual, when applying transfer learning. The objective of the transfer learning is to replace the last fully connected layers to adapt to the new visual recognition tasks while preserving the other parameters learned by ImageNet [26].

When we initialize the network with pre-trained parameters, this is called a pre-trained network. It is very usual because it can accelerate the learning process and improve the generalization ability, leading to pre-trained networks working better than networks trained in a traditional way [26] [66]. The pre-trained networks must fine-tune the weights according to a specific task. Frequently, fine-tuning requires class labels for the new training dataset that the output layer depends on to be initialized. All layers, except the output layer, are initialized based on the pre-trained model. However, there are cases for which it is very difficult to obtain class labels for the new training dataset [26].

2.5. Performance metrics

To evaluate the models' performances, it is essential to calculate and analyze some metrics, such as Matthew's Correlation Coefficient (MCC), accuracy, balanced accuracy, precision, recall, and F1-score.

The MCC is a classification performance measure specifically tailored for unbalanced data. A MCC of + 1 represents a perfect prediction, while a MCC of 0 represents a random prediction, and if the MCC is – 1 represents an inverse prediction [67]. The MCC can be calculated with the following formula (where TP is the number of true positives predictions, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives) [68]:

$$MCC = \frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}} \tag{2.1}$$

Accuracy corresponds to the percentage of correct predictions and can be calculated as [68] [69]:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.2}$$

The balanced accuracy is appropriate to use when the dataset is unbalanced, and the best value corresponds to 1 and the worst value corresponds to 0. It is calculated by the next formula [69]:

$$BA = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) \tag{2.3}$$

The precision metric corresponds to the ability of the model to not classify a positive sample as a negative sample, and the best performance correspond to the precision value equal to 1, and the worst is equal to 0 [69]:

$$Precision = \frac{TP}{TP+FP} \tag{2.4}$$

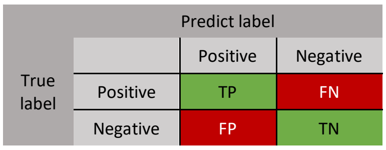
The recall measures the ability of the model to find all positive samples, and best value is 1 and the worst value is 0 [69]:

$$Recall = \frac{TP}{TP+FN} \tag{2.5}$$

The F1-score is the harmonic mean of precision and recall and its best value corresponds to 1 and the worst value corresponds to 0 [69]:

$$F1 = 2\ \times \left(\frac{Precision\times Recall}{Precision+Recall}\right) \tag{2.6}$$

The confusion matrix is used to evaluate the model, because through the confusion matrix we could observe the elements that were correctly or incorrectly classified, and which the classes that are confused each other. The diagonal values of confusion matrix correspond to the number or percentage of elements corrected classify, and the values outside of diagonal of confusion matrix correspond to the elements that were wrongly classified [69]. In Figure 2.20 it is represented a confusion matrix example [69].



		Predict label	
		Positive	Negative
True label	Positive	TP	FN
	Negative	FP	TN

Figure 2. 20 *Example of binary confusion matrix, where TP is the number of true positives predictions, TN is the true negatives, FP is the false positives and FN is the false negatives.*

CHAPTER 3 STATE-OF-THE-ART | Deep Convolutional Neural Networks applied to EGD

The scientific community and the endoscopists are increasingly interested in improving the EGD exam through DL. There are several studies that prove the ability of CNNs in the monitorization of blind spots. However, the CNNs investigations in the detection of upper GI landmarks are still in the beginning, as they are very recent studies.

In this chapter, several studies developed in this area are shown. In section 3.1., studies developed with a fully available dataset that contain classes of gastric landmarks (the *Kvasir* dataset) are presented. In section 3.2. is presented a study made with a recent update of the *Kvasir* repository, called *HyperKvasir* dataset. The section 3.3. contains studies that use CNNs pre-trained over the ImageNet repository to classify datasets of gastric landmarks through transfer learning. Section 3.4. consists of a study where blind spots are already monitored and projected over a grid model. Finally, the last section consists of a system that uses a CNN and deep reinforcement learning (DRL) to monitor the EGD exam in real-time.

3.1. Automatic anatomical classification of upper GI landmarks of *Kvasir* dataset using CNNs

The *Kvasir* dataset is a collection of GI images that became available in 2017 thanks to the Medical Multimedia Challenge offered by MediaEval. The images were collected in the Baerum Hospital in Norway, which has a large gastroenterology department. The images were taken with an endoscope and verified by certified endoscopists. The images have a matrix size that varies between 720×576 and 1920×1072 [70] [71].

The initial *Kvasir* dataset has 4 000 images, which are categorized into eight categories and each category has 500 images. Three categories represent anatomical locations, three other categories represent pathological states, and the final two categories represent removed lesions. The anatomical landmarks correspond to “geo-referred” spots that may exhibit ulcers and or inflammations. The anatomical landmarks are pylorus, z-line and cecum (only pylorus and z-line belongs to the upper GI tract) [70] [71]. The pylorus (see Figure 2.1) is the junction between the stomach and duodenal bulb, and a sphincter regulating the emptying process of the stomach into the duodenum. The z-line is located at the same level as the gastroesophageal junction (see Figure 2.1) [72].

Pogorelov et al. [71] presented the *Kvasir* dataset to enable the scientific community to have a reliable and publicly available dataset, because annotated datasets containing medical images are hardly available, making reproducibility and comparison of approaches almost impossible. The aim of the *Kvasir* dataset is to develop and evaluate the algorithmic analysis of images and the scientists with the same dataset can easily compare approaches and experimental results. Pogorelov et al. developed an initial multi-class detection experiment with *Kvasir* dataset as a baseline for future studies.

Pogorelov et al. used several configurations to classify the *Kvasir* dataset. The researchers split the dataset randomly in two equal parts for training and testing, with each class containing 250 images.

Many different combinations of features were made (Joint Composite Descriptor, Tamura, Color Layout, Edge Histogram, Auto Color Correlogram and Pyramid Histogram of Oriented Gradients) and then used the Random Forest (RF) and Logistic Model Tree (LMT) classifiers. RF is a classifier that uses many individual decision trees that operate in ensemble to predict the output class [73]. LMT is another classifier that combines logistic regression with decision tree learning [74]. The combination that showed better performance was the combination of all features with LMT classifier, with an accuracy of 93.70%, MCC of 71.10% and F1-score of 74.70%.

In this study, two different CNNs were trained, one with three convolutional layers and the other with six convolutional layers. The CNN with three layers had an accuracy of 95.90%, MCC of 43.00% and F1-score of 45.30%. The CNN with six layers had an accuracy of 91.40%, MCC of 60.20% and F1-score of 65.10%. The CNN with six layers had a better performance than the CNN with three layers in terms of detection, but not in terms of speed.

The pre-trained Inception-v3 network was used to classify the *Kvasir* dataset and obtained an accuracy of 92.40%, MCC of 64.90% and F1-score of 69.30%. The Inception-v3 had better performance than the other two CNNs. This study showed good results and can be a good starting point for other researchers.

Several research groups participated in the MediaEval Medico Challenge and obtained good results by using CNNs to classify the *Kvasir* dataset: Agrawal et al. [75] and Petscharning et al. [76]. The challenge objective is to improve the methods of multimedia-assisted diagnosis in the domain of endoscopic imaging, especially in the detection of gastric diseases and anatomical landmarks. The performance metric used to evaluate the algorithms proposed by the participants of the challenge was the MCC. The *Kvasir* dataset was updated and the second version consists of 8 000 images in 8 classes, which each class has 1 000 images instead of 500 images as the first version [72].

- **Agrawal et al.** [75] used 3 200 out of the 4 000 samples for training. The set of features used was provided by the challenge and included features such as Tamura, Color Layout, Edge Histogram and Auto Color Correlogram (baseline features). Each of these features is a global descriptor of the images characteristics that could be useful in final classification or not. Agrawal et al. used the VGG-16 network to extract further features. The researchers used the outputs/features of the first fully connected layer, which had a dimensionality of 4 096 neurons. Then, they also used the Inception-v3 network to extract features of the penultimate layer, which had a dimensionality of 2 048 neurons.

  Agrawal et al. evaluated three different combinations of features and used a Support Vector Machine (SVM) to classify the *Kvasir* images. The SVM classifier has the objective to find the best hyperplane (maximal margin) between two classes of training samples in the feature space [77].

  The performance of each feature's combination was evaluated through F1-score, accuracy and MCC. The MCC, accuracy and F1-score were 81.60%, 95.90%, 83.80%, respectively, when combining baseline features with Inception-v3 features. The MCC, accuracy and F1-score were 78.50%, 95.30%, 81.20%, respectively, when combining baseline features with VGG-16 features. The last combination was the best performance and was the combination of baseline features with Inception-v3 and VGG-16, which were 82.60%, 96.10%, 84.70% to MCC, accuracy and F1-score, respectively. Then, the confusion matrix was calculated for the best performance (combination of all features) and concluded that z-line and esophagitis (pathology) classes were more confounded. These results show that CNNs have a good potential to extract critical features.

- **Petscharning et al.** [76] proposed an inception-like CNN architecture, which was based on the GoogLeNet. This architecture is capable of learning using only a small amount of training data. This architecture shows acceptable results with only 50 training examples per class.

  An image representing the proposed architecture is in Figure 3.1. The architecture presented two models/variants (Model A and Model B). The first layer of the model was a max pooling layer. In Model A, the max pooling layer had stride of 4 and window size of 5, while Model B had a stride of 2 and window size of 3. Then, they used three stacked inception-like modules to extract features. The layer after the three stacked was 1×1 convolution with 92 learnt filters. The next convolutional layer had different variants, the Model A had 1 024 neurons and convolutional size of 4×4, while the Model B had 1 024/2 048 neurons and convolutional size of 8×8. Then, the model had a 1×1 convolution layer and a dense layer with softmax activation.

  The Model A with 1 024 neurons had 2.8M parameters, the Model B with 1 024 neurons had 7.3M parameters and the Model B with 2 048 neurons had 16.5M parameters.

  The authors evaluated the detection performance and computational complexity of three variants of the proposed architecture: the amount of training data (10%, 50%, or 90% of training data) and the two different models of the CNN architecture (Model A and Model B). Petscharning et al. varied their architecture increasing predictive performance due to the increase of computational cost and number of trainable parameters.

  The results showed that performance increases with more training data available. The Model A had a small number of parameters relative to Model B, which makes the model faster. The Model A takes 2.25 ms per forward pass while the Model B takes 2.91 ms and 3.42 ms per forward pass to 1 024 and 2 048 neurons, respectively. The architecture GoogLeNet takes 14.16 ms per forward pass.

  In the detection, surprisingly, the Model B with 1 024 neurons shows better performance than the Model B with 2 048 neurons. This could be explained by the bigger size of the Model B with 2 048 neurons, which tends to better adapt to the training data, generating overfitting. The Model B with 1 024 neurons and with 90% of training data was the combination with the best performance, which had 93.90% of accuracy, 75.50% of F1-score and 72.00% of MCC.

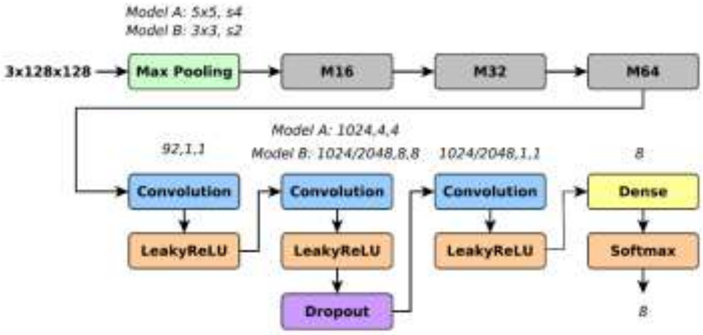


***Figure 3. 1** The inception-like CNN architecture (based on GoogLeNet). (Adapted from* [76]*)*

- **Cogan et al.** [70] is a study outside of the MediaEval Medico Challenge. Cogan et al. used Inception-ResNet-v2 (a CNN architecture that uses the inception modules with skip connections used in residual blocks [55]), Inception-v4, and NASNet models to classify anatomical landmarks and the other categories of the *Kvasir* dataset. The dataset was divided into training (85% of the images) and validation (15% of the images).

  The images have many variants such as, capture angle, brightness, center point and zoom. Before the classification of *Kvasir* dataset, due to these variants, Cogan et al. preprocessed the images with edge removal, contrast enhancement, filtering, color mapping and scaling. All this preprocessing framework was called Modular Adaptive Preprocessing for Gastrointestinal tract Images (MAPGI), which improved the images quality.

  The accuracies of Inception-ResNet-v2, Inception-v4, and NASNet models were respectively 98.48%, 98.45% and 97.35%. The performances of Inception-ResNet-v2 and Inception-v4 were similar. The performance of NASNet could be improved if the dataset had bigger size, as the NASNet has a large number of parameters (88.9 million) and with a relatively small number of training images, the model is more likely to overfit the training data.

  The Inception-v4 performance was evaluated with a confusion matrix, which showed that the model only misclassified z-lines into esophagitis. This happens because esophagitis is a disease that occurs in the z-line. The Inception-v4 had a good and consistent performance in 3 independent evaluations with averages of 93.80% of precision, 93.90% of recall, 99.10% of specificity, 93.80% of F1-score and 92.90% MCC.

3.2. Automatic anatomical classification of upper GI landmarks of *HyperKvasir* dataset using CNNs

The second version of the *Kvasir* dataset was extended to 110 079 images and 23 classes, in which 10 662 images were labeled and 99 417 were unlabeled. This extension of the second version of the *Kvasir* repository is called *HyperKvasir*. The *HyperKvasir* dataset has three classes of anatomical landmarks of upper GI (z-line, pylorus, retroflex stomach), one more class than the *Kvasir* dataset (retroflex stomach). The retroflex stomach means that the endoscope is retroflexed, looking back to visualize the cardia and fundus in the upper parts of the stomach (see Figure 2.2 and Figure 2.3) [78].

Borgli et al. [78] performed a series of experiments to give example baseline results to be used by future researchers to compare and measure their results. The experiments consisted of using CNN architectures to classify the 23 classes of the *HyperKvasir* repository. Borgli et al. experimented with five different CNN architectures. To evaluate the architectures' performances, they used F1-score, precision, recall, and they calculated the MCC.

Borgli et al. experimented with architectures pre-trained over the ImageNet dataset, which were ResNet-50, ResNet-152 and DenseNet-161 to classify the 23 classes. The DenseNet-161 architecture outperformed the other architectures with a precision of 64.00%, a recall of 61.60%, an F1-score of 61.90%, and an MCC of 89.90%. The ResNet-50 and ResNet-152 had an MCC of 82.60% and 89.80%, respectively.

Then, Borgli et al. used an architecture that combines the ResNet-152 and DenseNet-161 approach by averaging the output of both models as the final prediction. They also combined the ResNet-152 model and DenseNet-161 model with a multilayer perceptron (MLP) to estimate the best way to average the output of each model. The combination of ResNet-152 with DenseNet-161 outperforms the five experiments with a precision of 63.30%, a recall of 61.50%, an F1-score of 61.70%, and an MCC of 90.20%. The combination of the ResNet-152 model and DenseNet-161 model with an MLP had a precision of 61.20%, a recall of 60.60%, an F1-score of 60.50%, and an MCC of 90.20%.

Confusion matrices were made for each experiment and were concluded that there is confusion between pathological classes (namely, Barrett's and esophagitis) with z-line.

Borgli et al. demonstrated the quality of the available *HyperKvasir* dataset and with these experiments provided a baseline to future researchers.

3.3. Automatic anatomical classification of upper GI landmarks of private datasets using CNNs

- **Takiyama et al.** [79] created a CNN-based, diagnostic-oriented system based on a GoogLeNet architecture.

  A repository of 27 335 EGD images was used for network training. The images were categorized into four anatomical locations: larynx, esophagus, stomach and duodenum. Subsequently, the images corresponding to the stomach were classified into three regions: upper, middle and lower regions. The validation repository had 17 081 EGD images, which were independent of the training repository.

  A probability score ranging from 0% to 100% was created for each output image, indicating the probability that an image belongs to each of the anatomical classification, Figure 3.2. The classified CNN images were manually evaluated by two gastroenterology specialists to see if they were correctly classified.

  Performance was evaluated by recall, which showed a good performance in the classification of the four anatomical locations. The classification of the larynx was 93.90% recall, of the esophagus was 95.80% recall, of the stomach regions were 98.90% recall, and of the duodenum was 87.0% recall. These results show that this architecture has a robust performance and potential to be used in a computer-assisted diagnostic system of EGD.

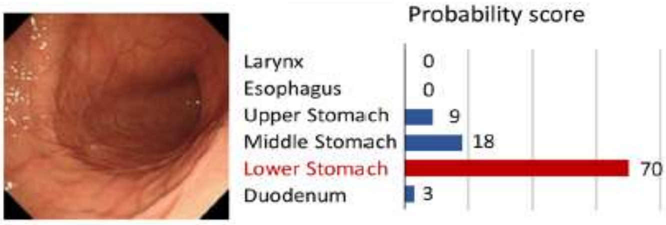


Figure 3. 2 *A flow chart with output probability score using CNN to classify the anatomical location, in this case the image corresponds to the lower stomach. (Adapted from* [79]*)*

- **He et al.** [80] tested many CNNs to classify anatomical locations according to the balance between the British and Japanese (SSS) guidelines. The dataset used contained 3 704 EGD images that were acquired from Tianjin Medical University General Hospital from two different gastroscopes. He et al. used white light imaging and LCI technique because these procedures provide very similar images in terms of tissue appearance.

  This study classified 11 landmarks, which are divided into antegrade view and retroflex view. There are eight landmarks from antegrade view: pharynx, esophagus, squamocolumnar junction, middle-upper body, lower body, antrum, duodenal bulb, and duodenal descending. There are three landmarks from a retroflex view: fundus, middle-upper body and angulus. The data was divided into 12 classes: 11 landmarks and 1 class to unqualified images (NA).

  Before the classification with CNNs, He et al. extracted the regions of interest of the images and then, the images were anatomical annotated by a clinical expert and a medical imaging doctoral student.

  The CNNs used to classify anatomical locations were: ResNet-50, Inception-v3, VGG-11-bn, VGG-16-bn and DenseNet-121. DenseNet-121 was the one that showed the best performance with an accuracy of 88.11%, followed by Inception-v3 (87.97%), VGG-16-bn (87.81%), ResNet-50 (87.72%) and VGG-11-bn (87.43%). Then, He et al. used DenseNet-121 to evaluate individual landmark classification and to measure the performance used F1-score accuracy and confusion matrix. From the observation of the confusion matrix, it is possible to understand that errors were mainly caused by three factors: landmarks were misclassified to NA; NA were misclassified to landmarks; landmarks with similar appearance are easily misclassified to each other.

This study demonstrated a good performance by many CNNs, and with individual landmarks evaluation was possible to detect the sources of errors.

The works by Takiyama et al. study and by He et al. study showed that CNNs have potential in classifying EGD images according to anatomical location. This ability to recognize anatomical locations is a major step because it proves that it may be possible to use CNN systems in the diagnosis of GI diseases. However, the image repository is from a single institute, which may skew the results because the network may not behave the same with another database. On the other hand, the images vary depending on the viewing angle and the expansion of the organ, so it would be beneficial to create a panoramic mapping during EGD, controlling the presence of blind spots.

3.4. Automatic anatomical classification of upper GI landmarks using CNNs and reconstruction of a grid model to the stomach

Wu et al. [4] created a CNN system armed with a grid model for the stomach to be able to control blind spots. Wu et al. used 26 anatomical locations. The use of more anatomical locations/classification classes makes the study more reliable than previous studies.

Wu et al. created a CNN system capable of detecting EGC during EGD and, at the same time, recognizing gastric locations better than endoscopists.

The proposed system is represented in Figure 3.3. First, the RF filters blurry images, making a pre-selection of the images. Clear images were further used for the training and testing of the ResNet-50 and VGG-16 networks. When running the proposed system on the videos, images were captured at 2 fps (frames per second). All frames were filtered by the RF classifier. Then, only the clear frames can enter into the ResNet-50 and VGG-16 networks.

The ResNet-50 network is responsible for early gastric cancer identification, where clear images were classified into malign or benign. The VGG-16 network is responsible for classification of gastric location, and clear images were classified into 10 or 26 gastric locations by two endoscopists for the training and testing network.

To train the ResNet-50 network, 3 170 malign images and 5 981 benign images were collected. To train the VGG-16 network to monitor blind spots, 24 549 images from different parts of the stomach were collected. The number of malign images is smaller than the number of benign images because malign cases are relatively rare compared with benign cases. The number of images of the different anatomical locations varies widely. The models VGG-16 and ResNet-50 were pre-trained with ImageNet repository and through transfer learning they are used to classify the cancer (benign or malign) and to classify anatomical locations.

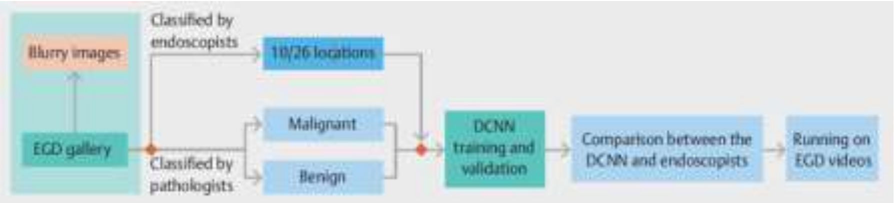

Blurry images
Classified by endoscopists
10/26 locations
EGD gallery
Malignant
Classified by pathologists
Benign
DCNN training and validation
Comparison between the DCNN and endoscopists
Running on EGD videos

Figure 3. 3 *Flowchart of the data preparation, training and testing procedure of Wu et al. (Adapted from [4])*

The anatomical locations used for classification were divided into two groups, one with 10 anatomical locations and the other with 26 anatomical locations. The 10 anatomical locations were esophagus, squamocolumnar junction, antrum, duodenal bulb, descending duodenum, lower body in forward view, middle-upper body in forward view, fundus, middle-upper body in retroflexed view and angulus. The 26 anatomical locations were the esophagus, squamocolumnar junction, antrum (in greater curvature, posterior wall, anterior wall and lesser curvature), duodenal bulb, descending duodenum, lower body in forward view (in greater curvature, posterior wall, anterior wall and lesser curvature), middle-upper body in forward view (in greater curvature, posterior wall, anterior wall and lesser curvature), fundus (in greater curvature, posterior wall, anterior wall and lesser curvature), middle-upper body in retroflexed view (in posterior wall, anterior wall and lesser curvature), angulus (posterior wall, anterior wall and lesser curvature).

The performance of VGG-16 in the classification of gastric locations was evaluated through comparison with the performance of endoscopists. A group of endoscopists had 10 experts, 16 seniors and 9 novices. The CNN accuracy was 90.00% to classify the 10 anatomical locations and 65.88% to classify the 26 anatomical locations. While the endoscopist accuracy to classify the 10 anatomical locations was 90.22%, 86.81% and 83.30% for experts, seniors and novices respectively. The endoscopists accuracy to classify the 26 anatomical locations was 63.76%, 59.26% and 46.47% respectively.

In the detection of EGC, the CNN system achieved a recall of 94.00%, an accuracy of 92.50% and a specificity of 91.00%.

The grid model for the stomach generates a virtual stomach model, which, initially, is uncovered/transparent and when EGD is performed, the virtual stomach model is filled. The CNN model captures images and fills them into the corresponding part in the stomach model, reminding endoscopists of blind spots' presence during EGD.

The strength of this study is the comparison of CNN model performance with the clinician's performance to classify anatomical locations, and the results are very interesting because it is visible that the CNN model achieves identify the anatomical locations better than clinicians. However, there are some limitations in the study by Wu et al. When CNNs were applied to unprocessed videos and mucosa was not washed clean, some errors occurred. Therefore, it is important to train CNN to recognize mucosa that is poorly prepared. On the other hand, the CNN system was only quantitatively tested in images, not in videos. So, it is necessary to keep collecting data to obtain credible results in real-time EGD unprocessed videos.

3.5. A real-time system based on CNN and DRL for monitoring upper GI landmarks

In 2019 Wu et al. [16] created a WISENSE system, a real-time quality improving system based on the CNN and DRL. WISENSE monitors blind spots, EGD duration time and generates photodocumentation during EGD. This is the first study using DL and DRL. The WISENSE system was tested on a randomized controlled trial and proved to be very efficient.

All the EGD images and videos were from Renmin Hospital of Wuhan University and were in white light view. The instruments used to collect data were from two vendors: Olympus Optical Co., Tokyo, Japan and Fujifilm, Co., Kanagawa, Japan.

The system was trained and tested on still images. The system was built around two mature CNN models (VGG-16 and DenseNet). The VGG-16 and DenseNet had the objective to classify 27 classes: 26 anatomical locations and one unqualified image (images that are not classified due to the absence of an anatomical landmark). The anatomical landmarks were the esophagus, squamocolumnar junction, antrum (in greater curvature, posterior wall, anterior wall and lesser curvature), duodenal bulb, descending duodenum, lower body in forward view (in greater curvature, posterior wall, anterior wall and lesser curvature), middle-upper body in forward view (in greater curvature, posterior wall, anterior wall and lesser curvature), fundus (in greater curvature, posterior wall, anterior wall and lesser curvature), middle-upper body in retroflexed view (in posterior wall, anterior wall and lesser curvature), angulus (posterior wall, anterior wall and lesser curvature). To train the model, 34 513 EGD images were used, which were labelled by seniors with 1-5 years of EGD analysis experience and three experts with more than 5 years of EGD experience.

The VGG-16 and DenseNet were tested three times, with 2 160 images (80 per class) each time. The accuracy was 88.70% and 83.76% to VGG-16 and DenseNet respectively. The performance of CNNs was also evaluated with the corresponding confusion matrix. The VGG-16 was chosen to develop WISENSE because it had better accuracy.

In addition to the study of the VGG-16 and DenseNet architectures, Wu et al. also considered the application of deep reinforcement learning techniques (DRL). The idea behind the adoption of DRL was that of taking advantage of context information gathered during the EGD exam to improve classification performance. DRL is a branch of DL, it is a combination between DL and reinforcement learning. DRL uses DL to leverage its perception ability in visual tasks and uses reinforcement learning in order to perform complex decisions in a dynamic environment. EGD judgments must be based on more than one frame due to dynamic and constantly changing views in the human body, and CNNs only analyze frames independently. Therefore, DRL could help combat this problem, as DRL has good potential to make decisions in these dynamic environments. The DRL was developed in this study with the objective to use sequentially captured frames to predict gastric locations and reduce the noise in real EGD videos.

DRL learns tasks with rewards (when make a correct action) and punishments (when make a wrong action) and creates a self-learning feedback loop with these states. A state refers to a snapshot of everything in an environment at a certain time. In this study, a state is characterized as the set of labels of the previous nine consecutive images predicted by the CNN, and all gastric sites previously activated by DRL at a certain time. This state was projected into a 10×26 matrix to use as input for DRL (see Figure 3.4), where the abscissa numbers are the 26 anatomical landmarks corresponding to the classes of the unqualified images and the ordinate represents in which frames such images appear. Each frame appears in different times and with respective class/abscissa predicted by CNN. The frames are represented in Figure 3.4 by cubes, which appear to be falling from top to bottom as the EGD video is played. The confidence of CNN's prediction is represented by the color shade of the cubes, the white color represents the higher confidence.

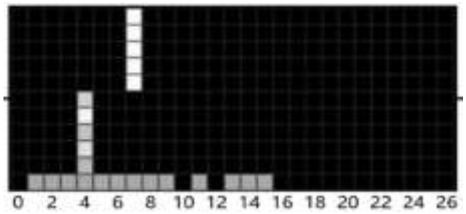


Figure 3. 4 *The DRL input state projected into a 10×26 matrix. (Adapted from* [16]*)*

The DRL will make the decision according to the state/matrix. Initially, during the training phase, the DRL model takes actions randomly and receives a reward for a correct decision and receives a punishment for a wrong decision. These successive decisions are stored in an experience pool. Then, based on the previous experiences, the model takes the optimal action in a state.

The DRL model was trained on 50 virtual EGD and tested on 30 real EGD videos. After 74 epochs, the DRL model had 91.40% of accuracy.

The WISENSE system uses the VGG-16 with DRL to monitor blind spots. To test the WISENSE performance a total of 107 real stored EGD videos were used. The WISENSE was configured to process images at 2 fps in videos. The monitoring of blind spots by WISENSE presented an average accuracy of 90.02% and a separate accuracy for each anatomical location ranging from 70.21% to 100%.

A randomized controlled trial was made to evaluate the WISENSE, where 324 patients were recruited and randomized (153 patients belonged to the WISENSE group and other 150 patients belonged to the control group). The randomized controlled trial concluded that the blind spot rate was lower in the WISENSE group compared with the control group, 5.86% vs 22.46% respectively.

The WISENSE system had a very good performance, since it monitors blind spots in real-time during EGD. This system could help endoscopists mitigate variations in their performances. The WISENSE system considerably decreases the blind spot rate and shows improvements in EGD photodocumentation.

3.6. Conclusion

The *Kvasir* dataset promoted CNNs exploration in landmark detection, with several studies showing promising results. Pogorelov et al. uses architectures with 3 and 6 convolutional layers, which presented good performances. However, Pogorelov et al. concluded that using a pre-trained architecture like the Inception-v3 gave better performance. Agrawal et al. used CNN architectures (VGG-16 and Inception-v3) to extract features and obtained accuracies above 95.00%. These results show the ability of CNNs to extract the features that better represent the images. Petscharning et al. used an inception-like CNN architecture focusing on variables, such as the number of neurons in the network and the size of the training repository. Petscharning et al. concluded that better performances are obtained with large training datasets and many parameters could compromise the network performance, generating overfitting. It is important to maintain a balance between the number of parameters and the size of the training repository. Cogan et al. tested several architectures pre-trained over ImageNet to classify the *Kavasir* repository. In this study, Inception-v4, Inception-ResNet-v2, and NASNet networks were tested, which all had accuracies above 97.00%. However, the NASNet performance is noteworthy because it is an architecture that uses CNNs with RNN controllers recursively. The NASNet considers the previous states due to its controller, which could be interesting for monitoring blind spots during EGD exams. This study shows the potential that RNNs architectures could have in detection of landmarks.

The Borgli et al. used the *HyperKvasir* repository that has 3 anatomical landmarks and tested several architectures (ResNet and DenseNet models) as a baseline for future studies and to show the dataset potential.

All previous studies only classified 2/3 anatomical landmarks. However, it is important to evaluate the CNNs performances with more anatomical landmarks.

Takiyama et al. used a superior number of EGD images and anatomical landmarks (collected from a hospital) than the *Kvasir* repository and used a GoogLeNet architecture, which showed a good performance.

He et al. used several CNN architectures pre-trained with ImageNet repository (ResNet-50, Inception-v3, VGG-11-bn, VGG-16-bn, DenseNet-121) and used 11 anatomical landmarks, requiring greater efficiency by CNN architectures. However, all CNN architectures had accuracies above 87.00%, which means that, with an increase of anatomical landmarks, the CNNs continue to perform well.

Wu et al. increased the anatomical landmarks to 26 and compared the performance with only 10 anatomical landmarks with the same architecture (VGG-16). It is noticeable that the accuracy decreases greatly with the increase of anatomical landmarks: the accuracy with 10 landmarks was 90.00%, while with 26 landmarks it went to 65.90%. The complexity of the architecture increases with more landmarks. Later, in another study, Wu et al. improved the VGG-16 accuracy to 88.70% and added DRL to the VGG-16 architecture, having been the first randomized test with a CNN system in EGD videos in real-time.

In Table 3.1 it is shown a summary of studies presented in the state-of-art to see the evolution and comparing the performance studies.

Observing the Table 3.1, in the first studies (*Kvasir* dataset), it is visible that the Inception architectures are the most explored and have the best performances. However, Inception modules with residual blocks (Inception-ResNet-v2 architecture) were similar perform to the Inception architectures. The experiments with the *HyperKvasir* repository showed that the junction of DenseNet with ResNet models obtained good performances.

In later studies, with larger datasets, the complexity of the networks also increased, and the Inception architectures continued to have the highest accuracy values. However, DenseNet architecture outperformed the Inception-v3 architecture in the He et al. study. Wu et al. explored DenseNet and VGG-16 architectures and the VGG-16 architecture had the best performance.

Table 3. 1 *Summary of the detection gastric landmarks studies, where ACC corresponds to the accuracy of the respective study, MCC corresponds to the Matthew's Correlation Coefficient, and the RC corresponds to the recall.*

Studies	Technique	Dataset	Performance	Anatomical Landmarks	Task
Pogorelov et al.[71]	Inception-v3	*Kvasir* repository	92.40% (ACC)	2 anatomical landmarks	Classification of 8 classes in the *Kvasir* repository
Agrawal et al.[75]	Baseline Features + Inception-v3 Features + SVM classifier		95.90% (ACC)		
	Baseline Features + VGG-16 Features + SVM classifier		95.30% (ACC)		
	Baseline Features + Inception-v3 Features + VGG-16 Features + SVM classifier		96.10% (ACC)		
Petscharning et al.[76]	Inception-like CNN		93.90% (ACC)		
Congan et al.[70]	Inception-v4		98.40% (ACC)		
	Inception-ResNet-v2		98.50% (ACC)		
	NASNet		97.40% (ACC)		
Borgli et al. [78]	ResNet-50	*HyperKvasir* repository	82.60% (MCC)	3 anatomical landmarks	Classification of 23 classes in the *HyperKvasir* repository
	ResNet-152		89.80% (MCC)		
	DenseNet-161		89.90% (MCC)		
	ResNet-152 + DenseNet-161		90.20% (MCC)		
	ResNet-152 + DenseNet-161 + MPL		90.20% (MCC)		
Takiyama et al.[79]	GoogLeNet	44 416 EGD images from a hospital	93.90% (RC)	6 anatomical landmarks	Anatomical landmarks detection
He et al.[80]	ResNet-50	3 704 EGD images from a hospital	87.70% (ACC)	11 anatomical landmarks	
	Inception-v3		88.00% (ACC)		
	VGG-11-bn		87.40% (ACC)		
	VGG-16-bn		87.80% (ACC)		
	DenseNet-121		88.10% (ACC)		
Wu et al.[4]	VGG-16	24 549 EGD images from a hospital	90.00% (ACC)	10 anatomical landmarks	Anatomical landmarks detection + stomach grid model
			65.90% (ACC)	26 anatomical landmarks	
Wu et al.[16]	VGG-16	34 513 EGD training images	88.70% (ACC)	26 anatomical landmarks	Anatomical landmarks detection in real-time
	DenseNet		83.80% (ACC)		
	VGG-16 + DRL	80 EGD videos from a hospital	91.40% (ACC)		

CHAPTER 4 DESIGN

In this chapter, the design of our study is shown. In section 4.1., the available repositories, as well as the repositories that were selected for the proposed classification task are described. In section 4.2., the models/architectures that were experimented to classify upper GI images are presented.

4.1. Data

DL models' performance is greatly influenced by the characteristics of the dataset. It is crucial to choose good sources with reliable labels. The quality and amount of data influences the model's behavior, and it is important to prepare the data to obtain a balanced dataset that is representative of each class. The sources of our dataset are from available repositories.

Firstly, we started by searching datasets that contain both upper GI tract images with healthy anatomical sites and with pathologies. An overview of the datasets with images and videos from upper GI tract is represented in Table 4.1, where it is possible to observe the findings and the number of frames and videos of each repository. However, only two repositories were used: *HyperKvasir* [78], and *GASTROLAB* [81]. The notes in Table 4.1 express the reasons why the respective repositories were not used.

The *HyperKvasir* [78] repository contains 110 079 images and 374 videos collected during gastro- and colonoscopy examinations and they are in white light. This dataset contains anatomical landmarks and pathologies. Among the 110 079 images contained in the dataset, 10 662 are labeled and 99 417 are unlabeled. The majority of images have a matrix size of 768×576. The videos are labeled by a gastroenterologist and most videos have a matrix size of 720×576. Besides that, *HyperKvasir* have images with segmentation masks and bounding boxes (1 000 images from the polyp class) [78].

The *HyperKvasir* dataset contain 16 classes from upper GI tract: the anatomical landmarks are z-line, pylorus, and retroflex stomach; the pathological finds are barrett´s, barrett's short segments, esophagitis, esophagitis grade A, esophagitis grade B-D, polyps, ulcer, gastric antral vascular ectasia, varices, and cancer gastric banding perforated; the therapeutic interventions; and the quality of mucosal view are reduced view and optimal view [78]. The objective of our work is to classify upper GI landmarks with or without pathologies, so, our dataset only was considered the images from pathologies or classes which it is possible to identify the anatomical region.

The *GASTROLAB* [81] repository contains images and videos for educational purposes, it does not provide specific information regarding the patients from which data were collected. The dataset is only organized according to the anatomical zones and the pathologies. According to what we collected from *GASTROLAB* repository we have the next classes (the images and videos in each class include healthy and pathological tissue): antrum, antrum and pylorus, body, cardia, duodenal bulb, duodenum, esophagus, fundus, papilla vateri, pylorus, and stomach. We only collected white light images and videos. The majority of images from *GASTROLAB* have a matrix size of 1 210×475.

The labeled videos and images contain healthy and pathological anatomical zones. Table 4.2 shows the number of images and videos collected from *HyperKvasir* and *GASTROLAB* repositories, and in Table 4.2 are represented the final classes used in our task.

The frames were extracted from videos at 1 frame per second (fps). The total of images and extracted frames obtained is represented in Figure 4.1 with the respective anatomical zone.

In Figure 4.1, the collected images and frames correspond to 15 classes. However, the 15 classes are very unbalanced. So, to tackle class unbalance, from a total of 15 classes we selected 9 classes. The final classes are esophagus, z-line, fundus, cardia, retroflex stomach, body, antrum and pylorus, duodenal bulb, and duodenum. The distribution of the 9 classes is represented in Figure 4.2 and Table 4.2.

Table 4. 1 *List of upper GI datasets. (Adapted from* [78]*)*

Dataset	Findings	Size	Notes
Endoscopy Artifact detection 2019	Endoscopic artifact classes (bubbles, motion blur, etc.)	5 138 images	Unlabeled images according to the anatomical landmarks
KID	Angiectasia, bleeding, inflammations, polyps	2 371 images and 47 videos	Video capsule endoscopy
GASTROLAB	GI lesions and GI healthy landmarks	431 labeled images and 85 labeled videos *	Open academic
WEO Clinical Endoscopy Atlas	GI lesions	152 images	Low quality, it is educational
Atlas of Gastrointestinal Endoscope	GI lesions	1 295 images	Not available anymore
El salvador atlas of gastrointestinal video endoscopy	GI lesions	5 071 video clips	Low quality, it is educational
Kvasir	Z-line, pylorus, cecum, esophagitis, polyps, ulcerative colitis, dyed and lifted polyps, dyed resection margins	8 000 labeled images (1 000 images for each class)	Open academic
HyperKvasir	Anatomical landmarks, quality of mucosal views, pathological findings, therapeutic interventions	10 662 labeled images; 99 417 unlabeled images; 374 labeled videos; 1 000 images with segmentation masks from polyps' class	Open academic

Benchmark dataset for a digestive tract diagnostic support system	Pathological findings and mucosal views	1 354 700 images;
		1 779 upper GI precise annotations;
		21 680 upper GI imprecise annotations
			By request

AIDA-E challenge [82]	Stomach	60 lesion images 28 no lesion images	Chromoendoscopy Modality

* The number of GI images and videos are undetermined, however, for our classification task there are those that are mentioned in the table.

Table 4. 2 *The number of images and videos collected by the HyperKvasir and GASTROLAB repositories. The anatomical classes include healthy and pathological tissue. There are represented the final classes used in our task.*

	Number of images		Number of videos		
Classes	***HyperKvasir***	***GASTROLAB***	***HyperKvasir***	***GASTROLAB***	**Final Classes**
Antrum	-	55	4	9	Antrum and Pylorus
Antrum and pylorus	-	13	-	1	
Pylorus	999	2	6	2	
Body	-	37	-	15	Body
Cardia	-	22	-	9	Cardia
Duodenal bulb	-	13	-	6	Duodenal bulb
Duodenum	-	71	13	13	Duodenum
Esophagus	94	130	9	22	Esophagus
Esophagus and cardia	-	-	-	3	-
Fundus	-	26	-	4	Fundus
Papilla vateri	-	10	-	-	-
Retroflex stomach	764	-	2	-	Retroflex stomach
Stomach	-	52	3	-	-
Z-line	1 595	-	3	-	Z-line
Pylorus, duodenal bulb, and descending duodenum	-	-	-	1	-

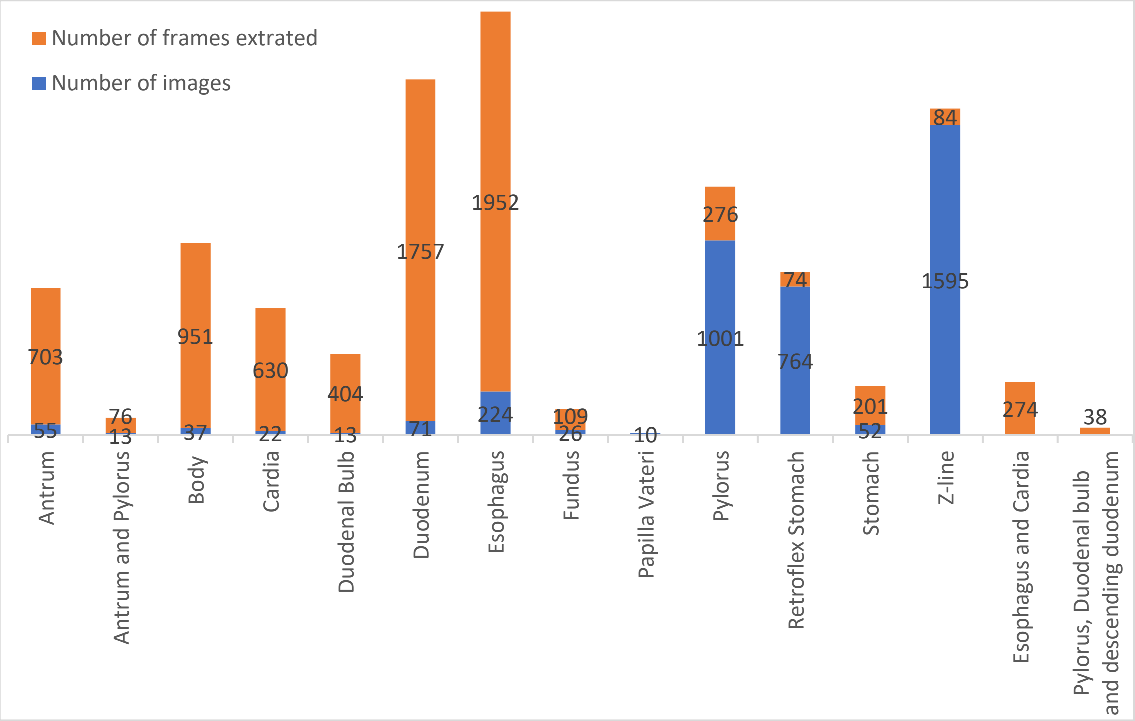


Figure 4. 1 *Total number of images and frames extracted from HyperKvasir and GASTROLAB repositories with respective anatomical zone.*

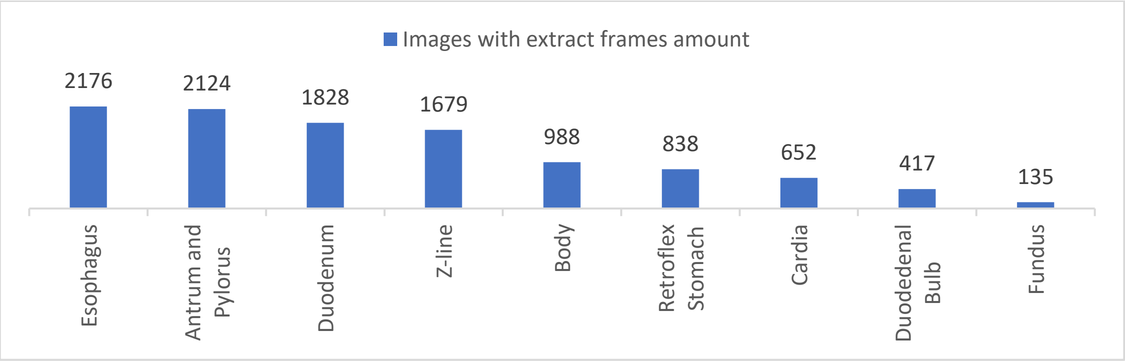


Figure 4. 2 *Final classes distribution (esophagus, antrum and pylorus, duodenum, z-line, body, retroflex stomach, cardia, duodenal bulb, and fundus).*

4.2. Classification models

As a first analysis, we used our dataset to train pre-trained architectures that evidenced good results in the literature to detect upper GI landmarks. The pre-trained architectures that we used with our dataset were ResNet-50, DenseNet-121, and VGG-16. In all experiments, the frames were resized to 224×224 before training the networks. All pre-trained networks were implemented a *Python* framework, using *Tensorflow* (version 2.1.0) and *Keras* (version 2.3.1). The specifications of each experiment are explained in the sections below.

4.2.1. Pre-trained ResNet-50 models, pre-trained DenseNet-121 models, and pre-trained VGG-16 models

- **Pt ResNet-50 experiment** is based on pre-trained ResNet-50 architecture by *HyperKvasir* baseline [78]. We used the same learning rate and batch size, that was respectively 0.001 and 32. However, we adjusted the architecture to our task of classifying 9 different upper GI classes. We took the convolutional layers from pre-trained ResNet-50 with ImageNet weights and froze them to avoid destroying any information. Then, we added a global average pooling layer and a dense layer on top of the frozen layers to predict the respective classes (see Figure 4.3). The model was trained for 40 epochs, and we used Adam optimizer. Adam optimizer is an optimization algorithm that updates the trainable weights based on our dataset, this method can be used instead SGD [83].

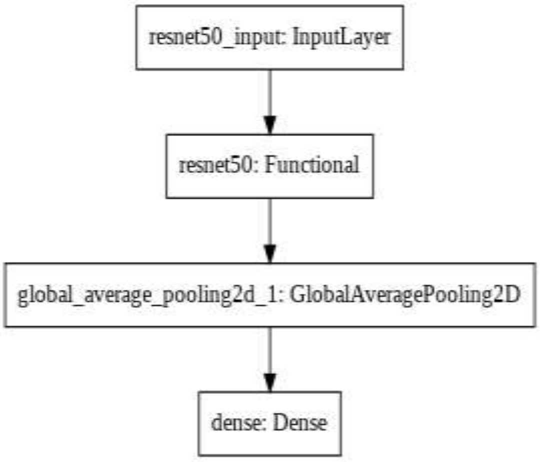

resnet50_input: InputLayer
resnet50: Functional
global_average_pooling2d_1: GlobalAveragePooling2D
dense: Dense

Figure 4. 3 *Pt ResNet-50 architecture.*

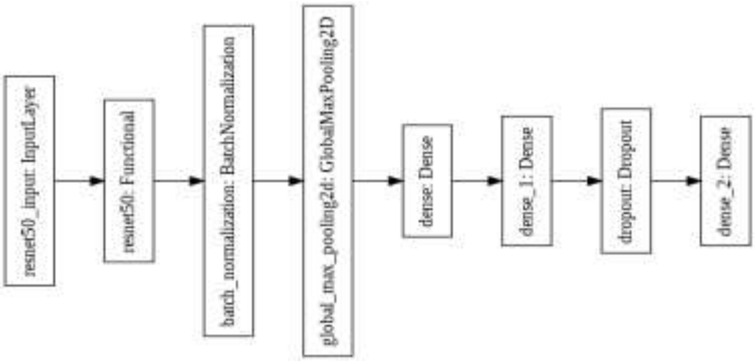
- **CW ResNet-50 experiment** has the same architecture and parameters described in Pt ResNet-50 experiment, we used the same learning rate and batch size, that was respectively 0.001 and 32. The model was trained for 40 epochs, and we used Adam optimizer. However, CW ResNet-50 has the addition of a *class_weight* (CW) parameter in the *fit* method [84], which is used to indicate the model to "pay more attention" to classes with less samples, weighting the loss function during the train model. This is a strategy to try to attenuate the unbalanced dataset effect. The weights were selected according to the inverse of the number of images contained in each class. For example, the esophagus class has approximately 16 times more data than the fundus class, so, the value of *class_weight* for the esophagus class is the inverse of 16 and the value of *class_weight* to the fundus class is 1. Consequently, during the train, the model will "pay more attention" to the fundus class (with less data) than the esophagus class (with more data).
- **DA ResNet-50 experiment** has the same architecture of Pt ResNet-50 experiment, however, the training data is different. DA ResNet-50 architecture was trained with a learning rate of 0.001 and a batch size of 32. The model was trained for 80 epochs, and we used Adam optimizer. In this experiment, the training data is different: we apply data augmentation (DA) to help reduce overfitting.

  Data augmentation application allows dataset balancing with the amount of data increase by adding modified copies of already existing data. We applied 18 random transformations to each class to balance the dataset, described in Table 4.3. We applied Gaussian noise, rotations, flipping, transposing, cropping, adjusting saturation, and adjusting brightness. So, data augmentation was only used to train the network. To validate, we did not use data augmentation. Table 4.3 represents the example of data augmentation that was applied in an experiment. From the observation of Table 4.3 is possible to note that with the application of data augmentation the classes stayed balanced with approximately 2106 images per class.
- **Bn ResNet-50 experiment** was trained with a learning rate of 0.001 and a batch size of 32. The model was trained for 40 epochs, and we used Adam optimizer. We took the convolutional layers from pre-trained ResNet-50 with ImageNet weights and froze them to avoid destroying any information. Then, we added new layers on the top of the pre-trained ResNet-50 base model, Figure 4.4. We added a batch normalization layer (Bn), we swapped the global average pooling layer for the global max pooling layer since max pooling is a noise suppressant. We added two more dense layers and we added a dropout layer to the model.

  The batch normalization layer have the objective to stabilize the network by normalization of the input features, reducing overfitting [85]. The batch normalization allows maintaining the mean of output close to 0 and the output standard deviation close to 1 [86]. The dropout layer is used to a regularization technique with the objective to reduce overfitting, this layer randomly switching some percentage of neurons of the network. The dropout layer dropped some weights and consequently could avoid overfitting. We only switched off 20% of the neurons, because if switching off more than 50% can provoke bad predictions [85].

***Table 4. 3** The number of training images for each class with respective data augmentation transformations.*

Transformations	Classes	Antrum and pylorus	Body	Cardia	Duodenal bulb	Duodenum	Esophagus	Fundus	Retroflex stomach	Z-line
Gaussian noise	**Mean and standard deviation of 0.25**	117	117	117	107	117	117	115	117	117
	Mean and standard deviation of 0.50	117	117	117	107	117	117	115	117	117
	Mean and standard deviation of 1.00	117	117	117	108	117	117	115	117	117
Rotates in 90°		117	117	167	107	117	117	115	117	117
Transpose		117	117	117	107	117	117	115	117	117
Flip	**Horizontal flipping**	117	117	117	108	117	117	115	86	117
	Vertical flipping	117	92	117	107	117	117	115	117	117
Saturation	**Saturation of 1.20**	117	117	167	107	117	117	115	117	117
	Saturation of 1.40	117	117	117	108	117	117	115	117	117
	Saturation of 1.50	117	117	117	107	117	117	115	117	117
	Saturation of 1.70	117	117	117	107	117	117	115	117	173
	Saturation of 2.00	117	117	112	108	117	117	115	86	117
Brightness	**Brightness of 0.10**	117	117	55	107	73	117	115	117	117
	Brightness of 0.20	117	92	117	107	117	117	115	117	117
	Brightness of 0.30	61	117	117	108	117	103	115	117	117
	Brightness of 0.35	117	117	117	107	117	117	115	118	117
	Brightness of 0.40	118	117	112	107	117	117	115	202	117
0.90 central cropping		172	167	55	108	161	131	115	93	61
Total of images		**2106**	**2106**	**2072**	**1932**	**2106**	**2106**	**2106**	**2106**	**2106**



resnet50_input: InputLayer
resnet50: Functional
batch_normalization: BatchNormalization
global_max_pooling2d: GlobalMaxPooling2D
dense: Dense
dense_1: Dense
dropout: Dropout
dense_2: Dense

***Figure 4. 4** Bn ResNet-50 architecture.*

We repeated the previous four experiments for pre-trained DenseNet-121 and pre-trained VGG-16:

- **Pt DenseNet-121 experiment** consists of using convolutional layers from pre-trained DenseNet-121 architecture with ImageNet weights. We froze the convolutional layers to avoid destroying any information. Then, we added a global average pooling layer and a dense layer on top of the frozen layers to predict the respective classes, such as in the Pt ResNet-50 experiment. We used a learning rate of 0.001 and a batch size of 32. The model was trained for 60 epochs, and we used Adam optimizer.

- **CW DenseNet-121 experiment** has the same architecture and parameters described in Pt DenseNet-121 experiment, we used the same learning rate and batch size, that was respectively 0.001 and 32. The model was trained for 60 epochs, and we used Adam optimizer. However, CW DenseNet-121 has the addition of a *class_weight* parameter in the *fit* method. The weights were selected according to the inverse of the number of images contained in each class.
- **DA DenseNet-121 experiment** has the same architecture of Pt DenseNet-121 experiment. DA DenseNet-121 architecture was trained with a learning rate of 0.001 and a batch size of 32. The model was trained for 80 epochs, and we used Adam optimizer. However, the training data is different: we apply data augmentation, such as in DA ResNet-50 experiment (see Table 4.3).
- **Bn DenseNet-121 experiment** was trained with a learning rate of 0.001 and a batch size of 32. The model was trained for 40 epochs, and we used Adam optimizer. We took the convolutional layers from pre-trained DenseNet-121 with ImageNet weights and froze them to avoid destroying any information. Then, we added new layers on the top of the pre-trained DenseNet-121 base model, such as in Bn ResNet-50 experiment. We added a batch normalization layer, we swapped the global average pooling layer for the global max pooling layer. We added two more dense layers and we added a dropout layer to the model.
- **Pt VGG-16 experiment** consists of using convolutional layers from pre-trained VGG-16 architecture with ImageNet weights. We froze the convolutional layers to avoid destroying any information. Then, we added a global average pooling layer and a dense layer on top of the frozen layers to predict the respective classes, such as in the Pt ResNet-50 experiment. We used a learning rate of 0.001 and a batch size of 32. The model was trained for 60 epochs, and we used Adam optimizer.
- **CW VGG-16 experiment** has the same architecture and parameters described in Pt VGG-16 experiment, we used the same learning rate and batch size, that was respectively 0.001 and 32. The model was trained for 60 epochs, and we used Adam optimizer. However, CW VGG-16 has the addition of a *class_weight* parameter in the *fit* method. The weights were selected according to the inverse of the number of images contained in each class.
- **DA VGG-16 experiment** has the same architecture of Pt VGG-16 experiment. DA VGG-16 architecture was trained with a learning rate of 0.001 and a batch size of 32. The model was trained for 80 epochs, and we used Adam optimizer. In this experiment, the training data is different: we apply data augmentation, such as in DA ResNet-50 experiment, Table 4.3.
- **Bn VGG-16 experiment** was trained with a learning rate of 0.001 and a batch size of 32. The model was trained for 40 epochs, and we used Adam optimizer. We took the convolutional layers from pre-trained VGG-16 with ImageNet weights and froze them to avoid destroying any information. Then, we added new layers on the top of the pre-trained VGG-16 base model, such as in Bn ResNet-50 experiment. We added a batch normalization layer, we swapped the global average pooling layer for the global max pooling layer. We added two more dense layers and we added a dropout layer to the model.

4.2.2. Autoencoder-based models

The *HyperKvasir* repository has 99 417 unlabeled images, so, we used these unlabeled images to try to extract representative features of the digestive system via the use of autoencoders architectures. This is an unsupervised learning task because we do not have labels to the images and the objective is to find/learn a high-level and compact representation of the features of the EGD images with autoencoders architectures. The 99 417 unlabeled images from *HyperKvasir* are from the upper and lower GI tract in white light.

Autoencoder is an unsupervised learning technique, which consists of a reconstruction of the original input. Autoencoders covert a higher dimension of input data to a lower dimension code and then reconstruct the original input data using this lower dimension code. The autoencoders are constituted of three parts: encoder, code, and decoder, Figure 4.5. The encoder part covert the input data into a smaller dimension code and the decoder part convert the lower dimension code to the original input [87]. The autoencoder network is trained to minimize the reconstruction error, which measures the differences between original input images and output images (reconstructions of input images). The reconstruction error is the loss function of the autoencoder, which is given by the mean-squared error (MSE) to our experiments [87]. MSE consists of the average of the square of the difference between the input data and the output data [88]: $MSE =$

$\frac{1}{N}\sum_{i=1}^{N}(x - x')^2$, where N is the number of samples, x is the original input, and x' is the reconstruction output. The autoencoder train consists to minimize the difference between the input and the output, $x \approx x'$.

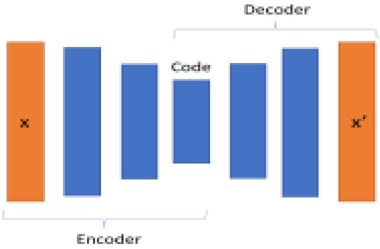


Figure 4. 5 Autoencoder architecture, where x is the input data and x' is the output data (reconstruction of input data).

Autoencoder architectures were tested in our task. The objective of our task consists to train convolutional autoencoder architectures according to the unlabeled GI images, and then use the encoder part, with the weights adjust according to the unlabeled images, to classify labeled EGD images. So, in our work we have two training phases: 1) one to train the convolutional autoencoder and 2) another one to train the classifier that uses features learned by the autoencoder. In the next paragraphs are described the autoencoder architectures experiments:

- **Convolutional encoder (CE experiment)** consists of using the encoder layers from the convolutional autoencoder model to classify labeled EGD images. The idea of this architecture is to use pre-trained layers with weights adjusted according to the GI images instead of using a pre-trained architecture with images from ImageNet, which are not related to digestive anatomical zones.

  The convolutional autoencoder was trained with unlabeled GI images. The encoder part has convolutional layers followed by the max pooling layers. The decoder part has convolutional transpose layers followed by the upsampling layers to increase the dimension of input and consequently reconstructed the input images [89] (see in Figure 4.6 A)). The convolutional autoencoder architecture was trained using 64 and 0.001 as batch size and learning rate, respectively. We used 10 epochs and the Adam optimizer to train.

  Then, we used the encoder layers to classify labeled EGD images - CE architecture (see Figure 4.6 B)). In the CE model, we used a learning rate and a batch size of 0.001 and 32, respectively. We took the encoder layers and froze them to avoid destroying any information. Then, we added the global average pooling layer and a dense layer on top of the frozen layers to predict the respective classes. The model was trained over 60 epochs, and we used the Adam optimizer. The CE model has the addition of a *class_weight* parameter in the fit method.
- **Concatenation of convolutional encoder with CW ResNet-50 (CE + CW ResNet-50 experiment)** consists in concatenating the encoder section (CE model) with the CW ResNet-50 model. We added a global average pooling layer on the top of each concatenation section (CE section and CW ResNet-50 section). Then, with a concatenate layer, we concatenated CW ResNet-50 with CE (see Figure 4.7). Then, we added a dense layer on top of the frozen layers to predict the respective classes, (see Figure 4.7). The model was trained for 40 epochs, and we used Adam optimizer. In the CE + CW ResNet-50 model, we used a learning rate and a batch size of 0.001 and 32, respectively.
- **VGG-16 based encoder experiment** consists to train the encoder layers from autoencoder constructed based in VGG-16 architecture to classify labeled EGD images.

  The autoencoder based on VGG-16 architecture was trained with unlabeled GI images (see Figure 4.8). The autoencoder architecture was trained using 64 and 0.001 as batch size and learning rate, respectively. We used 10 epochs and the Adam optimizer to train.

  Then, we used the encoder layers to classify labeled EGD images - VGG-16 encoder architecture. In the VGG-16 encoder model, we used a learning rate and a batch size of 0.001 and 32, respectively. We took the encoder layers and froze them to avoid destroying any information. Then, we added the global average pooling layer and a dense layer on top of the frozen layers to predict the respective classes. The model was trained over 40 epochs, and we used the Adam optimizer. The VGG-16 encoder model has the addition of a *class_weight* parameter in the fit method.
- **Concatenation of VGG-16 based encoder with CW ResNet-50 (VGG-16 encoder + CW ResNet-50 experiment),** consists in concatenating the encoder section (VGG-16 based encoder), with the CW ResNet-

50 model. We added a global average pooling layer on the top of each concatenation section (VGG-16 based encoder and CW ResNet-50) and then with a concatenate layer, we concatenated CW ResNet-50 with VGG-16 based encoder. Then, we added a dense layer on top of the frozen layers to predict the respective classes. The model was trained for 40 epochs, and we used Adam optimizer. We also used the same learning rate and batch size, 0.001 and 32 respectively.

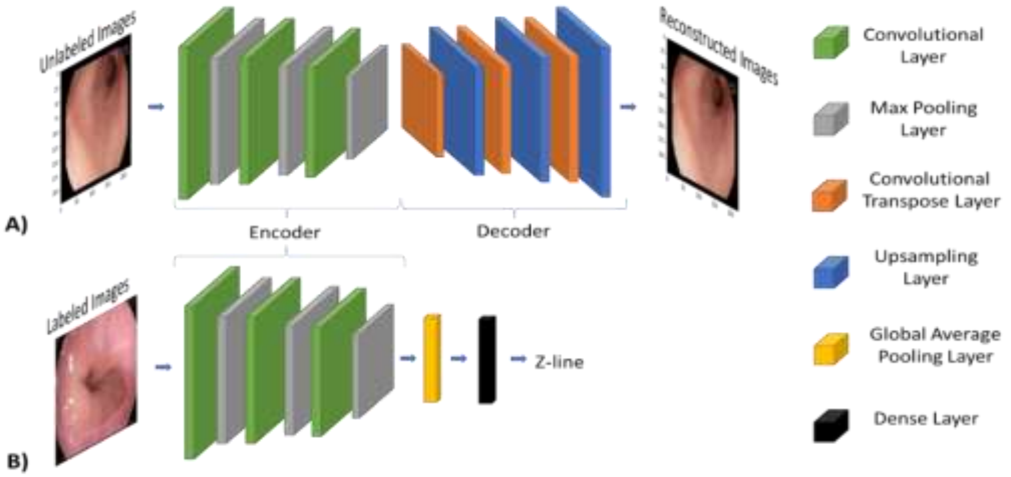


Figure 4. 6 *A) The convolutional autoencoder architecture, which was trained according to the unlabeled GI images from HyperKvasir repository. B) The CE model, which corresponds to the encoder part of A). The CE was trained to classify labeled anatomical zones. We froze encoder layers to avoid destroying any information learning in A).*

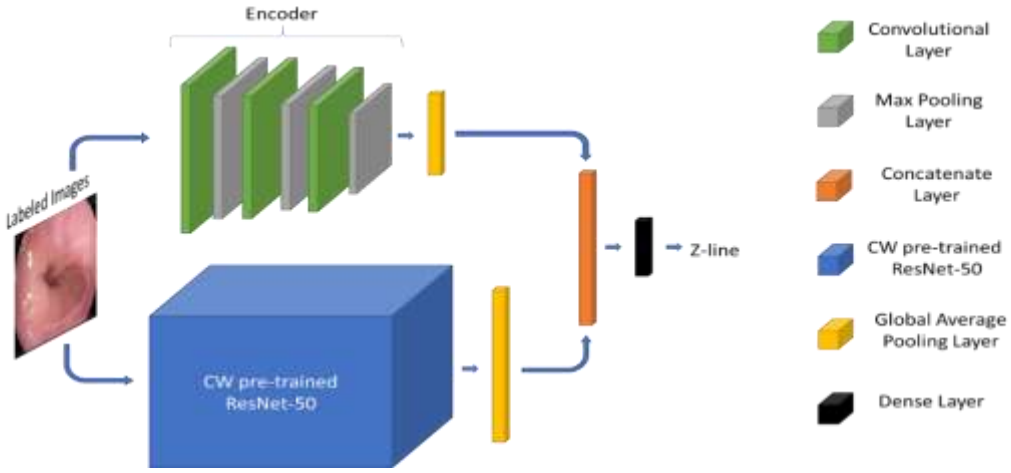


Figure 4. 7 *Concatenation of CE with CW ResNet-50 (CE + CW ResNet-50).*

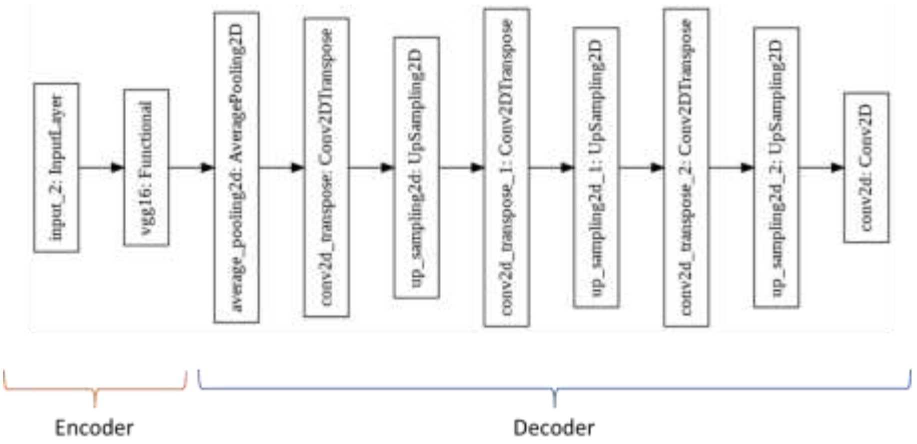
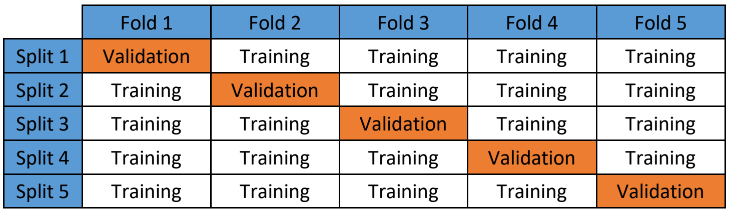

input_2: InputLayer
vgg16: Functional
average_pooling2d: AveragePooling2D
conv2d_transpose: Conv2DTranspose
up_sampling2d: UpSampling2D
conv2d_transpose_1: Conv2DTranspose
up_sampling2d_1: UpSampling2D
conv2d_transpose_2: Conv2DTranspose
up_sampling2d_2: UpSampling2D
conv2d: Conv2D
Encoder
Decoder

Figure 4. 8 *Autoencoder based on VGG-16 architecture.*

CHAPTER 5 EXPERIMENTS

5.1. Experimental methodology

To validate our models, we applied 5-fold cross-validation [90] [91], which consists in dividing the data into five folds and in each split have one validation fold and four training folds, Figure 5.1. In each split, the validation fold is rotative: in the first split, the first fold is used for validation, and the other four folds are used to train the model. In the second split, the second fold is used for validation, and the other four folds are used to train the model. This iterative process is repeated until each fold has been used for validation, which results in five splits, Figure 5.1. The 5-fold cross-validation technique minimizes the bias of the model because instead of making a single split, we make different splits with different validation and training folds combinations. All folds are used for both training and validation and the metrics are more robust since they result from an average of five splits metrics. In our dataset, note that the frames of the same patient are in the same fold.



	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Split 1	Validation	Training	Training	Training	Training
Split 2	Training	Validation	Training	Training	Training
Split 3	Training	Training	Validation	Training	Training
Split 4	Training	Training	Training	Validation	Training
Split 5	Training	Training	Training	Training	Validation

Figure 5. 1 *5-fold cross-validation method.*

The metrics that were used to evaluate the performance of each experiment were macro-average, weighted-average, and micro-average precision, recall, and F1-score for each split. The macro-average and weighted-average calculate metrics for each label and find their average, while micro-average calculate metrics globally by counting the total true positives, false negatives and false positives. The macro-average does not take label imbalance into account while weighted-average does [69].

Additionally, we calculate accuracy, balanced accuracy, and MCC for each split. The balanced accuracy avoids inflated performance estimates on imbalanced datasets [69].

To obtain the final architecture metrics an average of five splits for each metric was performed. The accuracy curves were made to discuss the results. To understand the behavior of individual classes the confusion matrix, precision, recall, and F1-score were calculated for each class. The confusion matrices of each split were summed to obtain the final confusion matrix of the respective experiment.

5.2. Results

In this chapter, the performances of our architectures are shown. The pre-trained ResNet-50, DenseNet-121, and VGG-16 models' performances are described in sections 5.2.1., 5.2.2., and 5.2.3., respectively. In section 5.2.4., the encoder models' performances are discussed. Finally, in section 5.2.5 a brief summary of the results is shown.

5.2.1. Pre-trained ResNet-50 models

Of the four pre-trained ResNet-50 architectures (Pt ResNet-50, CW ResNet-50, Bn ResNet-50 and DA ResNet-50), the one that performed the best was the CW ResNet-50. The performance values of the Pt ResNet-50 architecture are very close to the CW Resnet-50 architecture. However, the pre-trained ResNet-50 architecture improved its performance with the addition of the *class_weight* parameter: the MCC rises from 64.35% to 65.06% and the weighted-average of the precision, of the recall, and of the F1-score are a litter superior in the CW ResNet-50 architecture (see Table 5.1). However, it is perceptible that the *class_weight* parameter does not improve considerably the performance of ResNet-50 architecture because the metrics only differ less than 1% (see Table 5.1).

In Figure 5.2 A) and Figure 5.2 B) are represented the sum of the 5 splits confusion matrices of Pt ResNet-

50 and CW ResNet-50 experiments, respectively. With confusion matrices comparison we conclude that Pt ResNet-50 and CW ResNet-50 architecture have very similar confusion matrices, concluding that the addition of *class_weight* parameter does not guarantee a significantly higher number of true positives.

Looking at the confusion matrices in Figure 5.2 A) and Figure 5.2 B), it is clear that some classes are harder to identify than others. The duodenal bulb class is confused with the antrum and pylorus class with a percentage of 35.49%. This may be due to the fact of these classes belonging to successive anatomical zones. Although these anatomical zones are not similar, there are some video labeled as antrum and pylorus which contain frames of the duodenal bulb. In CW ResNet-50 architecture, 40.74% of the fundus class frames are confused with the body class. This surrounding classification can be due to fundus and body zones belonging to successive anatomical zones and both classes containing equal pathologies, which contributes to the classes having similar frames.

The duodenal bulb and fundus classes have the worst performances, they have only 17.89% and 40.57% of F1-score, respectively, in CW ResNet-50. These classes only contain *GASTROLAB* repository frames, which have worse visual quality than *HyperKvasir* repository frames, which could contribute to the lower performance of these classes. On the other hand, in both classes, most of the frames are pathologies, which makes the classification process more difficult. In all splits, the duodenal bulb class performs the worst. However, the fundus class does not have the same behavior in all splits. In split 3, the F1-score is 71.11% while in split 4 is 0.00%, in CW ResNet-50 architecture. This may be because the split 4 validation fold contains only frames from one patient with a pathology that is not represented in the remaining folds, which contributes to the model not being able to identify the Fundus class in split 4, and this reduces the F1-score weighted average of five splits from Fundus class.

On the other hand, retroflex stomach and z-line classes have the best performances, 92.64% and 89.85% F1-score respectively. These classes contain only frames from the *HyperKvasir* repository, which has a higher visual quality. Besides that, these classes, relatively the other classes, contain a low number of frames extracted from videos (see Figure 4.1 in 4.1 section), which could explain the better performance of these classes. The images have better visual quality because many extracted frames are blurry due to the video movement.

The behavior of the accuracy curves during training are similar in all splits (see Figure 5.3). However, split 3 is the one that has the best performance, reaching an accuracy of 76.42% in CW ResNet-50 architecture. On the other hand, split 1 performance is the worst, having an accuracy of 61.90%. However, with accuracy curves observation, it is perceptible that in all splits the trained models are affected by overfitting.

To try to improve the performance of the Pt ResNet-50 architecture, data augmentation was employed. The use of data augmentation in the training of Pt ResNet-50 architecture did not improve the performance, which have a MCC of 60.90%. Contrary to what we expected, the Bn ResNet-50 architecture has the worst performance, the MCC was only 37.03%. So, the CW ResNet-50 architecture continues to have the best performance of all pre-trained ResNet-50 experiments.

Table 5. 1 *Metrics values of Pt ResNet-50 and CW ResNet-50 models.*

Models	Weighted-average			Balanced Accuracy	MCC
	Precision	Recall	F1-score		
Pt ResNet-50	69.55%	69.53%	68.56%	59.89%	64.35%
CW ResNet-50	70.65%	70.01%	69.26%	61.50%	65.06%

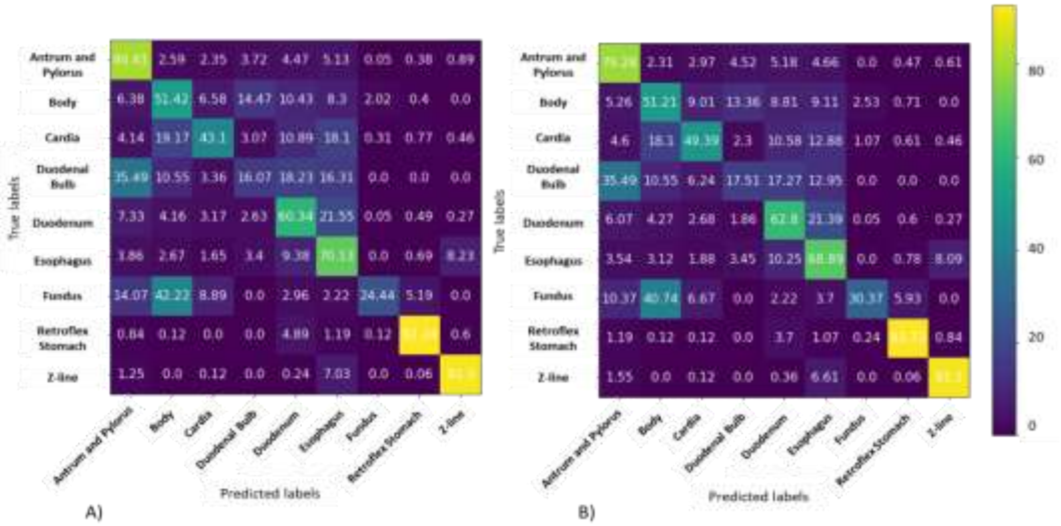


Figure 5. 2 *A) Sum of the 5 splits confusion matrices of Pt ResNet-50 model. B) Sum of the 5 splits confusion matrices of CW ResNet-50 model. The confusion matrices are in terms of percentage.*

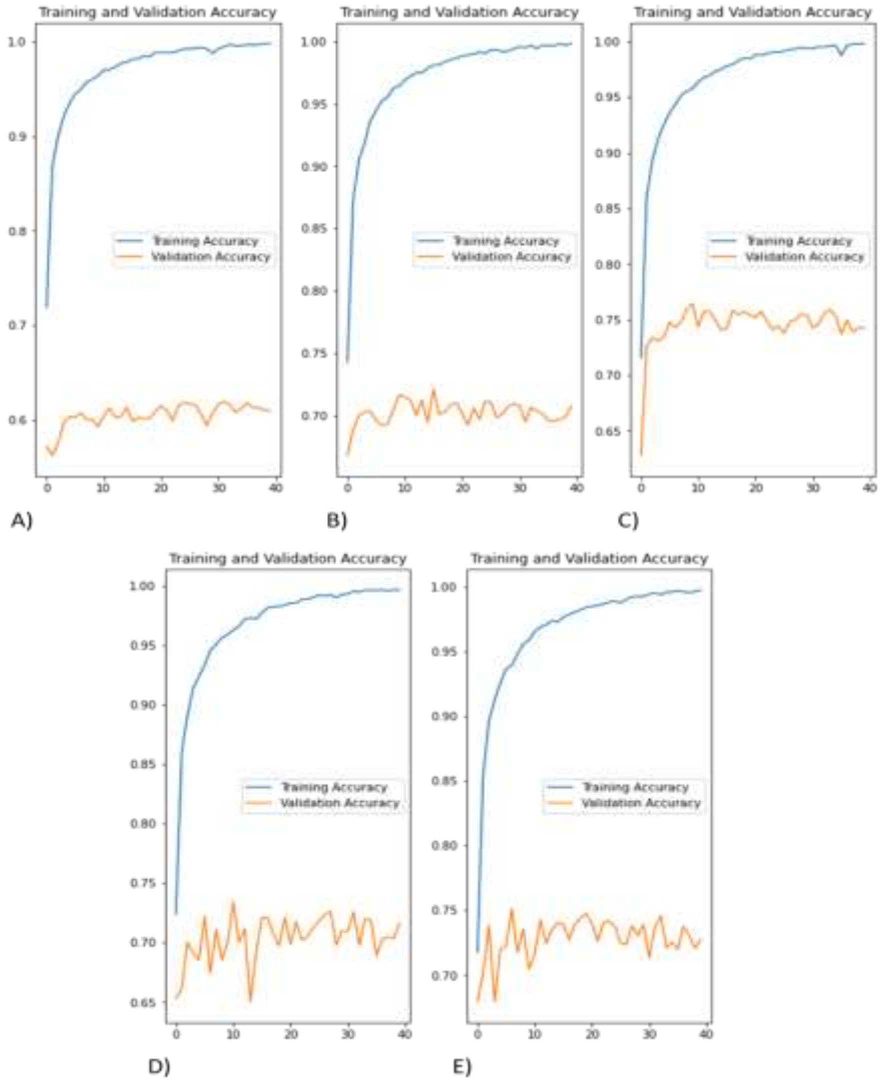


Figure 5. 3 *Accuracy curves during the train of A) split 1, B) split 2, C) split 3, D) split 4 and E) split 5 in CW ResNet-50 model.*

5.2.2. Pre-trained DenseNet-121 models

The Pt DenseNet-121 and the CW DenseNet-121 architectures have very similar performance values. However, unlike ResNet-50 architectures, the Pt DenseNet-121 architecture performed better than CW DenseNet-121. In this case, the *class_weight* parameters actually decrease the performance of the DenseNet-121 architecture, since the MCC drops from 56.40% to 54.85% (see Table 5.2). In addition, the weighted-average of the precision, of the recall, and of the F1 -score are superior in the Pt DenseNet-121 model (see Table 5.2).

Comparing the confusion matrices of the Pt ResNet-50 and CW ResNet-50 architectures with Pt DenseNet-121 and CW DenseNet-121 architectures, it is verified that the ResNet-50 architectures performed better, as shown in Figure 5.2 A), Figure 5.2 B) and Figure 5.4 A), Figure 5.4 B), respectively.

As for the ResNet-50 architecture, the duodenal bulb class continues to be confused with the antrum and pylorus class with a considerable percentage, 34.05% with the Pt DenseNet-121, as shown Figure 5.4 A). It is also noticeable that this duodenal bulb class is classified as the esophagus class with a percentage of 20.62%, and only 15.59% of frames are classified correctly. The frames of the duodenal bulb class may be confused with esophagus frames because they contain similar frames, as both zones appear as a tubular structure with an orifice. In all splits, the duodenal bulb class is confused with antrum and pylorus class except for split 4 where 86.11 % of the frames of this class are confused with the class esophagus. Only in splits 1 and 4, there is greater confusion between the duodenal bulb and the esophagus.

On the other hand, with the Pt DenseNet-121 architecture, the fundus class is less confused with the body class when compared to the CW ResNet-50 architecture, 8.15%, and 40.74%, respectively (Figure 5.4 A) and Figure 5.2 B)). However, with the Pt DenseNet-121 architecture, the fundus class is confused with the esophagus class with a percentage of 28.15% and splits 1 and 4 are the ones that contribute the most to this percentage.

There are several classes that are confused with the esophagus class with the Pt DenseNet-121. For example, the duodenum class is also confused with the esophagus class in split 1 with a percentage of 88.89%, which may be because there are frames with a similar structure to the esophagus since both are tubular zones.

On the other hand, the cardia class is confused with the esophagus class with a percentage of 23.93% in Pt DenseNet-121, as shown in Figure 5.4 A). Split 4 is the split that contributes most to this confusion because 83.21% of frames in split 4 are confused with the esophagus class. In split 4 the cardia validation data is more similar to the esophagus training data because the majority of cardia training data have the endoscope visible in the frames while in cardia validation data the endoscope is not visible. Besides that, the cardia and esophagus are proximal zones so the different views of frames could provoke sabotage.

On the other hand, the Pt DenseNet-121 architecture has more difficulty than CW ResNet-50 architecture in distinguishing the cardia and body classes. The Pt DenseNet-121 confuses 22.85% of cardia frames to body class and CW ResNet-50 confuses 18.10%. These classes contain very similar frames because they are successive anatomical zones and are captured in coincident views (retroflex and antegrade views). The cardia and body classes have different views in the same class which can be difficult for the model to learn.

In the Pt DenseNet-121 architecture, the classes that present the best performances are retroflex stomach and z-line, similarly to what happened with the CW ResNet-50 architecture, with F1-scores of 83.52% and 90.26% respectively. The worst classes are the fundus and duodenal bulb with F1-scores of 26.28% and 12.03% respectively.

In all splits the accuracy curves have the same behavior. Split 3 is the one with the best performance, reaching an accuracy of 71.59%, and split 1 having the worst performance, reaching only 56.43% (see Figure 5.5). However, with accuracy curves observation, it is perceptible that in all splits the trained models are affected by overfitting.

To try to improve the performance of Pt DenseNet-121 architecture, data augmentation was employed. The use of data augmentation in the training of Pt DenseNet-121 architecture did not improve the performance, which have a MCC of 47.60%. Once again, the dropout and Bn layers did not improve the performance of the architecture (MCC of 50.65%).

Table 5. 2 Metrics values of Pt DenseNet-121 and CW DenseNet-121 models.

Models	Weighted-average			Balanced Accuracy	MCC
	Precision	Recall	F1-score		
Pt DenseNet-121	64.50%	62.46%	61.93%	52.46%	56.40%
CW DenseNet-121	63.82%	61.04%	61.17%	52.78%	54.85%

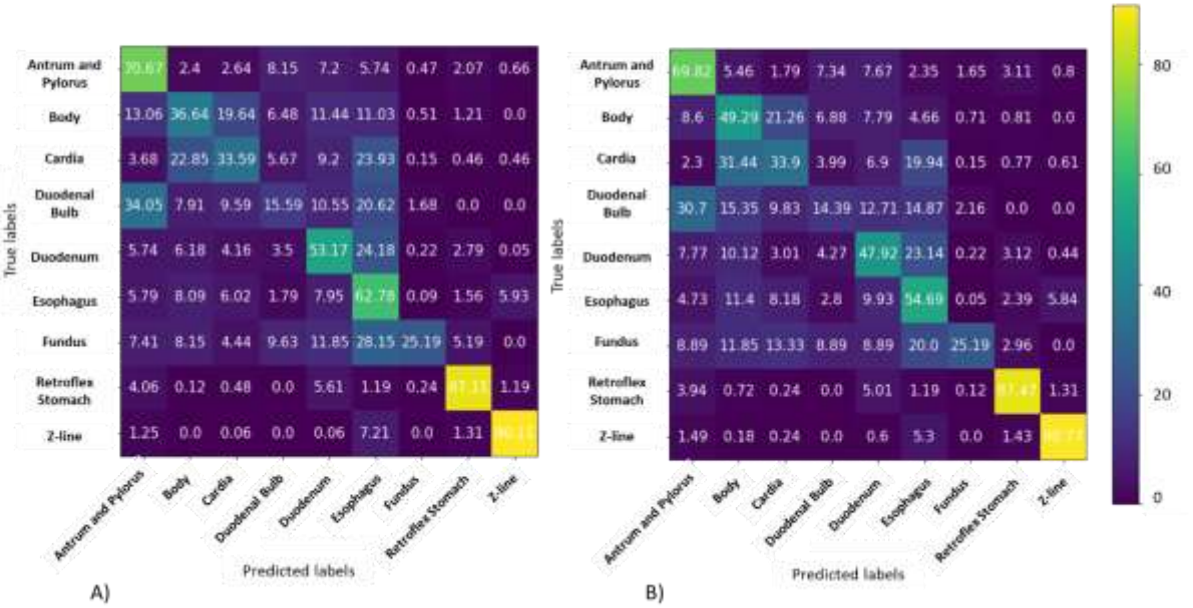


***Figure 5. 4** A) Sum of the 5 splits confusion matrices of Pt DenseNet-121 model. B) Sum of the 5 splits confusion matrices of CW DenseNet-121 model. The confusion matrices are in terms of percentage.*

5.2.3. Pre-trained VGG-16 models

The Pt and CW VGG-16 architectures are the ones with the best performance compared with the other VGG-16 architectures. The Pt and CW VGG-16 architectures present very similar performances, they have MCC values of 60.15% and 60.03% respectively (see Table 5.3). However, CW VGG-16 architecture presents better balanced accuracy, precision, recall, and F1-score values (see Table 5.3).

In Figure 5.6 A) and Figure 5.6 B) are represented the sum of the 5 splits confusion matrices of Pt VGG-16 and CW VGG-16, respectively. With confusion matrices comparison we conclude that Pt VGG-16 and CW VGG-16 architecture have very similar confusion matrix, concluding that the addition of *class_weight* parameter does not guarantee a significantly higher number of true positives.

Observing the CW VGG-16 confusion matrix in Figure 5.6 B), only 41.41% of the frames of the cardia class are correctly classified (15.18% of frames are confused with the esophagus and 14.57% of frames are confused with the duodenum). However, with the CW VGG-16 architecture, the cardia class is less confused with the body class (13.96% of frames), while with the CW ResNet-50 and Pt DenseNet-121 architectures are confused 18.1% and 22.85% of frames, respectively (Figure 5.2 B) and Figure 5.4 A)).

The duodenal bulb class continues to be confused with the antrum and pylorus class (20.62% of frames), but in a lower percentage than CW ResNet-50 and Pt DenseNet-121, 35.49% and 34.05% respectively (Figure 5.2 B) and Figure 5.4 A)).

On the other hand, in the CW VGG-16 architecture, the duodenal bulb class is also confused with the duodenum with a considerable percentage of 26.38%, as they are zones close to each other.

The duodenal bulb class is also confused with the esophagus (16.07% of frames), but in a lower percentage than the Pt DenseNet-121 architecture (20.62% of frames).

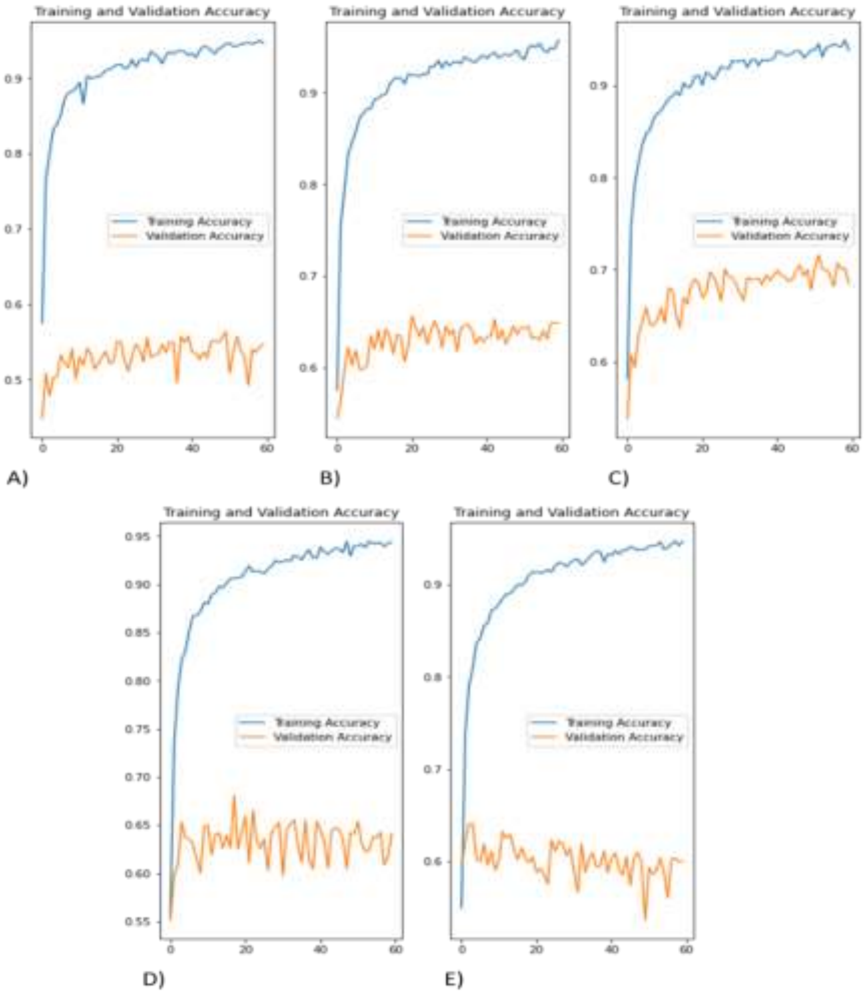


Figure 5. 5 Accuracy curves during the train of A) split 1, B) split 2, C) split 3, D) split 4 and E) split 5 in Pt DenseNet-121 model.

It is also notorious the confusion between the duodenum and esophagus class (23.19% of frames), but in a lower percentage than Pt DenseNet-121 (24.18% of frames).

Similarly to the CW ResNet-50 architecture, the confusion between the fundus class and body class is notorious, but in a smaller percentage, from 40.74% to 31.11% respectively.

In general, as with ResNet-50 and DenseNet-121 architectures, in VGG-16 the classes confused each other because they are proximal zones; both classes have the same pathologies; both classes have coincident views; both classes contain tubular zones; and some videos of different classes contain the respective anatomical zone and the successive anatomical zone.

In the Pt VGG-16 architecture, the classes that present the best performances are retroflex stomach and z-line, similarly to what happened with the ResNet-50 and DenseNet-121 architectures, with F1-scores of 87.22% and 85.15% respectively. The worst classes are the fundus and duodenal bulb with F1-scores of 26.23% and 20.18% respectively.

In all splits, the accuracy curves present the same behavior, with split 5 being the one with the best performance, reaching an accuracy of 73.39%, and split 1 having the worst performance, reaching only 59.56% (see Figure 5.7). However, with accuracy curves observation, it is perceptible that in all splits the trained models are affected by overfitting.

The use of data augmentation in the training of Pt VGG-16 architecture did not improve the performance (MCC of 57.05%) in the same way as the pre-trained ResNet-50 and pre-trained and DenseNet-121. The addition of the dropout and Bn layers did not improve the performance of the VGG-16 architecture. The Bn VGG-16 architecture presents worse performances (MCC of 52.80%) than Pt and CW VGG-16 architectures.

However, comparing the Bn VGG-16 with the Bn ResNet-50 and the Bn DenseNet-121, the Bn VGG-16 architecture is the one with better results.

Table 5. 3 *Metrics values of Pt VGG-16 and CW VGG-16.*

Models	**Weighted-average**			**Balanced Accuracy**	**MCC**
	Precision	**Recall**	**F1-score**		
Pt VGG-16	65.81%	66.01%	65.03%	56.54%	60.15%
CW VGG-16	66.15%	66.02%	65.45%	57.13%	60.03%

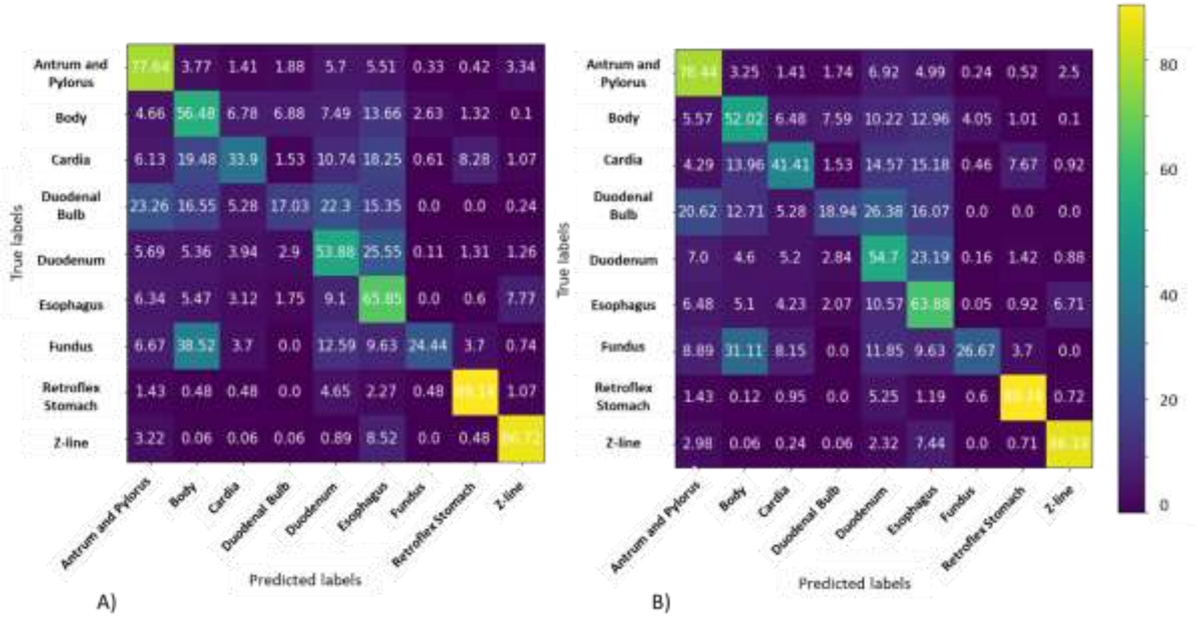


Figure 5. 6 *A) Sum of the 5 splits confusion matrices of Pt VGG-16 model. B) Sum of the 5 splits confusion matrices of CW VGG-16 model. The confusion matrices are in terms of percentage.*

5.2.4. Autoencoder-based models

The CW ResNet-50 is the model with the best performance of all previous experiments. However, to improve the CW ResNet-50 performance, we decided to concatenate encoder architectures: CE + CW ResNet-50 and VGG-16 encoder + CW ResNet-50. The encoder layers were pre-trained with GI unlabeled images. The idea of encoder models is to use encoder parts pre-trained with anatomical images instead of images from ImageNet that are not similar to EGD images.

The CE and VGG-16 encoder architectures without concatenation do not present good performance, however their performances improve considerably with concatenation (see Table 5.4). In the case of CE it is possible to observe that the concatenation increases the metrics values to approximately double, for example, the MCC value increases from 32.85% to 64.88% (see Table 5.4). The VGG-16 encoder has a MCC of 40.10%, and with the concatenation with CW ResNet-50 increase to 65.08% (see Table 5.4).

It would be expected that VGG-16 encoder architecture had more impact than CE architecture, because the VGG-16 encoder architecture was pre-trained with 14 714 688 trainable parameters (with unlabeled GI images). While with CE was pre-trained with only 2 192 parameters.

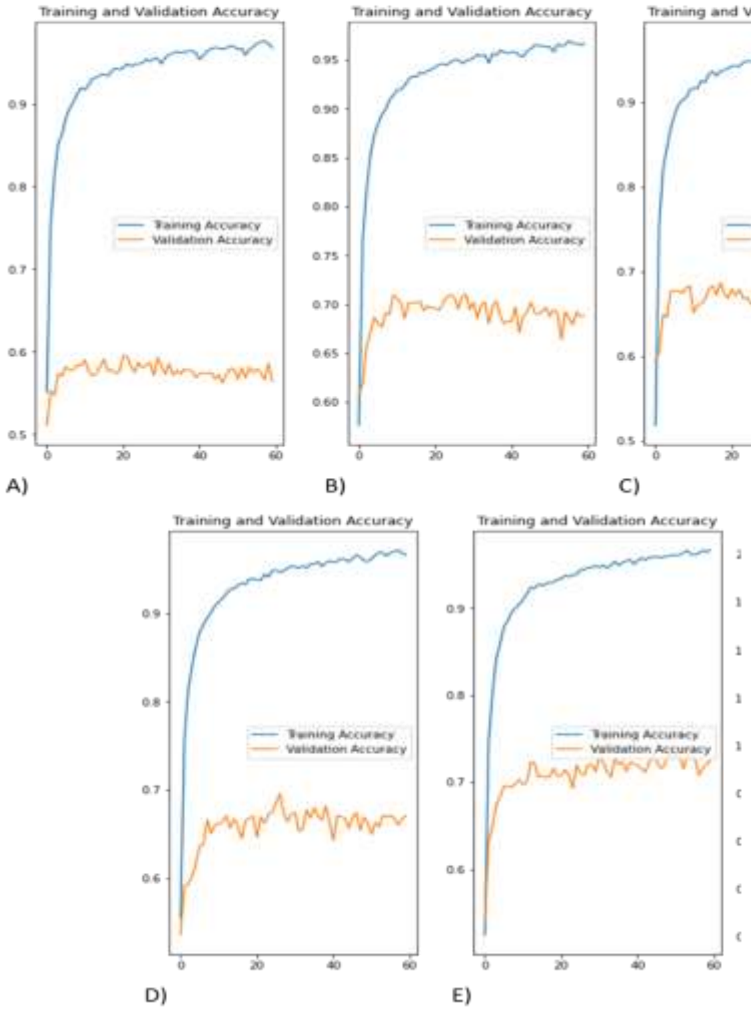


Figure 5. 7 Accuracy curves during the train of A) split 1, B) split 2, C) split 3, D) split 4 and E) split 5 in CW VGG-16 model.

The concatenations do not improve the CW ResNet-50 classification of anatomical zones. Comparing the average metrics values of concatenation models with the CW ResNet-50 model, it is visible that the values are close, which indicates that the encoder layers do not improve significantly our classification task, as seen in Table 5.4. The balanced accuracy of CE + CW ResNet-50, VGG-16 encoder + CW ResNet-50 and CW ResNet-50 are 61.32%, 60.60% and 61.50% respectively. The metrics values represented in Table 5.4, show that CE + ResNet-50 and VGG-16 encoder + ResNet-50 architectures only have less than 1% of the difference in their performances, which proves that the junction of encoder layers with CW ResNet-50 do not have a notable impact.

Comparing the confusion matrices of the CE + CW ResNet-50, Figure 5.8 A), and VGG-16 encoder + CW ResNet-50 models, Figure 5.8 B), the percentages of frames classified in the respective classes are similar. It is possible to note that the duodenal bulb class is confused quite frequently with the antrum and pylorus class, 30.70% and 34.29% using the CE + CW ResNet-50 and VGG-16 encoder + CW ResNet-50 respectively. A similar trend applies for the fundus class, which is often confused with the body class, 38.52% and 51.11% of the frames, at CE + CW ResNet-50 and VGG-16 encoder + CW ResNet-50 respectively. In general, the confused classes are the same that were confused with CW ResNet-50 architecture, which proves one more time that the encoder layers do not help our task.

Figure 5.9 shows the accuracy curves of the 5 splits for the VGG-16 encoder + CW ResNet-50 architecture, split 3 is the one that achieves a higher accuracy value, 76.99 %, and split 1 is the one that achieves a lower accuracy value, 62.27%. With accuracy curves observation, it is perceptible that all splits have overfitting as happened in others experiments.

Table 5. 4 *Metric values of the CE, VGG-16 encoder, CE + CW ResNet-50, VGG-16 encoder + CW ResNet-50 and CW ResNet-50 models.*

Models	Weighted-average			Balanced Accuracy	MCC
	Precision	Recall	F1-score		
CE	39.27%	32.04%	32.45%	36.91%	32.85%
VGG-16 encoder	54,42%	46,75%	47,19%	45,29%	40,10%
CE + CW ResNet-50	71.13%	69.77%	69.11%	61.32%	**64.88%**
VGG-16 encoder + CW ResNet-50	70.81%	70.01%	69.47%	60.60%	**65.08%**
CW ResNet-50	70.65%	70.01%	69.26%	61.50%	**65.06%**

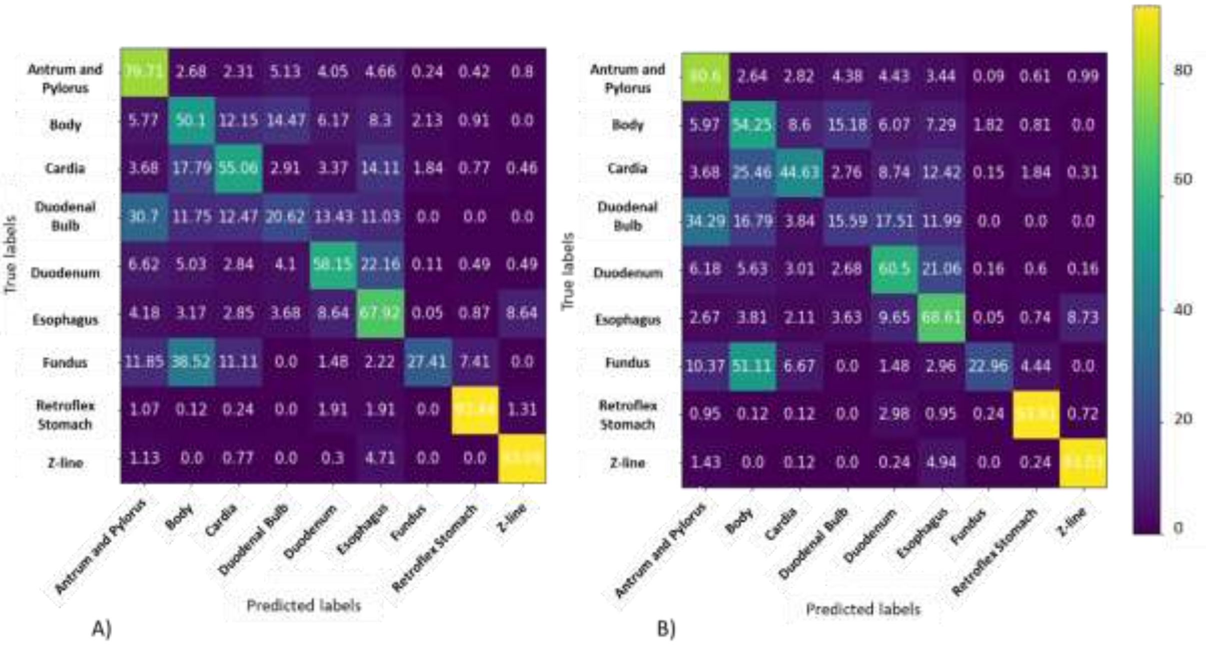


Figure 5. 8 *A) Sum of the 5 splits confusion matrices of CE + CW ResNet-50 model. B) Sum of the 5 splits confusion matrices of VGG-16 encoder + CW ResNet-50. The confusion matrices are in terms of percentage.*

5.2.5. Summary of the results

Analyzing Table 5.5, it is verified that among ResNet-50, DenseNet-121 and VGG-16 architectures (without encoder layers), the model that presents the best performance is the CW ResNet-50, then the Pt ResNet-50 architecture, and the third-best architecture is DA ResNet-50. These architectures present the following MCC values respectively, 65.06%, 64.35%, and 60.90%. Therefore, the ResNet-50 architectures are the ones that best result in the classification of anatomical zones. VGG-16 architectures perform better than DenseNet-121 architectures. In general, the application of data augmentation and the addition of dropout and Bn layers worsened the results of pre-trained ResNet-50, pre-trained DenseNet-121, and pre-trained VGG-16 architectures. On the other hand, the *class_weight* parameter did not harm the performance of pre-trained architectures but had little impact, slightly improving classification performance, as for example, in the ResNet-50 architecture.

In order to improve CW ResNet-50, we made encoder parts concatenations, which did not have the expected impact on CW ResNet-50 architecture performance. However, concatenation models have better performance than Pt, DA ResNet-50 and Bn ResNet-50 architectures (see Table 5.5). The encoder architecture that had better results was VGG-16 encoder + CW ResNet-50 with 65.08% of MCC, which is a litter better than the CW ResNet-50 experiment (MCC of 65.06%).

Through accuracy curves, it is perceptible that overfitting occurs in all models. In a general way with a confusion matrix, we conclude that the classes are confused because they are subsequent anatomical zones; they have equal pathologies, which contributes to the classes having similar frames; sometimes the classes have coincident views; the tubular zones sometimes are confused, like esophagus and duodenum; and some videos of different classes contain the respective labeled anatomical zone and the successive anatomical zone. Some classes have many types of images inside the respective class, which is referred to as intra-class variability. This complicates the model task because inside of one class there are pathologies, healthy tissue, different views and quadrants, and blurry frames. It is important to represent the classes with this variability because all these scenarios occur during the EGD exam, however, it makes our task more difficult.

On another hand, it is perceptible that the classes where most data are from *GASTROLAB* repository have less visual quality and consequently the respective classes are more confounded with other classes, like duodenal bulb and fundus. In contrast, the classes that contain majority *HyperKvasir* data have the best performances, like z-line and retroflex stomach classes. Besides that, z-line, and retroflex stomach classes relatively the other classes have a bigger portion of images comparative to the extracted frames portion. The images have better visual quality because many extracted frames are blurry due to the video movement, which could explain the better performance of these classes.

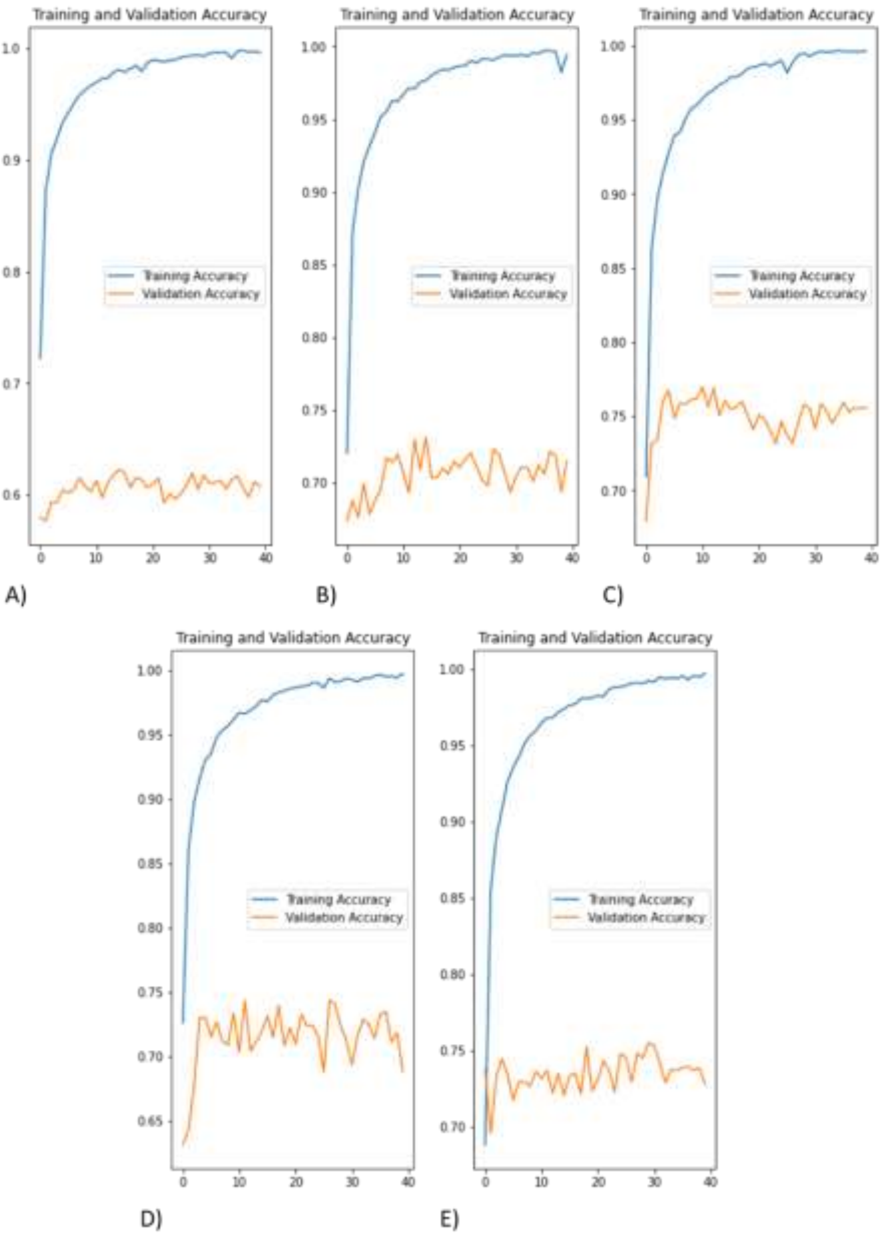


Figure 5. 9 *Accuracy curves during the train of A) split 1, B) split 2, C) split 3, D) split 4 and E) split 5 in VGG-16 encoder + CW ResNet-50 model.*

Table 5. 5 *Average metrics of all splits of the pre-trained ResNet-50 models, DenseNet-121 models, VGG-16 models, and encoder models.*

Models		Macro-average			Weighted-average			Micro-average			Accuracy	Balanced Accuracy	MCC
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score			
ResNet-50	Pt	63.22%	59.89%	59.91%	69.55%	69.53%	68.56%	69.53%	69.53%	69.53%	72.40%	59.89%	**64.35%**
	DA	57.75%	57.80%	56.20%	65.57%	66.47%	64.91%	66.47%	66.47%	66.47%	69.56%	57.80%	60.90%
	CW	64.10%	61.50%	61.50%	70.65%	70.01%	69.26%	70.01%	70.01%	70.01%	71.79%	61.50%	**65.06%**
	Bn	32.54%	33.49%	31.18%	43.15%	47.61%	43.00%	47.61%	47.61%	47.61%	49.78%	33.49%	37.03%
DenseNet-121	Pt	54.93%	52.46%	52.07%	64.50%	62.46%	61.93%	62.46%	62.46%	62.46%	65.16%	52.46%	56.40%
	DA	45.80%	47.39%	44.64%	58.49%	54.54%	55.01%	54.54%	54.54%	54.54%	59.61%	47.39%	47.60%
	CW	53.36%	52.78%	51.63%	63.82%	61.04%	61.17%	61.04%	61.04%	61.04%	64.29%	52.78%	54.85%
	Bn	43.99%	43.90%	42.56%	56.13%	57.75%	55.58%	57.75%	57.75%	57.75%	59.90%	43.90%	50.65%
VGG-16	Pt	58.88%	56.54%	56.22%	65.81%	66.01%	65.03%	66.01%	66.01%	66.01%	68.19%	56.54%	60.15%
	DA	53.88%	56.26%	53.47%	63.69%	62.95%	62.32%	62.95%	62.95%	62.95%	65.45%	56.26%	57.05%
	CW	58.85%	57.13%	56.89%	66.15%	66.02%	65.45%	66.02%	66.02%	66.02%	68.44%	57.13%	60.03%
	Bn	51.01%	50.93%	49.66%	60.23%	59.65%	59.06%	59.65%	59.65%	59.65%	62.84%	50.93%	52.80%
Encoder	CE	30,63%	30,29%	27,09%	39,27%	32,04%	32,45%	32,04%	32,04%	32,04%	45,45%	36,91%	32,85%
	VGG-16 encoder	43,28%	45,29%	40,14%	54,42%	46,75%	47,19%	46,75%	46,75%	46,75%	50,47%	45,29%	40,10%
	CE + CW ResNet-50	62,85%	61,32%	60,58%	71,13%	69,77%	69,11%	69,77%	69,77%	69,77%	72,14%	61,32%	**64,88%**
	VGG-16 encoder + CW ResNet-50	63,58%	60,60%	60,55%	70,81%	70,01%	69,47%	70,01%	70,01%	70,01%	72,45%	60,60%	**65,08%**

CHAPTER 6 CONCLUSION

6.1. Final remarks

The aim of this work was to test new systems able to monitor blind spots during the EGD exam that use CNNs to classify landmarks. In state-of-art studies, only supervised learning approaches were used to classify EGD images, and in our work, we used unsupervised learning to help our task. This choice was motivated by the fact that, the *HyperKvasir* repository has a considerable number of GI unlabeled images, which we used for our unsupervised learning task.

Besides that, in state-of-art studies, the anatomical landmarks classes only considered healthy tissue, whereas the problem tackled in this work present a higher intra-class variability (different pathologies, different views, different quadrants, etc.). In this sense, this work represents a significant step towards the application of CNN-based architectures for anatomical landmark detection in real-world environments.

First, we considered the use of different CNN architectures to classify anatomical landmarks from EGD exam. We considered pre-trained ResNet-50, pre-trained DenseNet-121, and pre-trained VGG-16 architectures and by the transfer learning techniques, we adjust the pre-trained architecture to our task. To reduce overfitting, we applied different strategies: we used class weights to reduce the unbalanced data effect; we applied data augmentation to balance the data; we applied dropout and batch normalization layers to stabilize the networks. However, the only technique that did not harm the networks' performance was the application of class weights. The architecture with better performance was CW ResNet-50 with 65.06% of MCC and accuracy of 71.79%. In the study by Borgli et al. [78], which uses the *HyperKvasir* dataset the ResNet-50 has an MCC of 82.60%. However, in study of Borgli et al. [78] the classes were different, they classify pathologies and healthy zones separately.

Further different techniques have been applied in the attempt to reduce overfitting, including the use of latent space representations learned from a large dataset of unlabeled images. We applied unsupervised learning with convolutional autoencoder architectures, concatenated encoder layers of pre-trained convolutional autoencoder architectures with CW ResNet-50 architecture. This allowed to use layers with weights pre-trained using GI images, which was expected to guarantee improvements in the CW ResNet-50 architecture. However, the use of features learned in an unsupervised fashion via the application of autoencoder did not provide a significant boost in the classification performance. We obtained a performance of 65.08% MCC and 72.45% accuracy with VGG-16 encoder + CW ResNet-50, which is closer to CW ResNet-50 performance.

In all experiments is perceptible the confusion between classes. The classes that are mostly confused correspond to subsequent anatomical zones; they have the similar pathologies, which contributes to the classes having similar frames; sometimes the classes have coincident views; and the tubular zones sometimes are confused, like esophagus and duodenum. In addition, it is possible to note that the classes where most data are from *GASTROLAB* repository have less visual quality and consequently the respective classes are more confounded with other classes, like the duodenal bulb and fundus.

This work is differentiating from the other studies because we joined supervised and unsupervised learning (encoder layers with ResNet-50) to classify EGD landmarks. Besides that, we used landmarks classes closer to reality (such as, with and without pathologies).

6.2. Future work

The first future goal would be to improve the quality of the dataset by provide more refined labels for the images, thus considering a higher number of classes. This process would lower the intra-class variability of the data, thus possibly allowing to better classification performance. For example, as the class cardia includes at this moment images from different views. It would be interesting instead to use one class for cardia, to use two different classes for cardia (antegrade view cardia and retroflex view cardia). This will facilitate our task without losing any information. However, this idea implies having a specialist in the EGD exam available to provide refined annotations.

The second future goal is to develop a system that uses CNN with Recurrent Neural Networks (RNNs) to classify GI landmarks, thus performing classification leveraging the information contained in various, contiguous video frames. The objective is to use RNNs to combine the information extracted from the sequence of images in order to improve the classification. This is motivated by the observation that neighboring images collected during a sequential EGD are highly correlated, especially when the endoscopist is attempting to abide by the rules of a specific exam protocol. In this sense, RNNs can be used as a valuable tool to classify sequences of features extracted from neighboring images, thus leveraging all the contextual information. Thus, we will build a DL model that uses convolutions to train/extract the features that best characterize and define the images and uses the RNN architecture to consider the temporal/sequential dimension of images.

REFERENCE

[1] World Health Organization, “CANCER TODAY,” *Internacional Agency for Reasearch on Cancer*, 2018. https://gco.iarc.fr/today/online-analysis-multi-bars?v=2018&mode=cancer&mode_population=countries&population=900&populations=900&key=total&sex=0&cancer=39&type=0&statistic=5&prevalence=0&population_group=0&ages_group%5B%5D=0&ages_group%5B%5D=17&nb_items=1.

[2] A. Goulart, “CANCRO DO ESTÔMAGO: FATORES DE RISCO, INVESTIGAÇÃO E TRATAMENTO,” 2019. https://www.hospitaldatrofa.pt/noticias-e-eventos/noticias/cancro-do-estômago-fatores-de-risco-investigação-e-tratamento/.

[3] M. Venerito, A. Link, T. Rokkas, and P. Malfertheiner, “Review: Gastric cancer—Clinical aspects,” *Helicobacter*, vol. 24, no. S1, pp. 1–5, 2019, doi: 10.1111/hel.12643.

[4] L. Wu *et al.*, “A deep neural network improves endoscopic detection of early gastric cancer without blind spots,” *Endoscopy*, vol. 51, no. 6, pp. 522–531, 2019, doi: 10.1055/a-0855-3532.

[5] D. E. Guggenheim and M. A. Shah, “Gastric cancer epidemiology and risk factors,” *J. Surg. Oncol.*, vol. 107, no. 3, pp. 230–236, 2013, doi: 10.1002/jso.23262.

[6] S. Yalamarthi, P. Witherspoon, D. McCole, and C. D. Auld, “Missed diagnoses in patients with upper gastrointestinal cancers,” *Endoscopy*, vol. 36, no. 10, pp. 874–879, 2004, doi: 10.1055/s-2004-825853.

[7] N. K. Choudhry, R. H. Fletcher, and S. B. Soumerai, “Systematic review: The relationship between clinical experience and quality of health care,” *Ann. Intern. Med.*, vol. 142, no. 4, pp. 260–273, 2005, doi: 10.7326/0003-4819-142-4-200502150-00008.

[8] American Society of Clinical Oncology, “Stomach Cancer: Risk Factors,” 2019. https://www.cancer.net/cancer-types/stomach-cancer/risk-factors.

[9] European Society for Medical Oncology and Anticancer Fund, “Cancro do estômago,” 2012.

[10] Therithal info, “Anatomy of the Stomach.” http://www.brainkart.com/article/Anatomy-of-the-Stomach_21938/.

[11] P. C. MD, “Gastric Cancer: Overview,” *Gastroenterol. Clin. North Am.*, vol. 42, no. 2, pp. 211–217, 2013.

[12] K. Yao, “The endoscopic diagnosis of early gastric cancer,” *Ann Gastroenterol*, p. 26:11, 2013.

[13] P. Moutinho, “Endoscopia digestiva alta.” https://www.saudebemestar.pt/pt/clinica/gastrenterologia/endoscopia-digestiva-alta/.

[14] Rede Hospital da Luz, “Endoscopia digestiva alta.” https://www.hospitaldaluz.pt/pt/guia-de-saude/dicionario-de-saude/e/188/endoscopia-digestiva-alta.

[15] K. et al. Veitch, A., Uedo, N., Yao, “Optimizing early upper gastrointestinal cancer detection at endoscopy,” *Nat Rev Gastroenterol Hepatol*, vol. 12, pp. 660–667, 2015.

[16] L. Wu *et al.*, “Randomised controlled trial of WISENSE, a real-time quality improving system for monitoring blind spots during esophagogastroduodenoscopy,” *Gut*, vol. 68, no. 12, pp. 2161–2169, 2019, doi: 10.1136/gutjnl-2018-317366.

[17] X. Liu, C. Wang, J. Bai, and G. Liao, “Fine-tuning Pre-trained Convolutional Neural Networks for Gastric Precancerous Disease Classification on Magnification Narrow-band Imaging Images,” *Neurocomputing*, vol. 392, pp. 253–267, 2020, doi: 10.1016/j.neucom.2018.10.100.

[18] S. Shinozaki, H. Osawa, Y. Hayashi, A. K. Lefor, and H. Yamamoto, “Linked color imaging for the detection of early gastrointestinal neoplasms,” *Therap. Adv. Gastroenterol.*, vol. 12, pp. 1–10, 2019, doi: 10.1177/1756284819885246.

[19] C. Spada *et al.*, “Performance measures for small-bowel endoscopy: A European Society of Gastrointestinal Endoscopy (ESGE) Quality Improvement Initiative,” *United Eur. Gastroenterol. J.*, vol. 7, no. 5, pp. 614–641, 2019, doi: 10.1177/2050640619850365.

[20] S. Marques, M. Bispo, P. Pimentel-Nunes, C. Chagas, and M. DInis-Ribeiro, “Image Documentation in Gastrointestinal Endoscopy: Review of Recommendations,” *GE Port. J. Gastroenterol.*, vol. 24, no. 6, pp. 269–274, 2017, doi: 10.1159/000477739.

[21] et al. JL, Hartman M, Lau L, “Duration of Endoscopic Examination Significantly Impacts Detection Rates of Neoplastic Lesions During Diagnostic Upper Endoscopy,” *Gastrointest Endosc*, p. 73(4S):AB393, 2011.

[22] Sociedade Portuguesa de Endoscopia Digestiva, "Normas de Avaliação e Garantia da Qualidade da Endoscopia Digestiva em Portugal," 2009.

[23] M. D. Ribeiro, *Cancro Gástrico em Portugal*. 2018.

[24] W. Nash, T. Drummond, and N. Birbilis, "A review of deep learning in the study of materials degradation," *npj Mater. Degrad.*, vol. 2, no. 1, pp. 1–12, 2018, doi: 10.1038/s41529-018-0058-x.

[25] S. Huang, J. Yang, S. Fong, and Q. Zhao, "Artificial intelligence in cancer diagnosis and prognosis: Opportunities and challenges," *Cancer Lett.*, vol. 471, no. September 2019, pp. 61–71, 2020, doi: 10.1016/j.canlet.2019.12.007.

[26] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016, doi: 10.1016/j.neucom.2015.09.116.

[27] M. Nielsen, *Neural Networks and Deep Learning*. 2019.

[28] C. Nicholson, "A Beginner's Guide to Multilayer Perceptrons (MLP)," *2012*. https://wiki.pathmind.com/multilayer-perceptron.

[29] Y. Li, B. Sixou, and F. Peyrin, "A review of the deep learning methods for medical images super resolution problems," vol. 1, pp. 1–14, 2020, doi: 10.1016/j.irbm.2020.08.004.

[30] S. Polamuri, "DIFFERENCE BETWEEN SOFTMAX FUNCTION AND SIGMOID FUNCTION," *2017*. https://dataaspirant.com/difference-between-softmax-function-and-sigmoid-function/.

[31] T. Wood, "Softmax Function." https://deepai.org/machine-learning-glossary-and-terms/softmax-layer.

[32] B. Tóth, "How do forward and backward propagation work?," 2018. https://tech.trustpilot.com/forward-and-backward-propagation-5dc3c49c9a05.

[33] P. Pandey, "Understanding the Mathematics behind Gradient Descent.," 2019. https://towardsdatascience.com/understanding-the-mathematics-behind-gradient-descent-dde5dc9be06e.

[34] "Gradient Descent & Back Propagation." http://test.basel.in/product/gradient-descent-back-propagation/.

[35] C. Sng, "An Introduction to Gradient Descent," 2020. https://medium.com/@sngcharis10/an-introduction-to-gradient-descent-4c04b4c063bd.

[36] J. Brownlee, "Overfitting and Underfitting With Machine Learning Algorithms." https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/.

[37] A. PAI, "CNN vs. RNN vs. ANN – Analyzing 3 Types of Neural Networks in Deep Learning," 2020. https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/.

[38] et al. Y. LeCun, L. Bottou, Y. Bengio, "Gradient-based learning applied to document recognition," *Proc. IEEE 86*, vol. 11, pp. 2278–2324, 1998.

[39] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way," 2018. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[40] M. Zeiler, "Hierarchical Convolutional Deep Learning in Computer Vision," New York University, 2014.

[41] and A. C. I. Goodfellow, Y. Bengio, "Deep Learning," *Cambridge MIT Press*, vol. 1, 2016.

[42] S. Das, "CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more...," 2017. https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5.

[43] M. ul Hassan, "VGG16 – Convolutional Network for Classification and Detection," 2018. https://neurohive.io/en/popular-networks/vgg16/.

[44] S.-H. Tsang, "Review: VGGNet — 1st Runner-Up (Image Classification), Winner (Localization) in ILSVRC 2014," 2018. https://medium.com/coinmonks/paper-review-of-vggnet-1st-runner-up-of-ilsvlc-2014-image-classification-d02355543a11.

[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *32nd Int. Conf. Mach. Learn. ICML 2015*, vol. 1, pp. 448–456, 2015.

[46] "VGG-16 | CNN model," 2020. https://www.geeksforgeeks.org/vgg-16-cnn-model/.

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[48] Google Developers, "Multi-Class Neural Networks: Softmax."

https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax.

[49] B. Raj, “A Simple Guide to the Versions of the Inception Network.” https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202.

[50] L. A. dos Santos, “GoogleNet,” 2020. https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/deep_learning/googlenet.

[51] “Inception Module.” https://deepai.org/machine-learning-glossary-and-terms/inception-module.

[52] C. Szegedy *et al.*, “Going deeper with convolutions,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

[53] S.-H. Tsang, “Review: Inception-v4 — Evolved From GoogLeNet, Merged with ResNet Idea (Image Classification),” 2018. https://towardsdatascience.com/review-inception-v4-evolved-from-googlenet-merged-with-resnet-idea-image-classification-5e8c339d18bc.

[54] S.-H. Tsang, “Review: Inception-v3 — 1st Runner Up (Image Classification) in ILSVRC 2015,” 2018. https://medium.com/@sh.tsang/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c.

[55] M. Längkvist, L. Karlsson, and A. Loutfi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” *Pattern Recognit. Lett.*, vol. 42, no. 1, pp. 11–24, 2014, doi: 10.1016/j.patrec.2014.01.008.

[56] J. Brownlee, “A Gentle Introduction to the ImageNet Challenge (ILSVRC),” 2019. https://machinelearningmastery.com/introduction-to-the-imagenet-large-scale-visual-recognition-challenge-ilsvrc/.

[57] A. SACHAN, “Detailed Guide to Understand and Implement ResNets,” 2017. https://cv-tricks.com/keras/understand-implement-resnets/.

[58] H. Kumar, “Skip connections and Residual blocks,” 2018. https://kharshit.github.io/blog/2018/09/07/skip-connections-and-residual-blocks.

[59] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2016.

[60] V. KURAMA, “A Review of Popular Deep Learning Architectures: DenseNet, ResNeXt, MnasNet, and ShuffleNet v2,” 2020. https://blog.paperspace.com/popular-deep-learning-architectures-densenet-mnasnet-shufflenet/.

[61] S.-H. Tsang, “Review: DenseNet — Dense Convolutional Network (Image Classification),” *2018*. https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803.

[62] “DenseNet-121,” 2017, [Online]. Available: https://www.kaggle.com/pytorch/densenet121.

[63] N. Radwan, “Leveraging Sparse and Dense Features for Reliable State Estimation in Urban Environments Noha Radwan,” no. July, 2019, doi: 10.6094/UNIFR/149856.

[64] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning Transferable Architectures for Scalable Image Recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 8697–8710, 2018, doi: 10.1109/CVPR.2018.00907.

[65] Y. Bengio, “Deep learning of representations for unsupervised and transfer learning,” *JMLR Work. Conf. Proc*, vol. 7, pp. 1–20, 2011.

[66] Y. B. D. Erhan, “Why does unsupervised pre-training help deep learning?,” *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, 2010.

[67] scikit-learn developers, “sklearn.metrics.matthews_corrcoef.” https://scikit-learn.org/stable/modules/generated/sklearn.metrics.matthews_corrcoef.html?highlight=matthews_corrcoef#sklearn.metrics.matthews_corrcoef.

[68] Wikipedia, “Matthews correlation coefficient,” 2021. https://en.wikipedia.org/wiki/Matthews_correlation_coefficient.

[69] scikit-learn developers, “3.3. Metrics and scoring: quantifying the quality of predictions.” https://scikit-learn.org/stable/modules/model_evaluation.html#.

[70] T. Cogan, M. Cogan, and L. Tamil, “MAPGI: Accurate identification of anatomical landmarks and diseased tissue in gastrointestinal tract using deep learning,” *Comput. Biol. Med.*, vol. 111, no. April, p. 103351, 2019, doi: 10.1016/j.compbiomed.2019.103351.

[71] K. Pogorelov *et al.*, “Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection,” *Proc. 8th ACM Multimed. Syst. Conf. MMSys 2017*, pp. 164–169, 2017, doi: 10.1145/3083187.3083212.

[72] P. H. Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland,

Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, “The Kvasir Dataset,” 2017. https://datasets.simula.no/kvasir/#data-collection.

[73] M. Belgiu and L. Drăgu, “Random forest in remote sensing: A review of applications and future directions,” *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 24–31, 2016, doi: 10.1016/j.isprsjprs.2016.01.011.

[74] N. Landwehr, M. Hall, and E. Frank, “Logistic model trees,” *Lect. Notes Artif. Intell. (Subseries Lect. Notes Comput. Sci.*, vol. 2837, pp. 241–252, 2003, doi: 10.1007/978-3-540-39857-8_23.

[75] T. Agrawa, R. Gupta, S. Sahu, and C. E. Wilson, “SCL-UMD at the medico task-mediaeval 2017: Transfer learning based classification of medical images,” *CEUR Workshop Proc.*, vol. 1984, pp. 3–5, 2017.

[76] S. Petscharnig, K. Schoffmann, and M. Lux, “An inception-like CNN architecture for GI disease and anatomical landmark classification,” *CEUR Workshop Proc.*, vol. 1984, pp. 0–2, 2017.

[77] M. E. Mavroforakis and S. Theodoridis, “A geometric approach to support vector machine (SVM) classification,” *IEEE Trans. Neural Networks*, vol. 17, no. 3, pp. 671–682, 2006, doi: 10.1109/TNN.2006.873281.

[78] H. Borgli *et al.*, “HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy,” *Sci. Data*, vol. 7, no. 1, p. 283, Dec. 2020, doi: 10.1038/s41597-020-00622-y.

[79] H. Takiyama *et al.*, “Automatic anatomical classification of esophagogastroduodenoscopy images using deep convolutional neural networks,” *Sci. Rep.*, vol. 8, no. 1, pp. 1–8, 2018, doi: 10.1038/s41598-018-25842-6.

[80] Q. He *et al.*, “Deep learning-based anatomical site classification for upper gastrointestinal endoscopy,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 15, no. 7, pp. 1085–1094, 2020, doi: 10.1007/s11548-020-02148-5.

[81] “GASTROLAB - the Gastrointestinal Image Site.” www.gastrolab.net.

[82] and M. C. F. Riaz, F.B. Silva, M. Dinis-Ribeiro, “Part of the AIDA-E (Analysis of Images to Detect Abnormalities in Endoscopy) challenge.” https://aidasub-chromogastro.grand-challenge.org/home/.

[83] J. Brownlee, “Gentle Introduction to the Adam Optimization Algorithm for Deep Learning,” *2017*. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

[84] “Model training APIs.” https://keras.io/api/models/model_training_apis/.

[85] R. DWIVEDI, “Everything You Should Know About Dropouts And BatchNormalization In CNN.” https://analyticsindiamag.com/everything-you-should-know-about-dropouts-and-batchnormalization-in-cnn/.

[86] Keras, “BatchNormalization layer.” https://keras.io/api/layers/normalization_layers/batch_normalization/.

[87] J. JORDAN, “Introduction to autoencoders.,” *2018*. https://www.jeremyjordan.me/autoencoders/.

[88] GeeksforGeeks, “Python | Mean Squared Error,” *2019*. https://www.geeksforgeeks.org/python-mean-squared-error/.

[89] H. Kumar, “Autoencoder: Downsampling and Upsampling,” 2019. https://kharshit.github.io/blog/2019/02/15/autoencoder-downsampling-and-upsampling.

[90] M. Grootendorst, “Validating your Machine Learning Model,” 2019. https://towardsdatascience.com/validating-your-machine-learning-model-25b4c8643fb7.

[91] Krishni, “K-Fold Cross Validation,” 2018. https://medium.datadriveninvestor.com/k-fold-cross-validation-6b8518070833.