UNIVERSITAT DE BARCELONA

FUNDAMENTALS OF DATA SCIENCE MASTER'S THESIS

Domain Adaptation in Wireless Capsule Endoscopy Diagnosis

Author:
Èlia FICAPAL

Supervisor:
Dr. Santi SEGUÍ

A thesis submitted in partial fulfillment of the requirements for the degree of MSc in Fundamentals of Data Science

in the

Facultat de Matemàtiques i Informàtica

June 28, 2019

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Matemàtiques i Informàtica

MSc

Domain Adaptation in Wireless Capsule Endoscopy Diagnosis

by Èlia FICAPAL

Convolutional neural networks have been proved to reach excellent results in image classification, but also to require large amounts of data. Unfortunately, there are plenty of domains and applications where the availability of labelled datasets is limited. For instance, privacy issues and the small amount of tests performed leads to low availability of data in many medical image analysis problems. This obstacle is faced by a current project developed in collaboration with the company Corporate Health, which aims to make diagnosis out of the frames obtained from the Wireless Capsule Endoscopy (WCE) procedure.

This thesis will focus on finding a way to train models that generalise well to unseen target domains. In particular, images from different sources will be used for training and testing. Triplet loss is the secret weapon that will play a key role in improving the results obtained from the domain adaptation experiments.

The achievements of this work cover both training an algorithm that classifies accurately enough the images from the target domain, as well as proving that triplet loss is indeed helpful for domain adaptation in our specific scenario.

Acknowledgements

First of all, I would like to thank my supervisor Dr. Santi Seguí for giving me the opportunity to participate in this enriching project, as well as guiding and motivating me during the whole process.

Secondly, special thanks to Pablo Laíz who has also been a source of inspiration and constant help.

Also, I want to thank my eternal colleague in this project, Stefan Ivanov, who has shared with me many hours of hard work, laugh and learning.

Finally, I would like to express my gratitude to the talented students and professors from the Data Science research team in Universitat de Barcelona for preparing seminars, which made me learn more about deep learning in a different and pleasant way.

Chapter 1

Introduction

1.1 Wireless Capsule Endoscopy

Wireless Capsule Endoscopy (WCE) is a procedure that uses a tiny wireless camera to record internal images of the digestive tract for use in medical diagnosis (Iddan et al., 2000). The camera sits inside a capsule, also called pill cam, which has the size of a pharmaceutical capsule and is equipped with a light source, camera, lens, radio transmitter and battery. This capsule is swallowed by the patient and takes a large number of images, which are sent to an external device in order to be analysed.

By contrast, traditional endoscopy works inserting a large and flexible tube, that is equipped with a camera, through the throat or the rectum. The main advantage of WCE over traditional endoscopy is that it reaches the small intestine. Moreover, it is considered the best tool for the diagnosis of bleeding (Eliakim, 2004; Mustafa et al., 2013) and is giving good results for diagnosis of tumours (Cobrin, Pittman, and Lewis, 2006; Urgesi et al., 2012), mobility disorders (Malagelada et al., 2011) and chronic abdominal pain (Yang et al., 2014).

All these applications together with being a non-invasive product that allows the full visualisation of the entire endoluminal tract makes WCE a very powerful tool. However, its application is limited due to one main problem: around 200K images are produced and their visualisation is a tedious task that needs to be done by experts such as physicians.

1.2 The project

This is the main motivation of a big project developed in collaboration with the private company Corporate Health and other students and researches from Universitat de Barcelona. It consists on developing machine learning models to find several dysfunctions (such as polyps, blood, ulcers, etc.) in the images obtained from wireless capsule endoscopy.

Intelligent systems learn from data to recognise patterns, predict outcomes and make decisions. When loads of data are available, like in some image classification scenarios, these systems can even achieve better levels of performance than humans (He et al., 2015b). Their strength lies in their ability to process large amounts of examples and obtain a detailed estimate of what does or does not constitute the object of interest. In order to train an intelligent system, a so-called training set is required. It is composed of a set of object examples, for example images taken from patients, and a matching set of labels, which could take values "healthy", "at risk" or "disease". Then the system is trained to classify as accurately as possible the inputs of the training set according to their labels, and the main objective is to generalise well to new unseen inputs.

However, sometimes the training data is not an accurate representation of the population and thus generalisation is complicated. For example, if the unseen data is distributed differently as the training data, then the classifier of the intelligent system will not perform well. Domain adaptation is a sub-field within machine learning that is concerned with accounting for these type of changes (Kouw, 2018). The population of interest is called the target domain, for which labels are usually not available and training a classifier is not possible. However, if data from a similar population is available, it could be used as a source of additional information.

This thesis will focus on the domain adaptation part of the project. The motivation relies on the fact that different kinds of camera are used in the WCE procedure. For instance, the PillCam SB3 is the new version of PillCam SB2 (Given Imaging, Ltd., Yokneam, Israel), and thus slightly different images are produced. Figure 1.1 shows how these cameras look like.

PillCam SB2 PillCam SB3

FIGURE 1.1: Images of camera capsules from https://www.medtronic.com and (Ciuti, Menciassi, and Dario, 2011)

In the majority of medical imaging problems, only small amounts of labelled data are available due to privacy issues. This is a great challenge that this project in particular is also facing. The specific situation in this thesis is that there is a dataset containing images from PillCam SB3 that needs to be labelled. Luckily, there is available another dataset which contains SB2 images and has already been labelled. As previously introduced, the labelling process is usually done by experts and may take some time. Hence, it is very appealing to train a model with images from SB2 and the minimum possible amount of SB3 images and use it to obtain the missing labels of the SB3 images.

1.3 Goals and Methodology

In this thesis, two main goals are defined. On one hand, the most immediate goal is to correctly classify the images coming from the PillCam SB3. To do so, a model must be trained and then tested on SB3 images. If the test accuracies are high enough, then the model could use by physicians to classify the images without looking at all the frames, which leads to a significant save of time.

The models that will be considered are deep learning algorithms which look for vector representations for the images, the so-called embeddings, and then classify them according to certain defined classes. triplet loss can be introduced in order to train the network and obtain good embeddings, and the other goal of this thesis is to prove that it significantly improves the results on domain adaptation.

The methodology followed in this thesis consists on considering two models: a deep learning model and a counterpart that uses triplet loss. These models are first both trained and tested using images from the PillCam SB2 in order to check

the performance when source and target domains are the same. Even though it does not take part in the goals of the project, this point is also important and not straightforward. The complication relies on the complexity of the images recorded with WCE, the small size of the training dataset in comparison with other computer vision problems and the number of classes that are considered, which vary between 6 and 7.

After that, the same already trained models are tested using images from another camera, PillCam SB3. Since the target domain has changed, the results are probably not good enough in order to classify the unlabelled images. In order to improve the accuracy, some SB3 images are labelled and added to the training dataset that was initially uniquely composed of SB2 images.

With this, good enough results are expected to be achieved in order to help classifying all the images in the SB3 dataset. Moreover, since the tests will be performed using both models, the benefit of using triplet loss will be studied.

Chapter 2

Background

2.1 State of the Art

Since WCE was invented in 2000, several techniques were developed to work with the produced images. Initially, the methods relied on hand-crafted image representations for each different pathology or object of study. For instance, colour is a main feature for detecting intestinal content or bleeding, whereas structure and texture analysis determine the existence of polyps in a frame. However, the main drawback of these kind of methods is that when a new event of interest in the images is considered and needs to be detected, it has to be characterised by an experts and the main features must be extracted in order to create a whole new method from scratch. In the literature, methods using features such as colour (Basheer and Hajmeer, 2000; Li et al., 2009), texture (Yuan and Meng, 2014; Khatib, Werghi, and Al-Ahmad, 2015; Li and Meng, 2012) and shape (Iwahori et al., 2015; Bae and Yoon, 2015) can be found.

Computer vision is one of the areas that has advanced the most due to deep learning. More specifically, Convolutional Neural Networks have produced great results in image classification tasks (Krizhevsky, Sutskever, and Hinton, 2012; Szegedy et al., 2014; Simonyan and Zisserman, 2015) and a few cases are also reported for WCE (Zou et al., 2015; Yu et al., 2015; Yuan and Q.-H. Meng, 2017). These methods solve the problem explained above presented by the hand-crafted methods but require, in general, a large labelled dataset for training.

Until now, state-of-the-art methods for image classification, focusing on WCE frames, have been reviewed. On the other hand, the use of triplet loss has been proved to be effective for domain adaptation in the embedding space (Deng, Zheng, and Jiao, 2018).

2.2 Neural Networks and Deep Learning

In 1943, the concept of neural network arose in the attempt to simulate processes occurring in the brain (Palm, 1986). Neural networks are a set of algorithms designed to recognise patterns from observational data which help to cluster and classify. They consist of layers which, in turn, are composed of individual units called nodes or neurons. The nodes of each layer are connected to nodes of the next layer, as it can be seen in Figure 2.1.

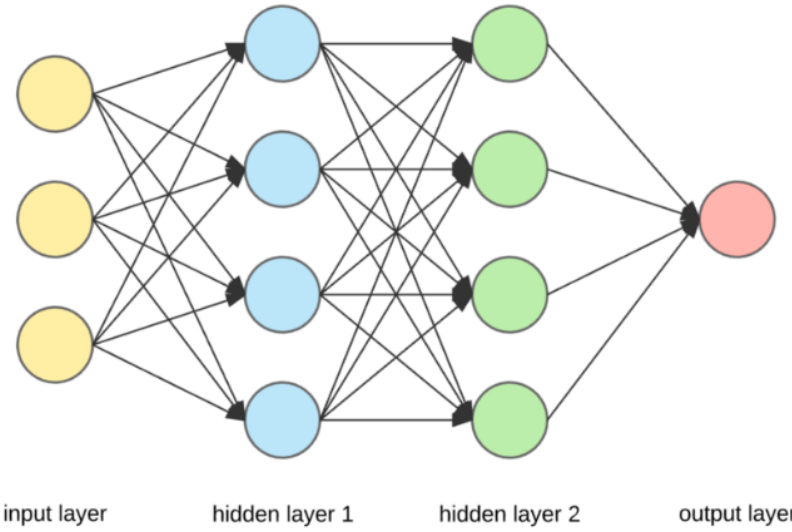

FIGURE 2.1: Example of a Neural Network from https://medium.com .

Each node combines the data it receives as input with a set of weights, assigning significance with regard to the task the algorithm is trying to learn. These input-weight products are summed and then the sum is passed through a node's so-called activation function. It is used to determine to what extent that signal should progress further through the network to affect the ultimate outcome, which could be, for instance, an act of classification. Following this process, each layer's output is simultaneously the subsequent layer's input, starting from an initial input layer receiving the initial data until the output layer.

From 1986 to mid 90's, important developments related to neural networks were made, such as convolutional neural networks (LeCun and Bengio, 1998), unsupervised learning (Bengio, 1991) and Restricted Boltzmann Machine (RBM) (Hinton and Sejnowski, 1986). However, new machine learning methods were emerging at that time and neural networks were not trusted since they seemed too intuition-based. Moreover, they required a computational cost that computers had trouble to achieve.

Finally, (Hinton, Osindero, and Teh, 2006) was published and it was seen as a significant enough breakthrough to rekindle interest in neural nets. Also, Moore's law helped computers to become dozens of times faster due to GPUs. This led to making learning using neural networks with large datasets and many layers much more feasible.

That brought Deep Learning, where technological structures of neural networks become more complex and are able to solve a wide range of tasks that could not be effectively solved before, for instance image classification, speech recognition and natural language processing.

2.3 Convolutional Neural Networks

One of the challenges of computer vision problems is that the inputs can become very big. Since images have to be analysed, a computational representation must be defined in order to fit them to a computer. This representation is given by $m \times n \times 3$ arrays of numbers, called pixels. The values m and n make reference to the height and width of the image in pixels, respectively, whereas the number 3 accounts for

the RGB values. Each number in this array is given a value from 0 to 255, which describes the pixel intensity at that point, and they are the only available inputs to the computer. Thus, if for example we are working with RGB images which have size 256×256, it gives input features of dimension $256 \cdot 256 \cdot 256 = 196.608$.

Apart from the need of reducing the input size, if a fully connected network was used the total number of weights for each layer may become huge. This comes with a risk of overfitting if not enough data is available. Also, the computational and memory requirements to train a neural network can be even infeasible. This is why the convolution operation is introduced in order to make the training of neural networks on large images feasible and it is one of the fundamental building blocks of a convolutional neural network.

Moreover, natural images are not just random combinations of values in an array, but they present strong correlations at different levels. For instance, it is intuitively obvious that the value of a pixel is not independent of the values of its neighbouring pixels. They also present local invariance, which means that visual structures, such as a person, can be present on any place of the image at any scale. Thus, image location is not important, but the relative positions of geometric and photometric structures can be very meaningful.

Convolutional neural networks are neural networks that make the explicit assumption that the inputs are images, which allows encoding properties into the architecture. They are composed, as a normal neural network, of a sequence of layers, which are normally convolutional layers, pooling layers, fully-connected layers, batch normalisation layers or dropout layers.

2.3.1 Convolutional Layer

Combining all the previous considerations and the experiments from Dr. Hubel and Dr. Wiesel, who worked on the area of sensory processing, the convolutional layers were introduced. They are based on convolutions, which are mathematical operations that combine two input images, the original image to process and a so-called kernel, to form a third image. Mathematically, given a convolution kernel K, also called filter, represented by a MxN array, the convolution of an image I with K is

$$output(x,y) = (I \otimes K)(x,y) = \sum_{m=0}^{M-1} \sum_{n=1}^{N-1} K(m,n) I(x-n, y-m). \tag{2.1}$$

The output of image convolution in (2.1) is an image that might represent some kind of information that was present in the image in a very subtle way. For example, a typical used kernel is an edge detector which highlights the edges of visual structures and attenuates smooth regions.

Convolutions $\otimes$ are linear operators and thus the application of successive convolutions can always be represented by a single convolution. This is why a non linear activation function is applied after each convolution. Figure 2.2 illustrates an example of how to apply a convolution. Moreover, the values of a kernel in convolutional layer, which are actually free parameters, must be learnt during training in order to perform the optimal information extraction to classify the image.

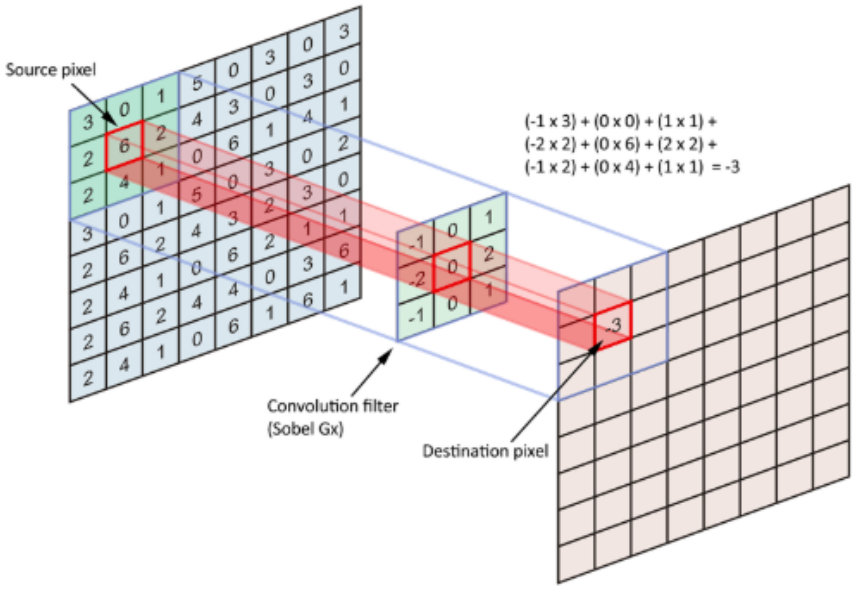

FIGURE 2.2: Diagram of the application of a convolution with kernel size 3×3 from https://github.com/DataScienceUB/DeepLearningMaster2019
.

Apart from the kernel size, the stride and the padding have to be defined. The stride defines the amount by which the filter shifts, hence it controls how the filter convolves around the input image. On the other hand, padding defines how the border is handled and prevents the size of the volume to decrease too fast when applying convolutional layers. The motivation for this is that, in the early layers of a network, it is better to preserve as much information about the original input volume so that those low level features can be extracted.

2.3.2 Fully-Connected Layer

A fully-connected layer, also called dense layer, is characterised by connecting each of its neurons to all the neurons in the previous layer. These kind of layers make a neural network feedforward, meaning that each layer can only be connected to the next layer and cycles are not allowed. Thus, information goes only in one direction.

The main drawback with dense layers for classifying images is that they require a large amount of weights to be optimised. This leads to many problems such as slow training time and chances of overfitting.

2.3.3 Batch Normalisation Layer

These layers are used in order to assess the so-called internal covariance shift problem. It can be explained as the distribution of the activations in the intermediate layers constantly changing during training. This leads to a slow training because, in every training step, each layer must learn to adapt itself to a new distribution. Apart from reducing the training time, it also results in an increase of stability of the gradient and reduces overfitting.

As the name suggests, it works by normalising the inputs by adjusting and scaling the activations. The idea behind this process is to force the input of every layer to have approximately the same distribution in every training step.

2.3.4 Pooling Layer

Pooling layers are used in order to reduce the spatial dimensions of the images, but not depth, on a convolutional neural network. Since the spatial amount of information is lower, then the computation performance increases. Moreover, less spatial information leads to having less parameters and lower chances to overfit. The intuitive reasoning behind this layer is that once a specific feature is known to be in the original value, its exact location is not as important as its relative location to the other features.

There are several options in this category. For instance, Maxpooling is the most popular one. It applies a filter and a stride of the same length to the input volume and outputs the maximum number in every subregion that the filter convolves around. Other options for pooling layers are average pooling and L2-norm pooling.

2.3.5 Dropout Layer

Finally, dropout layers are added in order to overcome the overfitting problem (Srivastava et al., 2014). The idea is that they deactivate a random set of neurons in a layer by setting the corresponding weights to zero, for instance as in Figure 2.3. Thus, it forces the network to be redundant in terms of being able to provide the right classification or output for a specific example, even if some of the activations are dropped out. It makes sure that the network is not getting too fitted to the training dataset and thus assesses the overfitting problem. This kind of layer, however, should only be used during training and not testing.

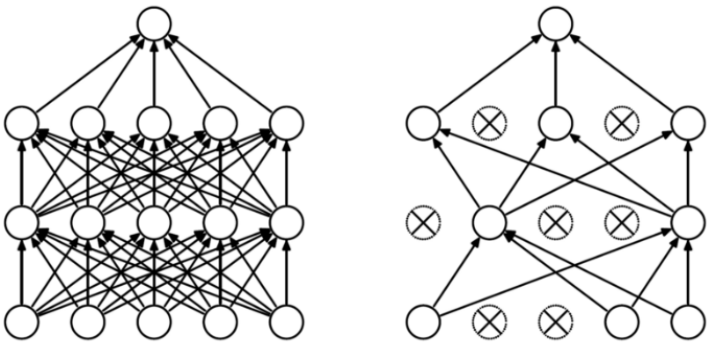

FIGURE 2.3: Dropout example in a neural network from https://medium.com.

Chapter 3

Proposed Architectures

As previously stated in the introduction, one of the main goals of the project is to determine whether the use of triplet loss improves the domain adaptation results. Thus, two strategies are proposed and they will be explained in this chapter. In both cases, the methods are divided in two parts: the extraction of a vector representation of each image, a so-called embedding, and the classification.

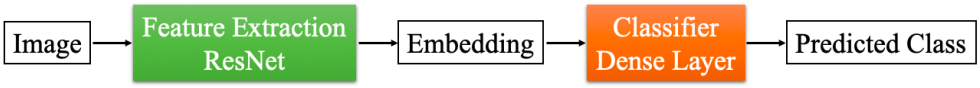

FIGURE 3.1: General model structure.

As it can be seen in Figure 3.1, the embeddings will be obtained using a ResNet and the classification task will be performed by a dense layer. The performance in classification will be evaluated using the cross-entropy loss. In the second approach, the triplet loss is introduced in order to separate the embeddings belonging to different classes during training.

This chapter focuses on briefly explaining both approaches and their main similarities and differences. However, a more detailed explanation about ResNet and triplet loss can be found in Chapter 4.

3.1 ResNet with Cross-Entropy Loss

The first approach consists of a ResNet stacked with a dense layer. The ResNet is used in order to extract the main features to create a rich vector representation of a given size of each image, the embedding. More specifically, the 50-layer ResNet from He et al., 2016 pretrained on ImageNet is considered. However, ResNet was originally designed to classify images into classes and thus the last layer is modified to introduce an embedding layer.

Once the embedding is computed, it is fed to the dense layer, which will have as many units as number of classes. It ouputs the scores of belonging to each of the classes which, in turn, can be converted into probabilities applying a softmax function. In order to measure the performance of the classification model and then optimise the parameters, the cross-entropy loss is used:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{C} t_i \log(s_i) \tag{3.1}$$

In (3.1), C is the total number of classes, s corresponds to the vector of scores obtained from the model and t is the target vector, which is a one-hot vector with one positive

class and $C-1$ negative classes. Thus, the elements s_i and t_i are the ground truth and the model score for each class $i \in C$, respectively. The output of (3.1) increases as the predicted probability diverges from the actual label, and it specially penalises the cases where predictions are confident and wrong.

Moreover, L2-regularisation is used to prevent overfitting. It consists on adding a regularisation term

$$\mathcal{L}_2 = \lambda \sum_{i=1}^{k} w_i^2 \tag{3.2}$$

to the loss function in order to include a measure of model complexity into the function to be minimised. In (3.2), w_i are the weights of the model and λ is an additional parameter added to allow control of the strength of the regularisation. In the experiments, $\lambda = 2 \cdot 10^{-5}$ will be used.

Altogether, the loss function for this approach is

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_2 \tag{3.3}$$

and the general pipeline is shown in Figure 3.2.

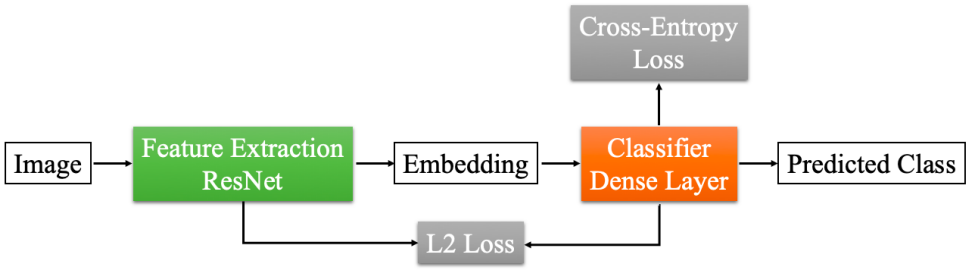

FIGURE 3.2: First model's structure.

3.2 ResNet with Triplet Loss

The second approach is very similar to the previous one, but now the the triplet loss is introduced. It is used in order to keep the embeddings of images belonging to the same class closer together in the Euclidean space than those from to different classes.

Thus, the previous equation (3.4) for the loss function is modified and results in

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{Triplet} + \mathcal{L}_2, \tag{3.4}$$

where $\mathcal{L}_{Triplet}$ is the triplet loss. The general pipeline of this approach is represented in Figure 3.3.

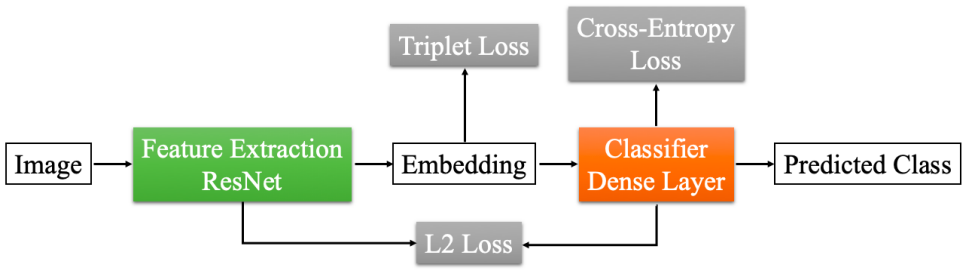

FIGURE 3.3: Second model's structure.

Chapter 4

Background of the Proposed Architectures

4.1 Residual Network (ResNet)

There is evidence (Simonyan and Zisserman, 2015; Szegedy et al., 2014; He et al., 2015c; Ioffe and Szegedy, 2015; Girshick et al., 2013; He et al., 2014; Girshick, 2015; Ren et al., 2015; Long, Shelhamer, and Darrell, 2014) that making a neural network deeper in an image recognition problem can significantly improve the results. However, adding layers may also lead to the problem of vanishing/exploding gradients. This means that as the gradient is back-propagated to earlier layers, repeated multiplications may make the gradient infinitively small. This problem can be assessed, for instance, using batch normalisation layers combined with Stochastic Gradient Descent (SGD) with backpropagation.

However, when deeper networks are able to start converging, a degradation problem has been exposed: with the network depth increasing, accuracy gets saturated and then degrades rapidly. Surprisingly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error.

This is intuitively unexpected because once a solution of a neural network is known, there is always a straightforward solution by construction of a deeper neural network obtained by just adding layers to the original one. It consists of copying the layers from the learned shallower model and set the added layers to identity mappings. Thus, a deeper model should produce no higher error than its shallower counterpart. However, in general, solvers are unable to find solutions that are as good or better than the constructed solution, or not in feasible time. Hence, the degradation problem suggests that the solvers might have difficulties in approximating identity mappings by multiple nonlinear layers.

In order to asses this problem, a deep residual learning framework was suggested (He et al., 2015a). To explain this framework, consider a few stacked layers and $\mathcal{H}(x)$ the desired underlying mapping fo be fit by them, with x denoting the inputs to the first of these layers. Assuming that the input and the output have the same dimensions, the residual function is defined as

$$\mathcal{F}(x) := \mathcal{H}(x) - x. \tag{4.1}$$

With the hypothesis that multiple nonlinear layers can asymptotically approximate complicated functions, then they can asymptotically approximate the residual functions (4.1) and this exactly what stacked layers are expected to approximate. Hence,

the original function becomes

$$\mathcal{H}(x) = \mathcal{F}(x) + x \tag{4.2}$$

As hypothesised, both forms should be able to asymptotically approximate the desired functions. However, with the residual learning reformulation (4.2), if identity mappings are optimal, the solvers may simply drive the weights of the multiple nonlinear layers towards zero to approach identity mappings. In general, identity mappings are not usually optimal, but the reformulation may help to precondition the problem. If the optimal function is closer to an identity mapping than to a zero mapping, it should be easier for the solver to find the perturbations with reference to an identity mapping than to learn the function as a new one.

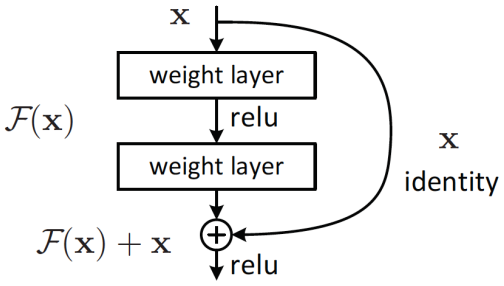

FIGURE 4.1: A building block in residual learning from (He et al., 2015a) .

The reformulation (4.2) can be realised by feedforward neural networks with shortcut skip connections, which means skipping one or more layers. In the case that is being treated, the shortcut connections simply perform identity mapping, and their outputs are added to the outputs of the stacked layers as in Figure 4.1. Identity shortcut connections add neither extra parameter nor computational complexity, and the entire network can still be trained end-to-end by SGD with backpropagation. Finally, the Residual Network (ResNet) is just a deep neural network with shortcut connections.

4.2 Triplet Loss

Sometimes, being able to have a variable number of classes is necessary in unsupervised learning. For instance, two unknown faces are compared in face recognition and it is necessary to decide whether they are the same person or not. In this case, triplet loss is a good way to learn embeddings for each face. In the embedding space, faces from the same person should be close together and form well separated clusters. Thus, the goal of the triplet loss is to keep examples with the same label close together in the embedding space, and separate those which have different labels.

To do so, triplets of examples are created, each of them composed of an anchor, a positive and a negative. In order to be a valid triplet, the positive example has to belong to the same class as the anchor, and the negative must belong to any other class. However, the train embeddings of each label are not supposed to be pushed to collapse into very small clusters. Hence, a margin is introduced in order to separate each class. In this way, the only requirement of the triplet loss is that given two examples of the same class and one of a different class, the negative should be further away than the positive by some margin.

This condition can be formalised as

$$\left\|f(x_i^a) - f(x_i^p)\right\|_2^2 + \alpha < \left\|f(x_i^a) - f(x_i^n)\right\|_2^2 \quad \forall \left(f(x_i^a), f(x_i^p), f(x_i^n)\right) \in \mathcal{T}, \tag{4.3}$$

where x_i^a, x_i^p and x_i^n are the anchor, positive and negative images, respectively, α is the margin, f is the embedding generator and $\mathcal{T}$ is the set of all possible triplets in the training set and has cardinality N. In the experiments, $\alpha = 0.2$ will be selected.

By applying the condition from (4.3) to each triplet, the triplet loss is obtained as follows

$$\mathcal{L}_{Triplet} = \sum_{i=1}^{N} \left[\left\|f(x_i^a) - f(x_i^p)\right\|_2^2 - \left\|f(x_i^a) - f(x_i^n)\right\|_2^2 + \alpha\right]. \tag{4.4}$$

Hence, the triplet loss minimises the distance between an anchor and a positive, both of which have the same identity, and maximises the distance between the anchor and a negative of a different identity, as shown in Figure 4.2.

FIGURE 4.2: Learning using triplet loss from (Schroff, Kalenichenko, and Philbin, 2015).

4.2.1 Triplet Mining

The condition stated in (4.3) is easily satisfied by the majority of possible triplets, which means using all of them do not help in training and thus convergence may be slow. Hence, it is convenient to choose triplets that violate (4.3), the so-called hard triplets. On one hand, hard positive triplets are those that given x_i^a, x_i^p is selected according to

$$\underset{x_i^p}{\operatorname{argmax}} \left\|f(x_i^a) - f(x_i^p)\right\|_2^2. \tag{4.5}$$

Analogously, hard negative triplets are those that given x_i^a, x_i^n is selected according to

$$\underset{x_i^n}{\operatorname{argmin}} \left\|f(x_i^a) - f(x_i^n)\right\|_2^2. \tag{4.6}$$

However, it is infeasible to compute (4.5) and (4.6) across the whole training set. Moreover, the most difficult images to label would generally be selected and this would lead to poorly training. Two approaches can be considered in order to assess this problem.

On one hand, triplets can be produced offline every a certain number of steps, using the most recent network checkpoints and computing (4.5) and (4.6) on a subset of data. In this case, a list of triplets is first produced. Then, batches of triplets of size B are considered, and each of them requires the computation of $3B$ embeddings. Also, the loss of the B triplets of each batch is computed and backpropagated into the network.

On the other hand, in (Schroff, Kalenichenko, and Philbin, 2015) triplets were generated online by selecting hard positive/negative exemplars from within a mini-batch. More specifically, given a batch of B examples and computing the B corresponding embeddings, a maximum of B^3 triplets can be obtained by all the available combinations. However, many of them may not be valid and thus will be discarded. In comparison with the previous strategy, online mining is much more efficient since it gives more triplets per batch of inputs and does not require any offline mining. In consequence, this is the one that will be used in our implementation.

Moreover, two approaches were introduced to out-perform in selecting triplets among the valid ones on which to compute the loss, the so-called batch all and batch hard strategies (Hermans, Beyer, and Leibe, 2017). If there are C different classes and K examples of each class are randomly sampled, then it results in an input batch of $B = CK$ examples. Batch all simply uses all possible $CK(CK - K)(K - 1)$ combinations of triplets and our experiments will be based on this approach. On the other hand, the batch hard strategy consists on selecting the hardest positive (4.5) and negative (4.5) for each anchor among the batch. This produces CK so-called moderate triplets, since they are the hardest within a subset of data.

Chapter 5

Numerical Experiments

5.1 Data

Images from two different sources will be considered in the experiments. The different sources correspond to different cameras, which are PillCam SB2 and SB3, respectively, and are shown in Figure 1.1. The main differences between these two cameras are that PillCam SB3 records with a higher image resolution and is able to increase the number of images from 2 to 6 frames per second.

The images are frames from videos recorded with the PillCams and can be classified according to (Seguí et al., 2016) as: turbid, bubbles, clear blob, wrinkles, dilated, wall or undefined. Examples of each class can be observed in Figure 5.1, where each column represents a different camera. Moreover, some description of the classes is given as follows:

1. Bubbles: The first row frames correspond to presence of bubbles, which can be identified as several white, yellow or green circular blobs. They appear due to agents used to reduce surface tension, similar to a detergent.

2. Clear blob: The second row frames correspond to clear blob, an open intestinal lumen. It is generally identified as a dark blob of variable size surrounded by intestinal wall.

3. Dilated: Third row images show the dilation in the gatrointestinal tract and this class is only considered in the SB2 PillCam.

4. Turbid: Fourth row frames show food in digestion. Their most remarkable characteristic is the presence of a wide range of green colours and a homogeneous texture.

5. Undefined: The images in the fifth row are visually undefined clinical events.

6. Wall: The sixth row images display frames without lumen presence. Thus, they are just orangish intestinal wall.

7. Wrinkles: Finally, last row frames contain a star-shape pattern that is produced by the pressure exerted by the nerve system. Wrinkles are usually present in the central frames of intestinal contractions.

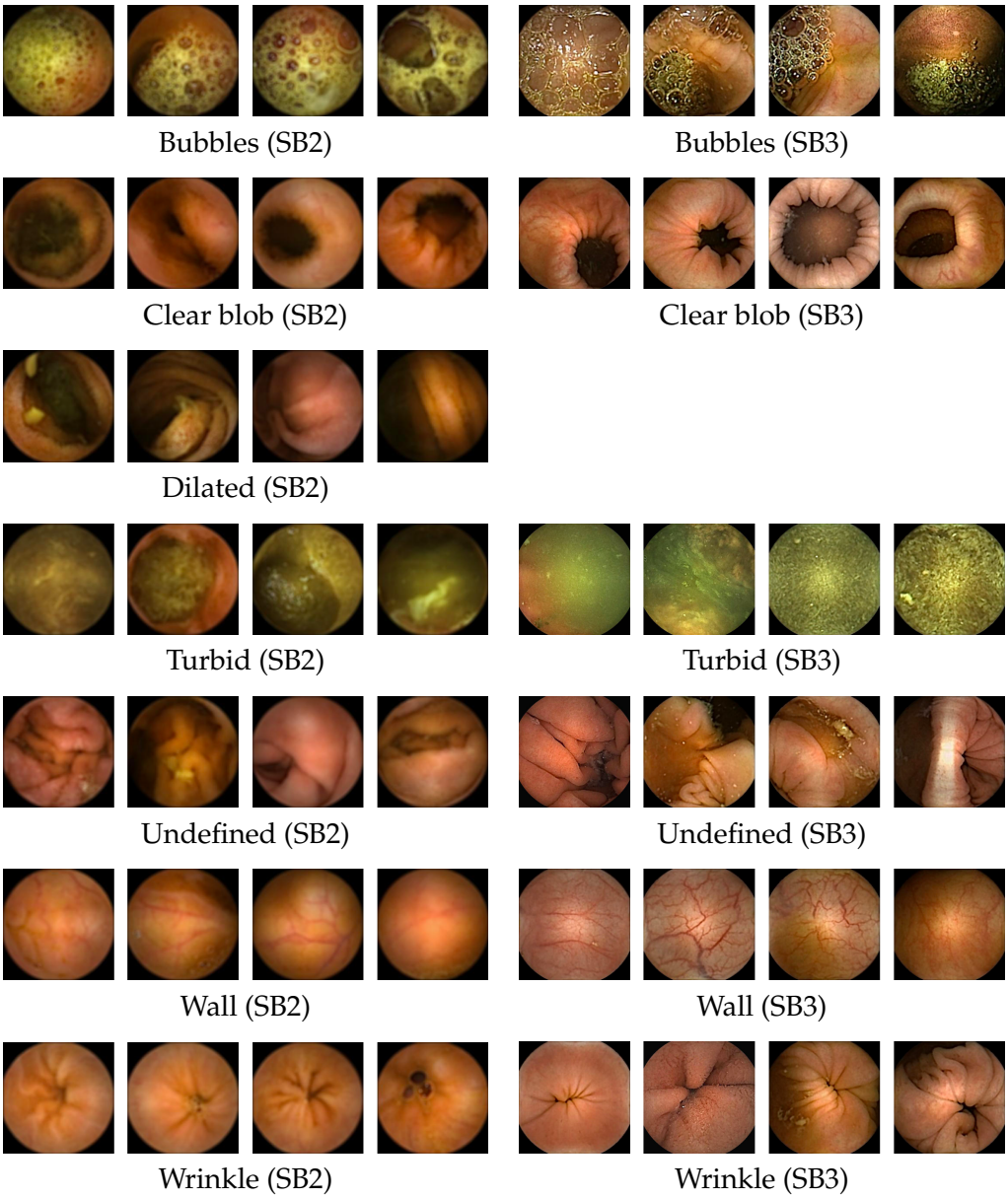

FIGURE 5.1: WCE image sample distributed where different classes are displayed by rows. The two different columns represent images obtained with PillCam SB2 (left) and PillCam SB3 (right).

Differences between images from both cameras can be appreciated in Figure 5.1. A very visually remarkable difference is the image resolution. Whereas RGB images with a resolution of 256×256 are obtained from SB2 PillCam, the resolution of SB3 images is 512×512.

Thus, two datasets are available: one obtained using PillCam SB2 and the other one using PillCam SB3. The first one is completely labelled, which means that each image is already classified according to the previously explained classes. On the other hand, the second one was initially completely unlabelled. However, a small amount was selected and manually classified in order to use them for training and testing in the experiments, which will be explained now.

5.2 Experiments

The main goal of the experiments is to show that triplet loss helps improving the results of domain adaptation. Having this in mind, the same experiments will be performed for the two approaches from the previous section. The experiments can be divided in two categories as follows.

The first category of the experiments consists on training and testing both approaches on SB2 images. More specifically, the models are trained using 2000 images per class and tested using 100 images per class. The second category are experiments with both models trained using the same SB2 images as before, but also adding (or not) some SB3 images. In this way, the training dataset will be composed of 2000 SB2 images per class and 0, 5, 10 or 25 SB3 images per class. Moreover, the testing is performed on 25 SB3 images per class. Note that, as it is shown in Figure 5.1, 7 and 6 different classes are considered for SB2 and SB3, respectively, since there are not SB3 images classified as dilated.

Going a little bit more into detail, each image in the training set is first resized to $256 \times 256 \times 3$ in order to be fed to the models, using a batch of 64 images. Then the ResNet is used to compute the embedding, which is a 2048 component vector. Finally, the class is predicted using a fully-connected layer. Figure 5.2 graphically shows this structure.

For both approaches, the well-known Stochastic Gradient Descent (SGD) is selected as optimiser with learning rate 10^{-4}. This choice is made due to its general stability during training and since it normally reaches the optimal solution, even though sometimes results in slow convergence

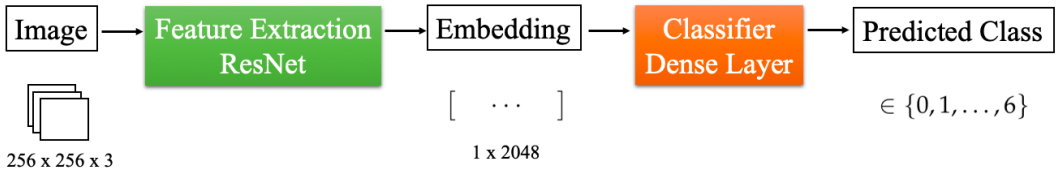

FIGURE 5.2: General structure of the models.

In order to analyse the results, the accuracies in testing will be shown in a table. Moreover, the normalised confusion matrix in testing will be computed in the most interesting cases and included as in Table 5.1.

	...	Class j	...
⋮			
Class i		$100P_j/R_i$	
⋮			

TABLE 5.1: General display of the normalised confusion matrix. P_j is the number of images classified by the model as class j, whereas R_i is the number of actual images belonging to class i.

Once the embeddings of the images in the test dataset are obtained by the model, their dimensionality can be reduced using Uniform Manifold Approximation and

Projection (UMAP) to two-dimensional vectors that will be used in order to generate plots. However, it is important to remark that these vectors are just a projection and thus not nearly all the information is included. Actually, if the embedding size is 2048, this means that 2046 components are lost.

These vectors will be used to obtain two kinds of plots. On one hand, a scatter plot using a different colour per class and a different marker if the images are correctly or mistakenly classified following the legend from Figure 5.3. Also, it is sometimes interesting to have a look at a scatter plot where the markers are the images themselves. Whereas the first plot is better for visualising the amount of well predicted classes, the second one helps to visually identify which features the model uses to match an image with a label.

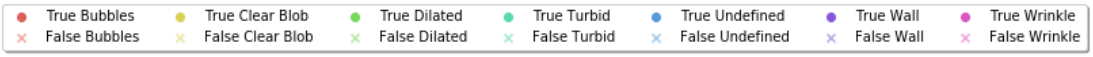

FIGURE 5.3: Legend of the plots.

5.2.1 Testing on PillCam SB2

First, both models are trained and tested using images from PillCam SB2. The training set consists of 14000 images, equally distributed between the 7 considered classes: bubbles, clear blob, dilated, turbid, undefined, wall and wrinkle. On the other hand, the test dataset is composed of 700 images, again equally distributed between classes.

The accuracy values in testing for both models are shown in Figure 5.2. The model using Triplet Loss gets better results, obtaining 95 % of accuracy, which is a very good result. On the other hand, the ResNet with Cross-Entropy Loss model reaches 87.71 %. However, this difference is not very meaningful in this thesis since the focus is on the domain adaptation part.

Model	Accuracy
ResNet with Cross-Entropy Loss	87.71
ResNet with Triplet Loss	95

TABLE 5.2: Accuracies in testing of the two models, using SB2 images for both training and testing.

Moreover, Tables 5.3 and 5.4 show the normalised confusion matrices. In both cases, the undefined class is the worst classified, which is quite intuitive since the images in this classes are more diverse. Finally, Figures 5.4 and 5.5 show the predicted embeddings distribution. Whereas the model using Triplet Loss splits the embeddings into 7 clearly distinguishable clusters, the other model seems to only distinguish three clusters. Only one of these clusters, the one containing bubble images, seems to be clearly defined.

	Bubbles	Clear Blob	Dilated	Turbid	Undefined	Wall	Wrinkle
Bubbles	94	0	0	5	1	0	0
Clear Blob	0	84	1	4	6	2	3
Dilated	0	6	88	1	4	1	0
Turbid	3	4	1	91	1	0	0
Undefined	0	8	4	1	75	10	2
Wall	0	0	0	1	4	92	3
Wrinkle	0	3	0	0	4	3	90

TABLE 5.3: Normalised confusion matrix in testing using the ResNet with Cross-Entropy Loss model.

	Bubbles	Clear Blob	Dilated	Turbid	Undefined	Wall	Wrinkle
Bubbles	93	0	0	5	2	0	0
Clear Blob	0	97	1	0	1	1	0
Dilated	0	1	97	1	1	0	0
Turbid	3	0	0	97	0	0	0
Undefined	1	2	2	1	89	5	0
Wall	0	0	0	2	3	95	0
Wrinkle	0	0	0	0	1	2	97

TABLE 5.4: Normalised confusion matrix in testing using the ResNet with Triplet Loss model.

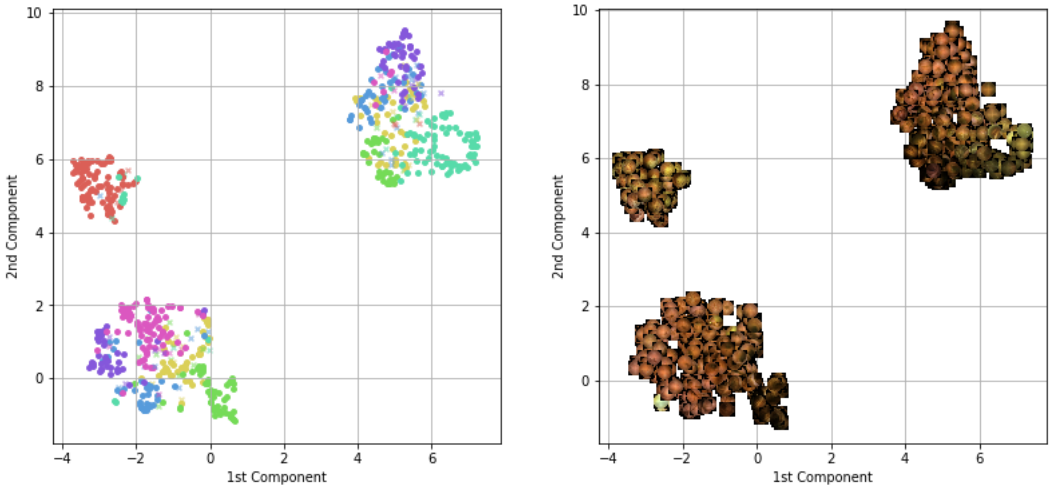

FIGURE 5.4: Embeddings distribution in testing using the ResNet with Cross-Entropy Loss model.

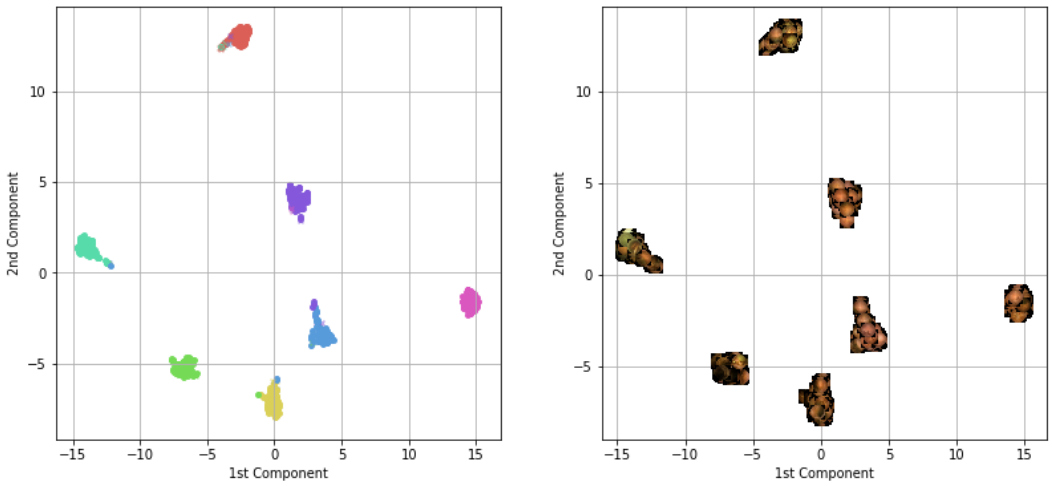

FIGURE 5.5: Embeddings distribution in testing using the ResNet with Triplet Loss model.

5.2.2 Testing on PillCam SB3

The weights of the previous trained models are saved and will be used as initial weights for the following experiments for the domain adaptation. As previously stated, the obtained results will be the important ones in order to assess the domain adaptation problem. Moreover, if good results are obtained, the models will be able to classify the current SB3 unlabelled dataset.

These experiments consist on training both models using the same SB2 images as before, but also adding some SB3 images. First of all, the models are directly tested on a test dataset composed of 150 SB3 images, equally distributed between the 6 classes: bubbles, clear blob, turbid, undefined, wall and wrinkle. Then the models will be trained adding 5, 10 and 25, respectively, SB3 images per class to the previously used training dataset composed of SB2 images. During the training process, 50 SB2 images and 14 SB3 images will be included in every batch, and data augmentation such as flips and rotations will be applied on the SB3 images.

Table 5.5 shows the obtained accuracies in testing for the experiments, which are also plotted in Figure 5.6. When testing the models trained with only with SB2 images, the accuracy values are 27.33 % and 42.67 % for each model, respectively, which are low values. However, by adding only 5 SB3 images the results start to get better. As expected, the more SB3 images are used to train the models, the better the accuracy. Moreover, highest accuracy values are obtained using the ResNet with Triplet Loss model, which achieves a maximum accuracy value of 89.33 %.

		# SB3 images in training			
		0	5	10	25
Model	ResNet with Cross-Entropy Loss	27.33	75.33	80	82
	ResNet with Triplet Loss	42.67	86	86.67	89.33

TABLE 5.5: Accuracies in testing for the two models, using SB2 images and 0, 5, 10 or 25 SB3 images when training.

FIGURE 5.6: Accuracies in testing for the two models, using SB2 images and 0, 5, 10 or 25 SB3 images when training.

Furthermore, more details are given about the experiments with 0 and 25 SB3 images in training, since they are the ones with worst and best results, respectively. On one hand, the normalised confusion matrices for the ResNet with Cross-Entropy Loss model are given in Tables 5.6 and 5.7. If no SB3 images are used for training, the vast majority of images are classified as turbid and also a few as wrinkle. If 25 SB3 images are added for training, much better results are obtained in the rest of classes. However, still low accuracies are obtained for the undefined class, as before, and clear blob. The embeddings distributions displayed in Figure 5.7 and 5.8 show how adding 25 SB3 images in training helps embeddings belonging to the same to closer together. However, not very distinguishable clusters are obtained in any of the cases.

	Bubbles	Clear Blob	Dilated	Turbid	Undefined	Wall	Wrinkle
Bubbles	0	0	0	100	0	0	0
Clear Blob	0	4	0	88	0	0	8
Dilated	0	0	0	0	0	0	0
Turbid	0	0	0	100	0	0	0
Undefined	0	0	28	56	12	0	4
Wall	0	0	8	92	0	0	0
Wrinkle	0	0	0	40	12	0	48

TABLE 5.6: Normalised confusion matrix in testing using the ResNet with Cross-Entropy Loss model.

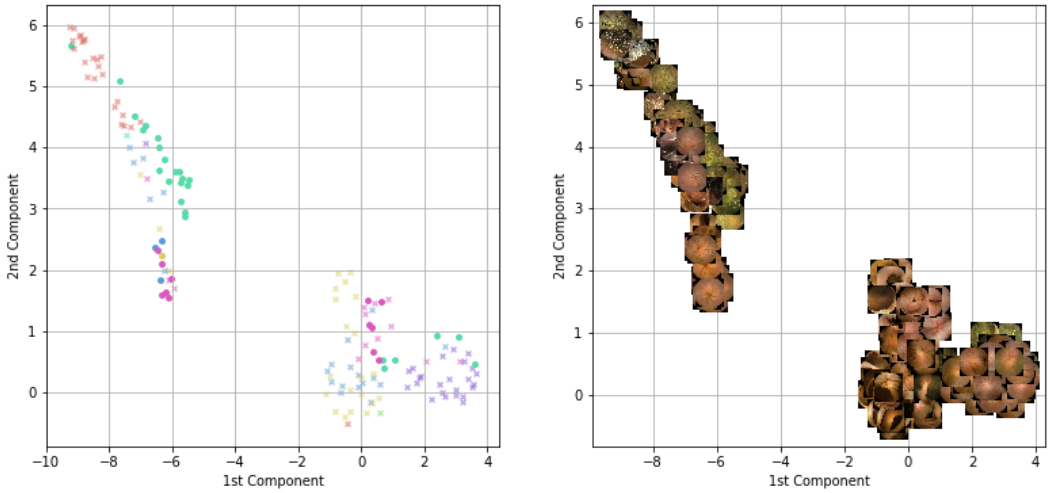

FIGURE 5.7: Embeddings distribution in testing using the ResNet with Cross-Entropy Loss model trained with SB2 images.

	Bubbles	Clear Blob	Dilated	Turbid	Undefined	Wall	Wrinkle
Bubbles	92	0	0	4	4	0	0
Clear Blob	0	64	0	0	24	0	12
Dilated	0	0	0	0	0	0	0
Turbid	8	0	0	88	0	0	4
Undefined	0	16	0	8	68	4	4
Wall	4	0	0	0	0	96	0
Wrinkle	0	0	0	4	0	12	84

TABLE 5.7: Normalised confusion matrix in testing using the ResNet with Cross-Entropy Loss model, adding 25 SB3 images in training.

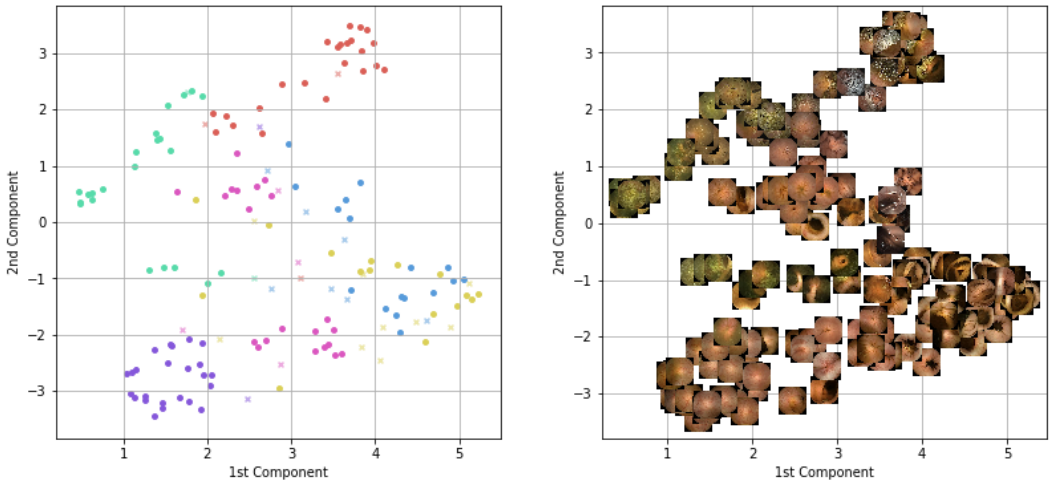

FIGURE 5.8: Embeddings distribution in testing using the ResNet with Cross-Entropy Loss model trained with SB2 images and 25 SB3 images.

Finally, the same tables and plots are included for the ResNet with Triplet Loss model. Firstly, Tables 5.8 and 5.9 display the normalised confusion matrices. In this

case, if no SB3 images are added in training then almost all the images are classified as turbid or wrinkle. However, if 25 SB3 images are added for training the accuracies improve significantly for all the classes, achieving 100% for the wall class and also very specially results for turbid and wrinkle images. The plots from Figure 5.9 and 5.10 show how adding SB3 images in training helps the clusters of each class to be more separated in the embedding space.

	Bubbles	Clear Blob	Dilated	Turbid	Undefined	Wall	Wrinkle
Bubbles	28	0	4	68	0	0	0
Clear Blob	8	12	48	32	0	0	0
Dilated	0	0	0	0	0	0	0
Turbid	12	0	0	88	0	0	0
Undefined	4	0	60	28	8	0	0
Wall	8	0	4	48	4	32	4
Wrinkle	0	0	8	4	0	0	88

TABLE 5.8: Normalised confusion matrix in testing using the ResNet with Triplet Loss model.

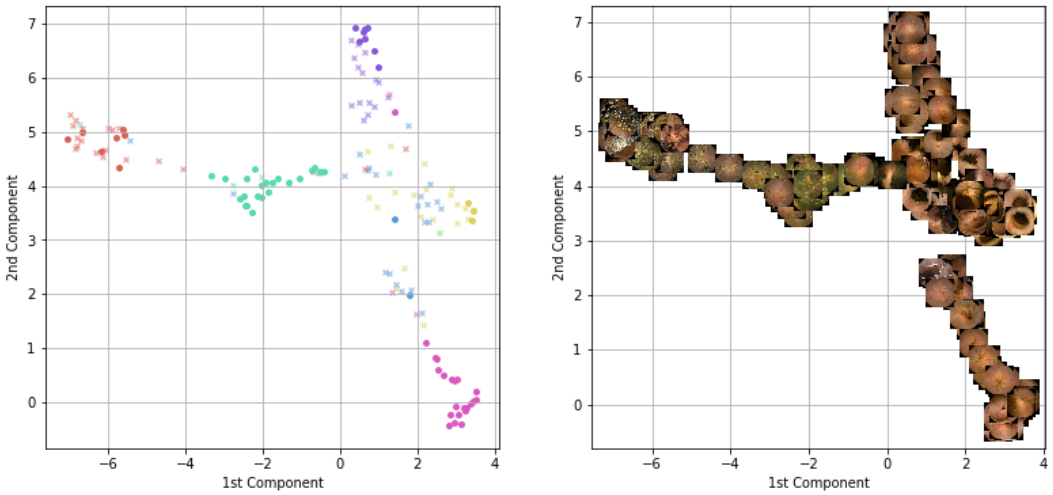

FIGURE 5.9: Embeddings distribution in testing using the ResNet with Triplet Loss model trained with SB2 images.

	Bubbles	Clear Blob	Dilated	Turbid	Undefined	Wall	Wrinkle
Bubbles	84	0	0	12	0	4	0
Clear Blob	0	80	0	0	12	0	8
Dilated	0	0	0	0	0	0	0
Turbid	4	0	0	96	0	0	0
Undefined	0	4	0	8	84	0	4
Wall	0	0	0	0	0	100	0
Wrinkle	0	0	0	4	4	0	92

TABLE 5.9: Normalised confusion matrix in testing using the ResNet with Triplet Loss model, adding 25 SB3 images in training.

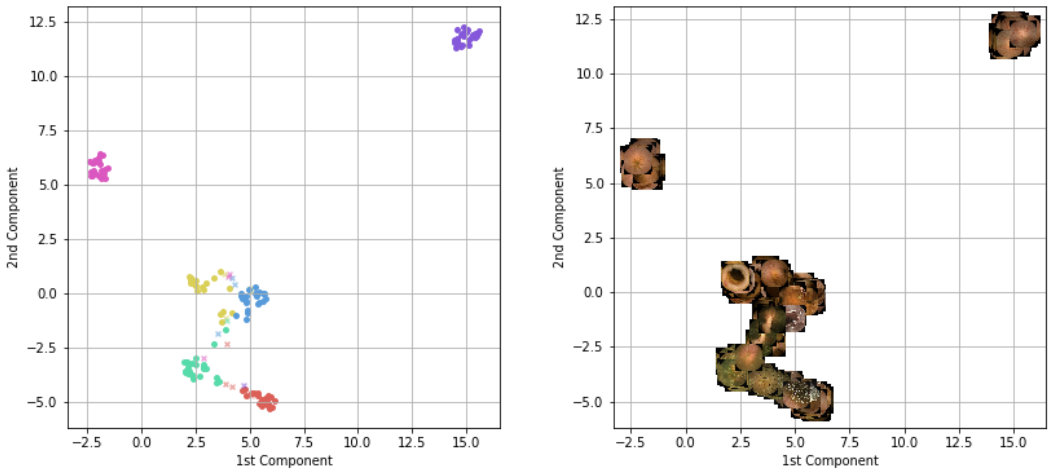

FIGURE 5.10: Embeddings distribution in testing using the ResNet with Triplet Loss model trained with SB2 images and 25 SB3 images.

As a general comment on all the previous results, all the plots have shown how triplet loss actually works for clustering in the embedding space. Moreover, in all the cases, better accuracy results were obtained with the second approach, which means that triplet loss also helps increasing accuracy in this domain adaptation scenario.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

To begin with, the goal of developing a model that is able to classify the images produced by PillCam SB3 has been fulfilled and is already in use. A maximum accuracy of 89.33% has been obtained, even higher for some specific classes such as turbid, wall or wrinkle. This result is very promising, since only 25 images per class from the target domain were included in the training dataset. However, it is necessary to say that the model should be used to classify images whose probabilities of belonging to a specific class are high, and with supervision of an expert for the rest.

On the other hand, the hypothesis that triplet loss improves the domain adaptation results has been proved to be true. In all the cases, better results were obtained using triplet loss to separate the embeddings. However, it is convenient to remark that even when training and testing on images from the same camera, the results were better if triplet loss was used.

Moreover, even though this was not an explicit goal of this thesis, 95% is a significantly good enough accuracy to say that the ResNet with Triplet Loss model has resulted in a very good method for classifying images from PillCam SB2.

6.2 Future Work

Many challenges are still open for the project, which faces complicated situations such as low availability of labelled datasets and developing classifiers for WCE images, which are usually very complex. More specifically, the following aspects could be studied in order to follow with the research line of this thesis.

Only 25 labelled images per class from PillCam SB3 is a very small number that could be increased. Including more, or more diverse, SB3 images in training would very likely lead to better accuracy results. Also, it would help answering a very interesting question: how many labelled images from the target domain are needed to be included in the training dataset in order to achieve the same accuracy as if testing on the source domain? In this case, this is equivalent to asking: how many labelled SB3 images have to be included in the training dataset in order to achieve 95% accuracy on the SB3 images?

Apart from actually classifying the images from PillCam SB3, this work could be used for other domain adaptation scenarios, very useful for this project. For instance, images from other procedures, such as colonoscopy, could be used for training models and classify WCE images.

In addition, some efforts could be dedicated to achieve better results with the ResNet with Cross-Entropy Loss model when testing on images from PillCam SB2.

In this way, the experiments in domain adaptation would be more deterministic on deciding whether triplet loss is useful or not.

Bibliography

Bae, S. and K. Yoon (2015). "Polyp Detection via Imbalanced Learning and Discriminative Feature Learning". In: *IEEE Transactions on Medical Imaging* 34.11, pp. 2379–2393. ISSN: 0278-0062. DOI: 10.1109/TMI.2015.2434398.

Basheer, I.A and M Hajmeer (2000). "Artificial neural networks: fundamentals, computing, design, and application". In: *Journal of Microbiological Methods* 43.1. Neural Computting in Micrbiology, pp. 3 –31. ISSN: 0167-7012. DOI: https://doi.org/10.1016/S0167-7012(00)00201-3. URL: http://www.sciencedirect.com/science/article/pii/S0167701200002013.

Bengio, Yoshua (1991). "Artificial Neural Networks and Their Application to Sequence Recognition". UMI Order No. GAXNN-72116 (Canadian dissertation). PhD thesis.

Ciuti, Gastone, Arianna Menciassi, and Paolo Dario (Jan. 2011). "Capsule Endoscopy: From Current Achievements to Open Challenges". In: *IEEE reviews in biomedical engineering* 4, pp. 59–72. DOI: 10.1109/RBME.2011.2171182.

Cobrin, Gena M., Robert H. Pittman, and Blair S. Lewis (2006). "Increased diagnostic yield of small bowel tumors with capsule endoscopy". In: *Cancer* 107.1, pp. 22–27. DOI: 10.1002/cncr.21975. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cncr.21975. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cncr.21975.

Deng, Weijian, Liang Zheng, and Jianbin Jiao (2018). "Domain Alignment with Triplets". In: *CoRR* abs/1812.00893. arXiv: 1812.00893. URL: http://arxiv.org/abs/1812.00893.

Eliakim, Rami (June 2004). "Wireless capsule video endoscopy: Three years of experience". In: *World journal of gastroenterology : WJG* 10, pp. 1238–9. DOI: 10.3748/wjg.v10.i9.1238.

Girshick, Ross B. (2015). "Fast R-CNN". In: *CoRR* abs/1504.08083. arXiv: 1504.08083. URL: http://arxiv.org/abs/1504.08083.

Girshick, Ross B. et al. (2013). "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *CoRR* abs/1311.2524. arXiv: 1311.2524. URL: http://arxiv.org/abs/1311.2524.

He, Kaiming et al. (2014). "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". In: *CoRR* abs/1406.4729. arXiv: 1406.4729. URL: http://arxiv.org/abs/1406.4729.

— (2015a). "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385. arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

— (2015b). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *CoRR* abs/1502.01852. arXiv: 1502.01852. URL: http://arxiv.org/abs/1502.01852.

— (2015c). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *CoRR* abs/1502.01852. arXiv: 1502.01852. URL: http://arxiv.org/abs/1502.01852.

— (Oct. 2016). "Identity Mappings in Deep Residual Networks". In: vol. 9908, pp. 630–645. ISBN: 978-3-319-46492-3. DOI: 10.1007/978-3-319-46493-0_38.

Hermans, Alexander, Lucas Beyer, and Bastian Leibe (2017). "In Defense of the Triplet Loss for Person Re-Identification". In: *CoRR* abs/1703.07737. arXiv: 1703.07737. URL: http://arxiv.org/abs/1703.07737.

Hinton, G. E. and T. J. Sejnowski (1986). "Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1". In: ed. by David E. Rumelhart, James L. McClelland, and CORPORATE PDP Research Group. Cambridge, MA, USA: MIT Press. Chap. Learning and Relearning in Boltzmann Machines, pp. 282–317. ISBN: 0-262-68053-X. URL: http://dl.acm.org/citation.cfm?id=104279.104291.

Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh (July 2006). "A Fast Learning Algorithm for Deep Belief Nets". In: *Neural Comput.* 18.7, pp. 1527–1554. ISSN: 0899-7667. DOI: 10.1162/neco.2006.18.7.1527. URL: http://dx.doi.org/10.1162/neco.2006.18.7.1527.

Iddan, Gavriel J. et al. (2000). "Wireless capsule endoscopy." In: *Nature* 405 6785, p. 417.

Ioffe, Sergey and Christian Szegedy (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *CoRR* abs/1502.03167. arXiv: 1502.03167. URL: http://arxiv.org/abs/1502.03167.

Iwahori, Yuji et al. (2015). "Automatic Detection of Polyp Using Hessian Filter and HOG Features". In: *Procedia Computer Science* 60. Knowledge-Based and Intelligent Information & Engineering Systems 19th Annual Conference, KES-2015, Singapore, September 2015 Proceedings, pp. 730 –739. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.08.226. URL: http://www.sciencedirect.com/science/article/pii/S1877050915023534.

Khatib, Alaa El, Naoufel Werghi, and Hussain Al-Ahmad (2015). "Automatic polyp detection: A comparative study". In: *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2669–2672.

Kouw, Wouter M. (2018). "An introduction to domain adaptation and transfer learning". In: *CoRR* abs/1812.11806. arXiv: 1812.11806. URL: http://arxiv.org/abs/1812.11806.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

LeCun, Yann and Yoshua Bengio (1998). "The Handbook of Brain Theory and Neural Networks". In: ed. by Michael A. Arbib. Cambridge, MA, USA: MIT Press. Chap. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258. ISBN: 0-262-51102-9. URL: http://dl.acm.org/citation.cfm?id=303568.303704.

Li, B. et al. (2009). "Intestinal polyp recognition in capsule endoscopy images using color and shape features". In: *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1490–1494. DOI: 10.1109/ROBIO.2009.5420969.

Li, Baopu and Max Q.-H. Meng (2012). "Automatic polyp detection for wireless capsule endoscopy images". In: *Expert Systems with Applications* 39.12, pp. 10952 – 10958. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2012.03.029. URL: http://www.sciencedirect.com/science/article/pii/S095741741200499X.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2014). "Fully Convolutional Networks for Semantic Segmentation". In: *CoRR* abs/1411.4038. arXiv: 1411.4038. URL: http://arxiv.org/abs/1411.4038.

Malagelada, C et al. (Nov. 2011). "Functional gut disorders or disordered gut function? Small bowel dysmotility evidenced by an original technique". In: *Neurogastroenterology and motility : the official journal of the European Gastrointestinal Motility Society* 24, 223–8, e104. DOI: 10.1111/j.1365-2982.2011.01823.x.

Mustafa, Barzin F et al. (2013). "Small bowel video capsule endoscopy: an overview." In: *Expert review of gastroenterology & hepatology* 7 4, pp. 323–9.

Palm, G. (1986). "Warren McCulloch and Walter Pitts: A Logical Calculus of the Ideas Immanent in Nervous Activity". In: *Brain Theory*. Ed. by Günther Palm and Ad Aertsen. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 229–230. ISBN: 978-3-642-70911-1.

Ren, Shaoqing et al. (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes et al. Curran Associates, Inc., pp. 91–99. URL: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf.

Schroff, Florian, Dmitry Kalenichenko, and James Philbin (2015). "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *CoRR* abs/1503.03832. arXiv: 1503.03832. URL: http://arxiv.org/abs/1503.03832.

Seguí, Santi et al. (2016). "Generic feature learning for wireless capsule endoscopy analysis". In: *Computers in Biology and Medicine* 79, pp. 163 –172. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2016.10.011. URL: http://www.sciencedirect.com/science/article/pii/S0010482516302712.

Simonyan, K. and A. Zisserman (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations*.

Srivastava, Nitish et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Szegedy, Christian et al. (2014). "Going Deeper with Convolutions". In: *CoRR* abs/1409.4842. arXiv: 1409.4842. URL: http://arxiv.org/abs/1409.4842.

Urgesi, Riccardo et al. (2012). "Increased Diagnostic Yield of Small Bowel Tumors with Pillcam: The Role of Capsule Endoscopy in the Diagnosis and Treatment of Gastrointestinal Stromal Tumors (GISTs). Italian Single-Center Experience". In: *Tumori Journal* 98.3. PMID: 22825512, pp. 357–363. DOI: 10.1177/030089161209800313. eprint: https://doi.org/10.1177/030089161209800313. URL: https://doi.org/10.1177/030089161209800313.

Yang, Liping et al. (Jan. 2014). "Increased Diagnostic Yield of Capsule Endoscopy in Patients with Chronic Abdominal Pain". In: *PLOS ONE* 9.1, pp. 1–4. DOI: 10.1371/journal.pone.0087396. URL: https://doi.org/10.1371/journal.pone.0087396.

Yu, Jiasheng et al. (2015). "A hybrid convolutional neural networks with extreme learning machine for WCE image classification". In: *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1822–1827.

Yuan, Y. and M. Q. . Meng (2014). "A novel feature for polyp detection in wireless capsule endoscopy images". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5010–5015. DOI: 10.1109/IROS.2014.6943274.

Yuan, Yixuan and Max Q.-H. Meng (Feb. 2017). "Deep Learning for Polyp Recognition in Wireless Capsule Endoscopy Images". In: *Medical Physics* 44. DOI: 10.1002/mp.12147.

Zou, Y. et al. (2015). "Classifying digestive organs in wireless capsule endoscopy images based on deep convolutional neural network". In: *2015 IEEE International*

Conference on Digital Signal Processing (DSP), pp. 1274–1278. DOI: 10.1109/ICDSP.2015.7252086.