UNIVERSITAT DE BARCELONA

Uncertainty, interpretability and dataset limitations in Deep Learning

Guillem Pascual i Guinovart

Aquesta tesi doctoral està subjecta a la llicència ***Reconeixement 4.0. Espanya de Creative Commons***.

Esta tesis doctoral está sujeta a la licencia ***Reconocimiento 4.0. España de Creative Commons.***

This doctoral thesis is licensed under the ***Creative Commons Attribution 4.0. Spain License.***

Doctor of Philosophy
Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

Uncertainty, interpretability and dataset limitations in Deep Learning

Guillem Pascual i Guinovart

30-09-2022

Director

Dr. Santi Seguí Mesquida
Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

Co-director

Dr. Jordi Vitrià Marca
Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

Abstract

Deep Learning (DL) has gained traction in the last years thanks to the exponential increase in compute power. New techniques and methods are published at a daily basis, and records are being set across multiple disciplines. Undeniably, DL has brought a revolution to the machine learning field and to our lives. However, not everything has been resolved and some considerations must be taken into account.

For instance, obtaining uncertainty measures and bounds is still an open problem. Models should be able to capture and express the confidence they have in their decisions, and Artificial Neural Networks (ANN) are known to lack in this regard. Be it through out of distribution samples, adversarial attacks, or simply unrelated or nonsensical inputs, ANN models demonstrate an unfounded and incorrect tendency to still output high probabilities. Likewise, interpretability remains an unresolved question. Some fields not only need but rely on being able to provide human interpretations of the thought process of models. ANNs, and specially deep models trained with DL, are hard to reason about. Last but not least, there is a tendency that indicates that models are getting deeper and more complex. At the same time, to cope with the increasing number of parameters, datasets are required to be of higher quality and, usually, larger. Not all research, and even less real world applications, can keep with the increasing demands.

Therefore, taking into account the previous issues, the main aim of this thesis is to provide methods and frameworks to tackle each of them. These approaches should be applicable to any suitable field and dataset, and are employed with real world datasets as proof of concept.

First, we propose a method that provides interpretability with respect to the results through uncertainty measures. The model in question is capable of reasoning about the uncertainty inherent in data and leverages that information to progressively refine its outputs. In particular, the method is applied to land cover segmentation, a classification task that aims to assign a type of land to each pixel in satellite images. The dataset and application serve to prove that the final uncertainty bound enables the end-user to reason about the possible errors in the segmentation result.

Second, Recurrent Neural Networks are used as a method to create robust models towards lacking datasets, both in terms of size and class balance. We apply them to two different fields, road extraction in satellite images and Wireless Capsule Endoscopy (WCE). The former demonstrates that contextual information in the temporal axis of data can be used to create models that achieve comparable results to state-of-the-art while being less complex. The latter, in turn, proves that contextual information for polyp detection can be crucial to obtain models that generalize better and obtain higher performance.

Last, we propose two methods to leverage unlabeled data in the model creation process. Often datasets are easier to obtain than to label, which results in many wasted opportunities with traditional classification approaches. Our approaches based on self-supervised learning result in a novel contrastive loss that is capable of extracting meaningful information out of pseudo-labeled data. Applying both methods to WCE data proves that the extracted inherent knowledge creates models that perform better in extremely unbalanced datasets and with lack of data.

To summarize, this thesis demonstrates potential solutions to obtain uncertainty bounds, provide reasonable explanations of the outputs, and to combat lack of data or unbalanced datasets. Overall, the presented methods have a positive impact on the DL field and could have a real and tangible effect for the society.

Acknowledgements

First of all, I would like to thank my wife, Aida Monsterrat. I am certain that without her constant and relentless encouragement and support, this thesis would not have been possible. She has been there in the best moments, but more than anything she has been a fundamental support when everything seemed impossible. The brainstorming sessions where each tried to help the other with their PhD have been invaluable, and, if that is not enough, she has read through all my publications and this manuscript. I cannot thank her enough for the dedication and effort that she has put into it.

I would also like to thank my parents. Thank you for all the support throughout the years, for giving me the means and encouragement to reach this goal. I am also grateful to all my family who, even if what I did seemed far-fetched and otherworldly, expressed their happiness for me.

It has been a pleasure working alongside my supervisors Santi Seguí and Jordi Vitrià, what I have learned from you goes beyond machine learning and deep learning. Thank you for the opportunity you have given me of working in this amazing group and university, and for all the guidance and advice. I must also thank every other professor and doctor I have come to meet during all these years, as from each of them I could learn and progress.

It would not have been the same without all the friends I met. Thank you Pablo Laiz for listening to me and being more than just a work colleague. I grew accustomed to the simple act of having lunch with every member of the department, PhD colleague or otherwise, and the discussions that emerged from that. Also thank you Pedro Herruzo and Laura Portell for all those interesting conversations we had and moments shared. And, of course, I cannot forget about you Alejandro Cartas, for the days in my office would have been much duller if not for your constant knocks on the door. Trust me, I have come to miss them. And now that we are talking about knocking the door, I enjoyed each and every one of the times you came to visit me, Carla Montserrat.

I cannot finish the acknowledgements without thanking Cristian Muriel. You have been there since the start, listened to me, gave irreplaceable advice, and distracted me when I needed it. You too, David Ballester, I must thank you for being there for me all these years.

Finally, I would like to acknowledge the Spanish Ministry of Universities for its Formación de Profesorado Universitario (FPU) FPU16/06843 grant that has financed the stipends of the first four years of PhD. Likewise, I gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used during this research.

Declaration

I declare that this thesis was composed by myself, that the work contained here is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Guillem Pascual i Guinovart)

Contents

List of Figures

List of Tables

Acronyms

ANN Artificial Neural Network.

AUC Area Under the ROC Curve.

BLSTM Bidirectional Long Short-Term Memory.

BRNN Bidirectional Recurrent Neural Network.

CAD Computer-Aided Detection and Diagnosis.

CADe Computer-Aided Detection.

CADx Computer-Aided Diagnosis.

CNN Convolutional Neural Network.

CRF Conditional Random Field.

CV Computer Vision.

DA Data Augmentation.

DL Deep Learning.

FCN Fully Convolutional Neural Network.

GAN Generative Adversarial Network.

GIANA Gastrointestinal Image ANAlysis.

GPU Graphics Processing Unit.

GRU Gated Recurrent Units.

IoU Intersection over Union.

LSTM Long Short-Term Memory.

M2O many-to-one.

MCC Matthews correlation coefficient.

NLP Natural Language Processing.

ReLU Rectified Linear Unit.

RNN Recurrent Neural Network.

ROC Receiver Operating Characteristic Curve.

RTL Relative Triplet Loss.

S2S sequence-to-sequence.

SSL Self-supervised Learning.

SVM Support Vector Machines.

t-SNE t-Distributed Stochastic Neighbor Embedding.

TL Triplet Loss.

UMAP Uniform Manifold Approximation and Projection.

VGG Visual Geometry Group.

ViT Visual Transformer.

WCE Wireless Capsule Endoscopy.

Chapter 1

Introduction

Contents

1.1 Background

Deep Learning (DL) is a sub-field of machine learning inspired by the human brain that aims to create algorithms for automatic learning. Since its adoption, it has brought a technological revolution unlike any we have seen so far (Sejnowski, 2018). From smart devices that can understand natural and talked speeches, to the point that they can partake in complex conversations, to computer-assisted diagnosis of all kinds of pathologies. It has brought advances to healthcare, self-driving cars, automatic map generation, artificial intelligence for games, artwork and music generation, and advancements in an endless myriad of other domains. Through its multiple variants, DL has undeniably impacted our lives, spreading from the academic world to everyday life and achieving feats that were considered hard or even impossible.

1.1.1 Artificial Neural Networks

While the term DL is relatively new, its core algorithm and foundations are not. Artificial Neural Networks (ANNs) are the basic building blocks behind this upturn, and although disputed, multiple researchers attribute its conception to as early as the 1940s. Even the form upon which our current algorithms are based took form in 1974, taking inspiration from the human biological neural network; a series of interconnected neurons communicating by means of electrical pulses, where different arrangements and signals give birth to divergent thoughts, images, and any other outcomes (Ohbayashi and Hirasawa, 1974).

A neuron receives inputs from several other neurons, accumulating them as its core, and computing a certain function of that accumulated energy before sending it off to another neuron. Mathematically, a neuron f processes $f(x) = \sigma(w \sum x)$, where σ is a usually a non-linear function, known as the activation function, and w is the strength of the connection between two neurons. This last parameter, or weight, simulates an increase or decrease in signal, exciting or inhibiting the neuron connection to another neuron.

ANNs, as in biological neural networks, do not consider neurons as individual components but arranged in groups. In fact, ANNs are composed of several layers of neurons, each with a pre-determined number of neurons, and each with a potentially different activation function. Typical ANNs, although certain algorithms betray the norm, neurons in a layer are not interconnected among themselves, only sending their outputs to the neurons in the subsequent layer. Through continuously stacking layers, ANNs gain more capacity to perform more complex tasks, but they also require a higher number of operations to compute the task. The number of parameters in an ANN quickly explodes into the thousands or millions. For instance, one layer with 1024 neurons fully connected to another layer with 1024 neurons, a rather common instance in ANNs, creates a dense network composed of $1024^2 = 1048576$ parameters. Thus, their capacity to model a wide variety of data comes at a cost, a much higher complexity than any other algorithm to date, to such extent that their usability until very recently was severely handicapped. It is not without reason that they did not shine until DL appeared, while other contemporary algorithms like Support Vector Machiness (SVMs) (Boser et al., 1992) received more attention and were more widespread Vapnik (1995, 1998). Only thanks to the monumental advances in hardware, and particularly parallel processing hardware such as Graphics Processing Units (GPUs), that DL has become the star it is today (Raina et al., 2009).

1.1.2 Deep Learning

DL is a natural extension to ANNs, keeping its inspiration and overall design, but exponentially increasing the number of neurons that can be used simultaneously. Modern computational power and advances in neural network architecture creation allow stacking of layers upon layers of neurons, creating denser and larger networks of interconnections that learn to adapt to their inputs, eventually producing the desired output. In fact, one of the many sources that gives DL its name, is the ability to create deeper models, with many more layers (Zhang et al., 2018). Even today, with the hardware advances and modern computing power, there exist DL models that have to train for a week on one hundred or more of the most advanced GPUs (Lin et al., 2021). One can only imagine how long it would have taken with the equipment available when ANNs, or even SVMs, first appeared.

Seeing their unparalleled performance in such a wide range of applications, a natural question arises: what is it that ANNs and DL do that other algorithms could not? Among the wide variety of valid answers, their ability to function as a class of universal function approximators is of particular interest (Hornik et al., 1989; Barron, 1993). That is, they can work with high dimensionality, a feat that most other algorithms would struggle with, while being able to approximate any underlying function given a series of inputs and outputs. In other words, ANNs have the ability to model an unknown function capable of mapping inputs, which can be images, videos, audio, sound, or any other signal, to the desired output.

Their success, though, must only be attributed to their capacity to stack millions of parameters into a single model. If anything, that is only a contributing factor to their success. ANNs and DL, through Convolutional Neural Networks (CNNs), opened the door to a new era of feature selection (LeCun et al., 1998). Researchers and programmers alike did no longer need to manually come up with features to describe their data, handcrafting complex heuristics that could aid in their objective; that task became fully automated through CNNs, having DL come up with the features that best worked for the particular dataset and objective. CNNs were, as a matter of fact, the starter gun in the race for ever-improving results.

Another characteristic that might have helped ANNs become what they are today, is their re-usability and composability. For example, parts of a model designed to detect common objects and entities in real-life images can then be used for historical document image analysis (Studer et al., 2019). A model trained to detect cars in a videogame can be re-used in real busy streets (Martinez et al., 2017), and a network that has learned to generate speech in English can be slightly altered to speak Spanish (Byambadorj et al., 2021). Researchers have formed a wide network through open-access publications, creating building blocks that other people can build upon. It is not hard to imagine that an environment that produces high-end products, makes them freely available and encourages their further development and refinement, can grow to become an authentic revolution (Braun and Ong, 2018).

1.2 Motivation

It would be easy for a reader to think that DL, through its many branches and variants, can perfectly solve all our existing problems. The truth, however, is that it comes with its own set of limitations. For each of its amazing properties, one must trade features that came for granted with traditional algorithms. This section explores such limitations and frames them in the context of this thesis.

1.2.1 Interpretability

Interpretability is the ability to reason about the results and to deduce what the model has used to arrive at its conclusion. Many fields rely on such capacity to perform business, such as banking, where they are hesitant towards implementing ANNs for their decision-making processes (Office of the Comptroller of the Currency, 2021; Bussmann et al., 2021). Probably, using ANNs could improve their existing predictive models, and perhaps obtain better profit margins, or make more accurate decisions in loan granting. What is stopping them is not the power behind ANNs, but rather its lacking interpretability. The problem comes when the bank has to explain to a client why their model has decided that they are not approved for the loan. An ANN, especially modern variants with high complexity, are like big black boxes; they receive an input—a set of numbers describing the client and their finances—and yield and output—whether the loan should be granted or not. Anything that happens inside, besides the fact that several mathematical operations take place, is unknown. Has it been their income? The number of properties in their name? Perhaps the number of children under their care? Obtaining a human-interpretable explanation is a hard-to-solve problem (Zhang et al., 2020). Decision trees, for instance, albeit their much more limited capacity, give a clear set of rules that explain how a certain decision has been reached (Quinlan, 1986).

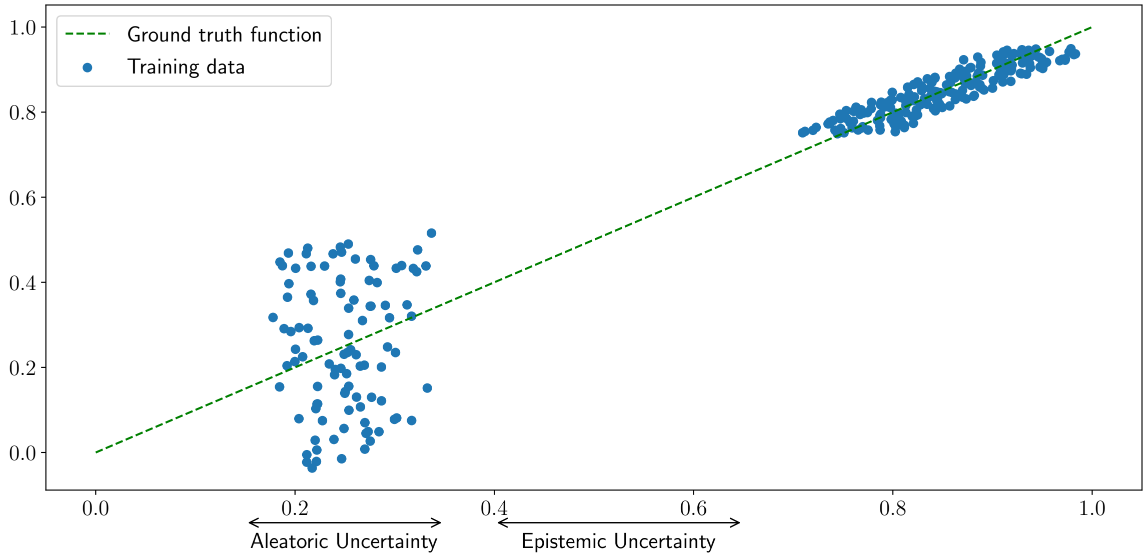


Figure 1.1: Graphical representation of uncertainty types. Here the ground truth function is the real and unobtainable distribution of the data, while the training data is the observed samples gathered from the distribution. Sampled data points from 0.2 to 0.3 are highly aleatoric, as they contain specific noise dependent on the sample itself. From 0.4 to 0.6, the lack of observed data indicates epistemic uncertainty.

1.2.2 Uncertainty

Closely related to interpretability, and sometimes explored in conjunction, is uncertainty. A standard ANN used in a setting as above will give a binary decision—yes or no—, and at most, the probability associated with that decision. A distinction must be made, however, that the probability of that decision is not an indication of the confidence in that decision (Guo et al., 2017). In fact, it can be easily proven that a network can give a high probability of a sample being correct for data that is either purposely manipulated (Goodfellow et al., 2014b), completely out of scope (Amodei et al., 2016), or outright nonsensical (Nguyen et al., 2014). Of course, that is not what the expected output should be for any of those inputs. Given an uncertain sample, the ANN should either notify that no clear answer can be given, or its output probability should indicate that it does not know where to assign it. Ideally, the end user should be able to discard outputs for which the network has made it clear that further classification would be useless. In other words, the problem with ANNs is that they do not give a measure of uncertainty in their output, which limits their applicable range of uses. What would happen if an autonomous car is presented with a plane blocking the road? These situations for which it has probably never been trained should still be handled correctly, without taking random actions with high probability.

When talking about uncertainty, it must be distinguished between its two principal different sources, as depicted in Figure 1.1. One treats the absence of information, epistemic, while the other regards the noise in the data, aleatoric (Kiureghian and Ditlevsen, 2009). On the one hand, epistemic uncertainty comes from the knowledge, or lack thereof, contained in our dataset. The databases used to train models might be incomplete, erroneous, or simply fail to capture all possible interpretations of the data. Given this lack of knowledge, our model can make wrong assumptions and apply noise to the model weights, eventually producing wrong outputs. Theoretically, given an infinite source of correct data, this kind of uncertainty could be solved. Models such as gaussian processes tackle this lack of data by providing higher bounds around areas that lack data.

On the other hand, aleatoric uncertainty describes the noise that occurs directly on the data itself. This kind of uncertainty cannot be solved with more examples, as it happens stochastically in the observations. Furthermore, aleatoric uncertainty can be further divided between homoscedastic uncertainty, where the noise is constant in all samples, and heteroscedastic uncertainty, where the noise depends on the particular data itself. In other words, homoscedastic uncertainty comes from the task itself, which could be modeled with a single parameter for the whole database. Heteroscedastic uncertainty, however, must be bounded individually for each sample.

In an ideal world, models should be able to cope with both types of uncertainties. They should be able to detect that its weights might contain erroneous assumptions due to epistemic uncertainty, and further be able to provide information with regards to its confidence in a certain task by considering the knowledge derived from aleatoric uncertainty, and particularly hetesorcedastic sources. Correctly bounded, a model could provide numerical or visual confirmation of its decision process, helping towards producing more robust models.

1.2.3 Model complexity

Lastly, but not least important, is the complexity of ANN models. This can be measured from two distinct points of view. One is the amount of data that a model must consume to perform adequately, and the other is the number of operations that a model must perform to obtain a certain performance level. Certainly, ANNs can be trained with any size of dataset, but the computer power revolution has brought the possibility of using huge datasets with enormous models. Often, using datasets with more data and of better quality allows making

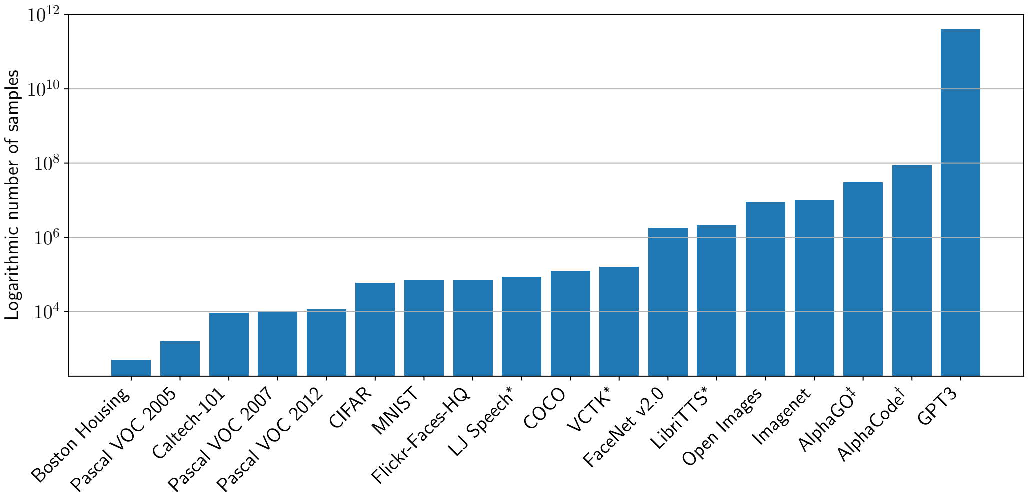


Figure 1.2: Most common and present datasets' samples count, shown in log-scale. For audio datasets, marked with *, the amount of samples is estimated as the cumulative seconds recorded instead of the number of samples. AlphaCode† is an underestimation which does not include the finetune dataset. Likewise, AlphaGO‡ does not consider the reinforcement training phase.

models with better results. As such, unsurprisingly, DL models are each time being trained with larger datasets (Patel and Thakkar, 2020). In fact, datasets starting at the far end of thousands of samples are commonly used nowadays, with some even reaching and surpassing the million mark (Byambadorj et al., 2021; Lin et al., 2021). While some works can obtain this data essentially free, as easily as downloading the entire English Wikipedia, not all fields are blessed like so. Gathering the data can be hard, but it can be even harder and more time-consuming to manually label all of them. For instance, images, videos, and sound, to name a few, are usually tagged with a description of their class, contents, or the appropriate feature that wants to be detected. For instance, in Wireless Capsule Endoscopy (WCE) it is rather straightforward to obtain huge datasets in the form of videos. Nevertheless, labeling the individual frames of several 12-hour long videos, comprised of thousands of images each, is a completely different story.

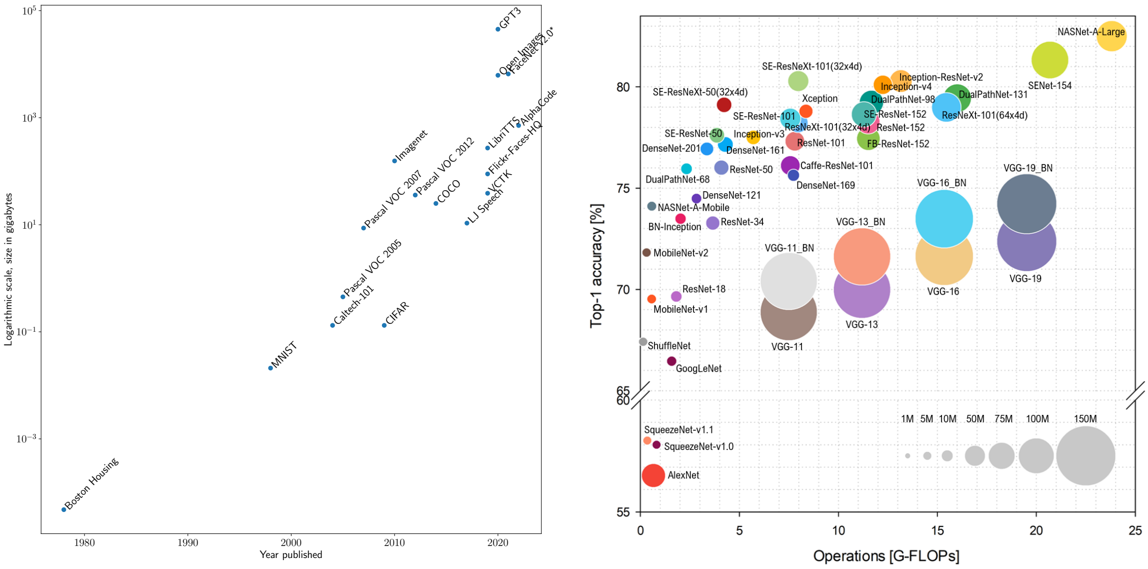
Figure 1.2 examines the complexity of several models from the standpoints outlined before. Figure 1.3a correlates the date when a dataset first appeared with the amount of data it uses. As can be observed, the plot is linear in log scale, which implies that sizes have been exponentially increasing over the years. Notably, the most recent advances in natural language have come with transformers models and, such as GPT3, which uses more than 40 billion tokens (Byambadorj et al., 2021). Likewise, Figure 1.3b is a study by Bianco et al. (2018) that compares the complexity of a model in the number of operations to its final accuracy. Again, the tendency is that better-performing models require a higher number of operations. More complex models, at the same time, often require larger and richer datasets to avoid common pitfalls like overfitting.

Thereafter, lacking data, and especially when combined with DL, usually leads to underperforming models. The model is prone to learn how to perfectly mimic the training dataset, which results in it later failing to generalize with new data. This behavior, which can be a product of a lack of data or too complex models, is referred to as overfitting (Brownlee, 2018). To put things into perspective, an overfitting model would overestimate its capability to predict new and unknown data, and produce completely wrong results. To picture the catastrophic effects this could have, we can go back to WCE example above. A system designed to detect polyps that can only recognize the exact same set of anomalies it has already seen would falsely claim that all patients are healthy. The implications can be, as seen, devastating.

Several techniques have been invented and applied to attempt to at least soften the problem. A common occurrence is applying what is known as Data Augmentation (DA), which mostly consists of using artificially altered versions of the data during the training process, effectively introducing veracious noise that augments the number of samples (Simard et al., 2003). Images, for example, can be altered through color jittering, by introducing hue, saturation, and brightness changes, by applying Gaussian or pepper noise on top of the image, by flipping and rotating it, and cropping and zooming, to name a few (Shorten and Khoshgoftaar, 2019).

ANN models also typically use some means of regularization. Most famously, L1 and L2 regularization are applied to prevent the network weights from growing too big, which in turn helps in preventing the model from adjusting too much to its input samples (Goodfellow et al., 2016). Similarly, although a completely different mechanism, are dropout layers. They aleatorically disable connections between neurons, effectively discarding weights and forcing the network to learn redundant information, which theoretically makes more robust models (Srivastava et al., 2014).

All these techniques, however, focus on overcoming a single problem—the lack of labeled data. Some, like DA, can be extended to cope with class imbalance. While not an optimal solution, they can be made to work by applying transformations to only the relevant classes. Nonetheless, ideal methods should be able to leverage all kinds of data, labeled or unlabeled, to produce more robust models, which do not suffer from overfitting. Thus, the focus is



(a) Size according to year presented

(b) Complexity to reach a determined accuracy, by Bianco et al. (2018)

Figure 1.3: Model complexity evaluated according to time and to desired accuracy. Figure (a) shows the most used and common datasets' sizes in gigabytes in log-scale. FaceNet v2.0, marked with *, is estimated from images count and sizes, as the dataset is private. Complimentary to the former, (b) plots the GFlops used per model as measured with the Top-1 accuracy in the Imagenet classification task.

shifted from having to acquire gigantic amounts of labeled data, which can be a hard task, to leveraging the use of any available information. One such way to incorporate that data is through the use of unsupervised models or Self-supervised Learning (SSL) (Liu et al., 2021a).

1.3 Objectives

It can be deduced from the previous section that using DL and training ANNs come with their own set of considerations and complications. The most prominent ones are, as outlined, the inability to interpret the results they provide, the lack of certainty or confidence in their decisions, and the amounts of data required. A broad definition of this thesis' aims would be to provide methods and algorithms that help tackle them.

Exploring solutions to provide uncertainty bounds should help in producing interpretable results, at least from the standpoint of reasoning about the confidence of the output. Additionally, uncertainty can be used to produce more robust models. A good candidate to explore this area and outline the advantages of uncertainty in satellite images. Due to their nature, images are captured at constant intervals without any other considerations, thus adverse conditions such as clouds, shadows due to the sun's position, or changing seasons, are common occurrences. All of them, particularly when not controlled in any way, are prominent sources of uncertainty. Moreover, the abundance of some elements, like woods, is much more dominant compared to human-made constructions like roads. Therefore, models that take into account these elements and provide interpretable results through uncertainty are a positive step towards overcoming two of the limitations pointed out in Section 1.2.

Further, leveraging contextual information in fields where the environment, such as satellite images, or the temporal axis, such as WCE, can be done through Recurrent Neural Networks (RNNs). Additional contextual information can aid in the learning process, which in turn helps in obtaining more accurate models. These models trained to learn from contextual information are more robust towards both lack of data and imbalances.

Similarly, fields where obtaining data is easier than labeling it might benefit from strategies that can leverage the whole dataset, including data that lacks labels. For instance, as stated before, a WCE video can easily span more than eight hours, which requires an enormous effort to label. Worse even, the resulting classes, particularly if looking for rare pathologies, are severely imbalanced. Unsupervised or semi-supervised techniques, like SSL, can be used to nullify those negative conditions.

To summarize, the goals of this dissertation are three:

1. **Produce uncertainty-aware models.** Create models that, aside from outputting a probability regarding its decision, also give a bound of its certainty. Additionally, this uncertainty must be interpretable for the user and provide tangible feedback. Effectively, this objective simultaneously tackles the problem of uncertainty and, up to a certain point, introduces interpretability.

2. **Create context-aware models.** As argued above, context can be used to train better models with lacking datasets, both in size and in class balance. As such, one of the aims of this thesis is to investigate the improvements that can be obtained with RNNs and explicitly use them for training models.

3. **Create methods to tackle data unavailability.** Lack of labeled data is battled by using SSL, a variant of supervised learning that can take into account unlabeled data. The aim is to obtain models that obtain better results than their standard

counterparts and that, by extension, are more resilient towards overfitting to the most-represented classes.

4. **Apply the above methods to real-world cases.** Finally, all methods must be demonstrated to work with a diverse set of datasets to show the applicability of the approaches explored. Specifically, uncertainty is applied to land cover segmentation, context-awareness to road extraction and WCE data, and SSL to WCE datasets.

1.4 Contributions

Our contributions to the respective fields mentioned are outlined as follows:

- An initial baseline and proof-of-concept with WCE and DL, in which the task of classification was augmented with handcrafted features to provide more information to the algorithm. It demonstrates how additional information combined with a carefully designed architecture can help CNNs provide more reliable results. Early and late fusion of features are explored to evaluate which creates more robust models in combination with a pretrained network. This work was published in Computers in Biology and Medicine (Seguí et al., 2016).

- We train a model that using uncertainty is capable of self-improvement by iteratively refining a land cover segmentation process and providing more confident results. Uncertainty combined with deep supervision is explored as a means to create robust algorithms that make informed decisions. The model not only provides segmentation of satellite images but is also able to inform the user of the level of confidence at each step, as well as a global certainty, in a clearly interpretable way through the use of heatmaps. The resulting model was presented in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (Pascual et al., 2018).

- Through the use of RNNs, we demonstrate that WCE images can obtain useful information from its context. In fact, the work shows that using RNNs is a positive step towards more robust classifiers with WCE and video data. This work focuses on refining the results obtained by a previous WCE model by taking the images it has a harder time to classify and proving that robustness can be achieved when looking at the context images around it. The work was published in Diagnostics (Reuss et al., 2022).

- WCE data is notorious for its lack of labeled samples and highly imbalanced classes, which, as proved by several works, is an important obstacle towards creating precise models. Our work shows a novel method for pseudo-label extraction from unlabeled videos, which combined with SSL enables the use of whole videos in a pre-train step. This additional phase, as demonstrated in the publication, serves as an initial parameter selection that creates more powerful and accurate models. The compressed representations obtained from the original frames are shown to contain rich information, making them useful for multiple downstream tasks. Moreover, having an initial semi-supervised phase acts as a regularization step, by which the final network avoids overfitting and generalizes much better than its non-SSL counterparts. This work is published in Computers in Biology and Medicine (Pascual et al., 2022).

- We demonstrate that creating a contrastive loss sensible to frames-position in a WCE video not only outperforms other methods, including SSL for videos and previous WCE SSL attempts, but also provides better generalization guarantees in databases with low amounts of data. This work elaborates on the fact that image similarity in a video can be considered relative to the distance between frames, instead of a hard-coded threshold over which images further than n-frames are considered dissimilar. It extends the method presented in its predecessor publication, extending the pseudo-labeling process and the SSL application to create more coherent embeddings, with time-information better extracted from the underlying videos. The results show encouraging progress in multiple downstream tasks. The publication was presented at the International Conference in Pattern Recognition 2022, and at the time of writing, is pending publication.

1.5 Outline

Related work, including state-of-the-art publications, is examined in Chapter 2. ANNs and DL models that had a big impact in the field, and are thoroughly used in this thesis, are explained along with publications in RNNs, uncertainty estimation, segmentation, WCE imaging, and SSL.

Then, the document is divided into three distinct working packages. Each is oriented towards one of the specific objectives of the thesis. Namely, in the first uncertainly is explored, the second focuses on RNNs used to build more robust models, and finally, SSL is applied to WCE data.

The first part is covered in Chapter 3, where uncertainty bounds and their applications are explored in several tasks. Its effect in deep supervision models and, specifically, in land cover segmentation is investigated.

The second part bridges the work with satellite images with RNNs. Chapter 4 introduces the pertinent theoretical frame for RNNs and its applications to satellite databases. Then, Chapter 5 showcases its application with WCE data.

Finally, Chapter 6 applies SSL to WCE in a first attempt to solve data-derived problems. Chapter 7 dives deeper in SSL and proposes a novel approach to extract better temporal information from WCE videos.

Finally, Chapter 8 contains the final conclusions of this thesis, including a summary of all the accomplishments, a proposal for future work that could expand on the work presented, and some closing remarks.

Chapter 2

Related Work

Contents

This chapter aims to give an overview of each of the three working packages that this thesis focuses on. Also, it first gives a background on ANNs and DL methods, particularly showcasing the most remarkable models and architectures. Then, the fields of uncertainty, RNNs, and SSL are covered. Likewise, the applications to which those techniques are applied, land cover segmentation and WCE, are also detailed in this chapter.

2.1 Artificial Neural Networks

As briefly explained in Section 1.1.1, an ANN imitates a biological neural network, using neuron interconnections as the parameters, or weights, that can be finetuned to learn a

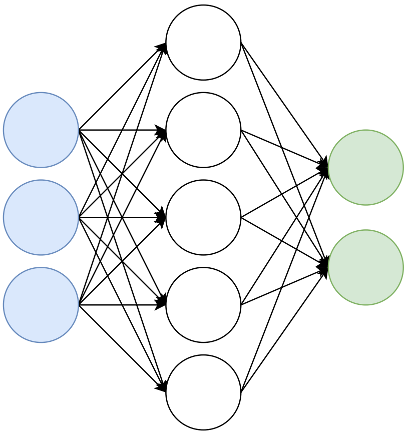


Figure 2.1: Schematic illustration of an Artificial Neural Network with three layers. The first layer, marked in blue, has three neurons, the middle one five neurons, and the output layer in green has two neurons. Represented by arrows, each neuron in a layer is connected to each neuron on the next one, forming weight.

particular task (Ohbayashi and Hirasawa, 1974). In fact, in matrix notation, denote $W^l_{i,j}$ the weights that connect the neuron i and j in the layers l and $l-1$, respectively. For instance, two layers l, m where all neurons are connected with one another would create a W matrix of size $N \times M$, where N is the number of neurons in l, and M the number of neurons in m.

As outlined before, each layer l computes the sum of the inputs x^l multiplied by the weights W^l, and then outputs the result of its activation function σ^l. Namely, a layer computes its outputs a^l as $a^l = \sigma^l(W^l x^l) = \sigma(z^l)$, which is then forwarded to the next layer. Composing for L layers, a network $f(x)$, where x is the initial input, can be viewed as the expression in Equation (2.1). Figure 2.1 depicts an schematic of a 3-layers ANN as expressed by the equation.

$$g(x) = \sigma^L(W^L \sigma^{L-1}(W^{L-1} \cdots \sigma^1(W^1 x) \cdots)) \tag{2.1}$$

To adjust the W parameters in the network, the model is trained by means of example repetition. A database's samples X and its respective labels Y are used as inputs for the first layer of the network, and the signals are propagated one layer at a time until they reach the final layer. The output of the ANN, that is, the result of the last layer, is compared against the label of the sample. The difference resulting from this computation is calculated through a cost function, also called error or loss, $E(X, W)$. This error is used as the basis for the next step, which attempts to modify the weights to obtain better outputs.

In a few words, each weight must be accounted for its contribution to the final error, which is done with an algorithm known as backpropagation (Rumelhart et al., 2013). After the forward part has been done, the error contribution for the last layer is computed as $\delta^L = \nabla_{a^L} E(X, w^L) \odot \sigma'(z^L)$, where $\odot$ denotes the Hadamard product and σ' the derivative of the activation function. Then, the error can be backpropagated to all previous layers by following the chain rule as shown in Equation (2.2). Finally, weights can be applied by moving in the negative direction of this error, Equation (2.3). The rate at which the weights change is usually further modified through α, the learning rate.

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) \tag{2.2}$$

$$w^l \rightarrow w^l - \alpha \delta^l (a^{l-1})^T \tag{2.3}$$

2.2 Deep Learning

DL is a multidisciplinary field that spans across a wide range of data types, applications, and uses. In fact, there is usually a cross-over between domains, like vision, text, and audio, where improvements in one can benefit the other. This study is mostly centered around Computer Vision (CV), but inevitably some references to other disciplines are done. It can be argued that one of the first models to step in what today is called DL was Alexnet (Krizhevsky et al., 2012). At its core, it is a CNN, a type of ANN that is invariant to scale and shift (Zhang et al., 1988; LeCun et al., 1998). CNNs receive their name from the use of convolutions with shared kernels, achieving a great capacity to filter input data with a low parameter count. Alexnet was not the first network to be run in a GPU, yet it was the first one to receive significant public attention. The concept is based on LeCun et al. (1998) LeNet, yet it takes advantage of modern hardware to introduce a much higher number of layers. The authors of Alexnet attributed its enhanced performance to the use of max-pooling (Scherer et al., 2010) in conjunction with reaching a much deeper depth. Additionally, Alexnet performed experiments to empirically test the best-performing activation functions, finding that deep networks benefited from the use of Rectified Linear Units (ReLUs) (Nair and Hinton, 2010) instead of the more commonly found hyperbolic tangents and sigmoid activations.

While Alexnet was already a significant improvement in the top-5 error (the percentage

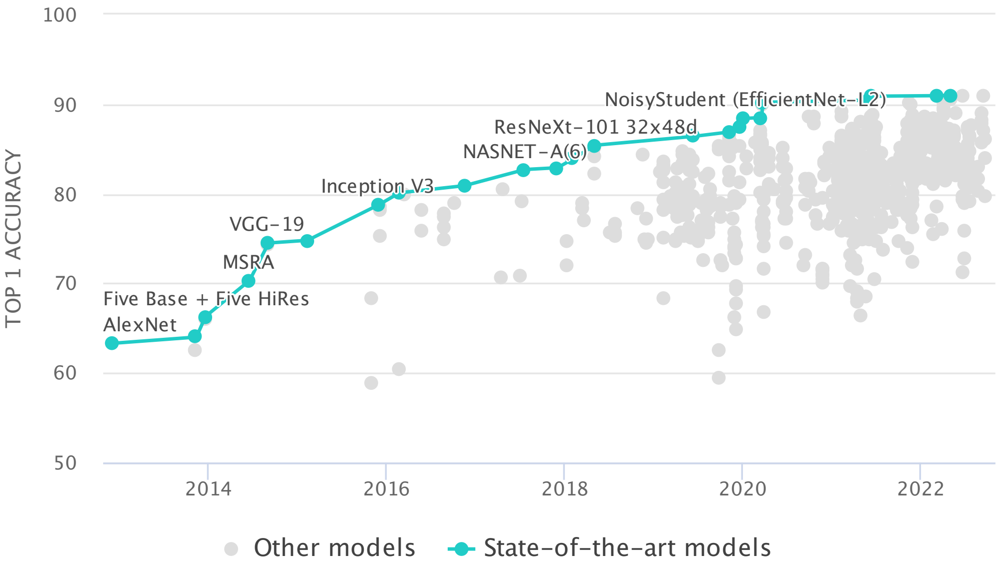


Figure 2.2: Overview of the models created for the Imagenet dataset along the years according to top-1 accuracy reproduced from Papers With Code (2022). The line in blue shows the state-of-the-art models for any given year, while the points in grey are other models developed in the same year.

of times the classifier failed to include a correct class amongst the top five guesses) in the ImageNet classification challenge (Deng et al., 2009), going from a 25.8% in 2011 to a 16.4% the next year, it only took two more years for another jump in the metric. Visual Geometry Group (VGG) was the spiritual successor, growing from the 8 layers that AlexNet proposed to two different variants, one with 16 layers and one with 19 (Simonyan and Zisserman, 2015). It was the latter version, known as VGG-19, that broke all records, setting the top-5 error at 7.3% in 2014. Architecture-wise, it shares many similarities with AlexNet, only differentiated by the use of various convolutional layers in sequence before applying the pooling operation. Notably, the number of parameters more than doubled, reaching 138M in total. Figure 2.2 shows an overview of the advances done in the Imagenet dataset according to the top-1 metric.

That same year, the best result was claimed by GoogLeNet (Szegedy et al., 2015), also called Inception, with a 6.7% top-5 error. The network, while deeper than both AlexNet and VGG-19, boasting a total of 22 layers, proposed a different approach to achieve better classification results. They argued that large networks are more prone to overfitting, and thus their solution sought to take advantage of wider steps with more spatial resolution.

Each step in the Inception network consists of several parallel convolutional layers, each with a bigger kernel size than the previous one. Combined with strategically placed poolings, the design allowed the network to capture details at different resolution levels, helping towards future generalization. To compensate for the larger number of layers and help in the learning progress, they introduced several intermediate classifications, using a technique that would be known as deep supervision (Wang et al., 2015).

In only one more year, the Imagenet challenge's leadership was once again snatched with a top-5 error of 3.57%. It was, in fact, a milestone in what DL represents, with the best model using a total of 152 layers. ResNet, the model in question, developed a technique to enable the training process of very deep networks (He et al., 2016). Conceptually, their network is almost equal to AlexNet and VGG proposals, yet the introduction of the so-called residual connections was groundbreaking. They grouped n convolutional layers into a single group and then used skip connections that directly linked the block's first layer input to the last layer's output. Essentially, this skip or residual connection bypassed any operation in that block, allowing information to be carried over unmodified. This seemingly straightforward alteration is what has allowed other networks to compose up to a thousand layers and still be able to converge without running into exploding or vanishing gradients (Li et al., 2018b; Balduzzi et al., 2017). While the application of skip connections to reach new depths was novel, the idea itself was not. Skip connections had been used before in U-Net for image segmentation (Weng et al., 2015), as will be detailed below in Section 2.3.2, and in parallel with Highway networks (Srivastava et al., 2015). The latter created an algorithm based on gating mechanisms that selectively allowed some information, or none, to be carried from previous layers. However, the design proved to be not as effective and as straightforward as residual connections in ResNet. Another noteworthy improvement of ResNet, aside from the lower error rate, is the reduced number of parameters with respect to its predecessors. With only 60.3M approximately and a never-seen-before depth, it successfully removed 1.7M compared to AlexNet.

During the coming years, many more efforts were poured towards obtaining better results (Bianco et al., 2018). Some of the research has focused on efficiency, producing models such as DenseNet (Huang et al., 2017) and MobileNet (Howard et al., 2017), which sacrifice some accuracy in exchange for a much lower parameter count, achieving greater operations per second and facilitating deployment in modest hardware. For instance, as the name indicates, MobileNet is targeted toward smartphones. Results-wise, the competition moved forward with ResNeXt (Xie et al., 2017), which empowers ResNet architectures by considering a wider residual block, and further with Squeeze-Excitation-based networks (Hu et al., 2020), which introduce a new layer capable of finding interdependencies in the convolutional layers' output channels. Architectures based on ResNet or derived from common CNN schemes peaked with the introduction of EfficientNet, which combined architecture search with a novel strategy of compound scaling of all dimensions (depth, width, and resolution) to surpass all other similar models (Tan and Le, 2019).

The next advancements, as hinted before, came from outside of the CV field. In fact, they originated in Natural Language Processing (NLP) and were later translated for CV. Without venturing too much into RNNs yet, which are explored in Section 2.3.3, all the traditional RNN architectures used for language translation suffer from a fatal failure point; they lose context on long sentences and fail to capture the correct meaning (Yang et al., 2020a). To investigate the nature of the issue, it must be considered that neural translation machines are, traditionally, a two-step process. The input phrase is first encoded into a set of one or more embeddings, and then is decoded by another section that outputs the target language tokens. Encoder-decoder schemes are not unique to NLP, and they are also found in CV. Be it phrases or images, these networks always break the input into smaller defining information, and then attempt to use this condensed definition to produce the output. Most problems in NLP arise during this decoding process, as the first tokens are forgotten by the time the network reaches the last ones. Notwithstanding, researchers found a way to circumvent such problems by introducing attention inside the decoder network (Vaswani et al., 2017). Self-attention is a mechanism through which the network is capable of using every single input token at a given timestep, distributing the weight it places on each token accordingly to the perceived importance. Such importance is automatically and mathematically determined through patterns between samples.

Arguably, NLP has seen another race towards better efficiency since the introduction of BERT (Devlin et al., 2018). The authors built on the idea of transformer models, a retake on encoder networks capable of retaining contextual information that was previously lost (Vaswani et al., 2017). BERT obtained new state-of-the-art results on eleven natural language processing tasks thanks to its ability to do both a forward and a backward pass during encoding, instead of the forward-only approach of other transformers and allowing it to extract more meaningful embeddings. Soon followed the GPT series models, which moved the transformer architecture to the decoders instead of the encoders and once again obtained state-of-the-art results. These transformer models also exceeded the previous models' complexity by large margins, with GPT-3 spanning over 175 billion parameters (Brown et al., 2020). Not much later, however, they were surpassed by Switch transforms, which broke the trillion parameters mark for the first time (Fedus et al., 2021).

Of course, the stunning results obtained in NLP did not take long to spark interest in other domains, CV not being an exception (Khan et al., 2021). Images are spatially and temporally coherent, as there is a clear relationship among adjacent pixels, which makes them prime candidates for transformer architectures. However, as considering each pixel individually could easily be prohibitive when the images increase in resolution, the first Visual Transformer (ViT) considered patches of 16 × 16 pixels (Dosovitskiy et al., 2020). The newfound success of transformers in CV can be attributed to their differences with CNNs, whereas the latter have limited spatial resolution to the kernel size, ViT has global information that can be shared across patches (Raghu et al., 2021). At the time this document was written, the best performing model in the Imagenet challenge was precisely a transformer model, CoCa, with a 91.0% top-1 accuracy (Yu et al., 2022).

2.3 Fields of investigation

Having covered the basic blocks for ANNs and DL, this section aims to explain the research done in each of this thesis' fields of interest. As such, uncertainty is covered in Section 2.3.2, segmentation in Section 2.3.2, RNNs in Section 2.3.3, and finally SSL in Section 2.3.4.

2.3.1 Uncertainty

Uncertainty is the ability of a model to procure not only a single prediction, but a distribution over predictions. As such, during classification, each sample would output a label and its confidence, while in regression settings, the mean would be accompanied by its variance (Tran et al., 2020). It is a critical measure that serves to build trust in the model decisions.

Having a confidence measure is important in all systems, and particularly in DL, where uncertainty can be found in the own model's weights or as an inherent part of the data. For instance, dataset shift is an example of the latter, where the data presented during inference deviates from that of training, and not necessarily by significant amounts (Hendrycks and Dietterich, 2019). Networks that are used with shifted data fail to capture the uncertainty of the samples, and not only are unable to inform of it, but they exhibit high confidence in the form of unnatural high classification probability (Ovadia et al., 2019; Nguyen et al., 2014). Data shift is not, however, the only situation where ANNs cannot behave properly. New classes might appear during test, classes' representation can change due to subpopulation shifts, or simply the labels themselves can shift (Tran et al., 2020); all these situations produce the same undesired results.

As briefly mentioned in Section 1.2.2, two different kinds of uncertainty are considered, that which comes from the model, epistemic, and aleatoric, which comes from the data.

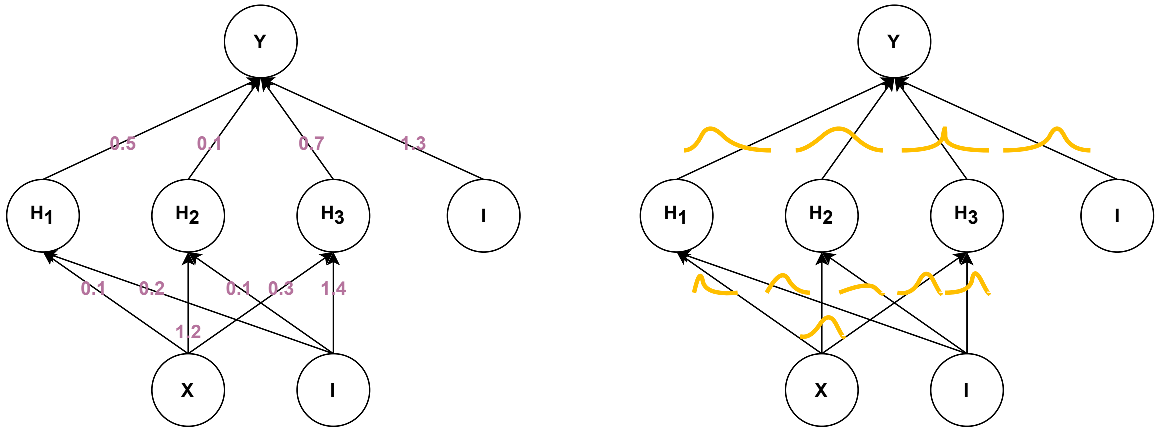
Through this thesis, the definitions taken for uncertainty are the ones proposed by Kiureghian and Ditlevsen (2009). All terms used to characterize uncertainty are described relative to model training, which is of utmost interest for this research. The authors further divide epistemic uncertainty into constant noise in the samples, homoscedastic, and data-dependent error, heteroscedastic.

On the one hand, there are algorithms other than DL that are inherently equipped with mechanisms to obtain epistemic uncertainty as part of its normal training process, such as Gaussian Processes (Williams and Rasmussen, 1995), and others that require little adaptation to output bounds, such as SVMs (Wang and Panos, 2015), or some whose parameters can be bounded, such as linear regressions (Altman and Gardner, 1988). In CV, other machine learning algorithms such as random fields were used for hand tracking by providing their contour and estimated bounds, the uncertain region where they might be found (Blake et al., 1993). Random fields have also been used to provide confidence in labeling processes by outlining the decision boundaries (He et al., 2004).

On the other hand, computing estimates for DL-based methods are inefficient or outright impossible if the number of parameters is high, as they do not provide bounds on their own. For once, ANNs usually report an inaccurately predicted probability of correctness relative to the observed frequency of correctness (Naeini et al., 2015). That is, empirically, the average number of times the network's classification matches the ground truth for a class is not equal to the average probability it outputs for that class. Efforts have been made to correct this behavior, as shown in studies that employ temperature scaling as a post-processing step to match both metrics (Guo et al., 2017). While such calibrations are useful and a positive step, the problem still remains that models lack a quantifiable measure of confidence.

Simplifying the common-case classification task, most models try to find a set of parameters θ^* that maximize the probability of correct classifications conditioned on data $\theta^* = \arg\max_\theta p(\theta|\mathbf{x}, \mathbf{y})$. Inadvertently, this results in the model outputting just one prediction per sample, which makes calculating bounds a hard task. Two classic solutions that enable estimating uncertainty, be it directly or through sampling, are (1) a probabilistic approach by estimating the full distribution for $p(\theta|\mathbf{x}, \mathbf{y})$, or (2) obtaining multiple θ^* through ensembles.

With regard to the first strategy, an attempt to provide estimates comes from Bayesian Neural Networks, as shown in Figure 2.3, which attempt to determine a distribution over the ANN weights $p(\theta)$ (Denker and LeCun, 1990; MacKay, 1992). To approximate the posterior distribution, which is multimodal and complex, a combination of local approximations



(a) ANN

(b) BNN

Figure 2.3: Comparison of (a) a standard ANN and (b) a Bayesian Neural Network. The former has scalar values for all its parameters, while the latter is defined by a distribution over each weight. This figure has been reproduced from Thakur (2022).

through variational inference and sampling can be used. A divergence measure such as Kullback–Leibler is used to optimize the divergence of a family of variational distributions with respect to the posterior (Kullback and Leibler, 1951). Then, Monte Carlo can be used to sample and estimate the final expectation, taking gradients for stochastic gradient descend (Robbins and Monro, 1951) and iteratively training the network. Interestingly, this approach can be generalized to Gaussian Processes (Neal, 1995; Lee et al., 2017b; Matthews et al., 2018).

Ensemble learning, on the other hand, skips all the approximate inference and instead aggregates over a collection of K models (Dietterich, 2000). Each model must not necessarily be a different machine learning algorithm, but it is required that they are separate training instances, each with its own random set of initial parameters and training process. Aggregating the combined classification outputs of all K samples provides a mixture distribution $p(y|x) = \frac{1}{K}\sum_{k=1}^{K} p(y|x, \theta_{\mathbf{k}})$, which can be used both to obtain a final classification output, such as the class that most models agree on, and at the same time can provide bounds and uncertainty information Lakshminarayanan et al. (2016).

Both methods, however, come at a great cost because the inference time and memory usage grow much more rapidly than that of a single model. Ideally, progress should be striving to improve the robustness and uncertainty bounds in single ANNs, which would bypass the issues that come with estimating whole distributions or using multiple models.

Efforts have already been made in that direction, for instance Liu et al. (2020) proposed a simple method named Spectral-normalized Neural Gaussian Process which uses distance awareness as a proxy for confidence estimation. To preserve input distance inside the ANNs hidden layers, which is normally lost, they use a spectral normalization layer. Similarly, Amersfoort et al. (2020) use a combination of radial basis function kernels and a novel loss function with centroids information to provide uncertainty estimates and reject out-of-distribution samples.

Aleatoric uncertainty, however, must be tackled in different ways. In non-Bayesian regression networks, homoscedastic uncertainty is often fixed as part of the model's weight decay, which means that any homogeneous noise from the data can be safely ignored (Kendall and Gal, 2017). However, its data-dependant counterpart, heteroscedastic uncertainty, requires more elaborate and specific solutions. In Bayesian networks, data-dependent uncertainty could be obtained by estimating individual mean and variance for each parameter in the conditional probability $p(y|x)$. Yet, when considering the whole database, these estimates summarize the conditional distributions into scalar values, which make them unable to model complex situations Nix and Weigend (1994). Instead, Kendall and Gal (2017) proposed the use of maximum a posteriori estimation inference, which unlike variational inference methods results in a single value for the model parameters θ^*. Regression is further developed into a method to introduce heteroscedastic uncertainty in classification tasks through a combination of loss attenuation, gaussian noise, and Monte Carlo sampling. Dropout has been used in Wang et al. (2018) to generate multiple samples at test time combined with entropy to obtain a proxy measure for uncertainty. A sampling-free method for aleatoric uncertainty estimation through approximated variance propagation was proposed by Postels et al. (2019), which does not require Monte Carlo sampling and is suitable both for regression and classification tasks. Later work by the same author proposes the use of density estimation in the latent space to once again provide uncertainty without sampling mechanisms (Postels et al., 2020). Variance decomposition has also been used to separate uncertainty estimates from the predictive variance of ensembles, effectively obtaining epistemic bounds (Egele et al., 2021).

2.3.2 Segmentation

Image segmentation can be described as a pixel-level classification problem, where each pixel must be assigned to a class. It is important to make a distinction between semantic segmentation and instance segmentation. While the first one attempts to assign each object to its semantic class, the second one tries to identify individual objects, assigning different

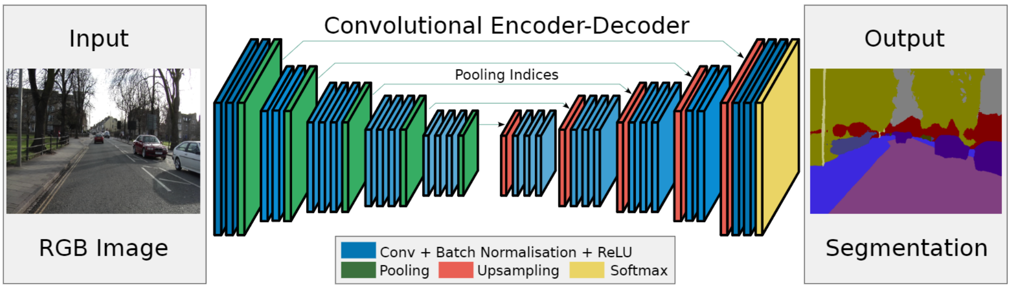


Figure 2.4: Badrinarayanan et al. (2015) proposed an architecture to perform semantic segmentation through an encoded-decoder architecture. An input image obtained for autonomous driving is passed through the encoder, sets of convolutions (blue blocks) ending in a pooling layer (in green). The resulting representation is used in the decoder network, a series of upsampling layers (in red) followed by convolutions. The segmentation output is obtained at the end of the decoder network. This image has been obtained from Badrinarayanan et al. (2015).

identifiers to instances of the same semantic (He et al., 2017; Li et al., 2016; Garcia-Garcia et al., 2018). In other words, the first type would differentiate humans from background in a street, but not tell a human apart from another, while the second one would find individual humans. Given this research focuses on the former class, semantic segmentation, the subsection is mostly focused on that same segmentation type. Figure 2.4 depicts an example of semantic segmentation for autonomous driving, showing that all cars get assigned to the same category.

Likewise, while some methods existed previous to the popularization of DL, their success was limited and not applied to the same extent as this thesis does. If there were enough computational resources, semantic segmentation could be done from bare ANNs or Markov random Random Fields. Even before those, segmentation could be done from intensity thresholds, iterative pixel classification, edge detection, and fuzzy methods to name a few (Pal and Pal, 1993). As hardware advanced and DL appeared, semantic segmentation was brought to the next level, especially so with Fully Convolutional Neural Networks (FCNs). As the name implies, FCNs are networks that consist solely of convolutions, erasing all dense operations. For instance, early attempts at FCN used modified networks, such as VGG-16 or GoogLeNet, where the last dense layer was replaced for convolutions. This seemingly minor modification had enormous ramifications, like allowing varying input and output sizes (Long et al., 2015). This first model by Long et al. (2015) used an early version of skip connections, although they did not name it as such. Their model proposed the use of skip connections to merge the features extracted at different levels and with varying kernel sizes, combining them into a single feature set. This FCN architecture obtained state-of-the-art results in all three different datasets tested. Further, ParseNet proposed the use of layer-averages to introduce some global context, which FCNs lacked (Liu et al., 2015). Famously and still in active use, Conditional Random Fields (CRFs) were proposed as a post-processing step for semantic segmentation (Chen et al., 2014). The output of a model is passed through a fully-connected CRF process, which improves the poor localization property of deep networks, helping in obtaining better boundaries between semantic objects.

Encoder-decoder architectures have already been mentioned as part of NLP models. FCNs have also employed the encoder-decoder scheme, and are in fact still found at its core nowadays. Usually, the encoder part is a renowned architecture such as VGG or ResNet, while the decoder is the same network but mirroring operations. As the encoder reduces the input resolution through both pooling and strides in convolutions, the decoder must recover the original size through unpooling or deconvolutions with dilation (Xu et al., 2014; Noh et al., 2015). SegNet follows the same idea of using deconvolutions and unpooling operations, but unlike previous architectures that required learning the upscaling operation, it proposes to use the same indexes used during pooling (Badrinarayanan et al., 2015) in the unpooling operations. In medical imaging, U-net set a precedent with the introduction of skip connections to encoder-decoder architectures (Weng et al., 2015), which allowed not only to recover the original resolution but to introduce fine details that were lost during the downsampling process.

Multiscale resolution through pyramids, decomposition of images into sets of exponentially lower resolution, also became relevant with networks such as FPN (Lin et al., 2016b) and PSPNet (Zhao et al., 2016), that process the input image at multiple scales before passing it through the actual convolutional network. By such means, it can detect objects and details at different resolutions before combining the features to obtain a final segmentation. Notably, most recent architectures have derived another mechanism to obtain features at different resolution levels. Dilated convolutions displace the pixels used by the kernel to effectively work at different space resolution (Yu and Koltun, 2015). DeepLab, for instance, combined dilated convolutions, spatial pyramid pooling, and probabilistic graphical models to obtain better and more accurate semantic segmentation (Chen et al., 2014).

Methods other than FCN also exist and have been widely used to achieve state-of-the-art results for semantic segmentation. For example, regional convolutional networks (RCNN) such as Masked-RCNN have found wide success by applying convolutions that conserve order at pixel-level (Minaee et al., 2020). Combinations of RNNs and CNNs also exist, as seen in ReSeg, that used four RNNs to sweep the image both vertically and horizontally, incorporating global context to the segmentation done by a VGG-16 model (Visin et al., 2015). Attention-based architectures have also been combined with existing ideas like pyramid pooling to extract fine and coarse features without using dilated convolutions (Li et al., 2018a). Amongst other novel architectures, active active contours models have been added into the mix by formulating new loss functions. Such models can output additional information on the area and size of segmentation results (Chen et al., 2019). And finally, a widely successful model named Gated-SCNN used a two-stream CNN architecture that separates shape information into a second independent branch to obtain a deeper understanding of the semantic information (Takikawa et al., 2019).

2.3.3 Recurrent Neural Networks

RNNs are a type of network specialized for sequential data and inputs that have a temporal axis, such as videos and audio. Unlike other ANN architectures where each input would use different weights, RNNs reuse the same weights for every element of the sequence, reducing the overall parameter count and introducing a recurrency that depends on the sequence (Ruineihart et al., 1985; Jordan, 1986). They are, like ANNs, a concept that has existed since a long time ago and has recently been rediscovered with DL. When they first appeared and in its original form, each element of the sequence would produce a hidden state in the RNN which was used for the next element, effectively carrying information in a memory-like implementation.

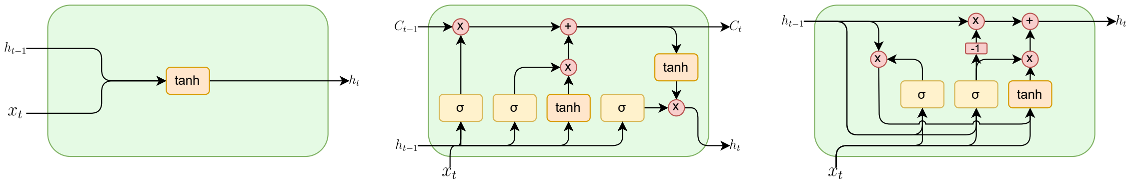
While the idea revolves around carrying a history that can be used in further steps, and is theoretically sound, its practical application soon discovered that it was not as straightforward (Pascanu et al., 2012). Standard RNN architectures suffer greatly as the sequence size, and therefore the memory requirements, grow in size. For instance, the most commonly encountered problems when trying to train vanilla RNNs were vanishing and exploding gradients (Bengio et al., 1994; Hochreiter, 1991). Both of these problems made it effectively hard to use the original RNNs in DL, as the number of parameters and complexity only exacerbate the issues.

The most prominent solution to these problems with RNNs was an architectural one, Long Short-Term Memory (LSTM), which changed the inner mechanism used to propagate and retain information (Hochreiter and Schmidhuber, 1997). Instead of trying to directly operate over the previous timestep state, LSTM allows the hidden state to be carried unmodified through all the passes, acting in essence as a residual connection would in a standard neural network. The network is entirely in charge of deciding which information should be carried as-is or if new information from the current step should be added. This is achieved through the use of three binary gates that mask the information. The forget gate is in charge of resetting information coming from previous steps, the input gate selects which new information must be incorporated, and finally the output gate can zero out any information. A seemingly inconspicuous change, adding the previous information instead of directly applying non-linear transformations, single-handedly solved the vanishing gradient problem.

LSTMs have been applied to a wide range of problems. Some of them, such as models and methods applied to CV, are closely related to this thesis. For instance, RNNs for video inputs excel at their task, as can be seen with architectures designed for action recognition (Muhammad et al., 2021) or autonomous driving (Gu et al., 2020). Of course, though, many other domains have seen benefits from LSTM, such as text-to-speech (Fan et al., 2014), and its close relative speech-to-text (Graves et al., 2013). Also worth mentioning, recurrence can be applied to single images by considering them as sequences of pixels, where each pixel in a row, and likewise column, depend on those preceding them. This phenomenon, especially for image generation, has been extensively explored in Row and Diagonal LSTMs by van den Oord et al. (2016). The authors also demonstrated with their PixelCNN architecture that recurrent behaviors could be emulated through the use of masked convolutions.

While LSTM makes RNNs able to work with longer sequences compared to its vanilla implementation, undeniably they come at a greater computational cost and added complexity. They have a larger number of operations, and each step in the process produces two outputs instead of one. Neural Machine Translation, which attempts to automatically translate from one language to another, succeeded in obtaining simpler alternatives with Gated Recurrent Unitss (GRUs) (Cho et al., 2014). This new unit managed to keep the number of operations at a lower, albeit similar, level, but most importantly simplified the architectural flow. GRU layers use the same, and unique, output as the hidden state that contains the historical information and as the input to the next layer. A review of the literature shows that LSTMs are generally more popular than GRUs, although there is no agreement on which is better. They are usually interchangeable, and using one or the other has a minor impact on the results (Cahuantzi et al., 2021; Khandelwal et al., 2016; Yang et al., 2020b; Mateus et al., 2021). As such, every and all applications that use LSTMs could potentially also employ GRUs. Figure 2.5 shows a visual representation of all three layers, the standard RNN, the LSTM, and the GRU. It shows the internal mechanisms and how RNNs have evolved from one architecture to the next.

RNNs were at first associated with processing sequences in their natural order. That is, sequences were input as-is and future information was not used. This approach is not only valid but necessary for fields such as simultaneous translation, real-time captioning, and



(a) RNN

(b) LSTM

(c) GRU

Figure 2.5: Illustration of the three main layers for RNNs: (a) shows the traditional architecture with a single operation inside, (b) the LSTM layer that introduces binary gates and the ability to carry information, and finally (c) represents the simpler GRU layer that has a single state instead of two.

any situation that requires immediately taking action based on past events. That leaves, however, much potential untapped in other fields. For instance, translation of whole documents, for which the original text is fully and readily available, can benefit from knowing the end of paragraphs before they are translated. In other words, processing backwards can contribute meaningful information, as demonstrated by Schuster and Paliwal (1997) with Bidirectional Recurrent Neural Network (BRNN). BRNNs stack two RNN layers, such that one receives the sequence in its natural order, and the other process it in backwards, or reversed, order. The same concept can be extrapolated to LSTM networks (Graves et al., 2005) and its GRU counterparts (Lu and Duan, 2017).

One problem that still plagues RNNs, irrespective of its implementation, is its undesired ability to overfit to the input data. The longer the sequences and the more complex the architecture, which usually comes with a larger number of parameters, the greater the probability of it overfitting. Several attempts have been made at tackling the problem, for instance by introducing dropout inside the recurrent layers (Zaremba et al., 2014), which randomly disables some of the internal connections between steps, or by merging RNN layers with CNNs (Shi et al., 2015), which should reduce the number of parameters while helping with generalization.

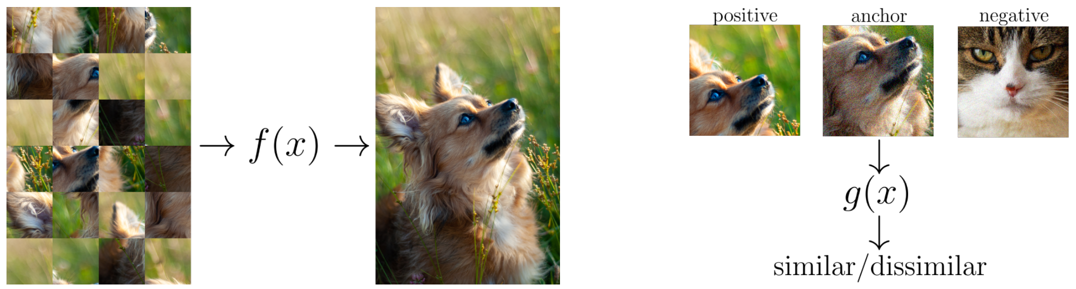
2.3.4 Self-supervised Learning

Often, research is focused on obtaining better performance in specific domains or problems, as has been seen with Imagenet for the case of CV. One such way to obtain better results is by creating more complex architectures, sometimes by introducing better data processing techniques, and often by throwing more computing power and training deeper and more complex networks. All of those are valid techniques—and have been covered in Section 2.2—but they leave out a different range of options unexplored. Not only that, but most often than not, they require enormous datasets with appropriately labeled data, which limits, even more, the applicable range for those algorithms.

As such, this section explores an alternative learning setting, SSL, that simultaneously tackles the lack of labeled data and serves to obtain better performance as a second step. Canonically, SSL is considered a variant of supervised learning, as it requires a supervisory signal to learn from the data (Liu et al., 2021b). The same way that ResNet networks trained on Imagenet have been used as a basis for other models not necessarily based on imagenet, SSL attempts to create a base model that can be finetuned for other tasks. This base model is not necessarily trained with the same source as the final and finetuned model and, most importantly, does not require that the labels accurately describe the underlying classes. SSL can be understood, thus, as a pre-train step that is designed to initialize an ANN with useful information. The same network, once pretrained, can be finetuned to perform a certain objective for a specific problem. As such, SSL serves to initialize the network parameters so that they can capture rich information about the data without necessarily performing a determinate task.

In other words, SSL proposes a paradigm change, from training a classifier by feeding it thousands if not millions of images, all accurately labeled, to first using a proxy that might not be labeled and then training for the original task but with fewer data constraints. One advantage of going in that direction is that the network has learned much more than simply the classification target. Perhaps it has learned to distinguish it from the background, to identify that an object and the same object but slightly blurrier are the same, that two perspectives of a single object are part of one, or perhaps that similar objects are conceptually closer than those that have different visual representations. This rich information deduced from the structure of the data can then be used to perform the original classification task, yet with additional robustness, as the network is now able to identify much more complex scenarios than it might have before. The secret lies in how to make the network learn any of those scenarios, for which several implementations can be found.

One of the first techniques used for SSL was a bottleneck-type architecture known as Autoencoders (Rumelhart et al., 2013; Kingma and Welling, 2014; Hinton et al., 2011). The basic working idea behind them is forcing an ANN to compress an input (such as an image) to a much smaller representation, and then having it reconstruct it. Arguably, this compressed information should be good and rich enough to enable obtaining faithful reconstructions of the input. However, the truth is that while the information is indeed compacted and compressed, it fails to capture specific and rich information of the images,



(a) Jigsaw

(b) Similarity

Figure 2.6: Comparison of two methods to obtain rich embeddings, (a) through input reconstruction and (b) by learning which images are similar. The first method trains an ANN $f(x)$ to output the order in which chunks of an image should be ordered to reconstruct the original picture. On the other hand, the similarity approach trains a network $g(x)$ to differentiate similar images (anchor and positive) from dissimilar images (anchor and negative).

instead tending to the mean representation (Bengio, 2009). Regrettably, that makes them unsuitable for SSL, as rich and specific information is required for the future finetuning step, as opposed to common or average information of a class. Variational autoencoders present a slight improvement by allowing the compressed information to be a distribution instead of discrete values (Kingma and Welling, 2014), which potentially extends its ability to capture more information than just the mean. Gatopoulos and Tomczak (2021) demonstrated with three different datasets that they are, indeed, a viable alternative for SSL.

Autoencoders pertain to a class of SSL named generative, as they learn from a single input and its task is to generate new data. Other methods that fall under this category are Generative Adversarial Networks (GANs) (Goodfellow et al., 2014a), as their dual discriminator-generator flow can be exploited to create a semi-supervised setting, on which one network learns the inherent structure while the other predicts the ground class (Odena, 2016).

While generative techniques rely on single input reconstruction or generation, on the other side of the spectrum lies contrastive learning (Falcon and Cho, 2020). The focus is lifted from a single-input and, instead, an input is contrasted against a modified version of itself. In other words, focusing on CV models, an image can be altered to produce two slightly different versions of it. One could be a cropped section, while the other could be a blurred or color-altered representation. The network would be trained with the objective of learning that these two representations are alike, as they represent the same image, whereas

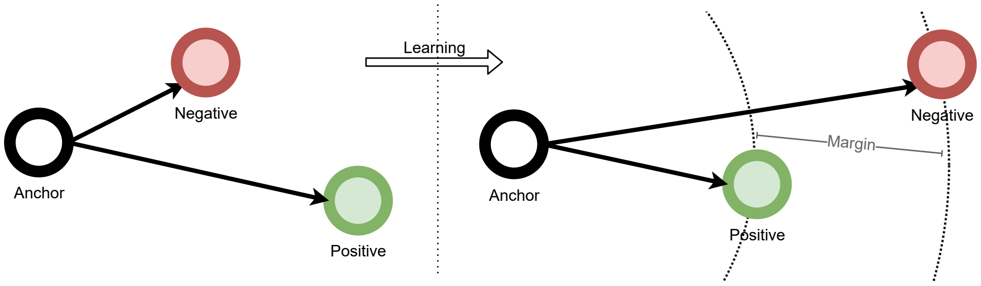


Figure 2.7: Schematic representation showing a triplet formed by three samples, the anchor, the positive sample of the same class, and the negative sample of any other class. Before training, the distances are not correlated to the classes, yet after the training process the TL has pushed away the negative and brought the positive closer.

they are dissimilar to any other image. Other alternatives to produce versions of the same image besides alternations exist, such as reordering chunks of an image, as done by Misra et al. (2016). For instance, reordering can be easily applied to other fields other than CV, such as audio (van den Oord et al., 2018). Figure 2.6 shows two examples of SSL, through reordering of chunks as explained above, and through metric-learning for similarity. Both methods would produce rich embeddings that enable finetuning for downstream tasks.

Recently, SimCLR was proposed as a SSL framework in a study that examined the use of different DA techniques and their combination for contrastive learning purposes (Chen et al., 2020a,b). Their contributions to the SSL field were two-fold. First, their detailed analyses serve to outline the best choices for future work in SSL. Further, their method is not limited to a study of image transformations but also performed an ablation study of contrastive losses to give insight on possible improvements. Finally, they proposed a novel architectural choice by which the embeddings used for the finetune stage were not the same as the final layers in the pretrain stage. That was achieved by introducing a series of projection layers during the SSL phase, which are then removed in the pretain stage. Their experiments showed that projection improved results in the final finetuning stage, while not being detrimental during SSL.

Contrastive SSL has been applied using a wide range of settings, all with the same purpose of creating a rich base. For instance, it has been used in videos by predicting the frames' order (Misra et al., 2016; Xu et al., 2019; Lee et al., 2017a), which forces the network to learn useful features to restore the correct order of the sequence. Object tracking has also been used as a proxy-task by either using motion maps, pairs of frames, or a set of explicit features such as pose, semantic and identity information (Pathak et al., 2017;

Wang and Gupta, 2015; Wang et al., 2019). A famously used loss function for contrastive learning is Triplet Loss (TL), which groups embeddings of images based on those images' similarity (Dong and Shen, 2018). It does so by creating triplets formed by two samples of the same class and one of another, forcing the similar samples together while pushing the negative afar until it reaches a minimum margin. Figure 2.7 shows an example based on one triplet, showing the desired effect after training is done, while Figure 2.6b showed a real-world triplet. Some publications have explored and developed innovative loss functions based on TL, such as learning sample and video invariants (Tschannen et al., 2015) to obtain better distance-based embeddings, and using multiple TL alongside regressions to take into account several views of a single action at once (Sermanet et al., 2018).

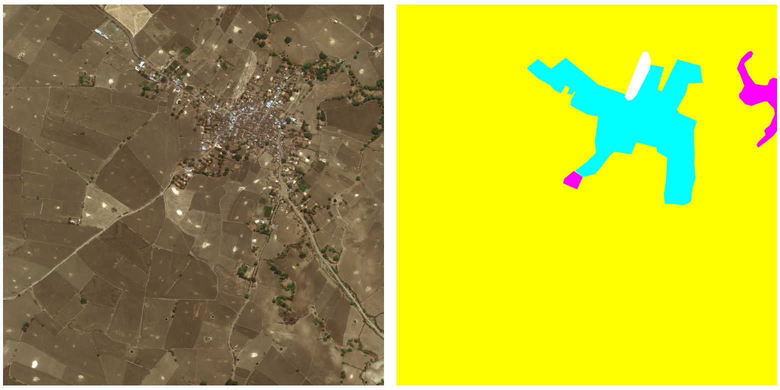
Lastly, SSL and specifically contrastive losses, have been demonstrated to be a good weight initialization process, improving the results obtained during finetuning (Sudowe and Leibe, 2016). It has been shown that these properties can also be leveraged for transfer learning, where the domain used during SSL is slightly different from the target one (Yang et al., 2020c; Spathis et al., 2020). Not only that, but research shows that it can also be used for domain adaptation, where simultaneous detection of two datasets under different shifts can be challenging for standard ANNs (Achituve et al., 2020; Saito et al., 2020).

2.4 Domain Specific Implementations

After having covered fundamental research and the basis for this thesis, including state-of-the-art research, this section covers the specific fields in which the following chapters are focused. As such, it starts by detailing the field of satellite imagery, delving into both land cover segmentation and road extraction, and, finally, ends by examining the medical field of WCE.

2.4.1 Satellite Images

Both subdomains explored in this subsection draw from a traditional CV setting, where the input data comes in the form of images. Specifically, images in this domain have their origin, as the name suggests, in satellites and are obtained at a high resolution, usually starting at two thousand pixels per side. For the purposes of this research, this section uses aerial images interchangeably with satellite ones. The main difference between aerial and satellite images is rooted in the height at which the images are taken, with satellite being much further from Earth and aerial obtaining finer details. Usually, aerial images can



(a) Satellite image (b) Segmentation mask

Figure 2.8: Example of a satellite image (a) and its corresponding land cover classification (b). Each pixel of (a) is classified according to its type and a given set of classes. For instance, here yellow indicates agriculture, aqua is urban land, fuchsia rangeland, and white is barren.

be augmented with other information, such as infrared channels, slopes, and orientation of surfaces to name a few. However, none of this additional information is utilized throughout this thesis. Overall, irrespective of their source, these images depict the Earth as seen from space at different zoom and detail levels. Commonly, both cases suffer from artifacts and occluded images due to clouds or other obstacles.

Land Cover Segmentation

Land cover segmentation is a semantic classification problem where each pixel must be attributed to its underlying type, ranging from big bodies such as water clusters, woods, plains, or civilization, to more fine-grained classes such as river, industrial zones, and beach, to name a few, as shown in Figure 2.8. Often, land cover segmentation suffers from a lack of labeled data. Obtaining satellite images is easier than aerial ones, as the process is automated and can be performed several times per day from several locations, but either of both cases is just a matter of capturing the Earth's surface. However, labeling those images is an entirely different problem, which is particularly aggravated the more classes and more detailed they are. For instance, Institut Cartogràfic i Geològic de Catalunya (2022) provides a manual land cover classification of the Catalan territory only every nine years. For an automated labeling process to reach the level of detail and correctness of a detailed manual process, such as the one above that spans nine years, would be a difficult—if not impossible—task. On the other hand, creating a coarse classification could be provided much more often if automatizing was considered, and the end result would still be good enough for most applications.

Inevitably, land cover segmentation draws most of its ideas from Section 2.3.2, and have followed the similar trend of CNNs' increased popularity with the publication of Long et al. (2015), and residual architectures like U-Net (Weng et al., 2015), PSPNet (Zhao et al., 2016), or newer iterations like RefineNet (Lin et al., 2016a). Both U-Net and RefineNet work with an encoder-decoder layout, but the latter introduces a late fusion mechanism of high-resolution feature maps with lower-resolution ones.

While land cover segmentation has a wide range of datasets available, it is undeniable that the Vaihingen dataset (ISPRS) has seen the most traction and usage in DL (Long et al., 2020, 2022). Even though its success, the dataset is composed of only 33 images of $2,500 \times 2,500$ pixels at a resolution of 0.09 m per pixel. Vaihingen is a clear example of the problem outlined above, that obtaining labeled satellite data is a hard and expensive process. Yet, even if it does not have an abundant number of images, it has been available since 2012 and has been extensively used as a benchmark and gold-standard for satellite image segmentation.

Vaihingen, along with other private datasets, has been used by several authors to solve semantic segmentation with high-resolution images (Sherrah, 2016; Wang et al., 2017; Nogueira et al., 2019; Nguyen et al., 2017). Sherrah (2016) proposed the use of FCNs without any downsampling, which avoids the use of deconvolutions or lossy upsamples and stops any details from being lost. Coupled with a pretrain stage with remote sensing data, they demonstrated that a no-downsampling FCN with CRF can achieve state-of-the-art results. Nogueira et al. (2019), on the other hand, employed downsampling and upsampling in a standard convolutional branch, along with dilated convolutions in parallel to perform overlapping patch-based classification. On the other side of the spectrum, Nguyen et al. (2017) used an encoder-decoder architecture with skip connections to carry high-resolution information. Similarly, Wang et al. (2017) took a ResNet-50 as the backbone network and added a decoding network with skip connections. Unlike all previous works, however, they did not simply add the high-resolution information as is. Their skip connections were masked with a gating mechanism derived through convolutions, so that the network should be able to learn which information needs to be added and which one can just be discarded.

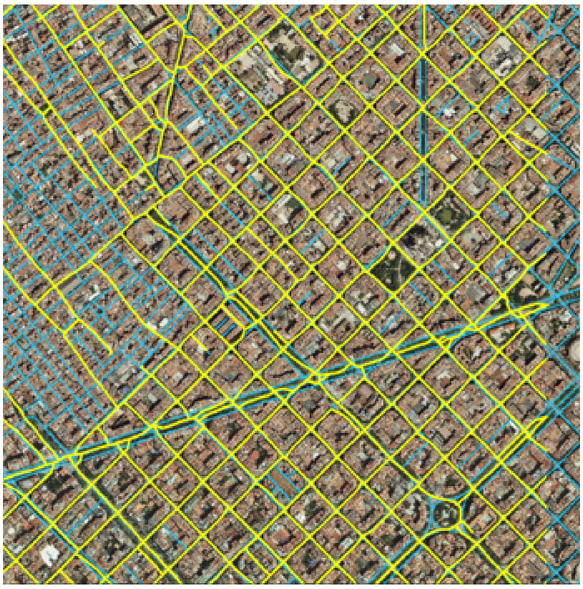


Figure 2.9: Representation of all roads visible in Barcelona from a Satellite image. Outlined in yellow are all the roads that a model trained to extract roads via graphs has been able to detect. The lack of yellow overlapping the blue ones indicates that the network has failed to detect valid roads.

Road extraction

The first publications in road extraction worked by defining a set of rules that defined what roads and junctions visually looked like and attempted to find them based on those features (Bajcsy and Tavakoli, 1976). Subsequent publications substituted these hardcoded and inflexible rules for hand-crafted features based on edges, shapes, and textures to detect these same elements (Mayer et al., 1997; Trinder and Wang, 1998; Treash and Amaratunga, 2000). Ultimately, foreground discrimination methods based on energy minimization, like snakes, were coupled with these feature-based mechanisms to obtain an even more flexible and powerful system (Mayer et al., 1997; Laptev et al., 2003)

With the advent of CNNs and FCNs, the most exploited approach to road extraction became semantic segmentation. An aerial image is classified according to a binary class, whether the pixel is a road or not (Mnih and Hinton, 2010; Costea and Leordeanu, 2016; Kaiser et al., 2017; Azimi et al., 2018; Aich et al., 2018). All of these architectures typically follow the same schemes described in Section 2.3.2, where encoder-decoder pairs are used in conjunction to skip connections to obtain the final road segmentation. Optionally, and usually done, the resulting image can be transformed to a graph of roads on a postprocessing step via superpixel creation with CRF (Wegner et al., 2015). Other means to create these superpixels have also been studied, such as thresholding and morphological thinning until a desired width (Cheng et al., 2017).

Segmentation-based methods can, as shown above, output a complete graph with roads and junctions. Yet, as evidenced by Mattyus et al. (2017), the resulting graphs are usually noisy due to the post-processing techniques. Several cleaning steps and heuristics can be used to aid in the process, but the graph can still be over-connected or accidentally remove connections between segments. To solve any issue that might arise from the conversion step, Bastani et al. (2018a), the creators of RoadTracer, proposed a network that directly outputs a graph instead of doing semantic segmentation. To such ends, it iteratively queries a network what action it would take given an aerial image. The network outputs which direction to go from that image, if any, and a graph exploration algorithm calculates the next image. An example of what such a network looks like is shown in Figure 2.9. The idea of going directly from image to graph, especially through iteration, quickly caught on and several more publications have exploited it successfully (Li et al., 2018c; Ventura et al., 2018; He et al., 2019, 2020, 2018; Bastani et al., 2018b).

2.4.2 Wireless Capsule Endoscopy

Traditional endoscopy is an obstructive intervention that causes great discomfort to the patients. Seeking to improve the experience, WCE only requires the patient to swallow a pill-shaped camera, which then travels through the person's digestive tract while recording everything and sending it to an external device, as shown in Figure 2.10. The discomfort is, thus, minimal, but comes at a price. The procedure often results in 12h or longer videos, which must then be reviewed in search of any anomaly. Manually checking each frame of the whole procedure is unfeasible for physicians, so Computer-Aided Detection and Diagnosis (CAD) systems have been created to ease the task. CADs are automatized systems that aid in the process by providing a second opinion which the physician can use to make the diagnosis (Suzuki, 2012).

CAD can be broken into two slightly overlapping categories, (1) Computer-Aided Detection (CADe), which is focused on locating lesions in medical images, and (2) Computer-Aided Diagnosis (CADx), which, aside from localizing, also characterizes the lesion. Paraphrasing Firmino et al. (2016), a CADx system geared towards polyps would distinguish between benign and malignant tumors besides locating them in a video. As such, CADx can be built from CADe systems—by using their outputs as a second stage model—or can be created as an end-to-end system. Here, the focus is on end-to-end CADx systems.

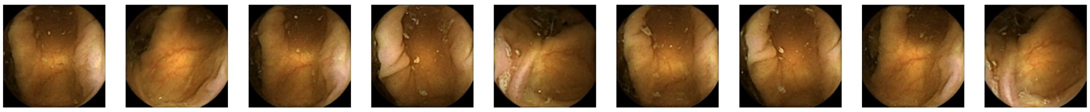


Figure 2.10: Series of nine images representing a continuous sequence from a WCE video recorded by a pillcam device. The images have been obtained after being sent to an external device and then recovered and processed.

One type of CADx system comes in the form of video summarization (Gilabert et al., 2022). Traditional WCE pillcams move at a slow rate and usually capture at a rate of 2 to 4 frames per second (Fernandez-Urien et al., 2014; Xavier et al., 2018), which means that they often have several segments where the images barely change from one to another. The interest in these close and look-alike images, medical or otherwise, is understandably non-existent, as they do not contribute towards anything. Also, modern capsules can capture at a rate of up to 35 images per second while in motion, reducing the risk of missing lesions, but they increment the number of similar images (Figueiredo et al., 2011). A good summarization system should point to a single frame of interest in these sequences and skip the rest, so that any pathology could still be detected while dramatically reducing the overall length to be reviewed.

CADx also comes in the form of automatic pathology detection. These systems can either pinpoint frames containing a single pathology of interest, or can do simultaneous detection of multiple pathologies. Such models were first developed with traditional means, such as superpixel creation coupled with SVMs (Fu et al., 2014), SVM with color invariants (Lv et al., 2011) or saliency maps (Yuan and Meng, 2015), or hand-crafted textures with classification and logic trees (Pogorelov et al., 2019). Polyp detection has been tackled through subdivision with SVMs (Alexandre et al., 2007), ulcer detection can be done through texture and color invariants (Yeh et al., 2014), and motility events can be identified through pattern recognition, color decomposition, and chromatic stability (Malagelada et al., 2008).

More recently, CNNs have been used in Seguí et al. (2016) to automatically find the best textures for motility events classification, instead of relying on hand-crafted features. Multiple other applications in WCE have also explored the use of CNN architectures, such as polyp detection (Iakovidis et al., 2018; Aoki et al., 2019; Nadimi et al., 2020), ulcer detection (V and Prashanth, 2020), bleeding (Caroppo et al., 2021; Khan et al., 2020), and celiac disease diagnosis (Wang et al., 2020). Overall, WCE saw a great increase in performance through CNNs, as is the case with most CV-related datasets.

With the modern networks outlined in Section 2.2, such as ResNet, more models were created to work with WCE data (Laiz et al., 2020; Jain et al., 2021; Yuan et al., 2020; Kundu and Fattah, 2019; Jain et al., 2020; Guo et al., 2022). Metric learning through TL was explored by Laiz et al. (2020), attention mechanisms to discriminate among features by Jain et al. (2021), and even more complex and deeper models such as DenseNet in Yuan et al. (2020). CADx has particularly enjoyed a higher models' production rate, as seen in bleeding detection, vascular lesions, ulcers, polyp, and tumors (Trasolini and Byrne, 2021; Attallah and Sharkas, 2021; Gilabert et al., 2022).

As formalized in the works of Yuan et al. (2020); Akay and Hess (2019), WCE related tasks suffer from a series of problems. First of all, a lack of labeled data, as the process is extremely time-involved. Moreover, there are specific domains such as polyp detection that also suffer from highly imbalanced classes (Laiz et al., 2020). As such, these works deduce that it is hard to produce models that do not overfit as a result of the imbalances, high inter-class variance, and high intra-class variance. Dropout layers, L1 or L2 regularization, and sampling mechanisms have been partially successful at coping with these issues (Kim and Lim, 2021). Other means such as TL to force better and more balanced embeddings have also been tried (Laiz et al., 2020).

In relation to Section 2.3.4, several works have employed semi-supervised or self-supervised learning for the medical field (Cheplygina et al., 2019; Azizi et al., 2021). Simulated post-operative MRI images have been successfully created with generative networks to augment the training phase (Pérez-García et al., 2021). Multi-organ segmentation alongside with pneumonia detection have also benefited from patch-reordering SSL (Navarro et al., 2021). WCE tasks such as detection of inflammatory and vascular lesions have also seen an increase in performance thanks to SSL (Vats et al., 2021). Unlabeled data has also been leveraged in the works of Guo and Yuan (2020) to detect several pathologies in WCE videos.

Chapter 3

Uncertainty for Land Cover Segmentation

Contents

This chapter presents a method to improve segmentation networks with uncertainty information, specifically for the case of satellite images with the objective of land cover classification. As has been reviewed in previous chapters, most segmentation approaches do not give any importance to the inherent uncertainty in the data or prediction's confidence during the process, which makes it harder to correctly interpret the results. Our method implicitly uses uncertainty to refine the segmentation process, incorporating details and new information where the network is uncertain of its work.

Uncertainty, as laid out in the following subsections, simultaneously tackles two of the problems this thesis aims to solve. First, it can be used to provide robust results, as the network learns insight into its own decision process and refines the segmentation output. This is achieved through automatically crafted per-pixel uncertainty at several levels of the process. Second, a final uncertainty heatmap is used to give a confidence value of the output classification.

This chapter is organized as follows. First, in Section 3.1 we present an overview of the method. Section 3.2 outlines the dataset used for this particular problem. Subsequently, Section 3.3 showcases the implementation details, such as hyper-parameters and training configurations, used during the training process. The results obtained are discussed in Section 3.4, and the conclusions are discussed in Section 3.5.

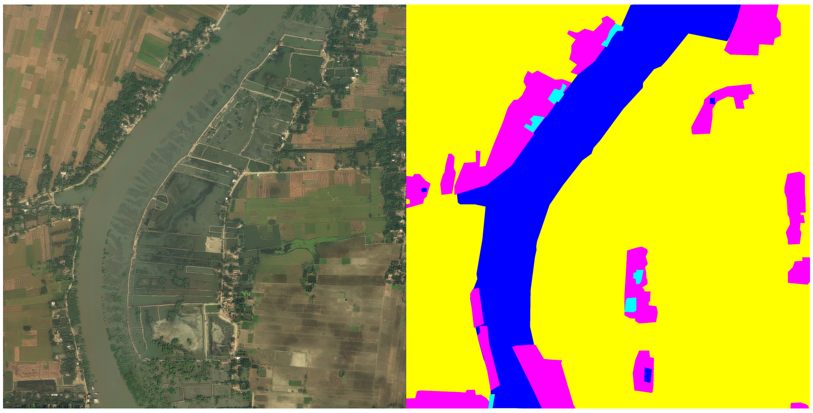
3.1 Approach

Image segmentation, if reduced to its minimal expression, is a classification task done at the pixel level. That is, each pixel of an image is assigned a class, and the model task is to predict that class given the image's features and its surroundings. Clearly, pixels that are close together in the image have a high probability of belonging to the same class, as objects have a natural gradient of features, such as color, instead of abrupt changes. Clearly, looking at a pixel as an isolated element is not enough, the algorithm must look for similarities and boundaries that separate it from other pixels.

Trying to distinguish pixel classes at a higher dimensionality is arguably harder than at lower dimensions, where pixels have been merged during a downsampling process. Of course, downsampling is a non-conservative process, during which information is inevitably lost. Lowering the size of an image, segmenting, and then upsampling the result is usually not enough, as too much information and classes would be lost in the process. Instead, our approach is inspired by U-Net, as presented by Weng et al. (2015), which downsamples the input images' features n times, produces an initial segmentation, and then iteratively upsamples it another n times, incorporating new information at each step.

This process, however, is still flawed, as each step done during the upsampling process demands the algorithm to complete redo the segmentation output from scratch. These upsampling steps require adding new information independently of how good the previous segmentation step already is. For instance, images containing big bodies of water do not need to iterate and refine the water segmentation, as it clearly will not change no matter how many times it is upsampled. For instance, Wang et al. (2017) proposed a Gated CNN in which each step of the upsampling process decides which pixels need updating and which not. In a sense, they propose the use of attention in the upsampling mechanism. Their solution, however, relies on the network finding the optimal pixels to update, which might not always be the case.

Instead of letting the network figure out where to spend its attention, we propose the use of uncertainty as the attention mechanism. Places where the network is unsure of what class



(a) Satellite image (b) Ground truth segmentation

Figure 3.1: Example of a satellite image and its corresponding semantic segmentation. Here, blue denotes water, aqua urbanization, yellow agriculture, and fuchsia rangeland.

should be given to a pixel should be revisited and refined, as they are likely sources of conflict and confusion. Using uncertainty, the responsibility and complexity of finding those pixels is taken out of the model, which is now provided with a guided mechanism. Additionally, this information can be output to the end-users, supplying a source of confidence in the results and empowering them to make decisions based on objective criteria.

Our approach attempts to model the heteroscedastic variant of aleatoric uncertainty, the sample-dependant noise, by introducing stochastic noise in the pixel-classification process and trying to isolate it from the ground truth. Throughout the implementation we follow the framework proposed by Kendall and Gal (2017). Uncertainty is derived through Monte Carlo sampling at each upsampling step instead of only at the output level, ensuring that each refining is accurate and reliable. To that end, our method relies on deep supervision as a mechanism to produce better segmentation. Each step of the upsampling process is used in the loss calculation, which makes the network simultaneously tune all its downsampled results and the uncertainty.

3.2 Dataset

The database used for this method was obtained during the DeepGlobe 2018 Challenge (Demir et al., 2018), and consists of a compilation of images with three different purposes in mind: road extraction, building detection, and land cover classification. For our particular

Table 3.1: Class representation in the DeepGlobe dataset. Each class is reported as the total count of pixels, where M means Million, and its overall percentage.

Class	Pixel Count	Proportion
Urban	642.4 M	9.35%
Agriculture	3898.0 M	56.76%
Rangeland	701.1 M	10.21%
Forest	944.4 M	13.75%
Water	256.9 M	3.74%
Barren	421.8 M	6.14%
Unknown	3.0 M	0.04%

use case, only the land cover set was employed. The data consists of 1,146 satellite images with a resolution of 2448 × 2448 pixels. Images come in RGB format with no additional information. The authors already propose in their publication training, validation, and test splits, each with 803, 171, and 172 images, respectively.

The segmentation task divides the available per-pixel classes into seven unique categories: 1) urban land, 2) agriculture land, 3) rangeland, 4) forest, 5) water (rivers, ocean, lakes, wetland, ponds), 6) barren land (mountain, land, rock, desert, beach, no vegetation), and 7) unknown (clouds and other artifacts). While most categories identify a single type of land, it can be seen that there are others, such as water and barren land, that serve as a grouping. In such cases where multiple types are found in a single class, high intra-class variance can be expected due to the varying representations they can have. Likewise, the unknown class contains all unexpected and undesired events, which again is a source of high uncertainty. Not only that, but as Table 3.1 shows, the per-pixel representation of each class is highly unbalanced, with more than 50% of them being agriculture. For instance, Figure 3.1 shows a sample from the DeepGlobe database with this exact problem. As can be seen, the river in Figure 3.1a is classified as the same class as the pond in its right, as evidenced in the provided ground-truth, Figure 3.1b. Classes like urban, shown in aqua, or small ponds (of which three are identifiable in the segmented mask) can be challenging to detect due to their small size. The smallest size considered is an area of roughly $20m \times 20m$.

3.3 Implementation

Given that the satellite images come in high-resolution, and are intractable at that resolution with the current GPUs, we first downsize them to a more realistic size. Through bilinear

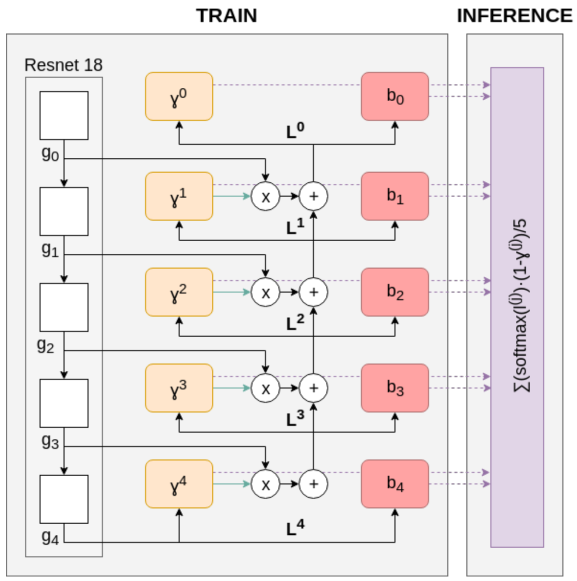


Figure 3.2: Proposed architecture with deep supervision via uncertainty attention mechanisms. Black arrows represent weighted connections between different layers. Green arrows represent forward-only weighted connections, where gradient flows in the backpropagation process are not allowed

interpolation, they are resized to 1024 pixels. Further, during the training process, 8 random crops of 256 × 256 pixels are obtained for each sample, so that the final input size is smaller. Standard DA techniques are used to cope with the low amount of data. Specifically, random rotation and flips are performed, gaussian noise is added, and random adjustments to the hue, contrast, and brightness are applied.

Once processed, an input image is first forwarded through our backbone network, a ResNet 18 model (He et al., 2016) pretrained with the Imagenet database. Instead of only taking the last output, before the global pooling, the outputs after each of the residual blocks, g_i, are kept (Figure 3.2). As such, during this first phase five representations are obtained, each at half the size of the previous one but doubling the number of filters used.

The method proceeds as follows. Starting at g_4, define b_4 as the parameter-preserving upscaled features that match the size of the previous g_3 block. In other words, b_4 is twice the size of g_4 with half the channels, as the only way to preserve the number of parameters while doubling the size is by halving the amount. Then, for every $i > 0$, the algorithm computes the logits l_i and uncertainty bound σ_i from the upsampled block b_i, as shown in Equation (3.1). Here, the operator $\circledast_C$ denotes the convolution operator with as many filters as the number of classes C and each consisting of a different 1×1 pixels kernel, as denoted by $W^{i,j}_{1\times1}$.

$$\begin{aligned} l_i &= b_i \circledast_C W^{(i,1)}_{1\times1} \\ \sigma_i &= b_i \circledast_C W^{(i,2)}_{1\times1} \end{aligned} \tag{3.1}$$

To compute a usable uncertainty value, $t \in [0, T)$ random samples, where T is a predefined constant, are drawn from a normal distribution by taking the logits l_i as the mean and the uncertainty bound σ_i as the standard deviation, Equation (3.2).

$$\hat{l}_{i,t} \sim \mathcal{N}(l_i, \sigma_i)\,. \tag{3.2}$$

The final uncertainty value γ_i at step i, which is used in the attention mechanism, is derived in Equation (3.3), where $\hat{l}_{i,t,c'}$ indicates the t-sampled logit from class c', and $\hat{l}_{i,t,c}$ is the logit vector of the winner class for each pixel and sample. This step is basically a Monte Carlo sampling of the T random samples.

$$\gamma_i = -log\frac{1}{T}\sum_t^T \exp(\hat{l}_{i,t,c} - \log\sum_{c'} \exp \hat{l}_{i,t,c'}) \tag{3.3}$$

Finally, the input to the next iteration and where the attention mechanism takes place is shown in Equation (3.4). As can be seen, the input from the current block is always carried to the next iteration by directly adding it, effectively treating it as a residual connection. The new information g_{i-1} coming from the upsampled block is masked through an element-wise multiplication (*), which creates the selective attention mechanism through uncertainty.

$$b_{i-1} = \gamma_i * g_{i-1} + b_i \tag{3.4}$$

It is important to remark that at this point, in Equation (3.4), the element-wise multiplication is modified to only allow the forward operation with respect to the uncertainty γ_i, but not the backward one. In other words, any gradient coming from the backpropagation algorithm is ignored, leaving γ_i unmodified. If gradients from upper layers were allowed to modify the uncertainty value, and by extension the bound σ_i, then the value obtained would no longer be purely computed from aleatoric uncertainty.

Further, to enable deep supervision and encourage the network to learn meaningful uncertainty values, a deep supervision loss L_x is introduced as shown in Equation (3.5). L_x simultaneously estimates appropriate values for the logits l_i while computing the uncertainty bound σ_i. Last, the final segmentation can be obtained as the softmax of the γ-weighted sum of all the intermediate probabilities, Equation (3.6). As the sum of γ_i is not bound, an average is computed as the final result by dividing by the number of refining blocks (5).

$$L_x = \sum_i \gamma_i \tag{3.5}$$

$$\frac{1}{5}\sum_{i=0}^{4} \text{softmax}(l_i * (1 - \gamma_i)) \tag{3.6}$$

Similarly, a final uncertainty output can be built from the aggregated uncertainty at all segmentation levels, Equation (3.7). Likewise, the average value is computed by dividing by the number of refining blocks.

$$\frac{1}{5}\sum_{i=0}^{4} \gamma_i \tag{3.7}$$

The L_x losses are optimized with WNAdam, an optimizer that at its core uses the same momentum strategy as Adam and, additionally, attempts to automatically find an appropriate initial learning rate by normalizing the weights of the network (Wu et al., 2018). A piecewise learning rate strategy is employed, diving the learning rate by 10 every 33 epochs, and for a total of 100 epochs.

3.4 Results

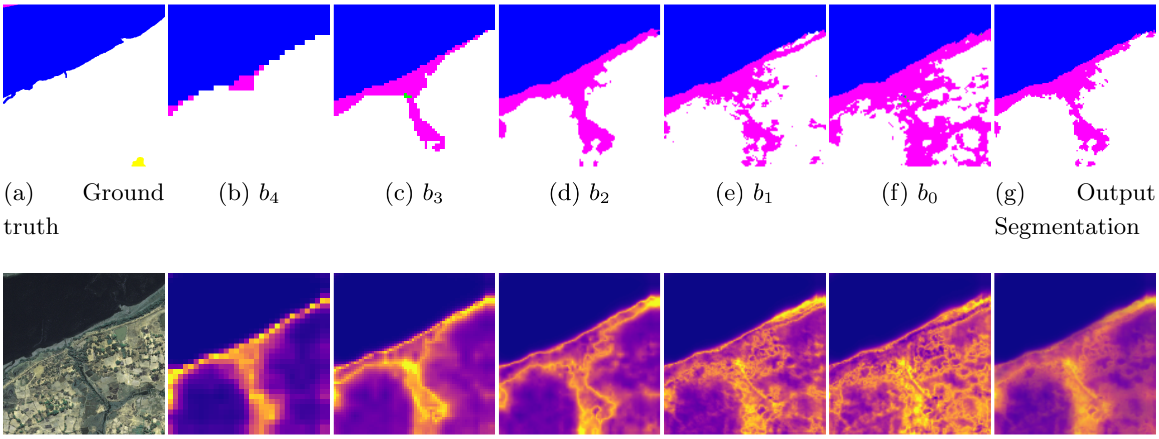
To evaluate the results the authors of DeepGlobe propose the use of Intersection over Union (IoU) averaged across all seven classes. As the name of the metric suggests, it evaluates the size of the overlap (intersection) between the prediction and the ground truth over the size of the union of both. In a sense, IoU measures how well the prediction matches the ground-truth, penalizing if it underestimates or overestimates the region. Another valid definition of IoU is in terms of the confusion matrix, where the intersection is equivalent to the true positive count, and the union is the sum of true positives, false positives and false negatives.

Table 3.2: Reported mean IoU (denoted as mIoU) in the DeepGlobe challenge, ordered as presented in the challenge leaderboard which uses the private test set. Note that models marked with * have not provided their final mIoU with the private set, thus only their public set results are shown.

Position	Model	mIoU
1	(Kuo et al., 2018)	0.5272
2	(Tian et al., 2018)	0.5224
3	(Seferbekov et al., 2018)	0.493
4	(Rakhlin et al., 2018)*	0.494
5	(Davydow and Nikolenko, 2018)*	0.4764
6	Ours	0.485
7	(Ghosh et al., 2018)*	0.507
8	(Samy et al., 2018)	0.428

Note that IoU, more commonly known as Jaccard Index, is a binary statistic designed to be used in two-classes scenarios. The approximation in the challenge consisted in computing the mean IoU, which bypasses the binary constraint but generates new problems. In fact, averaging worsens the results when taking into account the least balanced classes. Probably, the unknown class was taken out of the metric calculation at evaluation time to cope with the problem. For instance, considering the 7 classes this unknown class which only accounts for 0.04% of the data, would have been responsible for a 1/7th (14.29%) of the final score. Nevertheless, when considering only the remaining 6 classes, other minority classes like water, still represented higher importance (1/6 = 16.17%) than its pixel-count proportion (3.74%). Our model, evaluated with the evaluation set provided by DeepGlobe authors, obtained a 46.66% in the public test set, while the final test set, which was private until the challenge ended, gave a 48.50%. To avoid exploiting the metric, DeepGlobe only provided the mean IoU value, not giving the detailed per-class IoU.

Evaluation and inference was made with images at full size, 2048 × 2048 pixels, on an NVIDIA Titan X. The model can run in real time when deployed in production, taking an average of 250ms to produce the final segmentation with the uncertainty heatmaps. Figure 3.4 shows three outputs produced by the proposed method, selected to showcase cases where the uncertainty is useful to detect errors in the segmentation output. It can be observed in the first two rows that in places where the uncertainty is high, the pixels were misclassified. The last row showcases an example where the input image contains extreme variation and a high level of detail. These details were lost during the downsampling process,



(a) Ground truth (b) b_4 (c) b_3 (d) b_2 (e) b_1 (f) b_0 (g) Output Segmentation

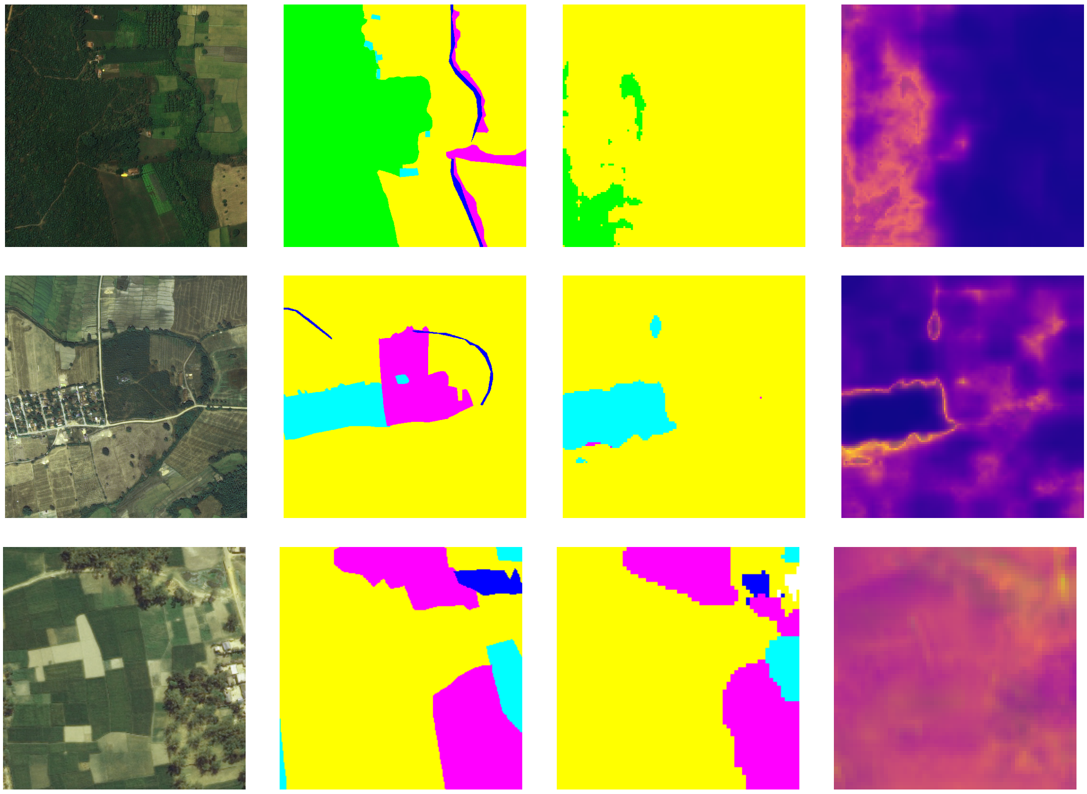
(h) Input image (i) γ_4 (j) γ_3 (k) γ_2 (l) γ_1 (m) γ_0 (n) Output Uncertainty

Figure 3.3: Upsampling process of a single segmentation task, where the uncertainty maps are used to incorporate new details at each step of the process. Subfigure (a) shows the ground truth segmentation, (b-f) the five internal downsampling blocks, (g) the network's segmentation output, (h) the input satellite image, (i-m) the internal uncertainty heatmaps for every downsampling block, and (n) the final uncertainty given to the user. Heatmaps use blue shades to indicate low uncertainty and bright yellow to indicate high uncertainty.

which prompted the network to assign high uncertainty even if the final output was correct.

Table 3.2 shows the challenge's final leaderboard as published after evaluation on the private set. Notably, our model scored sixth with barely any difference from the fifth to third places. Additionally, our model does not perform any additional cleaning or postprocessing step, unlike the works of Kuo et al. (2018); Rakhlin et al. (2018); Davydow and Nikolenko (2018); Ghosh et al. (2018), or use additional data like Tian et al. (2018).

As the challenge only measures the performance of the segmentation process, and not the uncertainty output, it is important to perform a qualitative analysis of the uncertainty heatmaps. As can be seen in Figure 3.3, the model has learned that the most common sources of uncertainty are boundaries in the input image. That is, places where the colors or textures severely change are prone to be of different classes. Thus, finding the exact place where one class ends and the other starts is uncertain and, probably, up to subjective interpretation. The upscaling process showed that the network progressively adds new features at these exact places while not modifying the rest, making good use of the proposed attention mechanism. As more details are added, the uncertainty does not necessarily



(a) Satellite (b) Ground Truth (c) Segmentation output (d) Uncertainty

Figure 3.4: Several examples of segmentation images obtained by the proposed model, including the uncertainty output. Column (a) shows the input image, column (b) the segmentation, and (c) the final uncertainty heatmap. Heatmaps use blue shades to indicate low uncertainty and bright yellow to indicate high uncertainty. Both the first and second rows are examples where the uncertainty output could be used to identify errors in the proposed segmentation, as they mark precisely the conflicting zones. The last row shows an example where, even if the output is correct, the uncertainty still shows a high value due to the complexity of the satellite image.

need to decrease everywhere, as the boundary between classes gets thinner and even more subjective.

Finally, the last uncertainty heatmap gives hints to the user with regards to the confidence with the segmentation task. In fact, it can be seen in Figure 3.3 that the miss-classified forest is clearly highlighted in the heatmap. Whether that miss-classification is due to a mislabel or an error produced by the model, is a question that could be easily answered with the additional information that uncertainty gives.

3.5 Conclusions

The method proposed in this working package leverages the use of uncertainty to (a) provide confidence in the output by supplying the end user with interpretable uncertainty heatmaps, and (b) iteratively refine land cover segmentation results by using uncertainty as the gating function in an attention mechanism.

We demonstrate that incorporating uncertainty in the upsampling process of segmentation tasks helps in obtaining more robust results. Particularly, when high imbalances exist or inaccuracy in labeling is expected, uncertainty does not only achieve state-of-the-art results, but also provides interpretable information to understand where it fails. The qualitative analysis shows that the addition of uncertainty heatmaps can be beneficial in deployed models, helping understand the model's outputs and decision process.

Finally, we conclude that the addition of uncertainty does help in providing confidence in the decision process and interpreting the results. Thus, the first working package, which centered around these two problems in DL, is addressed and proved to be solvable. The technique presented in this chapter can be extrapolated to any other databases and tasks, making uncertainty widely available in many more settings.

Future work could be centered around providing uncertainty heatmaps for image classification problems, where the output dimension is uni-dimensional instead of an image of the same resolution. One such way to do so could be by leveraging the techniques used in CAD interpretability techniques.

This work was published in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (Pascual et al., 2018).

Chapter 4

Road Extraction with Recurrent Neural Networks

Contents

In this chapter, we present a method to augment the contextual information used during road extraction in satellite images. Roads can be decomposed into sequences of movements and iteratively explored by a combination of deep learning and graph exploration, as did Bastani et al. (2018b), the authors of RoadTracer. We explore how to exploit the information about the trajectory that is inherent to all sequences of movements. For example, the final model should be able to tell if an agent is below a bridge or following based on its previous movements.

Section 4.1 argues that the additional data, which was previously manually input to the model, should help obtain more robust models. Then, Section 4.2 proposes an implementation of the proposed method, and Section 4.3 shows the results. Finally, conclusions are discussed in Section 4.4.

4.1 Approach

At a glance, our method works with sequences of satellite images. Particularly, a scheme is developed so that given an image where its centermost pixel is a road, the next set of plausible positions are deduced. This set of points is computed from any valid direction that results in an image still centered on a road. Iteratively, the process takes the next centered image and detects where the road leads to. An algorithm that explores the whole set of roads, capable of exploring junctions and exhausting all possible paths, can be implemented as a recurrent method, using backtracking to go back to unexplored paths when the current one ends.

The method so far requires that roads can be decomposed into a set of points. Instead of having long segments that connect two far-away junctions, our implementation breaks them into smaller segments. Given a determined distance D, a long segment between two points is decomposed into small segments of D length each. For example, the original segment $\big(n_{(x,y)}, n_{(x+p_x,x+p_y)}\big)$ is decomposed into a sequence of points $\big(n_{(x,y)}, n_{(x+D_x,x+D_y)}, n_{(x+2D_x,x+2D_y)}, ..., n_{(x+p_x,x+p_y)}\big)$, where $D_x = D\cos\alpha$, $D_y = D\sin\alpha$, and α the angle that unites the two original points.

We speculate that the use of RNNs, by using these sequences, could provide enhancements in terms of making the network aware of the context surrounding the road. For instance, a wide road such as a highway is unlikely to branch into a sandy road that ventures into the woods. Similarly, RNNs should help when traversing bridges that cross over a road, where non-context-aware models could mistakenly assume that turning while under a bridge is a valid maneuver.

Most architectures use RNNs as an additional layer in the model. The inputs are first processed through a set of convolutional layers and then fed to the LSTM or GRU layers, recursively iterating the data until t steps are done. Usually, either the input to each step is the output of the previous, or the data is pre-processed beforehand to be structured in sequences. While this approach does indeed work for the vast majority of cases, they pose a series of challenges for the road extraction task. The sequences grow exponentially large as more junctions are found in the process, and accounting for all possible sequences out of terabytes of images is infeasible. As such, the approach we suggest implements a novel mechanism where each step of the RNN process generates embeddings that depend on the previous temporal information.

To generate the embeddings, the neural network is presented with sequences of points and their corresponding satellite images centered on them. We allow for slight modifications with respect to the central point, as we consider the possibility of introducing noise to the sequences' points. Based on these images and the path already covered, the network must decide where to move next. At the same time, the graph exploring algorithm must take into account their decisions to consider where to explore next. Overall, the following must be taken into account:

1. Whether the point it currently stands on is really a road or not. It can be the case that a previous movement incorrectly places the next step outside the road. In such a case, continuing to extract road information would be wrong, thus the exploration process must backtrack to the last junction and take another road.

2. In cases where the center is on or close to a road, the network must evaluate where the next point in the road is. Namely, it must find the next point so that the next image is centered on the road baseline, and it must be accurate enough to detect turns and junctions. In fact, in case of junctions, it must be able to detect all of the possible outcomes, not only a single turn.

3. Both the model and the exploration algorithm must consider that, at some point, the road will end. As such, it must be able to encode the possibility that the exploration should be halted.

To decide where to go next, the network would have to predict over the whole range of valid turns, that is, the 360 degrees available. Predicting all possible angles over the whole continuous range would be close to impossible, thus the 360 degrees are discretized and categorized over 32 distinct directions. Each direction comprises 11.25 degrees, with the actual movement happening at the center of the arc. For instance, moving in direction $0 \geq N < 32$ implies that the next point will lie at $\alpha = 11.25 \cdot (N + 0.5)$ degrees. The distance D moved in direction N is fixed during the hyper-parameter search, and is measured in amount of pixels. A simplified overview of the sequence of events described can be seen in Algorithm 1.

As stated, however, the actual algorithm is much more complex than the presented version. Upon detailed inspection of Algorithm 1, one can notice that the algorithm is incapable of detecting more than one possible direction, as N is a single output that maps to the next destination. Instead, the algorithm needs to simultaneously perform detection on all 32 directions. Effectively, each of the 32 directions must map to the probability of it being a valid destination. The network and algorithm must work in conjunction to ensure that:

Algorithm 1 Simplified version of the method proposed for road extraction via RNNs. Here, f is a function that maps a satellite image centered at the point of interest to an action M, where 0 is stop and 1 continue, and a direction to follow $0 \geq N < 32$.

```
Require: $D > 0$
  $(x, y)$ 2D coordinate in world-space of the center-pixel
  $M \leftarrow 1$
  while $M \neq 0$ do
      $X \leftarrow \text{imageAt}(x, y)$
      $M, N \leftarrow f(x)$
      $\alpha \leftarrow {}^{360}/_{32}\,(N + 0.5)$
      $x \leftarrow x + D\cos(\alpha)$
      $y \leftarrow y + D\sin(\alpha)$
  end while
```

1. All 32 directions are parsed and only those that have a probability greater than a prefixed threshold are kept. Moreover, if the road is wide enough, several close directions might be good candidates, which the algorithm must prune, taking the middlemost only. Either at the architectural level or as a cleaning step, the model must avoid going back in the same direction it came from.

2. For each of the clean and valid directions, it must push the resulting coordinates into a queue, such that all directions are eventually considered. The algorithm at each step pops a set of coordinates and the LSTM states, and continues in that direction.

A more complete version of the actual decision process, focusing on the complete direction detection phase, is outlined in Algorithm 2. The backbone network extracts the embedding E_x, which does not depend on the sequence of points already seen and contains information about the environment. Then, two stacked GRU layers are used, each with its own internal state that keeps track of the recurrent state. They output the time-dependant embedding E_{RNN_2}, which incorporates the information that comes from the sequence's history. Finally, the classification heads, f and g, decide if the path has any continuation and which angles are available, respectively. Unlike the first version, here M only indicates that the current road cannot be followed any longer, but the process as a whole is not halted. Instead, the next coordinate is popped from the queue and backtraced to it.

However, as it stands, the process only ends when there are no more points to explore. While it would certainly explore the whole graph, the result would be highly unstable. If the threshold T, by which a direction is valid, is low, the process could potentially take

Algorithm 2 Detailed proposal for road extraction via RNNs. Here, LSTM_1 and LSTM_2 indicate two distinct LSTM cells, each implicitly carrying its internal hidden state. Additionally, f, g are two functions that map the embedding extracted from the LSTM cells to the action and the direction, respectively. It is assumed that g already produces a clean version of the available directions.

```
Require: $D > 0$
  $Q$ A queue with a valid 2D coordinate to start from, and LSTMs zero-states
  while not empty($Q$) do
      $(x, y), h_1, h_2 \leftarrow \text{pop}(Q)$
      $X \leftarrow \text{imageAt}(x, y)$
      $E_x \leftarrow \text{Backbone}(X)$
      $E_{\text{RNN}_1}, \hat{h}_1 \leftarrow \text{LSTM}_1(E_x, h_1)$
      $E_{\text{RNN}_2}, \hat{h}_2 \leftarrow \text{LSTM}_2(E_{\text{RNN}_1}, h_2)$
      $M \leftarrow f(E_{\text{RNN}_2})$
      if $M \neq 0$ then
          $N \leftarrow g(E_{\text{RNN}_2}) \in \mathbb{R}^{32}[0, 1]$
          for $i \leftarrow 0 \ldots 32$ do
              if $N^{(i)} \geq T$ then
                  $\alpha \leftarrow {}^{360}/_{32}\,(i + 0.5)$
                  $\hat{x} \leftarrow x + D\cos(\alpha)$
                  $\hat{y} \leftarrow y + D\sin(\alpha)$
                  $push(Q, (\hat{x}, \hat{y}), \hat{h}_1, \hat{h}_2)$
              end if
          end for
      end if
  end while
```

hours to end, as many candidates would be pushed to the queue Q. On the other hand, if a high T would be selected to avoid this problem, it could fall right on the other extreme. Too few candidates would be considered, and the process would end before time, leaving roads unextracted. To tackle this problem, the algorithm is shifted to make a trade-off between quality and time. The queue is switched for a priority queue, so that instead of popping the first element that was inserted, it chooses to pop the one with the highest probability of being a valid direction. This way, the end-user can visualize the results at several times during inference and can decide where to stop the extraction. The priority system ensures that the most obvious and easy roads are processed first, while those more dubious are only to be handled at the end, if at all.

During both the training and test processes, the network needs images for all possible moves that will happen when iterating, due to the nature of RNNs. As such, the input image is not limited to the region around the initial pixel, but includes a whole region around it. Thus, while an individual point on the road is a centered patch of $W \times W$ pixels, the network is actually fed with a wider box that covers the whole possible range of movement. For instance, if training is performed with at most S steps, each with a maximum distance of D, and the image size is W, then the input image's side size is $2 \times S \times D + W$. If the image contained a straight road spanning horizontally, that would mean the algorithm could choose to go either right or left, sticking to that same direction until S steps were done. Thus, as it cannot be predicted beforehand which of those directions it will go, both sides must allow for the whole range of movements, which gives $2 \times S \times D$. Then, at each extreme, W pixels of context must be available on each side. While not all sequences must necessarily be the same length, training with batches requires that all input has the same dimensions, which forces the crop to be based on the largest possible sequence S.

While the process is fully automatic, both Algorithm 1 and Algorithm 2 have already shown that there are several steps that depend on external input. To avoid spending too much time out of the GPU and performing calls to Python, which is an expensive operation that makes the whole process noticeably slower, paths during training are calculated ahead of time. Nevertheless, Python is still required several times along the process. For instance, when backtracking and restarting at an old point, the input image must be regenerated. Also, when the last point in the input image is reached, a new input image centered at this point must be generated. Using bigger images, especially during inference, can help reduce the context switches, but limitations with RAM size make it impossible to fit the whole area of interest at once.

4.2 Implementation

The dataset used for this project was obtained using the same procedure as outlined in Bastani et al. (2018b). Namely, the satellite images and the road segmentation were obtained through different means. The images came from Google Maps, and were downloaded through their public API, while the information contained on the image, which is not necessarily limited to roads, came from Open Street Map. As shown in Figure 4.1, even if the sources differ, the information matches close to $1:1$ given an equal zoom level (also known as pixel resolution). However, this is not always the case, and sometimes there are some slight misalignments, especially when trying to find a road center-line. Not to say that the dataset obtained by Bastani et al. (2018b) is different from any other that might

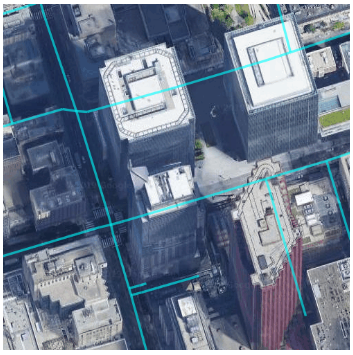


Figure 4.1: Sample image from Toronto, obtained from Google Maps, with ground truth roads from Open Street Maps in cyan. A road is hidden behind high constructions, which poses a serious problem for automatic extraction of roads.

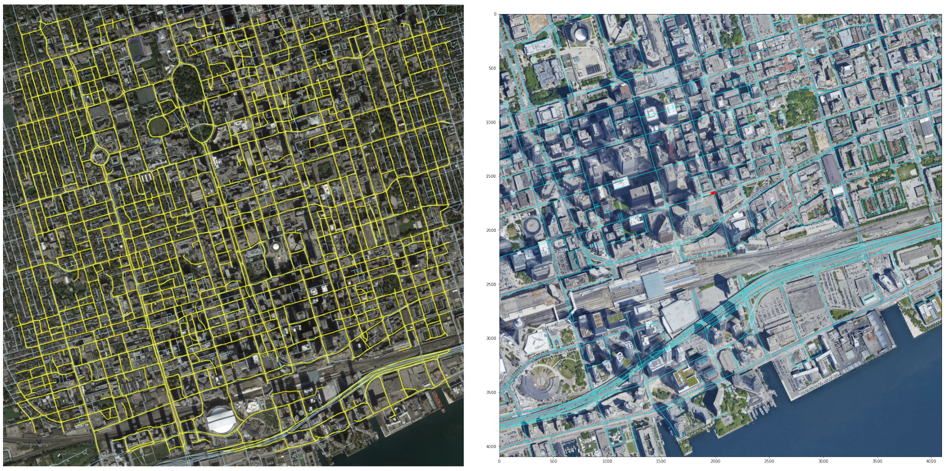
be downloaded at a later time. These images that depend on Google Maps are in constant change, as depicted in Figure 4.2.

For our experiments and dataset, we considered a step size $D = 20$ pixels to be enough coupled with a maximum of $S = 20$ steps. Only sequences with at least 15 points are considered, so that the network is always trained with long sequences and learns that most paths are long and do not always end. Through experimentation, the window size W was set to 200 pixels. The network was trained with Adam optimizer for a total of 100 epochs with a learning rate of 0.0003, decaying by 0.97 every 10000 steps. Points over the roads can be chosen either centered on them, or slightly altered by adding random noise, as depicted in Figure 4.3a and Figure 4.3b, respectively. An advantage of using RNNs, as depicted in both cases, is that the path can omit the angles to previous points, as the RNNs' memory should have already captured that information.

As outlined in Figure 4.4, the proposed architecture relies on a backbone network to process the satellite images, concretely the 50 layers ResNet pretrained with the Imagenet database. As previously mentioned, the resulting embeddings are fed into two stacked GRUs that compute the time-dependent ones. Further, they are fed into a dense layer with a ReLU activation and a linear dense to obtain the outputs for the next iteration.

4.3 Results

Considering that our method works at the graph level, and not via segmentation, we adopted the definitions from Bastani et al. (2018b). Traditional metrics used for road extraction,



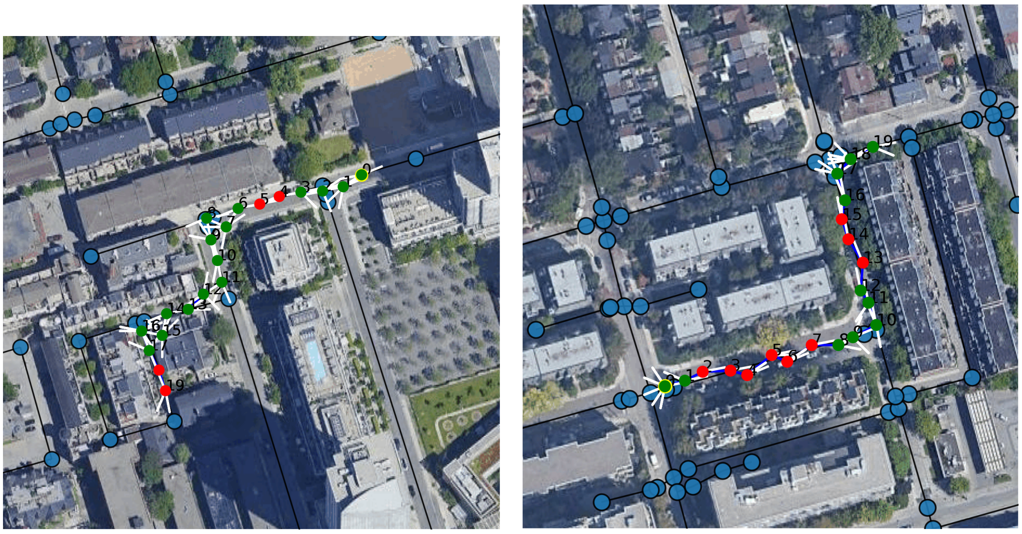
(a) Original

(b) Ours

Figure 4.2: Figure (a) shows the original image of Toronto as used on RoadTracer, while (b) depicts a subsection of that same image but obtained for this publication. As can be seen, there are major differences in lightning and even in the constructions themselves.

such as IoU and its derivatives, require comparing the extraction at the pixel level, which results in metrics being influenced by things such as the width of the line that defines a road. In segmentation-based models, where the network learns to classify all pixels of the road as such, it might not be a problem. However, for our work we would have to arbitrarily define a common width for all roads, which inevitably would lead to a false, probably underperfoming, result according to those metrics. Instead, Bastani et al. (2018b) proposed a metric that evaluates the quality of a graph of roads with respect to the ground truth based on the location and density of junctions. In a way, it measures the number of junctions the model has missed, the additional ones that are in reality non-existent, and the over-expanded ones (that is, junctions that have more roads connecting than in the ground truth). Their metric is better suited for actual road graphs as it does not require any rasterization step as metrics based on IoU.

As can be seen in Table 4.1, our model obtained comparable results to those of RoadTracer, yet it did not manage to surpass it. The table shows a comparison of the junctions that each model is capable of identifying in the city of Toronto and in Denver, in the column Correct, while outlining how many of them are incorrectly added (Extra) or missed from the ground truth (Missed). While the results are slightly inferior, there are notable differences between our implementation and RoadTracer. First, our model has a single starting position from which the whole graph is created, whereas their model requires a previous



(a) Without noise

(b) With noise

Figure 4.3: Visual representation of an input of the network. Paths can be constructed without noise (a), or with noise (b). The former has points that coincide with the baseline, shown as a black line, while the latter adds slight modifications. Both subfigures employ the same legend, points in blue are the ground truth nodes, and their discretization are shown in green if they are junctions or in red otherwise. White lines represent the angles to be learned, in this case only pointing towards unseen points.

segmentation to select an exhaustive collection of candidates to start with. Their method can, thus, afford to be more restrictive with the minimum probability required to follow an angle, as the roads left unexplored in one run can potentially be covered in another. Their method also relies on auxiliary input aside from the satellite image, as it uses rasterization of the graph explored so far, while ours only requires the satellite images themselves.

RoadTracer's authors also employed SP, as proposed by Wegner et al. (2013), and TOPO, by Biagioni and Eriksson (2013), to compare against previous models that did not use their metric. SP takes the shortest path from a set of predicted nodes to another set of real ones, and then computes which fraction of shortest paths lie at a maximum 5% distance of the real distance. TOPO, measures the number of nodes that can be reached from a given node in the ground truth, and compares that to the predicted graph. While they provide numeric results for SP, the exact numbers for the TOPO metric are not given. As such, and given that the results show that our model is not yet as performant as theirs, we have not yet procured ther results for SP.

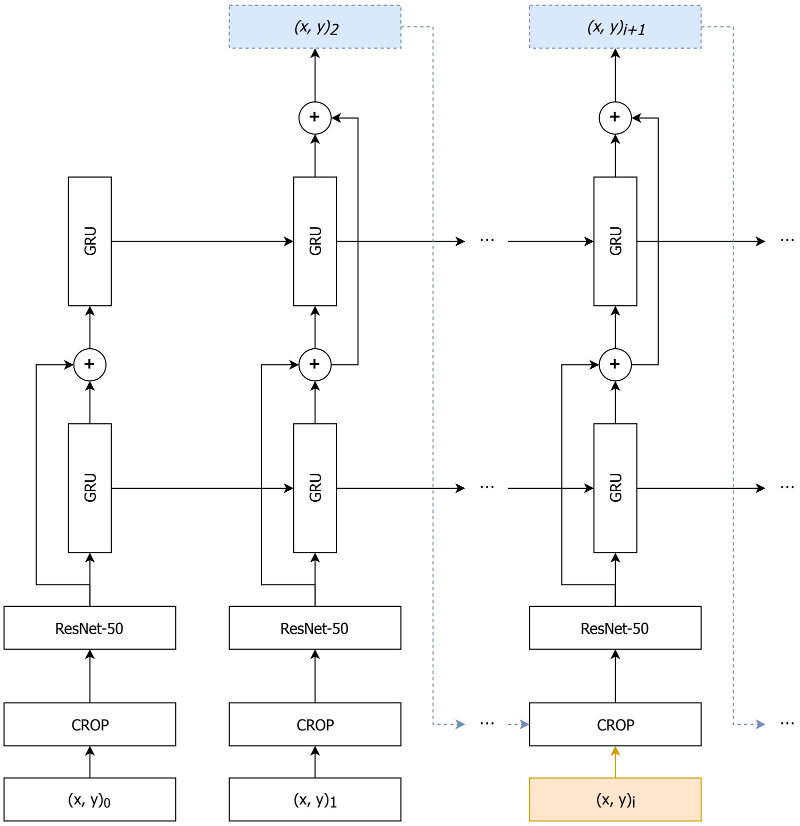


Figure 4.4: Schematic illustration of the model proposed, outlining the RNN process and the embedding extraction through ResNet 50. Nodes in blue are only considered during inference, as training already supplies a fixed sequence. Likewise, orange nodes are the pre-selected positions for training samples.

Included as part of the evaluation of the results is Figure 4.5, which shows the extracted roads for Denver. It is interesting to see that the network that was trained with noisy paths, Figure 4.5b was capable of finding more valid paths than the one trained with perfect paths, Figure 4.5a. We attributed the behavior to the extended diversity that the noisy network has seen, enabling it to recover from wrong turns while also making it more prone to discover junctions. Similarly, Figure 4.6 shows the final road network extracted from the city of Toronto.

Table 4.1: Junction metric evaluated on RoadTracer and our model for two test cities, Toronto and Denver. Each column indicates the number of junctions for that category out of the Total.

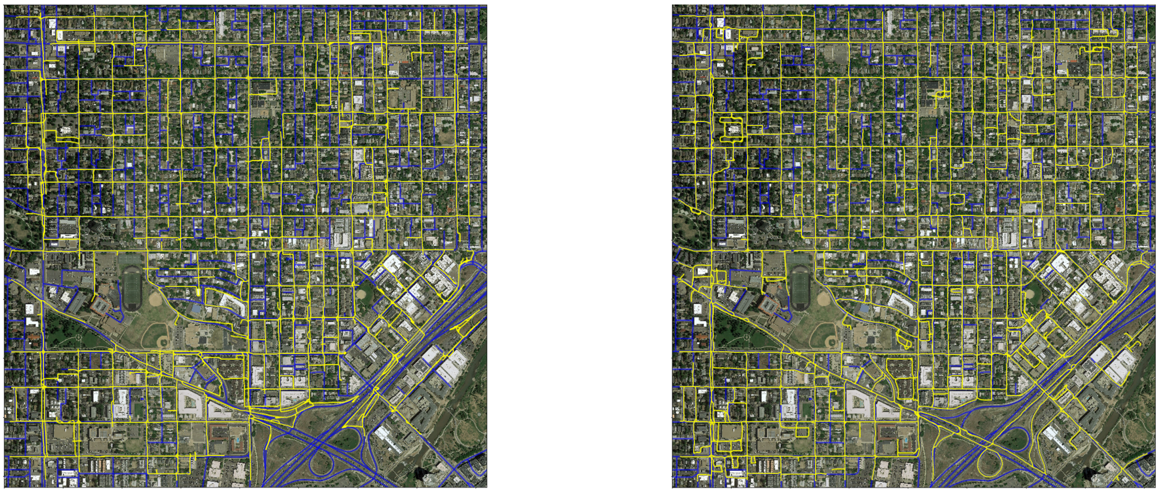
City	Model	Total	Correct	Wrong	Extra	Missed
Toronto	RoadTracer	223	207	0	145	16
	Ours	223	193	0	149	30
Denver	RoadTracer	421	393	0	41	28
	Ours	421	327	0	30	97

4.4 Conclusions

This work proposes the use of RNNs applied to road extraction. Instead of using CNN-based models and performing segmentation, we utilize the inherent information in the temporal axis to directly extract graphs of roads. The method is grounded in paths being discretized over a sequence of points at a determined distance from each other, so that their connections can be expressed in terms of angles. As such, each point of the sequence contains local information relative to the place it stands, and historical information with respect to the environment and path traveled so far.

The results obtained demonstrate that the method and its implementation are feasible and competitive with respect to state-of-the-art methods. Even though the final results are not superior to those of competing models, our model attains an almost equal result without requiring additional external information. Moreover, our model is capable of obtaining a faithful graph from a single initial point, instead of needing a whole list of candidates to make sure that each road is covered. Thus, this work demonstrates that using RNNs to solve problems that inherently contain a temporal axis can be extremely beneficial. They reduce the prerequisites needed to make road extraction work while not sacrificing much in terms of performance.

Experiments also confirmed that using paths with noise is overall positive for the final model. For instance, a model trained with noisy paths is capable of course-correction—if the deduced path deviates from the center-line, it will attempt to find it again—and are better at exploring junctions and dubious angles.



(a) Without noise

(b) With noise

Figure 4.5: Graphs created with the proposed method if trained (a) without noise, or (b) with noise. The ground truth is marked with blue lines, while outlined in yellow are the roads detected by the network and graph exploration algorithm.

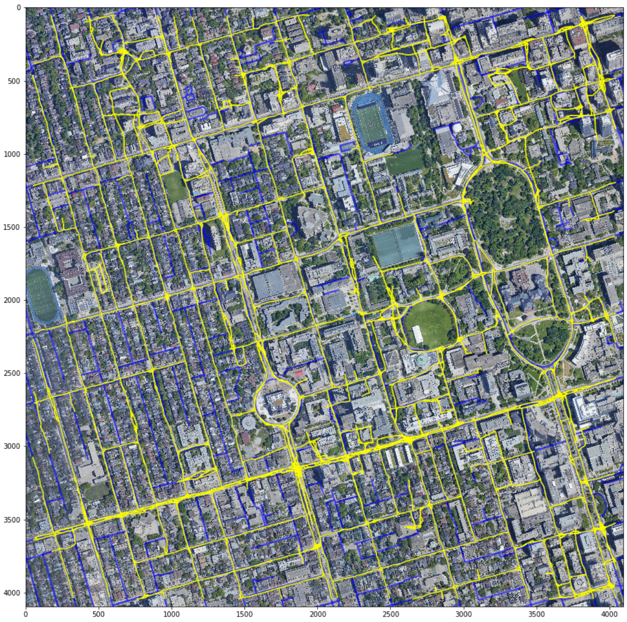


Figure 4.6: Graph of roads extracted from Toronto. The ground truth is marked with blue lines, while outlined in yellow are the roads detected by the network and graph exploration algorithm.

Chapter 5

Recurrent Neural Networks applied to Wireless Capsule Endoscopy

Contents

This chapter is centered around the application of RNNs in WCE databases. As a continuation of the work with RNNs for road extraction, LSTMs are used to demonstrate how using contextual information can be beneficial for other tasks. Namely, we show that a network trained over a difficult set of images containing polyps in conjunction with its surrounding frames can positively impact its classification accuracy compared to standard networks.

WCE is first introduced in Section 5.1, where a previous contribution is briefly used to explain basic and common concepts in WCE. Then, in Section 5.2 the approach is introduced, followed by the dataset in Section 5.3, and then the implementation in Section 5.4. Results are presented in Section 5.5 and finally the conclusion in Section 5.6.

5.1 Previous Work

To better understand WCE and the difficulties inherent to these types of datasets, a publication previous to the start of the thesis is used Seguí et al. (2016). It consists of a standard CNN with two dense layers to perform classification, thus it could be considered vanilla, in the sense that it does not use any kind of specialized layer or loss function. Seguí et al. (2016) studies alternatives to combat the lack of data by providing contextual information, albeit not in the form of temporal data.

The publication explored the effects and returns of increasing the dataset size, and outlined the importance of obtaining high-quality labeled data. In fact, they demonstrated an increase of up to 3% in accuracy when increasing the size from 10,000 samples to 100,000. However, relative to this chapter, they also investigated the effect of additional contextual information for small datasets. The publication presented a VGG inspired architecture that took two additional inputs in addition to the RGB image of a WCE frame. For each pixel in position (x, y) of the input image I, it computed the Laplacian $(L(x, y))$, Equation (5.1), and the highest eigenvalue in absolute value $(H(x, y) = \lambda_i)$ of the Hessian Matrix, $HM(x, y)$ in Equation (5.2).

$$H(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{5.1}$$

$$\begin{gathered} HM(x,y) = \begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix} \\ H(x,y) = \lambda_i \mid |\lambda_i| > |\lambda_j| \forall i \neq j \end{gathered} \tag{5.2}$$

While the Laplacian could certainly be inferred by the network itself after two layers of convolutions, it requires that it find the appropriate combination of filters. Moreover, it would take time, parameters, and compute power to compute the Hessian value, which is otherwise just fed as an input. The work argued and demonstrated that this additional information is beneficial for the overall performance. Specifically, for small datasets, it serves as a prior distribution and is able to increase, in their case, the performance by 2.6%.

Furthermore, the publication delved into the method used to merge these additional inputs. There are two options, first merging the information upfront and letting the network process it at once as a single blob, or alternatively making it process each of the inputs separately and merging them later. These two options are known as early-fusion and late-fusion networks, respectively. The authors empirically proved that late-fusion works better for WCE data, giving the network the freedom to learn independent features for each of the inputs. In a way, late-fusion works by extracting spatial features independently, and then combining the information and processing it on the fully connected layers at the end. Relative to this chapter, RNNs as implemented in most literature, can be seen as a late-fusion mechanism, as they operate after a set of transformations are already done to the image.

In conclusion, Seguí et al. (2016) showed that the addition of external information, even if it came from the image itself, aids in producing more robust classifiers. The main work presented in this chapter expands into this concept, re-using the late-fusion concept by merging temporal features after independently processing n contiguous frames.

5.2 Approach

A common practice in most DL fields, including WCE models and in particular polyp classification, is to use a backbone network, as has been shown in Section 2.4.2. As such, medical imaging models extensively use ResNet or more recent models such as DenseNet and its variants, coupled with a classification loss, typically a cross-entropy loss, at the end. As shown previously, however, the limited availability of data and class imbalances present in polyp detection often make this approach unrealistic. Trying to classify directly often results in highly overfitting models that are not suitable for deployment in CADx systems. While introducing specialized losses to counter unbalances, such as the TL, helps in tackling some issues, they still have an ample margin of improvement (Laiz et al., 2020).

This work proposes to augment each image with its surrounding eight images by means of RNNs. In the previous chapter, the input was not known beforehand, as exploring a graph of roads requires taking chances at new unknown paths. WCE data, however, consists of prerecorded videos, thus at any given frame of the video, it can be known what comes before and, most importantly, after. A model that works with these videos can take advantage of this fact and peek into the future of a frame to extract better contextual information. If the model had to work in real-time, for example by processing images at the same moment they were recorded, it would be completely impossible to process images ahead of time. As such, we propose to approach the model as a many-to-one (M2O) RNN or as a many-to-many (also called sequence-to-sequence (S2S)) network. These terms, borrowed from the system analysis field, refer to how many outputs are given after processing a sequence of inputs. For instance, in M2O a sequence is used to produce a single output, whereas in the second case, S2S, an output is produced for every element of the input sequence. In terms of classification, the first technique would produce a single class for the whole sequence, which could be whether the sequence as a whole has any polyp or not. On the other hand, the second would produce a prediction for each of the images in the sequences, specifically indicating if there is a polyp or not in each of them.

While using M2O means that the result would be more accurate, as every image has the most context possible and only a single decision must be taken, it would result in inexact locations. It would not be suitable for any system that requires knowing the exact image that contains polyps, as most CADx systems demand. The other network type, S2S, might be slightly harder to train, but the results would be exhaustive and suitable for real-time applications in CADx.

In a S2S setting, the whole input sequence would receive a classification, which traditionally would mean that images at the start of the sequences would receive less contextual information. RNNs work by remembering previous inputs, which for the first image on a sequence would be none. On the other extreme, images on the end would have the most context. While this holds true for standard RNNs, it can be argued that when working with previously known inputs, one could reverse the sequence so that the first image now would be the last and receive the most context. Precisely, BRNNs apply this context by using two RNNs, one that sees the sequences in their normal order, and one that processes it in backwards order, combining the information on each one (Schuster and Paliwal, 1997).

WCE is the perfect candidate for BRNN, as the videos are fully recorded once inspected by the model, unlike other fields like simultaneous language translations, which require immediate output and cannot tolerate waiting for the speaker to end. As such, in BRNNs the center-most image in the sequence will receive double the context that those on the extremes, but neither of the frames will have zero context, unlike standard RNNs.

This work uses Bidirectional Long Short-Term Memory (BLSTM) to process the embeddings resulting from a ResNet 50 model, as shown in Figure 5.1. It must be noted that our approach comes as a second phase, as it uses an already pretrained network that can accurately detect if an image is a polyp or not. In fact, our approach is a specialization for extreme cases where the original network might fail to detect a polyp, and should be used as a second opinion. The model itself breaks the temporal sequences into individual images which are then processed by the former, pretrained, network. As such, at this stage each frame is processed separately and each obtains its own set of features and, by extension, embedding. The BLSTM further down the line is where sequences are restored, bringing back the temporal axis again, and combining the information.

Unlike Seguí et al. (2016), the information extracted from context and the images them-

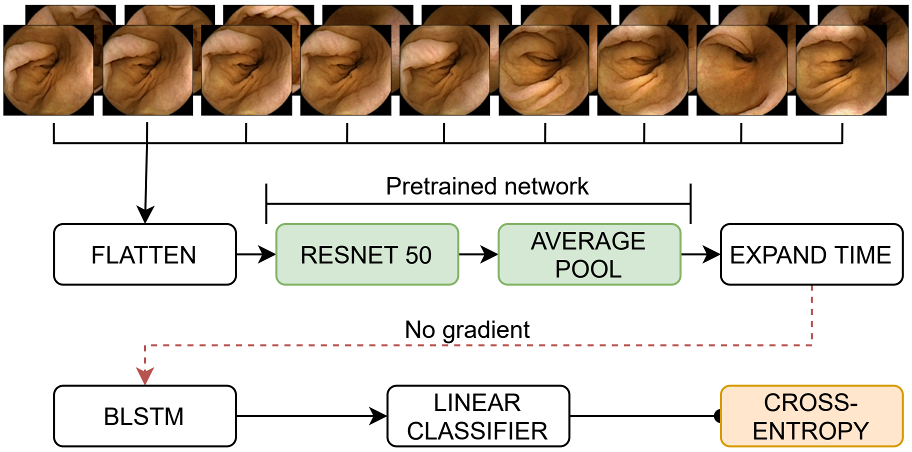


Figure 5.1: Schematics of the proposed architecture and method. The BLSTM layer indicates a Bidirectional Long Short-Term Memory. Before the backbone network, the RESNET 50, the input data is flattened over the batch axis in the FLATTEN block, removing the temporal axis. Likewise, EXPAND TIME recovers the temporal axis after it and before the BLSTM block. The LINEAR CLASSIFIER is a dense layer without any activation, and its outputs are fed into the CROSS-ENTROPY loss (in orange). The dashed red line denotes that the errors propagated from the classification loss do not change the embeddings learned by the pre-trained network (in green)

selves is not fed into dense layers, but instead is directly passed into a linear classifier. The BLSTM layer already acts as a means to merge the information and process it, thus making additional dense layers redundant for our case. While the pretrained model used a TL to learn better embeddings and avoid overfitting, the finetuning process introduced in this work does not need them. The network already begins the training process with good-enough embeddings, with no need to further modify them to function correctly. As such, the gradient coming from the BLSTM is removed and any alteration of the embeddings is prevented. In fact, enabling gradient here and allowing the embeddings to change would be detrimental, as the imbalance of the dataset would severely influence them and any potential information they held of polyps would be lost.

5.3 Dataset

The first step toward implementing an architecture that models our approach is the acquisition of a database. As we are not trying to create a model that works from zero, but one that can be used as a second step and starts from an already performant model, a big dataset is not required. Instead, it relies much more on the difficulty and quality of the samples. With that in mind, the dataset should be centered around polyps, especially those that are deemed hard to classify, so that a more robust model can be obtained.

To such means, the images are selected among the database used in Laiz et al. (2020). In fact, this dataset is thoroughly used both in this thesis and other publications from the same research group (Seguí et al., 2016; Laiz et al., 2019, 2020). The dataset contains 248,136 frames sampled from 120 procedures, each from different patients. These videos were recorded with Medtronic PillCam SB3 and PillCam Colon 2, obtaining 2,080 images with polyps, and 246,056 without. In total, a 0.85% of the images contain polyps, which clearly outlines one of the problems this thesis is trying to tackle.

The labeling process to obtain the samples is a two-step process. An initial report is produced by eight expert readers, endoscopy nurses with at least three months of experience, who tag potential polyp frames and possible other pathologies that require detailed revision. Then, two medical doctors (one gastroenterologist and one internal medicine) obtain the final version of the dataset by revising the images. The polyp's sizes were calculated with Rapid PillCam Software V9 and are reported in Table 5.1. The largest polyp was determined to be 16 mm. Tumors were considered positive, while any other pathology, like ileal lymphoid hyperplasia, bleeding, and diverticulitis, were discarded from the dataset.

To create the dataset used in this publication, the model from Laiz et al. (2020) was used to classify every single image, obtaining its probability to be a polyp. The images and its scores were submitted to a filtering process based on its ground truth class and the probability extracted from the model. All those that were previously labeled as polyps were kept for the database in this publication, but only the hard negatives—images not containing polyps and whose probability of being polyp are high—are considered. Selecting all positives ensures that the system does not lose its ability to detect polyps, even the easy ones, while discarding negatives that are easy to classify eliminates examples that would not provide any benefit to the network.

As such, the final database contains between 49 and 54 sequences for each of the 110 resulting videos. Table 5.2 shows a detailed outline of the resulting images and their classes, with only 1.9% of the sequences containing polyps. Each sequence, as introduced in the previous section, is conformed by the eight contiguous images around the filtered image for a total of nine images.

Table 5.1: Morphology - Polyp's size in the Polyp WCE dataset, as reported in Laiz et al. (2020).

		Morphology			Total
		Sessile	Pedunculated	Undefined	
Size	Small (2–6 mm)	65	4	19	88
	Medium (7–11 mm)	29	4	20	53
	Large (12+ mm)	8	3	13	24
Total		102	11	52	165

Table 5.2: Dataset structure and class distribution. Columns starting with Fold indicate the 5 different folds used during evaluation, as proposed in citetpolyp, with their corresponding Train and Test distribution. Here, Seq. is an abbreviation for Sequences.

	Total	**Fold 1**		**Fold 2**		**Fold 3**		**Fold 4**		**Fold 5**	
		Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
Videos	110	86	24	87	23	91	19	88	22	88	22
Sequences	5523	4326	1197	4361	1162	4569	954	4417	1106	4419	1104
Images	49,707	38,934	10,773	39,249	10,458	41,121	8586	39,753	9954	39,771	9936
Frames with polyps	2100	1772	328	1521	579	1733	367	1571	529	1803	297
Frames without polyps	47,607	37,162	10,445	37,728	9879	39,388	8219	38,182	9425	37,968	9639
Seq. with solely polyps	105	87	18	73	32	89	16	79	26	92	13
Seq. with at least one polyp	365	312	53	263	102	295	70	276	89	314	51

5.4 Implementation

To allow fair evaluations, the same cross-validation strategy as Laiz et al. (2020) was used. The dataset was divided into five-folds, filtering all images of the same patient to a single fold and ensuring that not only comparisons are valid, but also that a patient can not be found in the training and test set at the same time. Additionally, an ablation study was carried out to evaluate the improvements brought by each component. The addition of the BLSTM is compared to (1) the SSL model reported in Pascual et al. (2022), (2) the same model adding a trainable dense layer (SSL CNN), and 3) adding a trainable LSTM model (SSL LSTM).

Notably, all three implementations and our method are engineered to keep the same number of parameters, so that the comparisons are fair and cannot be attributed to an increase in parameters. The SSL CNN model adds a dense layer of 2048 units, the LSTM is comprised of 2048 hidden units, and the BLSTM can be broken down to two 1024-wide LSTMs. The baseline model does not add any additional parameters, but it should serve

Table 5.3: Mean and standard deviation of AUC ROC and Sensitivity (at given Specificity rates). All metrics have been obtained using a 5-fold cross-validation. The best performing system is highlighted in bold.

Network	**AUC** (%)	**Sens@Spec80** (%)	**Sens@Spec90** (%)	**Sens@Spec95** (%)
SSL	88.16 ± 1.83	81.40 ± 3.38	66.16 ± 3.38	42.85 ± 7.55
SSL CNN	92.84 ± 3.34	90.96 ± 5.96	84.04 ± 8.84	65.62 ± 16.65
SSL LSTM	92.86 ± 3.10	91.94 ± 5.47	81.78 ± 7.70	62.79 ± 11.94
SSL BLSTM	$\mathbf{93.83 \pm 2.65}$	$\mathbf{92.69 \pm 5.43}$	$\mathbf{84.44 \pm 6.59}$	$\mathbf{70.23 \pm 12.62}$

as a lower bound rather than an actual competitor. Overall, this guarantees that any gains are due to architectural changes and the addition of contextual information.

Following the evaluation strategy proposed by Laiz et al. (2020), each fold is evaluated through the use of Area Under the ROC Curve (AUC), where ROC stands for Receiver Operating Characteristic Curve (ROC). As such, each split is evaluated independently and then the mean and standard deviation of the five executions are reported. Furthermore, following the same guidelines, three sensitivity values at three different specificity thresholds, 80%, 90%, and 95%, are provided. These three thresholds serve to provide an idea of how many images a physician has to examine to achieve a certain performance. For instance, sensitivity at 80% specificity indicates the number of positive samples that would be detected if an 80% of the data was discarded according to the sentitivity measure.

5.5 Results

Several tests were designed to find the most appropriate batch size, which determines the number of sequences per batch. Up to 32, all strategies showed an increase in performance, but from there on, the margins were marginal for all but one model. BLSTM benefited from a larger batch size of 72 sequences, showing a noteworthy improvement and making it the only strategy that used a higher image count per batch. Sizes greater than 72 could not be tested due to hardware limitations.

After training all five splits and evaluating with AUC, the BLSTM model clearly outperforms the other methods. Particularly, Table 5.3 shows that it achieves a total of 93.83±2.65%, which represents an increase of 5.65% with respect to the baseline model, and of 0.99% compared to the finetune model without RNN layers. Not only does the BLSTM model beat any non-temporal model, but it also beats the single pass, LSTM, counterpart.

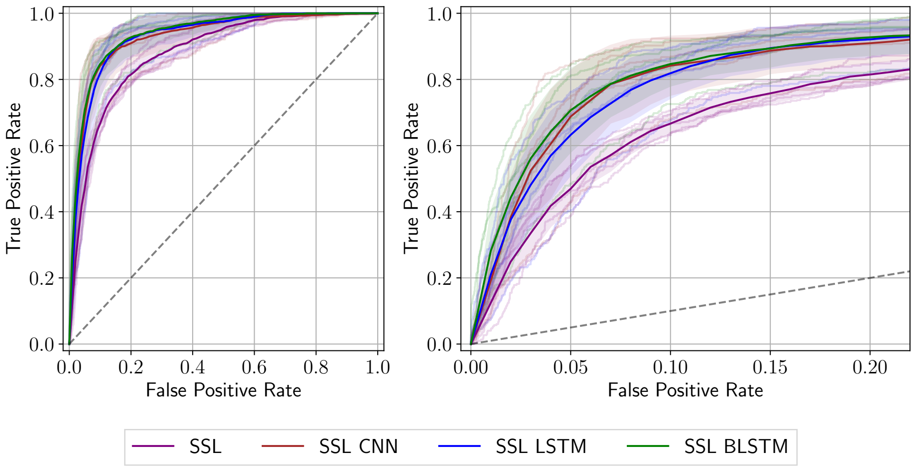


Figure 5.2: Receiver Operating Characteristic curve of the models studied. Each fold of the cross-validation split is shown in a softer version of its respective model color. The strong line represents the mean of those executions, while the background shade is the standard deviation. A magnified version is provided on the right.

Clearly, the addition of information from both ends of the sequence is not only beneficial but crucial, as the LSTM strategy achieves approximately the same performance as the SSL CNN one.

Table 5.3 also depicts the Sensitivy at Specifity at 80%, 90%, and 95% thresholds. As the results above, these ones show that the BLSTM model increased the sensitivity with respect to all other models, with the most prominent change found at 95% specificity. Discarding an 80% of the images according to its specificity value, a physician would be adle to detect up to a 92.69% of the polyps, while increasing the number until only a 5% of the dataset is revised, they would successfully identify 70.23% of them. The model enables physicians to detect more polyps with less time spent than any other of the models studied. Figure 5.2 shows an outline of the AUC, where the BLSTM curve stands out of the rest in the first 5%, once more proving the exact same behavior these metrics are reflecting.

Analyzing qualitative results is equally important to revise the numerical results, as it allows comprehending where the network fails to classify as well as identifying its forte. This is a crucial step for models intended for real-life deployment, such as CADx systems. Figure 5.3 depicts six different sequences annotated with the ground truth labels and the BLSTM model predictions. As can be seen, the network sometimes fails to accurately determine the frame where a polyp is no longer found in the labels (rows 2, 3, and 4), or misses the last one (row 5). Both could be attributed to the subjective decision of

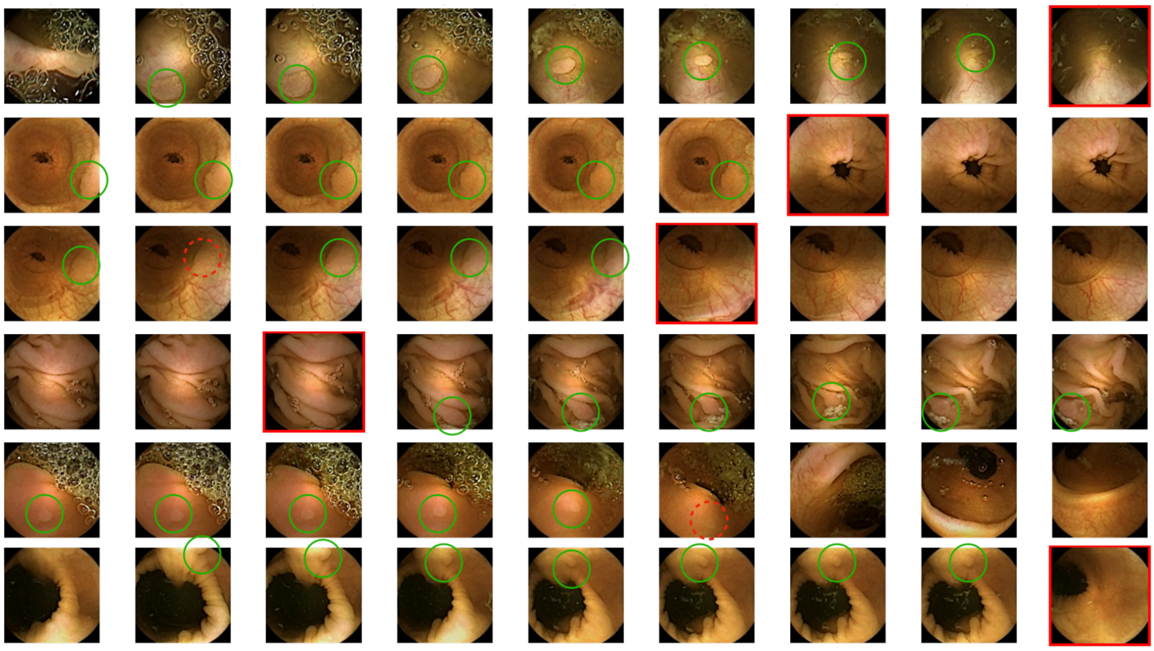


Figure 5.3: Image sequences. True positive detections (images correctly classified) are marked with a green circle, dashed red circles for False negatives (polyps classified as non-polyps) and images with a red border are False positives detections (non-polyps classified as polyps).

determining where a polyp is no longer shown. Some small parts, almost imperceptible, could still be present in the images and detected by the network. The third row shows the most concerning issue, as the network fails to notice a polyp in the second image of the sequence while both its left and right are correctly predicted. We believe this could be attributed to the same issue of lacking context that has been mentioned earlier. While BLSTM does give more context than the plain dense in SSL CNN or the additional LSTM, images at the sequences' extremes still suffer from potential lack of information.

5.6 Conclusions

The question raised of whether RNNs, are particularly BLSTM layers, could help produce more robust and reliable classifiers has been examined and developed, reaching results that not only show a positive step towards better models, but also reached new state-of-the-art results. From the qualitative analysis, it can be extracted from that the model makes no critical of fatal errors. While some frames are miss-classified, their immediate neighbors are not. A physician would rapidly notice the error and take the appropriate action. Overall, it can be assessed that the model meets the quality standards reported in the previous metrics.

We claimed and proved that using sequences of WCE images, instead of the frames in isolation, is beneficial. Physicians can obtain models that are much more reliable thanks to the surrounding contextual information, which empowers them to detect any issues that might have arisen through the automated classification process, such as the transition frames between a polyp appearing and disappearing from the capsule field of view.

This work was published in Diagnostics, Special Issue Capsule Endoscopy: Clinical Impacts and Innovation since 2001 (Reuss et al., 2022).

Chapter 6

Self-supervised Learning for Wireless Capsule Endoscopy

Contents

This chapter is centered around the problems derived from the lack of labeled data and severe class imbalances. Contrastive learning through SSL and unlabeled data is leveraged to produce better models that are robust against overfitting and have greater generalization capabilities. Particularly, the temporal axis of WCE datasets is exploited in a novel

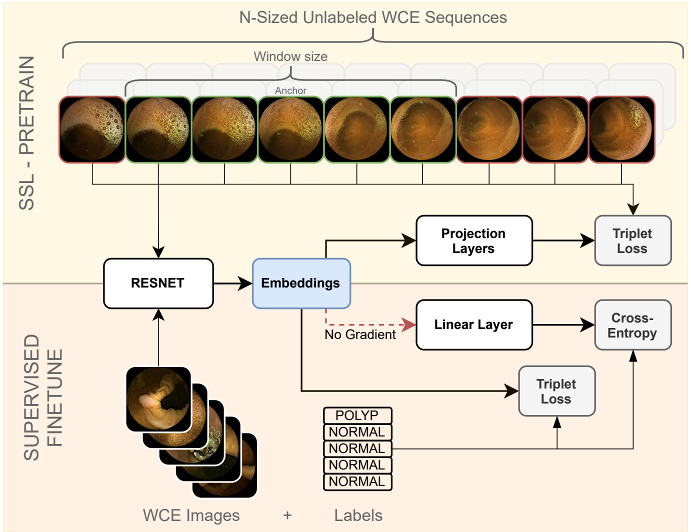
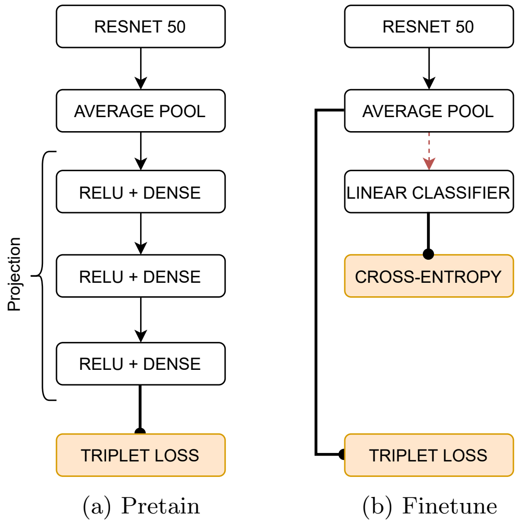


Figure 6.1: Overview of the proposed method, including the pretrain phase, in the upper half, and the final finetune phase in the lower half.

approach that considers the distance between frames as a proxy to introduce similarity between images.

6.1 Approach

The first consideration we must take into account is that the objective of this work is not merely obtaining a higher classification score in WCE datasets through added regularization, architecture tweaking, or parameter stacking. Instead, we seek to make existing networks and architectures more robust by introducing a preliminary step that uses unlabeled data. One such method to do so is through SSL. As has been explained previously, the most recent advances point to contrastive learning, where the network learns by contrasting two examples of the dataset. The standard constrastive setting, as explained by Chen et al. (2020a), compares an image to an augmented version of itself by, for example, introducing changes in hue, saturation, or brightness. This approach, however, is lacking when considering the inherent temporal structure of our data. WCE datasets come as sets of videos, where each video is a continuous stream obtained from a patient. Using contrastive learning based on single images input would completely ignore the temporal information, wasting hidden potential. The first requisite for our method is that it takes into account this inherent information and uses it to the model's advantage.



(a) Pretain (b) Finetune

Figure 6.2: Detailed illustration of the network architecture. The parameters obtained during pretain for ResNet are used in the finetune phase, while the projection layers are removed. Here, the dashed red line denotes that gradient is stopped and does not influence its parent layers.

In SSL, the training process is divided into two distinct phases. The first one, usually called pretrain, is where the SSL process really happens. The network is trained with pseudo-labeled data, aiming to produce embeddings, compact yet rich representations of the images that contain enough information to make them useful for other tasks. It is during the second stage, finetuning, where the embeddings are used for downstream tasks. For instance, a network pretrained on WCE data can then be finetuned to detect bleeding, polyps, motility events, or any other pathologies. Figure 6.1 shows an outline of the process described. The network is trained with SSL only once, and the resulting model can be reused for an unlimited number of applications, each using the rich embeddings for its own objective. This section is, consequently, divided into two, one for the SSL stage and one for the domain-specific training.

6.1.1 Self-supervised learning

Our proposal, thus, considers images not as an individual entities but as a sequence of N sequential frames in a video. Instead of altering an image and comparing it to itself, we proposed that an image is compared to another one of the sequence, deciding if they are similar or dissimilar based on the distance between them. For instance, an image and its immediate neighbors should be similar, while those on opposite extremes should be dissimilar. Formalized, given an N-samples sequence and an image a in the sequence, the $W < N$ frames to the left and to the right are considered similar, while any other image outside of this $2W + 1$ window is dissimilar. That is, given a, the similar images—also called positive—are all images i such that $d(a, i) < N$, where $d(\cdot, \cdot)$ indicates the number of frames between both images in the sequence. Naturally, images that are located close to the center of the sequence will have at most $2W$ positives samples, while those on the extremes will tend towards having only W.

The final objective is that the representation $f(x)$ of an image x, where f is the neural network, is closer to the representation of those images that are most close to it. In other words, the network will make similar images closer in the embedding-space, while those that are sufficiently away in the video will have distant representations. One way to achieve this effect is through contrastive losses, and specifically through TL. In the original formulation, as seen in Equation (6.1), an anchor sample a is taken along another of the same class p, called positive, and one of a different class, the negative n. The formulation eventually guarantees that the distance from the positive to the anchor will at least be α, the margin, units larger than the negative to the anchor.

$$TL = \max(||f(a) - f(p)||^2 - ||f(a) - f(n)||^2 + \alpha, 0) \tag{6.1}$$

In consequence, for the problem at hand, given an N-frames sequence, for each image a of the sequence, triplets will be formed by taking all distinct combinations of a positive sample p from the $2W$ window (not including itself) and a negative n from outside the range. Then, TL, as shown in Equation (6.1), will ensure that those images are close together.

Finally, to make the process of triplet selection smooth, pseudo-labels are introduced for each image. During labeling, it must be taken into account that each patient has a unique video, and that sequences from different videos might appear in a single batch during training. As such, a single image i will be identified by a unique pair: a) the video identifier $\gamma(i)$, which can be an incremental number, and b) the frame position within the video $\delta(i)$. Then, the pseudo-label $y(i)$ is constructed as shown in Equation (6.2), where M is a sufficiently large constant so that $\forall i, M > \delta(i)$.

$$y(i) = M\gamma(i) + \delta(i) \tag{6.2}$$

This formulation gives strong guarantees towards negative selection when multiple videos are involved, which positively impacts the applicability of our method. For instance, given frames i, j from different patients, it follows that $|y(i)-y(j)| \geq |M\gamma(i)-M\gamma(j)| \geq M > 2W$, given that $\gamma(i) \neq \gamma(j)$. Whereas the standard case where two images i, k come from the same video still reduces to the basic distance between frames that the method relies on, as $\gamma(i) = \gamma(k)$ and $d(i, k) = |y(i) - y(k)| = |\delta(i) - \delta(k)|$.

Although the pseudo-labeling process is robust, the method is bound to have false positives and false negatives nonetheless. The nature of WCE pillcams implies that sometimes they move slower or capture the same viewpoint for several frames. This might result in insufficiently large W parameters to capture this constant field of view, and inevitably some similar images might end up being considered negative even if they are not in reality. Likewise, two images that come from different sequences, or even videos, might not necessarily be distinct. For instance, it can happen that two videos contain almost identical sections, in which case the method will consider them negative even though from a visual standpoint, they are not. While the issue is real and acknowledged, it can be safely assumed that it will not negatively impact the training process, as the number of samples that fall into these two categories is several magnitudes lower than the total number of good triplets in the dataset. The network should, at most, treat it as noise and learn to ignore them.

Several considerations must be made and tested when attempting to use this method with a dataset. For instance, the parameters N and W must be empirically found through extensive evaluations. Also, one must consider if a single batch might contain multiple sequences or only one, and in the former case, whether they would come from a single video or multiple. While the method is prepared for multi-video and multi-sequence batches, it does not necessarily mean that it outperforms all other possibilities, such as restraining the sampling for triples to single videos. Lastly, another parameter to consider is the total number of sequences in a single batch, which might influence the training result. These reflections are analyzed in Section 6.3.

As shown in Figure 6.2a, we propose to follow standard network architectures, using a ResNet 50 as the backbone of the model and implementing three projection layers as proposed in SimCLR (Chen et al., 2020a). Using their same technique, TL is applied right after the last projection layer, which produces better results when using the last layer of the ResNet network in the finetuning process.

6.1.2 Supervised learning

After the SSL process, the network parameters are reused for downstream WCE tasks. To such end, the architecture is kept intact except for the projection layers, which are removed and not used, as shown in Figure 6.2b. This decision is made according to the findings of Chen et al. (2020a), where projection layers are used only for SSL. The embedding size reverts to the 2048 units, which is the number of channels output by the last global pooling of the ResNet 50 architecture.

Likewise, previous publications have explored the use of crossentropy and TL in WCE data. For instance, Laiz et al. (2020) showed that using crossentropy alone to perform classification is detrimental to the network, while adding a TL directly to the embeddings helps tackle it. Moreover, we argue that the TL helps by further specializing the embeddings that have been learned during the SSL phase. Thus, the finetuning model is composed of a standard TL loss, TL_{sup}, as defined in the original paper and with the specific domain dataset's labels, and a crossentropy loss, $L_{crossentropy}$. The final loss L_{sup} to be minimized is shown in Equation (6.3). Lastly, to further prevent the most represented class from skewing the embeddings, all gradient coming from $L_{crossentropy}$ is stopped after the linear classifier layer.

$$L_{sup} = TL_{sup} + L_{crossentropy} \tag{6.3}$$

6.2 Datasets

Three distinct datasets were used for the present work. The first one was solely used for the SSL step and consisted of unlabeled data, while the other two were used for the finetuning stage and served to validate the improvement in performance gained with SSL.

6.2.1 Self-supervised dataset

This dataset was acquired through 49 procedures with Medtronic PillCam SB2. We selected only the small intestine and colon segments and discarded any other image, ending with a total of 1,185,033 frames. These videos had not been labeled by anyone, professional or otherwise, and thus could not be used in any supervised process. However, the pseudo-labeling procedure outlined in this chapter made it a perfect candidate for SSL.

6.2.2 Polyps dataset

The polyp dataset, as introduced in Chapter 5, was used first as a proxy to evaluate the hyperparameter selection for SSL. During this usage, however, it is not meant to be used as a validation for the finetuning stage but rather only as a method to validate the parameters' quality.

Further, the dataset was used to evaluate the quality and richness of the embedding by performing polyp classification. While this dataset is composed of videos obtained by both Medtronic PillCam SB3 and Colon 2, which are slightly different from those of PillCame SB2, it does not pose a problem. Laiz et al. (2019) showed that using SB3 and SB2 simultaneously or even as a finetune step is possible, and further demonstrated that having mixed sources poses no problems for polyp detection (Laiz et al., 2020).

As laid out in the previous chapter, this is an extremely unbalanced dataset. The negative class (non-pathology) has a representation of over 99% of the total sample amount, which means that polyps are virtually non-existent. The difficulty in this dataset lies in creating models that can combat unbalances and are capable of successfully detecting new polyps during inference.

6.2.3 CAD-CAP WCE

This public dataset was created for the Gastrointestinal Image ANAlysis (GIANA) challenge (Dray et al., 2018). The images are divided into three classes: normal, inflammatory, and vascular lesion, each consisting of approximately 600 samples, for a total of 1,800 images.

This set poses an interesting problem different than the previous one. While the classes are indeed balanced, and thus the risk of overfitting into a single class is reduced, the total number of samples is extremely low. Consequently, it is quite probable that complex models fail at generalizing if trained inadequately. This dataset can, thus, be used to evaluate if the SSL method has produced rich embeddings, capturing enough information to supplement the lack of data in the downstream task.

6.3 Implementation

6.3.1 Preprocessing

Before delving into the details of the pretrain stage, data preprocessing is discussed. In particular, our method assumes that both pretrain and finetune use the same DA techniques. We have confirmed that mixing different DA in between the stages produces subpar results. As such, both use color jittering, grayscale conversion, and random rotations and flips.

Table 6.1: Hyperparameters tested during the self-supervised training, combining different Sequence Sizes (N) and Window Sizes (w). Resampling indicates that, in a single batch, all sequences come from the same video. Note that resampling only makes sense if N is smaller and multiple of the batch size.

Sequence Size	**Sequences per Batch**	**Window Size**	**Resample**	**AUC (%)**
9	8	3	No	93.51 ± 1.35
9	8	3	Yes	93.23 ± 1.78
9	8	6	No	93.49 ± 1.31
9	8	6	Yes	93.81 ± 2.12
18	4	3	No	93.68 ± 1.97
18	4	6	No	93.47 ± 1.11
18	4	6	Yes	92.91 ± 2.70
18	4	9	No	93.42 ± 1.62
18	4	9	Yes	93.62 ± 1.63
72	1	6	–	94.12 ± 1.35
72	**1**	**9**	**–**	$\mathbf{94.60 \pm 1.15}$
72	1	18	–	94.14 ± 2.12
72	1	32	–	94.53 ± 0.96

The data only consists of RGB channels and, if needed, images were resized to be 256 × 256 pixels by using bilinear interpolation without antialiasing. To eliminate any artifacts coming from specific datasets or videos, all images were processed through a circular mask with a radius of 128 pixels. This ensured that all pixels around the border were consistent and could not be used to identify images without using its internal structure.

DA techniques were only used during the training phase. Evaluation and inferences assume that images are not augmented in any way, except for resizing if the original size does not match the size used during training.

6.3.2 SSL Hyperparameter selection

The first step consisted in finding the optimal hyperparameter combination for the SSL method. To that end, the polyps dataset was used to evaluate how each parameter performs by means of a five-fold crossvalidation over randomly selected samples. Performance was evaluated using AUC computed from ROC. Note that this is not the same evaluation procedure that is used in the supervised setting, where whole videos are taken into account instead of random samples.

Initially, the upper bound of N was set to 72, which is the maximum that fits in our hardware's memory size. Next, as outlined in Table 6.1, all combinations of different window sizes W, N, and sources were tested. For instance, whether all sequences should come from a single video or from multiple was included in the tests.

All experiments were performed on an NVIDIA Titan Xp GPU and implemented in TensorFLow 2.4. The ResNet 50 used as the backbone for the network was pre-initialized with weights resulting from an Imagenet classification setting. All other layers, such as the projection layers, were randomly initialized. The model was trained for 2 hours and 30 minutes, which was roughly 21,000 batches with the highest $N = 72$. The learning rate was fixed to 0.1 and divided by 5 every 4,300 iterations. L2 weight decay was set to 0.0001, finding that any low value helped with regularizing the embeddings pre-projection (Chen et al., 2020a). The TL was trained using the batch all strategy with unnormalized embeddings and a margin of 0.2.

Table 6.1 demonstrates that increasing N, the sequence size, was beneficial, with a steady increase that capped at 72. Unlike lower settings, where both multi-video and single-video settings were tested, $N = 72$ implied that all images were in a single sequence, which inevitably meant that it used a single video as a source. It is worth mentioning, though, that improvements were sometimes found when using multiple videos. If enough hardware memory is available, readers and researchers are encouraged to try whether AUC increases in such cases.

While the best results for our particular task were achieved with single-video an $N = 72$, not all window sizes W achieved the same AUC. We found that using $W = 9$ introduced a good balance between the number of positive and negative samples, leaving enough hard positives and hard negatives to produce high-quality embeddings. In the same rhetoric, we believe that using a single sequence per batch, as opposed to multiple sequences, obtained better results due to not introducing as many easy negatives. Chances are that frames belonging to different sequences can be easily told apart, which makes the network not learn fine and minute differences.

The usage of projection layers was also evaluated in Table 6.2. Our findings were once again aligned with those of SimCLR, whereas introducing up to three dense layers during SSL greatly helped in obtaining better results (Chen et al., 2020a). In fact, the difference between using none and the best performing model was an increase in performance of 1.63%. Introducing more than three layers or adding more dimensionality was counterproductive, as the network used the new capacity to better model the data as opposed to the ResNet 50. As the projection layers are deleted during finetuning, this added capacity ended up hurting

Table 6.2: Study of the effect of adding several projection layers with a varying number of parameters. Each projection layer consists of a ReLU activation followed by a dense layer. All dense layers have the same amount of parameters (dimensionality).

Projection Layers	**Projection Dimensionality**	**AUC (%)**
0	–	92.97 ± 1.19
1	128	93.02 ± 1.39
2	128	94.09 ± 1.28
3	**128**	$\mathbf{94.60 \pm 1.15}$
3	256	93.56 ± 1.53
6	128	93.85 ± 1.80

the performance in downstream tasks. Thus, the best performance was obtained with 3 projection layers, each of 128 units.

To sum up, the exhaustive tests performed resulted in selecting $N = 72, W = 9$ and using 3 projection layers with 128 units. All the models used during this process were discarded; during the finetuning phase, all downstream tasks are trained directly from the SSL pretrained model.

6.4 Results

This section is divided into three. First, the embeddings produced by the SSL method are revised, and then the two supervised datasets used to evaluate the benefits of using SSL are exposed.

6.4.1 SSL embeddings

Given that the SSL method explored in this work is based on the proximity of two images within a sequence, it is expected that two images close together should have similar representations in the embedding space. Precisely, this is what the TL loss has been designed to do, thanks to the pseudo-labeling process and the introduction of the N, W parameters.

Figure 6.3 shows the eight closer embeddings to a randomly selected image, the anchor. Those eight candidates were selected by taking into account the euclidean distance in the embedding space. As shown, most of the closer embeddings pertain to images that were close to the anchor, which outlines the SSL process' capacity of extracting information useful to

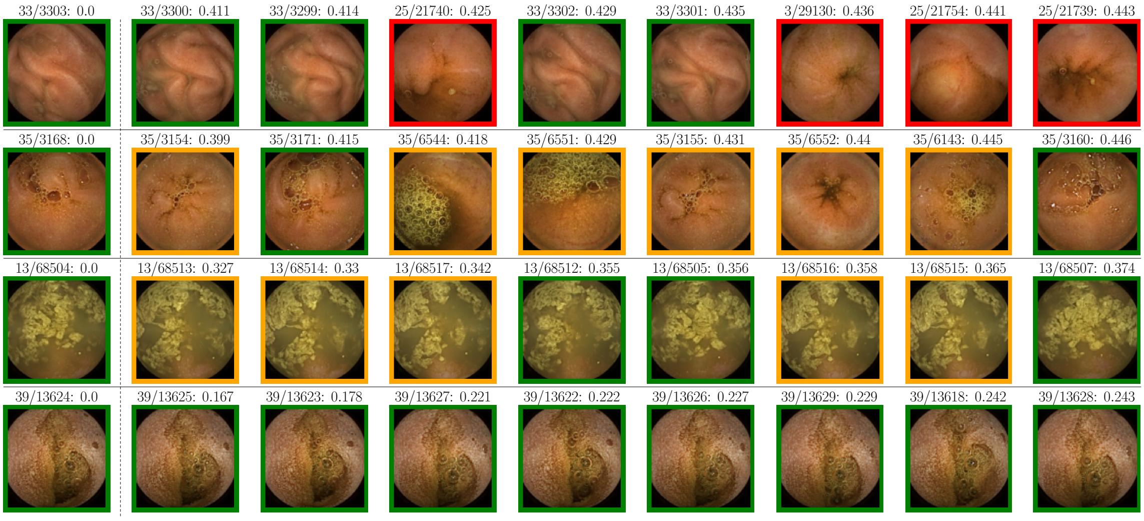
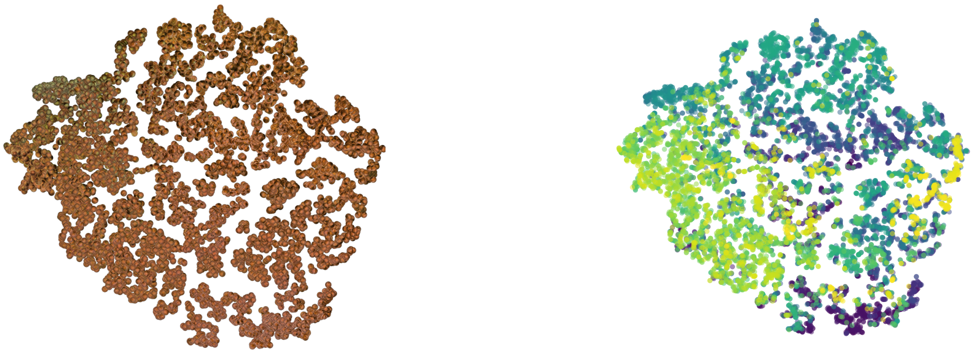


Figure 6.3: Given samples from the test set, shown in the first column, each row represents other samples in the set sampled by distance in the embedding space. Each image is titled as *video/frame: distance*, and framed in red if they come from a different video, orange if it is the same video, and green if, additionally to being in the same video, they are within w distance.

categorize which images are in close proximity. The network was able to find similar images even in different videos than those of the anchor, as seen in the first row, which shows that it did not simply learn which images are contiguous in a video. Likewise, the second row shows images that were far away in the same video, which again reinforces the notion that it learned based on features and not continuity. Overall, these results demonstrate that the model did overfit the samples' order. The model was capable of reasoning about the content and its structure, producing embeddings based on the image itself and not just the order. This is vital for SSL, as the embeddings must contain rich information about the image to be useful in downstream tasks.

Finally, the overall embeddings for a single video were analyzed through a t-Distributed Stochastic Neighbor Embedding (t-SNE) representation (Maaten and Hinton, 2008). This technique can project the embedding space, 2048 components, to a more manageable size. For instance, Figure 6.4 uses t-SNE to showcase the embeddings in a 2D projection of the space, which allows visual inspection of the results. First of all and most important, images that are similar–be it by color or structure–are close in the projected space, which is based on their embeddings, Figure 6.4a. This confirms the findings exposed so far, namely that the model indeed captured rich information. Secondly, and equally important, Figure 6.4b showcases the frames' order in the video. While retaining the whole order might not be important, it is useful to confirm that the images' neighbors mostly consist of frames that are close in the video. This can be seen by the smooth gradient of colors and overall similarity between one point and its nearest neighbors.



(a) Each embedding is represented with its corresponding image.

(b) Each embedding is colored according to its position in the video, following the *viridis* scheme. Images at the start of the video appear yellow, gradually turning purple as they get to the end.

Figure 6.4: t-SNE of the embeddings post-projections obtained from one WCE video after the pretrain phase. The representation shows (a) that visually alike images have close embeddings, and (b) that order is preserved.

6.4.2 Polyp dataset

This dataset was evaluated according to a previous publication which works with the same data. Laiz et al. (2020) proposed the use of ROC AUC to analyze the performance of their model, arguing that the predominant use of accuracy in the WCE field is flawed. Indeed, having more than 99% of the samples in a single class would make a classifier that only predicts this class seem unrealistically good. The effects of using accuracy are even more worrisome when one realizes this predominant class is the non-pathology one. Such a model, with more than 99% accuracy, would never detect a single polyp, having devastating effects in CADx systems.

Instead, AUC produces results that take into account the imbalance in data, and provide reliable metrics for the problem at hand. Likewise, Laiz et al. (2020) use Sensitivity at three different Specificity thresholds, 80%, 90%, and 95%, to provide more insight into the model's ability to correctly classify polyps. Moreover, these metrics are an indication of the amount

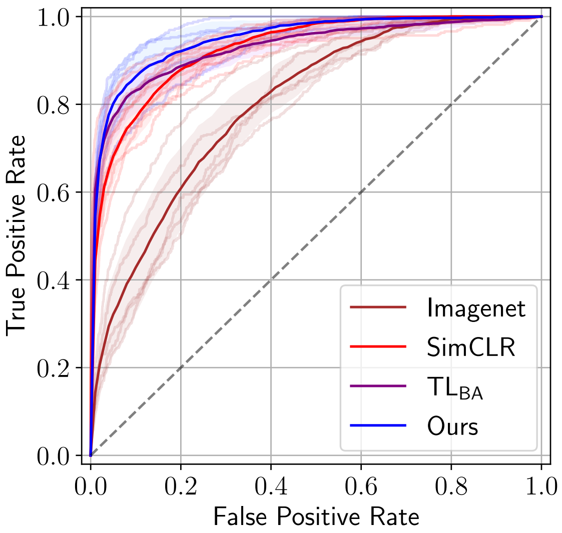


Figure 6.5: ROC curve for the four models tested for the polyp dataset. Each cross-validation split is shown in lighter versions of its corresponding model color, the mean ROC value is outlined in a darker color, and the standard deviation is provided as the background shade. True Positive Rate indicates the percentage of polyps correctly identified, while False Positive Rate is the percentage of non-polyps misclassified as polyps.

of work a physician would need to do to reach a certain detection rate.

The evaluation strategy is a five-fold crossvalidation, from which we report the mean and standard deviation for each metric. As is standard in WCE models, metrics were evaluated over all the frames in test videos and not over randomly selected samples. Also, following the procedure outlined in Laiz et al. (2020) and used in Chapter 5, the dataset's train and test spits are divided by patients instead of samples. This ensures that a single patient is found in only one of the sets at once, and prevents too similar images from being both trained and evaluated with. Failing to do so would produce overconfident results that would mislead towards its generalization ability.

All models compared in this section were trained on this exact same procedure and all share the same number of parameters, allowing for a fair comparison. A baseline, consisting of a ResNet 50 pretrained from imagenet, is provided as a means to compare against a basic result. Further, TL_{BA} from Laiz et al. (2020) was also compared against, which is a model that builds upon the same baseline by adding an additional TL before the classifier. Finally, we compared with the state-of-the-art model SimCLR (Chen et al., 2020a).

The results for polyp detection, as described, are outlined in Table 6.3. The proposed

Table 6.3: Performance comparison of several methods with the same parameter count. Imagenet refers to a ResNet 50 pretrained on the imagenet dataset and then finetuned with a cross-entropy loss over our dataset. SimCLR has been trained with NT-Xent as Chen et al. (2020a). $\mathrm{TL}_{\mathrm{BA}}$ is equivalent to Imagenet but trained with an additional triplet loss. Ours is the self-supervised network.

	AUC		**Sensitivity %**	
Model	**(%)**	**Spec. at 95%**	**Spec. at 90%**	**Spec. at 80%**
Imagenet	82.85 ± 5.72	37.75 ± 9.12	51.49 ± 11.09	66.71 ± 12.15
SimCLR (Chen et al., 2020a)	92.76 ± 1.62	68.13 ± 6.37	76.92 ± 5.40	87.91 ± 3.94
$\mathrm{TL}_{\mathrm{BA}}$ (Laiz et al., 2020)	92.94 ± 1.87	76.68 ± 4.93	82.86 ± 4.78	88.53 ± 3.76
Ours	$\mathbf{95.00 \pm 2.09}$	$\mathbf{80.16 \pm 6.97}$	$\mathbf{86.31 \pm 6.20}$	$\mathbf{92.09 \pm 4.63}$

SSL architecture outperformed all the other models, having a 12.15% more AUC than the baseline and 2.06% more than TL_{BA}, and reaching a total of $95.00 \pm 2.09\%$ AUC. The SSL method did indeed provide an edge towards the same model, with equal architecture, backbone network, and parameter count. Figure 6.5 provides more insight into the AUC ROC, which confirms that the network had higher true positive detections with a lower false positive rate. As can be seen, it improved over the other state-of-the-art SSL method, particularly at low false positive rates.

Also crucial were the improvements of Sensitivity at the three Specificity thresholds. As seen in Table 6.3, the proposed SSL method also outperformed all other models in these metrics. For instance, using our model a physician that checks only 20% of the data, discarding 80% of the negatives, would achieve a 92.09% polyp detection rate, which is an improvement of 3.56% over the next performing model.

Finally, we conducted a qualitative analysis of some inference results. Particularly, it is important for polyp detection to understand where the network fails to classify correctly. First, row a) in Figure 6.6 shows two cases where the model has incorrectly classified as polyps two non-pathology frames. We attributed this failure to the rugged texture and pinkish tones found in the images, which are usually associated with polyps. Similarly, row b) depicts two false negative samples. The network classified them as normal while they contained polyps, which have been circled to help the reader identify them. While the model incorrectly classified them, they are arguably two extremely difficult cases, where even a physician would have trouble detecting the anomalies.

6.4.3 CAD-CAP dataset

For the CAD-CAP dataset we followed the evaluation procedure outlined in the state-of-the-art publication for this particular set (Guo and Yuan, 2020). The samples are split into a four-split crossvalidation, reporting the mean and validation scores for the per-class Matthews correlation coefficient (MCC) and F1 scores, along with the overall precision, denoted as p_0, averaged among all three classes.

As reported in Table 6.4, a baseline composed of a ResNet 50 pretrained with Imagenet fails to correctly classify a large portion of the samples, achieving a low mean accuracy of only 69.98%, while the first model provided by Guo and Yuan (2020) already has a large advantage of 15.01% over it. At the same time, our model outperforms the ResNet baseline and all their baselines with a 92.77% accuracy. These gains were obtained solely by adding the SSL step to the ResNet baseline, without adding parameters or otherwise changing anything.

Notably, when comparing our model with the state-of-the-art model, SOTA in Table 6.4, our method reaches comparable results. While their implementation achieves 0.5% more accuracy, and in general, provides better MCC and F1 scores, it must be said that their

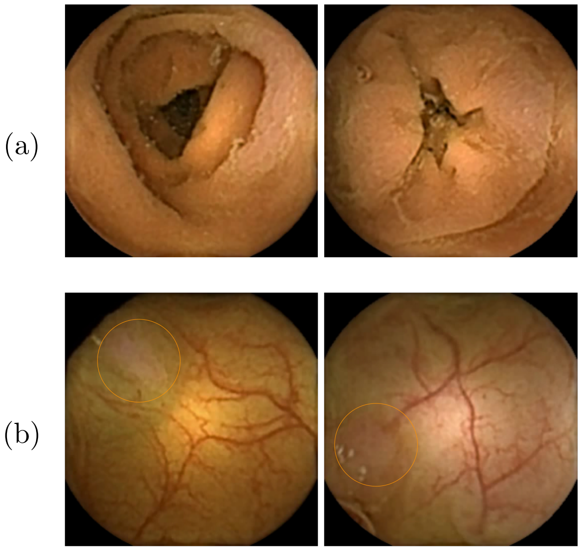


Figure 6.6: Random samples from the test set. Row a) shows two false positives, images inaccurately classified as polyps. Row b) depicts two false negatives. The polyps have been circled in orange to help with their identification.

Table 6.4: Per class and overall results of various methods in GIANA. ResNet is the same architecture as Ours but without the SSL step. Baseline 1 and 6 refer to the baselines reported by Guo and Yuan (2020), while the model with the same name is their semi-supervised performing implementation. Here $\mathbf{p_0}$ indicates the mean accuracy across all classes.

Method	**Class**	**F1-Score (%)**	**MCC (%)**	$\mathbf{p_0}$ **(%)**
ResNet	Normal	73.28 ± 3.57	60.58 ± 5.44	69.98 ± 1.35
	Inflammatory	65.19 ± 2.95	55.86 ± 1.77	
	Vascular	70.79 ± 4.60	65.35 ± 3.80	
Baseline 1 (Guo and Yuan, 2020)	Normal	94.92 ± 0.71	92.37 ± 1.07	84.99 ± 0.80
	Inflammatory	79.24 ± 1.55	68.72 ± 2.15	
	Vascular	80.75 ± 1.65	71.49 ± 2.57	
Baseline 6 (Guo and Yuan, 2020)	Normal	96.41 ± 0.84	94.61 ± 1.26	91.92 ± 1.71
	Inflammatory	88.98 ± 2.13	83.44 ± 3.24	
	Vascular	90.27 ± 2.78	85.75 ± 3.73	
Ours	Normal	95.00 ± 1.13	92.57 ± 1.66	92.77 ± 1.20
	Inflammatory	89.87 ± 1.65	84.99 ± 2.46	
	Vascular	90.26 ± 1.76	85.78 ± 2.37	
SOTA Guo and Yuan (2020)	Normal	97.41 ± 0.45	96.10 ± 0.69	93.17 ± 1.14
	Inflammatory	90.30 ± 1.56	85.43 ± 2.24	
	Vascular	91.69 ± 1.21	87.78 ± 2.06	

model was specifically hand-crafted for the CAD-CAP dataset. As such, they used semi-supervision with additional unlabeled data provided during the challenge, which we have not used for this task. Specifically, our model achieves similar results while not making use of the 1,807 additional images and providing a general framework for any downstream task.

6.5 Conclusions

In this project we have derived a method to enhance existing models without changing their architecture and keeping their number of parameters intact. This enables the creation of more robust predictors while working with the same amount of data, helping in tackling overfitting and failure to generalize.

The proposed SSL mechanism does not require of labeled data, which makes it suitable for any field where obtaining data might be easier than labeling it, as is WCE. The pseudo-labeling process is applicable to any stream of images, such as videos. It uses the similarity between close frames as a basis to extract important information from images.

Through this work we have demonstrated that SSL creates better models, specifically showing that the learned embeddings contain structural and rich information. This information is useful for downstream tasks, helping achieve new state-of-the-art results in polyp classification, with 95.00% ± 2.09 AUC, and building more reliable models in the CAD-CAP dataset.

Overall, we strongly believe this SSL method can be used to further reinforce the use of CADx systems, introducing more reliable models and thus raising the confidence in these kinds of automated systems. It is indeed proved that the time a physician would spend using models derived from this procedure would be lower, as they could discard a higher amount of negatives while achieving better detection rates.

The following chapter explores an improved version of this method, initially reported as future work, which takes even more advantage of the temporal axis. It would also be worth exploring how this method could be applied to other fields outside of WCE, be it medical or not.

This project has been published in Computers in Biology and Medicine (Pascual et al., 2022).

Chapter 7

Time-coherent Embeddings for Wireless Capsule Endoscopy

Contents

The previous chapter explored how SSL can be used to obtain more accurate and robust models by using unlabeled data. While it attempted to take some information from the temporal axis in WCE videos, it did so by imposing a hard-threshold in the relationship between images. This chapter explores an extension to the previous concept, lifting its limitations and allowing for a much refined control over the similitude metric. Overall, a novel loss that can produce better WCE SSL models is presented.

7.1 Approach

Previous attempts at using TL have mainly focused on the study of the margin parameter's role as a constant (Pascual et al., 2022; Laiz et al., 2020). Such usage either (1) groups all samples of the same class into a single group, where intra-class variance would be high, or (2) assumes that images close in a video are similar. However, with capture rates as low as 2 to 4 frames per second (Fernandez-Urien et al., 2014; Xavier et al., 2018), or as high as 35 (Figueiredo et al., 2011), it can be difficult to argue that a single constant can accurately capture where a frame stops being similar to its neighbors.

Instead of using a pre-defined margin parameter, this work aims to use a margin that dynamically changes according to the triplet. For instance, Zhou et al. (2020) created a variation of the TL, called ladder loss, which considered two margin parameters. One would be used to group items in the same primary group, while the other serves to model the distant relationship with the other group. We are, however, aiming to create virtually limitless margins. In fact, our parameter has to capture the relationship between an anchor, and its positive and negative pairs. That is, given a network $f(x)$ that learns embeddings $z^x = f(x)$ and an anchor-positive-negative triplet (x_a, x_p, x_n), we want to enforce the pair of embeddings z^a and z^p to be closer than the pair z^a and z^n.

Intuitively, given a sequence of images and a random anchor x_a, its most immediate neighbouring frame—x_i so that $d(x_i, x_a) = 1$, one at distance 1 from x_a—should be more similar than the any other closer one sitting at any greater distance. By this same rule, images at $d(x_i, x_a) = 2$ should be more similar than those at distance three or greater. However, unsurprisingly no strong guarantees can be given as the images get further away from the anchor.

Precisely this is what our modification of the TL, a novel contrastive loss called Relative Triplet Loss (RTL), imposes. Building from the idea of triplets (x_a, x_p, x_n) that TL proposes, our method redefines how triplets are formed by only requiring that $d(x_a, x_p) < d(x_a, x_n)$. In other words, it forms triplets where, relatively speaking, the anchor is more similar to the positive than the negative. An important aspect to understand is that unlike in TL, our loss does not use the term negative as dissimilar nor opposite. Negative just implies that the sample is not as similar to the anchor as the positive is. In fact, given a same anchor, an image can be both a positive and a negative sample in two different triplets, which is impossible with the traditional TL. Take for instance the sequence of x_i frames, where $0 \leq i \leq 10$. One valid triplet can be formed by (x_4, x_5, x_6), yet another one could be composed by taking (x_4, x_6, x_7). Here, given the same anchor x_4, it can be seen that x_6 is both a positive and a negative sample in the first and second triplets, respectively.

Our loss, thus, introduces a gradient of similarity and requires contrasting a sample with its triplet, instead of imposing a seemingly random constant.

Formally, the method starts by applying this same logic to the set of all possible combinations of (x_a, x_p, x_n) in an N-sequence, as shown in Equation (7.1). Extrapolating from TL and the intuitive concepts outlined before, relatively close images should have close representations.

$$\begin{gathered} d(z^a, z^p) < d(z^a, z^n) \ \forall x_p, x_n | d(x_a, x_p) < d(x_a, x_n) \\ x_p \neq x_a \end{gathered} \tag{7.1}$$

However, imposing this whole set of constraints to a DL model would make the model close to impossible to train. For this reason, instead of attempting to directly use them, the equations are rewritten in Equation (7.1) to a TL-like formulation. The original TL, as seen in Equation (7.2), uses the constant α to model the margin, while our RTL loss, in Equation (7.3), substitutes it for the expression $\delta\alpha_{(x_a,x_p,x_n)}$. As can be seen in Equation (7.4), α is now an always-positive computation that adapts based on the triplet elements' distances, while δ can be used to scale down the product if necessary, such as use-cases that employ a normalized embedding space.

$$TL = \max(d(z^a, z^p) - d(z^a, z^n) + \alpha, 0) \tag{7.2}$$

In fact, upon close examination, it can be observed that α gets bigger the further away the negative sample is from the anchor. The best-case scenario, where all samples are as close as possible, reduces to $\alpha = 1$, whereas for all other cases, it maximizes the margin whenever the negative or positive is too far away. Returning to the previous N-sequence example, the margin parameter for (x_4, x_5, x_6) would be $\alpha_{(x_4,x_5,x_6)} = 2 - 1 = 1$, whereas choosing any other further negative would result in an increase of the value, $\alpha_{(x_4,x_6,x_{7+|i|})} = (3 + |i|) - 2 = 1 + i > 1$.

$$RTL = \max(d(z^a, z^p) - d(z^a, z^n) + \delta\alpha_{(x_a,x_p,x_n)}, 0) \tag{7.3}$$

$$\alpha_{(x_a,x_p,x_n)} = d(x_a, x_n) - d(x_a, x_p) \tag{7.4}$$

As briefly mentioned before, long sequences could cause unwanted behavior. As samples are drawn further away from an anchor, the uncertainty in regards to their similarity grows;

we can no longer guarantee whether they will be similar or not. For instance, take the triplet $(x_{1000}, x_1, x_{10000})$. It is impossible to known—without manual intervention, of course—if they are similar or not. To prevent this kind of issues two solutions are derived.

First, one viable way to tackle the problem would be to limit the sampling mechanism. Without introducing any changes to the previous formulation, the data-sampling algorithm should only generate triplets where the anchor and its pairs conform to a maximum distance. While this method imposes a hard-limit, it is not by any means equal to the limitations that come with TL. The inner-workings would still be relative, not constant, and only the amount of combinations would be limited.

Secondly, another way to model the uncertainty would be by pondering our novel loss by the triplet inherent uncertainty. In fact, this method allows completely forgoing the margin parameter. The max operation can be changed for a softplus activation $(\log(1 + \exp(\cdot)))$, Equation (7.5), and the loss contribution is modelled to be exponentially smaller as the samples get further away from the anchor, Equation (7.6). Here RTL_{soft} denotes the softplus version of RTL.

$$RTL_{soft} = \gamma^{-1}_{(x_a, x_p, x_n)} \log(1 + \exp(d(z^a, z^p) - d(z^a, z^n))) \tag{7.5}$$

$$\gamma_{(x_a, x_p, x_n)} = (d(x_a, x_n) - 1) \cdot d(x_a, x_p) \tag{7.6}$$

Further combinations can also be created by merging Equation (7.3) and Equation (7.5). For instance, Equation (7.7) shows a formulation that penalizes uncertain embeddings with the γ term while keeping the max-margin calculus. A developer or user could also choose to add restrictions to the sampling mechanism while using this equation, effectively combining all the possible alternatives.

$$RTL_{mix} = \gamma^{-1}_{(x_a, x_p, x_n)} \max(d(z^a, z^p) - d(z^a, z^n) + \delta\alpha_{(x_a, x_p, x_n)}, 0) \tag{7.7}$$

7.2 Datasets

7.2.1 Self-supervised dataset

This work employs the same unlabeled dataset as Chapter 6, keeping all 49 videos and 1,185,033 with the same selection criteria. Using the same dataset allows us to perform fair

Table 7.1: CrohnIPI dataset description, grouped in both two classes (N/NP) and the whole seven subclasses. Note that Percentage of samples (%) is rounded for display purposes.

Class			Number of samples	Percentage of samples (%)
Non-Pathologic (61%)	N	Normal	2124	61.0
Pathologic (39%)	S	Stenosis	130	3.7
	U10	Ulceration 10mm	297	8.5
	U3-10	3-10mm ulceration	408	11.7
	AU	Aphthoid ulceration	251	7.2
	O	Edema	149	4.3
	E	Erythema	125	3.6

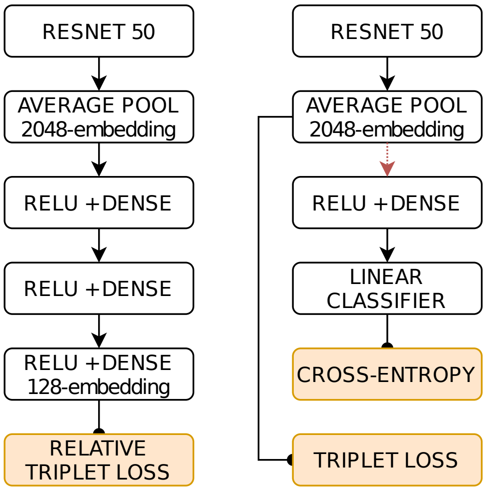
comparisons with respect to the previous SSL model, but at the same time it is the perfect candidate due to the nature of the data and its generous quantity.

7.2.2 CrohnIPI

After the pretaining phase, the model is finetuned with a specific downstream task. We have chosen the ChronIPI dataset, which is composed of 3484 WCE labeled images divided between non-pathology (NP) and pathology (P) (de Maissin et al., 2021). The latter one being subdivided into 6 separate instances of pathology. The labels were chosen according to three independent experts, gathering only those images for which at least two of the experts were concordant. An overview of the dataset composition can be seen in Table 7.1, including the general NP/P classes and their detailed groups.

The first challenge this dataset poses is its low amount of samples. In its binary classification setting the per-class representation can be said to be balanced, yet the intra-class variance is high. That is, images containing pathologies can be very varied from one another, which poses a significant obstacle for DL models. As such, this classification task can be useful to evaluate if the model can learn minute differences within the same class, and to compare its performance with the baselines given by the dataset's authors.

On the other hand, multiclass classification with the seven classes outlines a different challenge. It can be seen in Table 7.1 that the least represented class, erythema, barely represents the 3.6% of the data. In other words, the dataset is heavily unbalanced with some classes being severely underrepresented. Multiclass classification can be used to verify that the approach outlined in this chapter outperforms other models and, in particular, our previous SSL method.



(a) Pretrain. (b) Finetune.

Figure 7.1: Proposed architecture for both training stages. Finetune starts from the parameters learned during pretrain. The dotted red arrow indicated that no gradient flows through the classifier.

7.3 Implementation

Architecture-wise, as can be seen in Figure 7.1, we have chosen to keep the same number of parameters and layers as seen in Chapter 6. As demonstrated, using projection layers is beneficial during the SSL training, as it gives the network more freedom in the higher-order embeddings (Figure 7.1a). This, at the same time, reverberates in the finetune stage, where the model performs better in the downstream task after removing these layers (Figure 7.1b).

Overall, the only change with respect to the older architecture resides in the loss function. Where the old model employed a TL in the pretrain stage, this model substitutes it for the proposed RTL. It is important to consider that this loss is only used during SSL, as the downstream tasks do not consist of whole videos, thus sequences cannot be built by using the proposed method.

7.4 Results

All results were evaluated according to the method proposed by the CrohnIPI authors. The dataset was split into five distinct folds following the schema provided, where each patient is only found in a single set at the same time. Cross-validation was performed over the five sets, which allowed us to report the mean and standard deviation for all measures taken. It is noteworthy to say that the dataset authors did not provide such statistics for their evaluations even though they used same cross-validation strategy. As explained in Section 7.2, both binary and multiclass settings were explored.

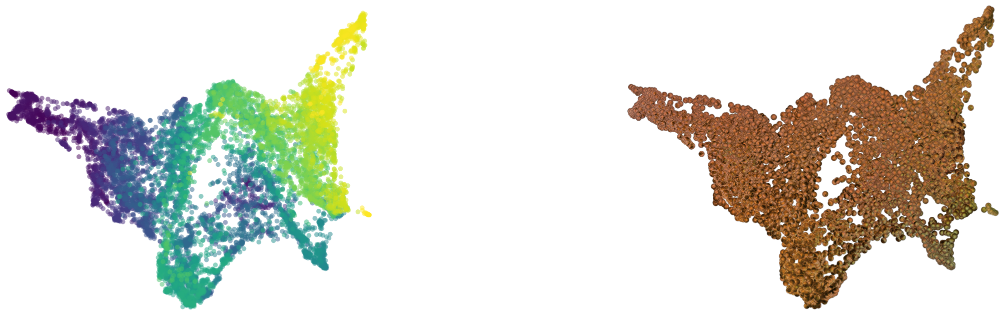
This section is divided into the SSL phase and the finetuning one. During the first one, the quality of the RTL learned embeddings is evaluated. In the second one, the embeddings are used for particular downstream tasks and the method is compared as a whole against other methods.

7.4.1 Self-supervised learning

The network was trained with the 49 videos from the unlabeled dataset, downsampled to 256×256 pixels from 320×320. Some standard data-augmentation techniques were applied per-image, namely random rotations and flips, and erasing any noise or video-identifying patterns from the borders through a mask of 128-pixels radius.

An Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018) representation of one whole video's embeddings was used to verify if the embeddings were coherent. Figure 7.2b clearly shows that similar images were grouped close together in the projected space. This is a clear indication that the network, and our proposed loss, properly learned how to group embeddings based on similarity. In fact, Figure 7.2a shows that the distribution of the samples is even consistent with the order in the video. A gradient of colors following the *viridis* colormap indicates that those images close in the video have similar embeddings. Overall, both images together confirm that the embeddings contain rich information that is not a product of overfitting, as the model is capable of preserving both order and similarity.

To provide some more insight into the embeddings' ability to preserve order and similarity, we produced a visualization that plots the distance of an anchor frame with respect to the rest of the video, as shown in Figure 7.3. We provide two comparisons, each taking the first and last frames considered in CADx systems—the entry and exit of the small bowel—as the respective anchors. Figure 7.3a compares the entry point with the rest of the video, while Figure 7.3b compares it from the other way around. Once again, it can be clearly seen that the distance steadily increases as the frames get further away. Importantly, as can be seen from the constant spikes as opposed to a completely linear relationship, distance is not only based on the position in the video but actual image resemblance is also kept into



(a) Viridis mapped transformation showing the order within the video.

(b) Reduction with ground truth images to showcase visual similarity.

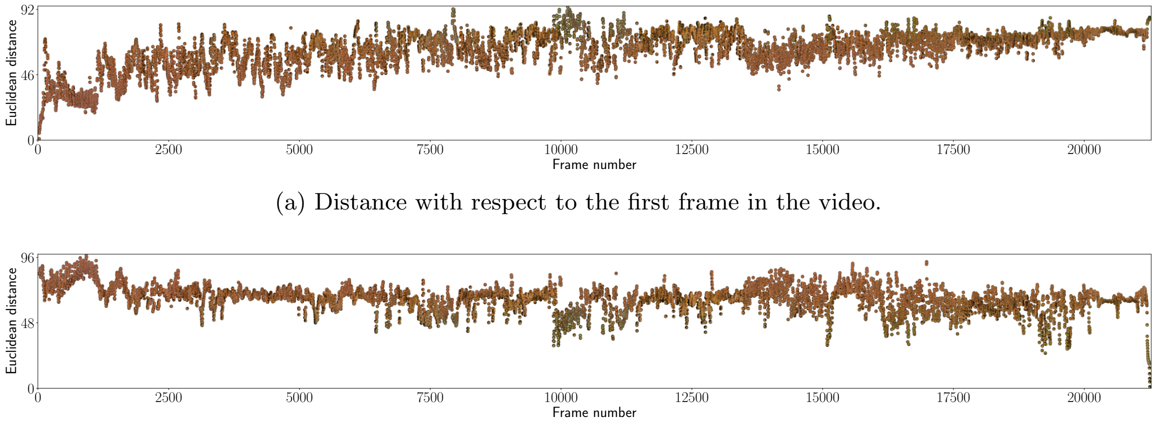
Figure 7.2: UMAP reductions of the 2048-sized embedding for a given WCE video after our SSL method has been applied in the pretrain phase.

the equation.

7.4.2 CrohnIPI - Two classes

The two-classes dataset was used to perform an ablation test of our proposed method. We evaluated all possible combinations of our RTL as well as versions with and without TL during the finetuning phase. The models without TL were not allowed to change the embeddings during their training. Table 7.2 outlines the results obtained after considering all possible combinations. As can be seen, the results include both the RTL and RTL_{soft} definitions and combines them with all the aforementioned variations, such as adding γ decay and limiting the maximum distance considered for triplets. Ultimately, results show that the best performing model was obtained with the variable margin RTL when considering only 9 frames around the anchor. That model in particular was trained with a normalized embedding space and fixing δ to 0.05. Based on the experiments performed we deduced that introducing decay into the best model did not further improve the results. A possible explanation for that behavior could be that the maximum distance being set at a low number already limits the number of uncertain triplets. Other downstream tasks and applications should properly check if the same conditions apply.

For completeness, Table 7.2 also reports the results obtained with a baseline model trained on the Imagenet dataset. In particular, it is a ResNet-50 model with the exact same number of parameters and directly finetuned on the CrohnIPI dataset. Notably, our



(a) Distance with respect to the first frame in the video.

(b) Distance with respect to the last frame in the video.

Figure 7.3: Distance, calculated as the euclidean distance between embeddings, of every frame in a video with respect to the (a) first and (b) last frames of that same video.

model was capable of surpassing the models provided by the CrohnIPI dataset's creators, the baseline, and the previous SSL implementation. When considering models without TL during the finetuning stage, it is of utmost importance that our model outperformed the baseline, as it indicates that the embeddings learned by RTL are much more useful than those of Imagenet, which are typically used as a start point in most WCE models.

Nevertheless, our method is still superior when TL is included in the finetune process, achieving better results than the other models. Table 7.2 might be misinterpreted, as it shows that model offers a degraded Sensitivity. However, it must be taken into consideration that metrics such as Specificity and Sensitivity, when considered along, are affected by data-unbalances. Table 7.2 also shows the results obtained for Sensitivity at different thresholds, which is a metric that is more robust toward unbalances and, at the same time, provides more insights. Considering Sensitivity at different Specificity levels $(80\%, 90\%, 95\%)$ can be translated to analyzing how many images a physician should check to detect a certain number of positives. Potentially, when building a CADx system we want to maximize the Sensitivity with respect to the number of images seen, and this is precisely what this metric measures. As can be seen, when discarding 95% of the images based on specificity, a 95.26% of all pathologies are detected. This number increases to 99.30% when removing only 80% of the data. Clearly, our proposed loss and method obtains better results than any other SSL system or model.

Table 7.2: Overview of the results obtained in the two-classes (N/NP) variant of the CrohnIPI dataset. Results are reported as mean and standard variation resulting of a 5-fold cross-validation. The models marked with * follow the same splits, but do not report these statistics. The models marked with † are evaluated on an older version of the database.

	Accuracy	**Specificity**	**Sensitivity**		**Sensitivity (%)**	
Model	**(%)**	**(%)**	**(%)**	**Spec. at 95%**	**Spec. at 90%**	**Spec. at 80%**
Vallée et al.*†(Vallée et al., 2019)	90.85	91.47	90.22	–	–	–
Vallée et al.*†(Vallée et al., 2020)	94.56	–	92.39	–	–	–
Maissin et al.*(de Maissin et al., 2021)	92.48	95.24	88.16	–	–	–
Imagenet Without TL	80.80 ± 1.05	95.12 ± 1.93	58.26 ± 5.68	10.64 ± 21.29	37.29 ± 30.61	78.01 ± 4.74
Imagenet	94.34 ± 1.94	96.81 ± 2.53	90.52 ± 6.16	57.47 ± 46.95	97.35 ± 0.95	98.39 ± 0.90
Pascual et al.(Pascual et al., 2022) Without TL	67.17 ± 2.69	52.81 ± 3.46	89.66 ± 2.21	39.11 ± 6.96	51.97 ± 6.67	67.51 ± 5.69
Pascual et al.(Pascual et al., 2022)	94.20 ± 1.34	92.57 ± 2.06	**96.79 ± 1.94**	94.04 ± 2.87	96.27 ± 1.57	96.27 ± 1.57
RTL Without TL	86.97 ± 0.91	89.34 ± 1.68	83.21 ± 1.51	68.29 ± 3.70	82.73 ± 2.41	91.33 ± 0.92
RTL_{soft} Softplus with decay	94.78 ± 0.85	96.96 ± 1.31	91.29 ± 2.29	94.34 ± 2.14	97.60 ± 0.79	99.11 ± 0.22
RTL_{soft} Max. dist. 9, softplus, no decay	94.80 ± 1.12	97.26 ± 0.74	90.97 ± 3.51	76.46 ± 38.24	97.73 ± 0.97	99.01 ± 0.31
RTL Margin with decay	94.66 ± 0.85	97.49 ± 0.49	90.24 ± 1.68	95.13 ± 1.99	97.50 ± 0.93	99.20 ± 0.25
RTL Max. dist. 9, margin, no decay	**95.38 ± 0.68**	**97.55 ± 0.72**	92.00 ± 1.82	**95.26 ± 1.72**	**98.21 ± 0.44**	**99.30 ± 0.35**

7.4.3 CrohnIPI - Seven classes

To evaluate the multiclass dataset we chose the F1 metric Sasaki (2015). Unlike accuracy and other metrics that rely on pre-defined thresholds, F1 is robust against data imbalances, which makes it the perfect candidate for this dataset. It is defined per-class and in terms of both precision and recall. The evaluation was performed using the same models as in the binary classification setting except for the models provided by CrohnIPI's authors; who did not perform this experiment. As already mentioned earlier, a baseline named Imagenet is provided, which consists of model trained with an equal architecture and parameter count as our proposed method, but finetuned from Imagenet.

The results can be seen in Figure 7.4, where each model is evaluated both with and without TL. As can be observed, the results are consistent with the ones observed before. Out of all the models without TL, our model outperforms the rest and does a particularly outstanding job in the most unbalanced class, erythema (E). When considering TL, both the Imagenet and Pascual et al. (2022) perform similarly, while our proposed method achieves a better or comparable score in all categories. Notably, in erythema detection our model achieves 14% more F1 over Pascual et al. (2022). Overall, it can be said that *RTL* makes a substantial difference in battling class imbalance, demonstrating the potential of our approach for battling class unbalances.

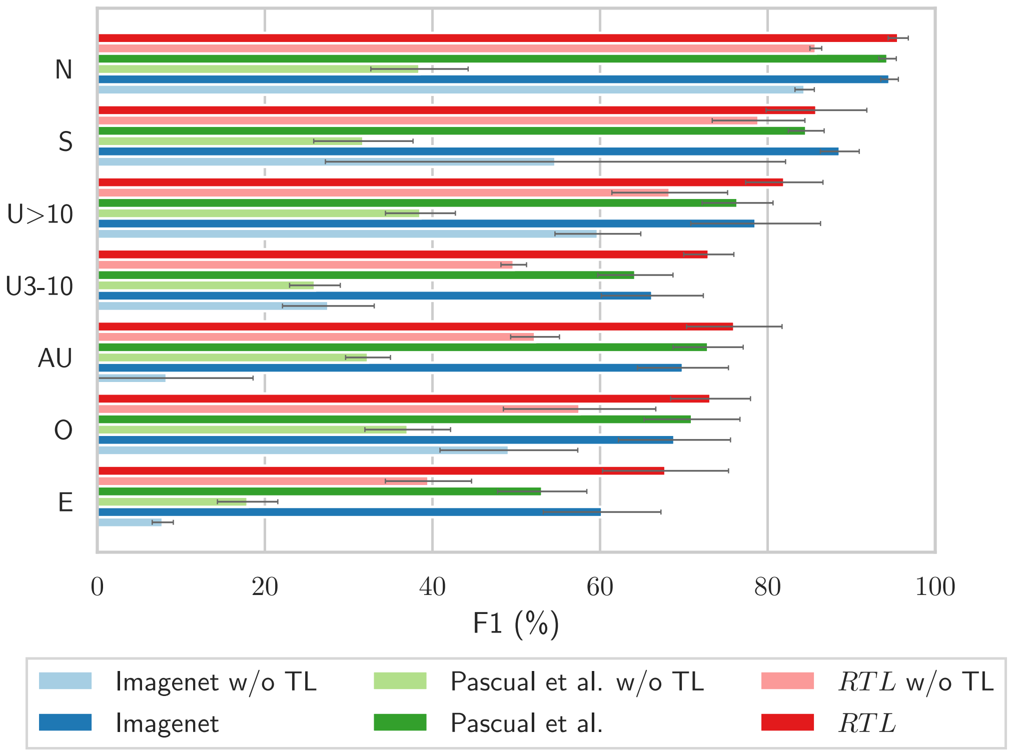


Figure 7.4: Per-class F1 score obtained from 5-fold cross-validation. Standard deviations are reported as black lines. Imagenet is model as described in the finetune architecture, with the ResNet-50 trained from an Imagenet pretrain instead of our SSL. Pascual et al. is the model by Pascual et al. (2022). Here *w/o TL* indicates the model does not use Tripet Loss during the finetune phase.

7.5 Conclusions

In this chapter, we proposed a new SSL method designed to extract the inherent information from the temporal axis in WCE videos. The method presents a new approach to similarity-based contrastive learning, where triplets are selected based on the relative distance between the anchor and the positive and negative samples.

The experiments demonstrated that our approach is capable of outperforming the traditional finetuning process, which starts from Imagenet, even without using an additional TL during finetuning tasks. Likewise, adding the TL yields superior results in both binary and multiclass classification datasets. Through quantitative and qualitative analysis it can be concluded that our method is capable of learning rich embeddings. These models could be deployed into CADx systems and would successfully reduce the time required to revise WCE videos while detecting more pathologies.

This work has been published in the International Conference in Pattern Recognition 2022. Currently, it has already been presented and it is due to be published in the proceedings.

Chapter 8

Conclusions

Contents

In this chapter, the final conclusions of the thesis are exposed. First, the contributions are contrasted with respect to the original objectives raised in Section 1.3. Then, the contributions made throughout the thesis are summarized on a chapter basis. Finally, some future work in relation to the previous sections is proposed.

It can be firmly said that the works published during this thesis have all been progressively exploring and advancing in the respective three objectives that were set for this academic period.

1. **Produce certainty-aware models.** Through uncertainty we have created a self-refining model, capable of improving its own predictions based on intermediate uncertainty values. Moreover, the final uncertainty is presented to the user.

2. **Create context-aware models.** RNNs have been used both for satellite images and WCE datasets to take into account the context. The former use-case leverages previous road-information to determine the next steps, while the latter looks ahead and behind in a WCE video to determine if an image contains polyps. Both models demonstrate that temporal information can be used as context through RNNs.

3. **Create methods to tackle data unavailability.** Lack of labeled data is battled by using SSL, a variant of supervised learning that can take into account unlabeled

data. The aim is to obtain models that obtain better results than their standard counterparts, and that, by extension, are more resilient towards overfitting of the most-represented classes.

4. **Apply the above methods to real-world cases.** Each of the topics was successfully applied to a dataset. Uncertainty has successfully been applied to land cover segmentation, obtaining useful information both for the network and the user. RNNs were used for road extraction and successfully employed to extract context from WCE data. And, finally, SSL has been demonstrated to work with WCE datasets that suffer from unbalances and from lack of data.

8.1 Summary of Contributions

In Chapter 3 we introduced a novel approach to use uncertainty in the context of semantic segmentation for computer vision. Our model is able to calculate bounds for uncertainty, which serves two purposes. First, it internally uses this uncertainty information to update the representations of the final segmentation. By iteratively refining the pixels that are more uncertain the model is able to obtain better results. In fact, we demonstrate that our approach surpasses other models that, albeit similar architecture-wise, ignore uncertainty. Equally important, the model is able to present this uncertainty information to the user. This additional output can create a positive feedback loop where the model can be further refined after manual intervention. Likewise, the heatmap output can provide confidence with the classification and empower the user to trust the network.

Chapter 4 explores further applications in the satellite image domain. The proposed method seeks to incorporate RNNs into the task of road-graph modeling, arguing that contextual information can be useful to create better graphs. The model is a mix between traditional graph exploring algorithms and ANNs which delegates the decision to explore to the ANN model. Results demonstrate that the architecture is viable and obtains comparable results to a CNN that is fed with the previous outputs.

In Chapter 5, we first presented an initial work in WCE, which both serves as an introduction to the common pitfalls in the field and as an initial explanation of the databases obtained through pillcams. Conveniently, this work is a bridge that unites the previous chapters in RNNs with the following ones in WCE. It aims to prove that traditional WCE models can greatly benefit from RNNs, as their nature ensures that image classification is not done in isolation. On the contrary, the proposed method uses the context around the images that were deemed the hardest to classify according to a pre-existing CNN model.

Through multiple experiments we demonstrated that BLSTM layers are instrumental in taking advantage of the temporal axis. Furthermore, a cross-validation strategy was used to prove that the model developed outperforms other models available in the literature.

Furthermore, SSL has been introduced in Chapter 6 as a means to compensate for the lack of data in the WCE field. We exhibited that TL can be used to generate rich embeddings based on the temporal axis of videos. To such end, sequences of contiguous images were formed and the network was trained to group samples by similarity. To demonstrate that the embeddings are indeed rich, they were visualized through a t-SNE projection. After verifying that they were correct, we applied them to two different downstream tasks. First, we achieved state-of-the-art results in polyp detection, which showed that our method is resilient against data unbalances. Following, the CAD-CAP dataset was used to verify that the model was also robust against low amounts of data.

Lastly, Chapter 7 deeps further into SSL applied to WCE videos. These methods build from the one presented in Chapter 6, where videos were divided into sequences and TL was used to find information-rich embeddings. Here we proved that allowing the triplets to be formed based on relative similarity, instead of having a predefined threshold, generates embeddings that are even more useful for downstream tasks. UMAP projections and distance-based plots verified that the model had learned to differentiate similar images while preserving enough contextual information about the order in the video. Further experiments with a Crohn-based dataset demonstrated that the embeddings were capable of generalization and suitable for both binary and multiclass classification. They proved to be useful to combat extreme unbalances and small datasets, and tests showed that this SSL technique can outperform the typical finetune setting based on the Imagenet dataset. Our method obtained state-of-the-art results in all domain-specific tasks tested and surpassed both the baseline and our previous model.

8.2 Future Work

This thesis has proposed methods to tackle the fields of uncertainty, interpretability, and augmented learning from the context of unlabeled data. This section aims to provide pointers towards possible improvements and open questions in the above implementations with the objective of making the presented works easy to continue and extend.

For instance, uncertainty has been applied as a means to iteratively improve satellite image segmentation. The method could easily be expanded to any other segmentation architecture, as it is not bound or limited to a specific segmentation domain. We strongly believe that the same method could be applied to virtually any architecture that employs deep supervision and not necessarily only to semantic segmentation. Thus, we propose to investigate whether uncertainty could be used to train better models and provide the user with meaningful and accurate confidence information. With regards to the latter, we employed heatmaps to make uncertainty human readable, thus an equivalent interpretation should also be researched.

Extracting contextual information from road network extraction through RNNs demonstrated that there are benefits to be explored. We achieved smaller and simpler architectures, with fewer inputs and preconditions during inference, yet the question of accuracy remains open. Currently, through stateful RNNs there are potentially infinite paths to explore, each of them maintaining a full chain of probabilities. If no hardware restrictions applied, a network could automatically expand a whole graph without the need of an auxiliary graph exploration algorithm. As such, a path could be expressed as the conditional probability of one point x_i given the previous ones $P(x_i|x_{i-1},...,x_1)$. This formulation allows to disregard the naive approach of taking a point along the most probable angle at each timestep, and instead borrow a popular NLP inspired method: beam search (Bahdanau et al., 2014).

Beam search takes a mixed approach between fully expanding all paths from a given point and naively taking the most-probable candidate independently. Computationally and time-wise, we can not afford to find the best path (the one that maximizes the overall probability) for each point, as it would imply expanding too many paths. Arguably, naively selecting the most probable node at each timestep is neither a good solution to obtain a good final product. Beam search does a k-step look-ahead by expanding the path up to k nodes, which then uses to select the most probable continuation taking into account all k steps.

Tightly related to beam search, and also mostly used in NLP, we also propose the borrow perplexity from Jelinek et al. (1977), which is often used as a measure of how difficult a language model is, or how well we can model the language. We propose to adopt it as a measure of how difficult a path is and, generalizing to the whole graph, to use it as a measure of the roads graph's quality.

Moving forward, RNNs were applied to WCE as a S2S classification task, meaning that each frame in a sequence of images was simultaneously classified via a BLSTM layer. Aside from improvements related to pathology detection, possible future work includes modifying the loss function or even the architecture so that results are coherent. As shown in Section 5.5, there are instances where the network suddenly fails to detect a polyp frame located between two polyp frames. Logically speaking, the probability of such an occurrence taking into account the nature of pillcams is extremely low. The network should be able to reach this same understanding and use it to improve the results and inform the user when no conclusive decision can be taken. It would also be interesting to analyze how this BLSTM network compares with the recent improvements in Transformer networks, and whereas the latter produces outputs that are more coherent with their surroundings.

Finally, the last two chapters of the thesis derived methods to work with unlabeled data through SSL. In fact, RTL is already a product derived from the first implementation of SSL for WCE, thus this section focuses on improvements and ideas for RTL. RTL formulation comes from the infeasibility of having a whole set of constraints on the relative similarity between samples, having to simplify the training process by introducing a TL-like formula. While we believe the limitations are still present, and will be for a long time, there is a wide range of losses other than TL that could be explored. Using or creating another function capable of better expressing the relative nature of our triplets would certainly produce even richer and more powerful embeddings. Another question that remains unanswered, though we are sure is viable, is applying RTL to fields other than WCE. For instance, we can see untapped potential in action recognition, where obtaining samples might be straightforward but, as in WCE, labeling them is much harder.

Appendix A

Research Outcome

This thesis has led to the publications summarized below:

Journal Submissions

- S. Seguí, M. Drozdzal, G. Pascual, P. Radeva, C. Malagelada, F. Azpiroz, and J. Vitrià. Generic feature learning for wireless capsule endoscopy analysis. Computers in Biology and Medicine, 79:163–172, 12 2016. ISSN 18790534. doi: 10.1016/j.compbiomed.2016.10.011

- G. Pascual, P. Laiz, A. García, H. Wenzek, J. Vitrià, and S. Seguí. Time-based self-supervised learning for wireless capsule endoscopy. Computers in Biology and Medicine, 146:105631, 7 2022. ISSN 0010-4825. doi: 10.1016/J.COMPBIOMED.2022.105631.

- J. Reuss, G. Pascual, H. Wenzek, and S. Seguí. Sequential models for endoluminal image classification. Diagnostics, 12:501, 2 2022. ISSN 20754418. doi: 10.3390/diagnostics12020501.

International Conferences

- G. Pascual, S. Seguí, and J. Vitrià. Uncertainty gated network for land cover segmentation. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2018-June, 2018. ISBN 9781538661000. doi: 10.1109/CVPRW.2018.00052.

- G. Pascual, S. Seguí, and J. Vitrià. Time-coherent embeddings for Wireless Capsule Endoscopy. International Conference in Pattern Recognition, 2022. The publication has been presented and is awaiting publication.

Bibliography

I. Achituve, H. Maron, and G. Chechik. Self-supervised learning for domain adaptation on point-clouds. *Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021*, pages 123–133, 3 2020. doi: 10.48550/arxiv.2003.12641.

S. Aich, W. van der Kamp, and I. Stavness. Semantic binary segmentation using convolutional networks without decoders. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 182–1824, 2018. doi: 10.1109/CVPRW.2018.00032.

A. Akay and H. Hess. Deep learning: Current and emerging applications in medicine and technology. *IEEE Journal of Biomedical and Health Informatics*, 23:906–920, 5 2019. ISSN 21682194. doi: 10.1109/JBHI.2019.2894713.

L. A. Alexandre, J. Casteleiro, and N. Nobre. Polyp detection in endoscopic video using svms. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4702 LNAI:358–365, 2007. ISBN 9783540749752. doi: 10.1007/978-3-540-74976-9_34.

D. G. Altman and M. J. Gardner. Calculating confidence intervals for regression and correlation. *British medical journal (Clinical research ed.)*, 296:1238–1242, 1988. ISSN 0267-0623. doi: 10.1136/BMJ.296.6631.1238.

J. V. Amersfoort, L. Smith, Y. W. The, and Y. Gal. Uncertainty estimation using a single deep deterministic neural network. *37th International Conference on Machine Learning, ICML 2020*, PartF168147-13:9632–9642, 3 2020. doi: 10.48550/arxiv.2003.02037.

D. Amodei, C. Olah, G. Brain, J. Steinhardt, P. Christiano, J. Schulman, O. Dan, and M. G. Brain. Concrete problems in ai safety. 6 2016. doi: 10.48550/arxiv.1606.06565.

T. Aoki, A. Yamada, K. Aoyama, H. Saito, A. Tsuboi, A. Nakada, R. Niikura, M. Fujishiro, S. Oka, S. Ishihara, T. Matsuda, S. Tanaka, K. Koike, and T. Tada. Automatic detection of erosions and ulcerations in wireless capsule endoscopy images based on a deep convolutional neural network. *Gastrointestinal Endoscopy*, 89:357–363.e2, 2 2019. ISSN 10976779. doi: 10.1016/j.gie.2018.10.027.

O. Attallah and M. Sharkas. Gastro-cadx: a three stages framework for diagnosing gastrointestinal diseases. *PeerJ Computer Science*, 7:1–36, 3 2021. ISSN 23765992. doi: 10.7717/peerj-cs.423.

S. M. Azimi, P. Fischer, M. Korner, and P. Reinartz. Aerial lanenet: Lane-marking semantic segmentation in aerial imagery using wavelet-enhanced cost-sensitive symmetric fully convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–20, 2018. ISSN 01962892. doi: 10.1109/TGRS.2018.2878510.

S. Azizi, B. Mustafa, F. Ryan, Z. Beaver, J. Freyberg, J. Deaton, A. Loh, A. Karthikesalingam, S. Kornblith, T. Chen, V. Natarajan, and M. Norouzi. Big self-supervised models advance medical image classification. 1 2021.

V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:2481–2495, 11 2015. ISSN 01628828. doi: 10.48550/arxiv.1511.00561.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, pages 1–15, 2014. ISSN 0147-006X. doi: 10.1146/annurev.neuro.26.041002.131047.

R. Bajcsy and M. Tavakoli. Computer recognition of roads from satellite pictures. *IEEE Transactions on Systems, Man and Cybernetics*, 6:623–637, 1976. ISSN 21682909. doi: 10.1109/TSMC.1976.4309568.

D. Balduzzi, M. Frean, L. Leary, J. P. Lewis, K. W. D. Ma, and B. McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? *34th International Conference on Machine Learning, ICML 2017*, 1:536–549, 2 2017. ISBN 9781510855144. doi: 10.48550/arxiv.1702.08591.

A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE TRANSACTIONS ON INFORMAI10N THEORY*, 39, 1993.

F. Bastani, S. He, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, and S. Madden. Machine-assisted map editing. *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL '18*, pages 23–32, 2018a. doi: 10.1145/3274895.3274927.

F. Bastani, S. He, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, S. Madden, and D. DeWitt. Roadtracer: Automatic extraction of road networks from aerial images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4720–4728, 2018b. doi: 10.1109/CVPR.2018.00496.

Y. Bengio. Learning deep architectures for ai. *Foundations and Trends in Machine Learning*, 2:1–27, 1 2009. ISSN 19358237. doi: 10.1561/2200000006.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5:157–166, 1994. ISSN 19410093. doi: 10.1109/72.279181.

J. Biagioni and J. Eriksson. Inferring road maps from global positioning system traces. *Transportation Research Record: Journal of the Transportation Research Board*, 2291: 61–71, 2013. ISSN 0361-1981. doi: 10.3141/2291-08.

S. Bianco, R. Cadene, L. Celona, and P. Napoletano. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6:64270–64277, 10 2018. ISSN 21693536. doi: 10.1109/ACCESS.2018.2877890.

A. Blake, R. Curwen, and A. Zisserman. A framework for spatiotemporal control in the tracking of visual contours. *International Journal of Computer Vision 1993 11:2*, 11: 127–145, 10 1993. ISSN 1573-1405. doi: 10.1007/BF01469225.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. Training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992. doi: 10.1145/130385.130401.

M. L. Braun and C. S. Ong. *Open Science in Machine Learning*, pages 343–365. Chapman and Hall/CRC, 12 2018. ISBN 9781315373461. doi: 10.1201/9781315373461-13.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020-December, 5 2020. ISSN 10495258. doi: 10.48550/arxiv.2005.14165.

J. Brownlee. *Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions*. 2018.

N. Bussmann, P. Giudici, D. Marinelli, and J. Papenbrock. Explainable machine learning in credit risk management. *Computational Economics*, 57:203–216, 1 2021. ISSN 15729974. doi: 10.1007/S10614-020-10042-0/FIGURES/5.

Z. Byambadorj, R. Nishimura, A. Ayush, K. Ohta, and N. Kitaoka. Text-to-speech system for low-resource language using cross-lingual transfer learning and data augmentation. *Eurasip Journal on Audio, Speech, and Music Processing*, 2021:1–20, 12 2021. ISSN 16874722. doi: 10.1186/S13636-021-00225-4/FIGURES/12.

R. Cahuantzi, X. Chen, and S. Güttel. A comparison of lstm and gru networks for learning symbolic sequences. 7 2021. doi: 10.48550/arxiv.2107.02248.

A. Caroppo, A. Leone, and P. Siciliano. Deep transfer learning approaches for bleeding detection in endoscopy images. *Computerized Medical Imaging and Graphics*, 88:101852, 3 2021. ISSN 18790771. doi: 10.1016/j.compmedimag.2020.101852.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. pages 1–12, 12 2014. doi: 10.48550/arxiv.1412.7062.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. *arXiv*, 2020a. ISSN 23318422. Simple, direct, works.

T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv*, pages 1–18, 2020b. ISSN 23318422.

X. Chen, B. M. Williams, S. R. Vallabhaneni, G. Czanner, R. Williams, and Y. Zheng. Learning active contour models for medical image segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June: 11624–11632, 6 2019. ISSN 10636919. doi: 10.1109/CVPR.2019.01190.

G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan. Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 55:3322–3337, 6 2017. ISSN 01962892. doi: 10.1109/TGRS.2017.2669341.

V. Cheplygina, M. de Bruijne, and J. P. Pluim. Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical Image Analysis*, 54:280–296, 5 2019. ISSN 13618423. doi: 10.1016/j.media.2019.03.009.

K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. *Proceedings of SSST 2014 - 8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, 2014. doi: 10.3115/V1/W14-4012.

D. Costea and M. Leordeanu. Aerial image geolocalization from recognition and matching of roads and intersections. *British Machine Vision Conference 2016, BMVC 2016*, 2016-September, 5 2016. doi: 10.48550/arxiv.1605.08323.

A. Davydow and S. Nikolenko. Land cover classification with superpixels and jaccard index post-optimization. *IEEE Computer Society Conference on Computer Vision and Pattern*

Recognition Workshops, 2018-June:280–284, 12 2018. ISBN 9781538661000. doi: 10.1109/CVPRW.2018.00053.

A. de Maissin, R. Vallée, M. Flamant, M. Fondain-Bossiere, C. L. Berre, A. Coutrot, N. Normand, H. Mouchère, S. Coudol, C. Trang, A. Bourreille, and T. Iv. Multi-expert annotation of crohn's disease images of the small bowel for automatic detection using a convolutional recurrent attention neural network. *Endoscopy International Open*, 09: E1136–E1144, 7 2021. ISSN 2364-3722. doi: 10.1055/a-1468-3964.

I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, R. Raska, and R. Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2018-June:172–181, 5 2018. ISSN 21607516. doi: 10.1109/CVPRW.2018.00031.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *IEEE conference on computer vision and pattern recognition*, pages 248–255, 3 2009. doi: 10.1109/cvpr.2009.5206848.

J. Denker and Y. LeCun. Transforming neural-net output levels to probability distributions. *Advances in neural information processing systems*, 3, 1990.

J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:4171–4186, 10 2018. doi: 10.48550/arxiv.1810.04805.

T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning 2000 40:2*, 40:139–157, 8 2000. ISSN 1573-0565. doi: 10.1023/A:1007607513941.

X. Dong and J. Shen. Triplet loss in siamese network for object tracking. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11217 LNCS:472–488, 2018. ISSN 16113349. doi: 10.1007/978-3-030-01261-8_28/COVER.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. 10 2020. doi: 10.48550/arxiv.2010.11929.

X. Dray, C. Li, J. Saurin, F. Cholet, G. Rahmi, J. L. Mouel, C. Leandri, S. Lecleire, X. Amiot, J. Delvaux, C. Duburque, R. Gérard, R. Leenhardt, F. Mesli, G. Vanbiervliet, I. N. Larmurier, S. Sacher-Huvelin, C. Simon-Chane, R. Olivier, and A. Histace. Cad-cap: une base de données française à vocation internationale, pour le développement et la validation d'outils de diagnostic assisté par ordinateur en vidéocapsule endoscopique du grêle. *Journées Francophones d'Hépato-Gastroentérologie et d'Oncologie Digestive (JFHOD)*, 50:000441, 2 2018. doi: 10.1055/s-0038-1623358.

R. Egele, R. Maulik, K. Raghavan, B. Lusch, I. Guyon, and P. Balaprakash. Autodeuq: Automated deep ensemble with uncertainty quantification. 10 2021. doi: 10.48550/arxiv.2110.13511.

W. Falcon and K. Cho. A framework for contrastive self-supervised learning and designing a new approach. *arXiv*, 8 2020.

Y. Fan, Y. Qian, F. Xie, and F. K. Soong. Tts synthesis with bidirectional lstm based recurrent neural networks. *Fifteenth annual conference of the international speech communication association*, 2014.

W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23: 1–40, 1 2021. doi: 10.48550/arxiv.2101.03961.

I. Fernandez-Urien, C. Carretero, E. Borobio, A. Borda, E. Estevez, S. Galter, B. Gonzalez-Suarez, B. Gonzalez, M. Lujan, J. L. Martinez, V. Martínez, P. Menchén, J. Navajas, V. Pons, C. Prieto, and J. Valle. Capsule endoscopy capture rate: Has 4 frames-per-second any impact over 2 frames-per-second? *World Journal of Gastroenterology : WJG*, 20:14472, 10 2014. ISSN 22192840. doi: 10.3748/WJG.V20.I39.14472.

P. N. Figueiredo, I. N. Figueiredo, S. Prasath, and R. Tsai. Automatic polyp detection in pillcam colon 2 capsule images and videos: Preliminary feasibility report. *Diagnostic and Therapeutic Endoscopy*, 2011. ISSN 10703608. doi: 10.1155/2011/182435.

M. Firmino, G. Angelo, H. Morais, M. R. Dantas, and R. Valentim. Computer-aided detection (cade) and diagnosis (cadx) system for lung cancer with likelihood of malignancy. *BioMedical Engineering Online*, 15:1–17, 1 2016. ISSN 1475925X. doi: 10.1186/S12938-015-0120-7/TABLES/5.

Y. Fu, W. Zhang, M. Mandal, and M. Q. Meng. Computer-aided bleeding detection in wce video. *IEEE Journal of Biomedical and Health Informatics*, 18:636–642, 2014. ISSN 21682194. doi: 10.1109/JBHI.2013.2257819.

A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez. A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*, 70:41–65, 9 2018. ISSN 1568-4946. doi: 10.1016/J.ASOC.2018.05.018.

I. Gatopoulos and J. M. Tomczak. Self-supervised variational auto-encoders. *Entropy (Basel, Switzerland)*, 23, 6 2021. ISSN 1099-4300. doi: 10.3390/E23060747.

A. Ghosh, M. Ehrlich, S. Shah, L. Davis, and R. Chellappa. Stacked u-nets for ground material segmentation in remote sensing imagery. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2018-June:252–256, 12 2018. ISBN 9781538661000. doi: 10.1109/CVPRW.2018.00047.

P. Gilabert, A. Watson, and H. Wenzek. Artificial intelligence to improve polyp detection and screening time in colon capsule endoscopy. *Scientific Reports, In Review*, 1 2022. doi: 10.21203/RS.3.RS-1278962/V1.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Advances in neural information processing systems*, 6 2014a. ISSN 16113349. doi: 10.48550/arxiv.1406.2661.

I. Goodfellow, B. Yoshua, and A. Courville. *Deep Learning*. MIT Press, 2016.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 12 2014b. doi: 10.48550/arxiv.1412.6572.

A. Graves, S. Fernández, and J. Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3697 LNCS:799–804, 2005. ISSN 03029743. doi: 10.1007/11550907_126/COVER/.

A. Graves, A. R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 6645–6649, 3 2013. ISSN 15206149. doi: 10.48550/arxiv.1303.5778.

Z. Gu, Z. Li, X. Di, and R. Shi. An lstm-based autonomous driving model using waymo open dataset. *Applied Sciences (Switzerland)*, 10:1–14, 2 2020. doi: 10.3390/app10062046.

C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. *34th International Conference on Machine Learning, ICML 2017*, 3:2130–2143, 6 2017. doi: 10.48550/arxiv.1706.04599.

X. Guo and Y. Yuan. Semi-supervised wce image classification with adaptive aggregated attention. *Medical Image Analysis*, 64:101733, 8 2020. ISSN 13618423. doi: 10.1016/j.media.2020.101733.

X. Guo, Z. Chen, J. Liu, and Y. Yuan. Non-equivalent images and pixels: confidence-aware resampling with meta-learning mixup for polyp segmentation. *Medical Image Analysis*, 78:102394, 5 2022. ISSN 13618415. doi: 10.1016/j.media.2022.102394.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:386–397, 3 2017. ISSN 19393539. doi: 10.48550/arxiv.1703.06870.

S. He, F. Bastani, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, and S. Madden. Roadrunner: Improving the precision of road network inference from gps trajectories. *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, pages 3–12, 11 2018. doi: 10.1145/3274895.3274974.

S. He, F. Bastani, S. Jagwani, E. Park, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, S. Madden, and M. A. Sadeghi. Roadtagger: Robust road attribute inference with graph neural networks. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pages 10965–10972, 12 2019. ISSN 2159-5399. doi: 10.48550/arxiv.1912.12408.

S. He, F. Bastani, S. Jagwani, M. Alizadeh, H. Balakrishnan, S. Chawla, M. M. Elshrif, S. Madden, and M. A. Sadeghi. Sat2graph: Road graph extraction through graph-tensor encoding. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12369 LNCS:51–67, 7 2020. ISSN 16113349. doi: 10.48550/arxiv.2007.09547.

X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2, 2004. ISSN 10636919. doi: 10.1109/CVPR.2004.1315232.

D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *7th International Conference on Learning Representations, ICLR 2019*, 3 2019. doi: 10.48550/arxiv.1903.12261.

G. E. Hinton, A. Krizhevsky, and S. D. Wang. Transforming auto-encoders. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6791 LNCS:44–51, 2011. ISBN 9783642217340. doi: 10.1007/978-3-642-21735-7_6.

J. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. 1991.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9: 1735–1780, 11 1997. ISSN 08997667. doi: 10.1162/NECO.1997.9.8.1735.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90020-8.

A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv*, 4 2017. doi: 10.48550/arxiv.1704.04861.

J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu. Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:2011–2023, 9 2020. ISSN 19393539. doi: 10.1109/TPAMI.2019.2913372.

G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger. Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:2261–2269, 8 2017. ISBN 9781538604571. doi: 10.1109/CVPR.2017.243.

D. K. Iakovidis, S. V. Georgakopoulos, M. Vasilakakis, A. Koulaouzidis, and V. P. Plagianakos. Detecting and locating gastrointestinal anomalies using deep learning and iterative cluster unification. *IEEE Transactions on Medical Imaging*, 37:2196–2210, 10 2018. ISSN 1558254X. doi: 10.1109/TMI.2018.2837002.

Institut Cartogràfic i Geològic de Catalunya. Cobertes del sòl. 2022. URL https://www.icgc.cat/Descarregues/Mapes-en-format-d-imatge/Cobertes-del-sol.

ISPRS. International society for photogrammetry and remote sensing. 2d semantic labeling - vaihingen data. URL https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-vaihingen.aspx.

S. Jain, A. Seal, A. Ojha, O. Krejcar, J. Bureš, I. Tachecí, and A. Yazidi. Detection of abnormality in wireless capsule endoscopy images using fractal features. *Computers in Biology and Medicine*, 127:104094, 12 2020. ISSN 18790534. doi: 10.1016/j.compbiomed.2020.104094.

S. Jain, A. Seal, A. Ojha, A. Yazidi, J. Bures, I. Tacheci, and O. Krejcar. A deep cnn model for anomaly detection and localization in wireless capsule endoscopy images. *Computers in Biology and Medicine*, 137:104789, 10 2021. ISSN 18790534. doi: 10.1016/j.compbiomed.2021.104789.

F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62:S63–S63, 12 1977. ISSN 0001-4966. doi: 10.1121/1.2016299.

M. Jordan. Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986. 1986.

P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55:6054–6068, 2017. ISSN 01962892. doi: 10.1109/TGRS.2017.2719738.

A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 2017-Decem: 5575–5585, 3 2017. doi: 10.48550/arxiv.1703.04977.

M. A. Khan, S. Kadry, M. Alhaisoni, Y. Nam, Y. Zhang, V. Rajinikanth, and M. S. Sarfraz. Computer-aided gastrointestinal diseases analysis from wireless capsule endoscopy: A framework of best features selection. *IEEE Access*, 8:132850–132859, 2020. ISSN 21693536. doi: 10.1109/ACCESS.2020.3010448.

S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah. Transformers in vision: A survey. *ACM Computing Surveys*, 1 2021. doi: 10.1145/3505244.

S. Khandelwal, B. Lecouteux, and L. Besacier. *COMPARING GRU AND LSTM FOR AUTOMATIC SPEECH RECOGNITION*. PhD thesis, LIG - Laboratoire d'Informatique de Grenoble, 2016.

S. H. Kim and Y. J. Lim. Artificial intelligence in capsule endoscopy: A practical guide to its past and future challenges. *Diagnostics*, 11:1722, 9 2021. ISSN 20754418. doi: 10.3390/diagnostics11091722.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

A. D. Kiureghian and O. Ditlevsen. Aleatory or epistemic? does it matter? *Structural Safety*, 31:105–112, 3 2009. ISSN 0167-4730. doi: 10.1016/J.STRUSAFE.2008.06.020.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 5 2012. ISSN 15577317. doi: 10.1145/3065386.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22:79–86, 3 1951. ISSN 0003-4851. doi: 10.1214/AOMS/1177729694.

A. K. Kundu and S. A. Fattah. Probability density function based modeling of spatial feature variation in capsule endoscopy data for automatic bleeding detection. *Computers in Biology and Medicine*, 115:103478, 12 2019. ISSN 18790534. doi: 10.1016/j.compbiomed.2019.103478.

T. S. Kuo, K. S. Tseng, J. W. Yan, Y. C. Liu, and Y. C. F. Wang. Deep aggregation net for land cover classification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2018-June:247–251, 12 2018. ISBN 9781538661000. doi: 10.1109/CVPRW.2018.00046.

P. Laiz, J. Vitria, and S. Segui. Using the triplet loss for domain adaptation in wce. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 399–405, 10 2019. ISBN 978-1-7281-5023-9. doi: 10.1109/ICCVW.2019.00051.

P. Laiz, J. Vitrià, H. Wenzek, C. Malagelada, F. Azpiroz, and S. Seguí. Wce polyp detection with triplet based embeddings. *Computerized Medical Imaging and Graphics*, 86, 2020. ISSN 18790771. doi: 10.1016/j.compmedimag.2020.101794.

B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 2017-December:6403–6414, 12 2016. ISSN 10495258. doi: 10.48550/arxiv.1612.01474.

I. Laptev, T. Lindeberg, H. Mayer, C. Steger, W. Eckstein, and A. Baumgartner. Automatic extraction of roads from aerial images based on scale space and snakes. *Machine Vision and Applications*, 12:23–31, 2003. ISSN 0932-8092. doi: 10.1007/s001380050121.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2323, 1998. ISSN 00189219. doi: 10.1109/5.726791.

H. Y. Lee, J. B. Huang, M. Singh, and M. H. Yang. Unsupervised representation learning by sorting sequences. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:667–676, 2017a. ISBN 9781538610329. doi: 10.1109/ICCV.2017.79.

J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. Deep neural networks as gaussian processes. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 11 2017b. doi: 10.48550/arxiv.1711.00165.

H. Li, P. Xiong, J. An, and L. Wang. Pyramid attention network for semantic segmentation. *British Machine Vision Conference 2018, BMVC 2018*, 5 2018a. doi: 10.48550/arxiv.1805.10180.

H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. *Advances in Neural Information Processing Systems*, 2018-Decem:6389–6399, 12 2018b. doi: 10.48550/arxiv.1712.09913.

Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:4438–4446, 11 2016. doi: 10.48550/arxiv.1611.07709.

Z. Li, J. D. Wegner, and A. Lucchi. Polymapper: Extracting city maps using polygons. *arXiv preprint arXiv:1812.01497*, 12 2018c. doi: arXiv:1812.01497v1.

G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:5168–5177, 11 2016a. doi: 10.48550/arxiv.1611.06612.

J. Lin, A. Yang, J. Bai, C. Zhou, L. Jiang, X. Jia, A. Wang, J. Zhang, Y. Li, W. Lin, J. Zhou, and H. Yang. M6-10t: A sharing-delinking paradigm for efficient multi-trillion parameter pretraining. 10 2021. doi: 10.48550/arxiv.2110.03888.

T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. 12 2016b. doi: 10.48550/arxiv.1612.03144.

J. Z. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax-Weiss, and B. Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, 2020-December, 6 2020. ISSN 10495258. doi: 10.48550/arxiv.2006.10108.

W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. 6 2015. doi: 10.48550/arxiv.1506.04579.

X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 6 2021a. ISSN 15582191. doi: 10.1109/TKDE.2021.3090866.

X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 6 2021b. ISSN 15582191. doi: 10.1109/TKDE.2021.3090866.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 07-12-June-2015:3431–3440, 6 2015. ISSN 1063-6919. doi: 10.1109/CVPR.2015.7298965.

Y. Long, G. S. Xia, S. Li, W. Yang, M. Y. Yang, X. X. Zhu, L. Zhang, and D. Li. On creating benchmark dataset for aerial image interpretation: Reviews, guidances and million-aid. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14: 4205–4230, 6 2020. ISSN 21511535. doi: 10.48550/arxiv.2006.12485.

Y. Long, G.-S. Xia, L. Zhang, G. Cheng, and D. Li. Aerial scene parsing: From tile-level scene classification to pixel-wise semantic labeling. 1 2022. doi: 10.48550/arxiv.2201.01953.

R. Lu and Z. Duan. Detection and classification of acoustic scenes and events. *Detection and Classification of Acoustic Scenes and Events*, 2017.

G. Lv, G. Yan, and Z. Wang. Bleeding detection in wireless capsule endoscopy images based on color invariants and spatial pyramids using support vector machines. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2011:6643–6646, 2011. ISBN 9781424441211. doi: 10.1109/IEMBS.2011.6091638.

L. V. D. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2625, 2008. ISSN 15324435.

D. J. MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4:448–472, 1992.

C. Malagelada, F. D. Iorio, F. Azpiroz, A. Accarino, S. Segui, P. Radeva, and J. R. Malagelada. New insight into intestinal motor function via noninvasive endoluminal image analysis. *Gastroenterology*, 135:1155–1162, 2008. ISSN 00165085. doi: 10.1053/j.gastro.2008.06.084.

M. Martinez, C. Sitawarin, K. Finch, L. Meincke, A. Yablonski, and A. Kornhauser. Beyond grand theft auto v for training, testing and enhancing deep learning in self driving cars. 12 2017. doi: 10.48550/arxiv.1712.01397.

B. C. Mateus, M. Mendes, J. T. Farinha, R. Assis, and A. M. Cardoso. Comparing lstm and gru models to predict the condition of a pulp paper press. *Energies 2021, Vol. 14, Page 6958*, 14:6958, 10 2021. ISSN 1996-1073. doi: 10.3390/EN14216958.

A. G. D. Matthews, J. Hron, M. Rowland, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 4 2018. doi: 10.48550/arxiv.1804.11271.

G. Mattyus, W. Luo, and R. Urtasun. Deeproadmapper: Extracting road topology from aerial images. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-October:3458–3466, 12 2017. ISSN 15505499. doi: 10.1109/ICCV.2017.372.

H. Mayer, I. Laptev, A. Baumgartner, and C. Steger. Automatic road extraction based on multi-scale modeling, context, and snakes. *Image (Rochester, N.Y.)*, pages 3–10, 1997.

L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv*, 2 2018.

S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1 2020. ISSN 19393539. doi: 10.48550/arxiv.2001.05566.

I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS: 527–544, 2016. ISBN 9783319464473. doi: 10.1007/978-3-319-46448-0_32.

V. Mnih and G. E. Hinton. Learning to detect roads in high-resolution aerial images. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence* *and Lecture Notes in Bioinformatics)*, 6316 LNCS:210–223, 2010. ISSN 16113349. doi: 10.1007/978-3-642-15567-3_16/COVER/.

K. Muhammad, Mustaqeem, A. Ullah, A. S. Imran, M. Sajjad, M. S. Kiran, G. Sannino, and V. H. C. de Albuquerque. Human action recognition using attention based lstm network with dilated cnn features. *Future Generation Computer Systems*, 125:820–830, 12 2021. ISSN 0167-739X. doi: 10.1016/J.FUTURE.2021.06.045.

E. S. Nadimi, M. M. Buijs, J. Herp, R. Kroijer, M. Kobaek-Larsen, E. Nielsen, C. D. Pedersen, V. Blanes-Vidal, and G. Baatrup. Application of deep learning for autonomous detection and localization of colorectal polyps in wireless colon capsule endoscopy. *Computers and Electrical Engineering*, 81:106531, 1 2020. ISSN 00457906. doi: 10.1016/j.compeleceng.2019.106531.

M. P. Naeini, G. F. Cooper, and M. Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015:2901, 6 2015. ISSN 2159-5399.

V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, pages 807–814, 2010. ISBN 9781605589077.

F. Navarro, C. Watanabe, S. Shit, A. Sekuboyina, J. C. Peeken, S. E. Combs, and B. H. Menze. Evaluating the robustness of self-supervised learning in medical imaging. 5 2021.

R. M. Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.

A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June-2015:427–436, 12 2014. ISSN 10636919. doi: 10.48550/arxiv.1412.1897.

D. M. Nguyen, W. Ding, A. Munteanu, N. Deligiannis, and Y. Liu. Hourglass-shapenetwork based semantic segmentation for high resolution aerial imagery. *Remote Sensing*, 9:522, 2017. doi: 10.3390/rs9060522.

D. A. Nix and A. S. Weigend. Estimating the mean and variance of the target probability distribution. *IEEE International Conference on Neural Networks - Conference Proceedings*, 1:55–60, 1994. doi: 10.1109/ICNN.1994.374138.

K. Nogueira, M. D. Mura, J. Chanussot, W. R. Schwartz, and J. A. D. Santos. Dynamic multicontext segmentation of remote sensing images based on convolutional networks. *IEEE Transactions on Geoscience and Remote Sensing*, 57:7503–7520, 10 2019. ISSN 15580644. doi: 10.1109/TGRS.2019.2913861.

H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. 5 2015.

A. Odena. Semi-supervised learning with generative adversarial networks. 6 2016. doi: 10.48550/arxiv.1606.01583.

Office of the Comptroller of the Currency. *Comptroller's Handbook: Model Risk Management.* 8 2021.

M. Ohbayashi and K. Hirasawa. Beyond regression : "new tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*, 117:289–297, 9 1974. ISSN 13488163. doi: 10.1541/IEEJIAS.117.289.

Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, 32, 6 2019. ISSN 10495258. doi: 10.48550/arxiv.1906.02530.

N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26:1277–1294, 9 1993. ISSN 0031-3203. doi: 10.1016/0031-3203(93)90135-J.

Papers With Code. Imagenet benchmark (image classification) — papers with code. 2022. URL https://paperswithcode.com/sota/image-classification-on-imagenet.

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. *30th International Conference on Machine Learning, ICML 2013*, pages 2347–2355, 11 2012. doi: 10.48550/arxiv.1211.5063.

G. Pascual, S. Segui, and J. Vitria. Uncertainty gated network for land cover segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2018-June, 2018. ISBN 9781538661000. doi: 10.1109/CVPRW.2018.00052.

G. Pascual, P. Laiz, A. García, H. Wenzek, J. Vitrià, and S. Seguí. Time-based self-supervised learning for wireless capsule endoscopy. *Computers in Biology and Medicine*, 146:105631, 7 2022. ISSN 0010-4825. doi: 10.1016/J.COMPBIOMED.2022.105631.

P. Patel and A. Thakkar. The upsurge of deep learning for computer vision applications. *International Journal of Electrical and Computer Engineering (IJECE)*, 10:538–548, 2 2020. ISSN 2722-2578. doi: 10.11591/IJECE.V10I1.PP538-548.

D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:6024–6033, 12 2017. ISBN 9781538604571. doi: 10.1109/CVPR.2017.638.

K. Pogorelov, S. Suman, F. A. Hussin, A. S. Malik, O. Ostroukhova, M. Riegler, P. Halvorsen, S. H. Ho, and K. L. Goh. Bleeding detection in wireless capsule endoscopy videos — color versus texture features. *Journal of Applied Clinical Medical Physics*, 20: 141–154, 8 2019. ISSN 15269914. doi: 10.1002/acm2.12662.

J. Postels, F. Ferroni, H. Coskun, N. Navab, and F. Tombari. Sampling-free epistemic uncertainty estimation using approximated variance propagation. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:2931–2940, 8 2019. ISSN 15505499. doi: 10.48550/arxiv.1908.00598.

J. Postels, H. Blum, C. Cadena, R. Siegwart, L. V. Gool, and F. Tombari. Quantifying aleatoric and epistemic uncertainty using density estimation in latent space. *CoRR*, 9 2020.

F. Pérez-García, R. Dorent, M. Rizzi, F. Cardinale, V. Frazzini, V. Navarro, C. Essert, I. Ollivier, T. Vercauteren, R. Sparks, J. S. Duncan, and S. Ourselin. A self-supervised learning strategy for postoperative brain cavity segmentation simulating resections. *International Journal of Computer Assisted Radiology and Surgery*, 82:1–9, 6 2021. ISSN 18616429. doi: 10.1007/s11548-021-02420-2.

J. R. Quinlan. Induction of decision trees. *Machine Learning 1986 1:1*, 1:81–106, 3 1986. ISSN 1573-0565. doi: 10.1007/BF00116251.

M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy. Do vision transformers see like convolutional neural networks? 8 2021. doi: 10.48550/arxiv.2108.08810.

R. Raina, A. Madhavan, and A. Y. Ng. Large-scale deep unsupervised learning using graphics processors. *ACM International Conference Proceeding Series*, 382, 2009. doi: 10.1145/1553374.1553486.

A. Rakhlin, A. Davydow, and S. Nikolenko. Land cover classification from satellite imagery with u-net and lovász-softmax loss. *IEEE Computer Society Conference on Com-*

puter Vision and Pattern Recognition Workshops, 2018-June:257–261, 12 2018. ISBN 9781538661000. doi: 10.1109/CVPRW.2018.00048.

J. Reuss, G. Pascual, H. Wenzek, and S. Seguí. Sequential models for endoluminal image classification. *Diagnostics*, 12:501, 2 2022. ISSN 20754418. doi: 10.3390/diagnostics12020501.

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22:400–407, 9 1951. ISSN 0003-4851. doi: 10.1214/AOMS/1177729586.

D. E. Ruineihart, G. E. Hint, and R. J. Williams. Learning internal representations by error propagation. *California Univ., San Diego, La Jolla (USA). Inst. for Cognitive Science*, 8506, 1985.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning Internal Representations by Error Propagation*, pages 399–421. 9 2013. ISBN 1558600132. doi: 10.1016/B978-1-4832-1446-7.50035-2.

K. Saito, D. Kim, S. Sclaroff, and K. Saenko. Universal domain adaptation through self supervision. *Advances in Neural Information Processing Systems*, 2020-December, 2 2020. ISSN 10495258. doi: 10.48550/arxiv.2002.07953.

M. Samy, K. Amer, K. Eissa, M. Shaker, and M. Elhelw. Nu-net: Deep residual wide field of view convolutional neural network for semantic segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2018-June: 267–271, 12 2018. ISBN 9781538661000. doi: 10.1109/CVPRW.2018.00050.

Y. Sasaki. The truth of the f-measure. *Teach tutor mater*, 1:1–5, 2015.

D. Scherer, A. Müller, and S. Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6354 LNCS:92–101, 2010. ISBN 3642158242. doi: 10.1007/978-3-642-15825-4_10.

M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45:2673–2681, 1997. ISSN 1053587X. doi: 10.1109/78.650093.

S. Seferbekov, V. Iglovikov, A. Buslaev, and A. Shvets. Feature pyramid network for multi-class land segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2018-June:272–275, 6 2018. ISBN 9781538661000. doi: 10.1109/CVPRW.2018.00051.

S. Seguí, M. Drozdzal, G. Pascual, P. Radeva, C. Malagelada, F. Azpiroz, and J. Vitrià. Generic feature learning for wireless capsule endoscopy analysis. *Computers in Biology and Medicine*, 79:163–172, 12 2016. ISSN 18790534. doi: 10.1016/j.compbiomed.2016.10.011.

T. J. Sejnowski. The deep learning revolution: Machine intelligence meets human intelligence. *MIT Press*, 2018.

P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain. Time-contrastive networks: Self-supervised learning from video. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1134–1141, 9 2018. ISBN 9781538630815. doi: 10.1109/ICRA.2018.8462891.

J. Sherrah. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. 6 2016. doi: 10.48550/arxiv.1606.02585.

X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, 2015-January:802–810, 6 2015. ISSN 10495258. doi: 10.48550/arxiv.1506.04214.

C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48, 12 2019. ISSN 21961115. doi: 10.1186/S40537-019-0197-0/FIGURES/33.

P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2003-January:958–963, 2003. ISSN 15205363. doi: 10.1109/ICDAR.2003.1227801.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 9 2015. doi: 10.48550/arxiv.1409.1556.

D. Spathis, I. Perez-Pozuelo, S. Brage, N. J. Wareham, and C. Mascolo. Self-supervised transfer learning of physiological representations from free-living wearable data. *ACM CHIL 2021 - Proceedings of the 2021 ACM Conference on Health, Inference, and Learning*, pages 69–78, 11 2020. doi: 10.1145/3450439.3451863.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. ISSN 1533-7928.

R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv*, 5 2015. doi: 10.48550/arxiv.1505.00387.

L. Studer, M. Alberti, V. Pondenkandath, P. Goktepe, T. Kolonko, A. Fischer, M. Liwicki, and R. Ingold. A comprehensive study of imagenet pre-training for historical document image analysis. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pages 720–725, 5 2019. ISSN 15205363. doi: 10.48550/arxiv.1905.09113.

P. Sudowe and B. Leibe. Patchit: Self-supervised network weight initialization for fine-grained recognition. *British Machine Vision Conference 2016, BMVC 2016*, 2016-September:2266–2270, 2016. doi: 10.5244/C.30.75.

K. Suzuki. A review of computer-aided diagnosis in thoracic and colonic imaging. *Quantitative Imaging in Medicine and Surgery*, 2:163, 9 2012. ISSN 2223-4292. doi: 10.3978/J.ISSN.2223-4292.2012.09.02.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:1–9, 9 2015. ISBN 9781467369640. doi: 10.1109/CVPR.2015.7298594.

T. Takikawa, D. Acuna, V. Jampani, and S. Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-October:5228–5237, 7 2019. ISSN 15505499. doi: 10.48550/arxiv.1907.05740.

M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:10691–10700, 5 2019. doi: 10.48550/arxiv.1905.11946.

S. Thakur. The very basics of bayesian neural networks. 2022. URL `https://sanjaykthakur.com/2018/12/05/the-very-basics-of-bayesian-neural-networks/`.

C. Tian, C. Li, and J. Shi. Dense fusion classmate network for land cover classification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2018-June:262–266, 11 2018. ISBN 9781538661000. doi: 10.48550/arxiv.1911.08169.

D. Tran, J. Snoek, and B. Lakshminarayanan. Practical uncertainty estimation and out-of-distribution robustness in deep learning. *Neural Information Processing Systems*, 2020.

R. Trasolini and M. F. Byrne. Artificial intelligence and deep learning for small bowel capsule endoscopy. *Digestive Endoscopy*, 33:290–297, 1 2021. ISSN 14431661. doi: 10.1111/den.13896.

K. Treash and K. Amaratunga. Automatic road detection in grayscale aerial images. *Journal of Computing in Civil Engineering*, 14:60–69, 2000. ISSN 0887-3801. doi: 10.1061/(ASCE)0887-3801(2000)14:1(60).

J. C. Trinder and Y. Wang. Automatic road extraction from aerial images. *Digital Signal Processing: A Review Journal*, 8:215–224, 1998. ISSN 10512004. doi: 10.1006/dspr.1998.0322.

M. Tschannen, J. Djolonga, M. Ritter, A. Mahendran, N. Houlsby, S. Gelly, and M. Lucic. Self-supervised learning of video-induced visual invariances. *In Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2015.

V. V and K. V. Prashanth. Ulcer detection in wireless capsule endoscopy images using deep cnn. *Journal of King Saud University - Computer and Information Sciences*, 9 2020. ISSN 22131248. doi: 10.1016/j.jksuci.2020.09.008.

R. Vallée, A. D. Maissin, A. Coutrot, N. Normand, A. Bourreille, and H. Mouchère. Accurate small bowel lesions detection in wireless capsule endoscopy images using deep recurrent attention neural network. *IEEE 21st International Workshop on Multimedia Signal Processing, MMSP 2019*, 9 2019. doi: 10.1109/MMSP.2019.8901788.

R. Vallée, A. D. Maissin, A. Coutrot, H. Mouchère, A. Bourreille, and N. Normand. Crohnipi: An endoscopic image database for the evaluation of automatic crohn's disease lesions recognition algorithms. *https://doi.org/10.1117/12.2543584*, 11317:61, 2 2020. ISSN 16057422. doi: 10.1117/12.2543584.

A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *ICML*, 4:2611–2620, 1 2016. doi: 10.48550/arxiv.1601.06759.

A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. 7 2018.

V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer New York, 1995. ISBN 978-1-4757-2440-0. doi: 10.1007/978-1-4757-2440-0.

V. N. Vapnik. *Statistical Learning Theory (Adaptive and Learning Systems for Signal Processing, Communications and Control Series).* Wiley-Interscience, 1998. ISBN 0471030031.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-December:5999–6009, 6 2017. ISSN 10495258. doi: 10.48550/arxiv.1706.03762.

A. Vats, M. Pedersen, and A. Mohammed. A preliminary analysis of self-supervision for wireless capsule endoscopy. *Proceedings - European Workshop on Visual Information Processing, EUVIP*, 2021-June:1–6, 2021. ISSN 24718963. doi: 10.1109/EUVIP50544.2021.9484012.

C. Ventura, J. Pont-Tuset, S. Caelles, K.-K. Maninis, and L. V. Gool. Iterative deep learning for road topology extraction. *arXiv preprint arXiv:1808.09814*, pages 1–13, 2018.

F. Visin, A. Romero, K. Cho, M. Matteucci, M. Ciccone, K. Kastner, Y. Bengio, and A. Courville. Reseg: A recurrent neural network-based model for semantic segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 426–433, 11 2015. ISSN 21607516. doi: 10.48550/arxiv.1511.07053.

G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338:34–45, 7 2018. doi: 10.1016/j.neucom.2019.01.103.

H. Wang, Y. Wang, Q. Zhang, S. Xiang, and C. Pan. Gated convolutional neural network for semantic segmentation in high-resolution images. *Remote Sensing*, 9:446, 5 2017. ISSN 20724292. doi: 10.3390/rs9050446.

L. Wang, C.-Y. Lee, Z. Tu, and S. Lazebnik. Training deeper convolutional networks with deep supervision. *arXiv*, 5 2015. doi: 10.48550/arxiv.1505.02496.

X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:2794–2802, 2015. ISBN 9781467383912. doi: 10.1109/ICCV.2015.320.

X. Wang and M. P. Panos. A survey of support vector machines with uncertainties. *Annals of Data Science 2015 1:3*, 1:293–309, 1 2015. ISSN 2198-5812. doi: 10.1007/S40745-014-0022-8.

X. Wang, A. Jabri, and A. A. Efros. Learning correspondence from the cycle-consistency of time. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:2561–2571, 2019. ISBN 9781728132938. doi: 10.1109/CVPR.2019.00267.

X. Wang, H. Qian, E. J. Ciaccio, S. K. Lewis, G. Bhagat, P. H. Green, S. Xu, L. Huang, R. Gao, and Y. Liu. Celiac disease diagnosis from videocapsule endoscopy images with residual learning and deep feature extraction. *Computer Methods and Programs in Biomedicine*, 187:105236, 4 2020. ISSN 18727565. doi: 10.1016/j.cmpb.2019.105236.

J. D. Wegner, J. A. Montoya-Zegarra, and K. Schindler. A higher-order crf model for road network extraction. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1698–1705, 2013. ISSN 10636919. doi: 10.1109/CVPR.2013.222.

J. D. Wegner, J. A. Montoya-Zegarra, and K. Schindler. Road networks as collections of minimum cost paths. *ISPRS Journal of Photogrammetry and Remote Sensing*, 108: 128–137, 10 2015. ISSN 0924-2716. doi: 10.1016/J.ISPRSJPRS.2015.07.002.

W. Weng, X. Zhu, O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9351:16591–16603, 5 2015. ISSN 16113349. doi: 10.1007/978-3-319-24574-4_28.

C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. *Advances in Neural Information Processing Systems*, 8, 1995.

X. Wu, R. Ward, and L. Bottou. Wngrad: Learn the learning rate in gradient descent. 3 2018. doi: 10.48550/arxiv.1803.02865.

S. Xavier, S. Monteiro, J. Magalhães, B. Rosa, M. J. Moreira, and J. Cotter. Capsule endoscopy with pillcamsb2 versus pillcamsb3: has the improvement in technology resulted in a step forward? *Revista espanola de enfermedades digestivas : organo oficial de la Sociedad Espanola de Patologia Digestiva*, 110:155–159, 3 2018. ISSN 1130-0108. doi: 10.17235/REED.2017.5071/2017.

S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:5987–5995, 11 2017. ISBN 9781538604571. doi: 10.1109/CVPR.2017.634.

D. Xu, J. Xiao, Z. Zhao, J. Shao, D. Xie, and Y. Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:10326–10335, 2019. ISBN 9781728132938. doi: 10.1109/CVPR.2019.01058.

L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. *Advances in neural information processing systems*, 27, 2014. doi: 10.5555/2968826.2969026.

S. Yang, Y. Wang, and X. Chu. A survey of deep learning techniques for neural machine translation. *arXiv*, 2 2020a. doi: 10.48550/arxiv.2002.07526.

S. Yang, X. Yu, and Y. Zhou. Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example. *Proceedings - 2020 International Workshop on Electronic Communication and Artificial Intelligence, IWECAI 2020*, pages 98–101, 6 2020b. doi: 10.1109/IWECAI50956.2020.00027.

X. Yang, X. He, Y. Liang, Y. Yang, S. Zhang, and P. Xie. Transfer learning or self-supervised learning? a tale of two pretraining paradigms. 6 2020c. doi: 10.48550/arxiv.2007.04234.

J.-Y. Yeh, T.-H. Wu, and W.-J. Tsai. Bleeding and ulcer detection using wireless capsule endoscopy images. *Journal of Software Engineering and Applications*, 07:422–432, 5 2014. ISSN 1945-3116. doi: 10.4236/jsea.2014.75039.

F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 11 2015. doi: 10.48550/arxiv.1511.07122.

J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, Y. Wu, and G. Research. Coca: Contrastive captioners are image-text foundation models. 5 2022. doi: 10.48550/arxiv.2205.01917.

Y. Yuan and M. Q. Meng. Automatic bleeding frame detection in the wireless capsule endoscopy images. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015-June:1310–1315, 6 2015. doi: 10.1109/ICRA.2015.7139360.

Y. Yuan, W. Qin, B. Ibragimov, G. Zhang, B. Han, M. Q. Meng, and L. Xing. Densely connected neural network with unbalanced discriminant and category sensitive constraints for polyp recognition. *IEEE Transactions on Automation Science and Engineering*, 17: 574–583, 4 2020. ISSN 15583783. doi: 10.1109/TASE.2019.2936645.

W. Zaremba, I. Sutskever, O. Vinyals, and G. Brain. Recurrent neural network regularization. 9 2014. doi: 10.48550/arxiv.1409.2329.

W. Zhang, J. Tanida, K. Itoh, and Y. Ichioka. Shift-invariant pattern recognition neural network and its optical architecture. *Proceedings of annual conference of the Japan Society of Applied Physics*, pages 2147–2151, 1988.

W. J. Zhang, G. Yang, Y. Lin, C. Ji, and M. M. Gupta. On definition of deep learning. *World Automation Congress Proceedings*, 2018-June:232–236, 8 2018. ISSN 21544832. doi: 10.23919/WAC.2018.8430387.

Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5:726–742, 12 2020. doi: 10.1109/TETCI.2021.3100641.

H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:6230–6239, 12 2016. doi: 10.48550/arxiv.1612.01105.

M. Zhou, Z. Niu, L. Wang, Z. Gao, Q. Zhang, and G. Hua. Ladder loss for coherent visual-semantic embedding. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pages 13050–13057, 11 2020. ISBN 9781577358350. doi: 10.1609/aaai.v34i07.7006.