University of Macedonia
Department of Business Administration
Master in Business Analytics and Data Science

Transfer learning: Applications in image and natural language data

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master in Business Analytics and Data Science

Student:
Mamalis Konstantinos Marios

Supervisor:
Tarabanis Konstantinos

MBADS

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗΝ
ΑΝΑΛΥΤΙΚΗ ΤΩΝ ΕΠΙΧΕΙΡΗΣΕΩΝ ΚΑΙ ΕΠΙΣΤΗΜΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

June 2023

To my parents

Acknowledgements

I would like to acknowledge my supervisor professors, Konstantinos Tarabanis and Evangelos Kalampokis, for their support and constant guidance throughout the creation of this thesis that could not have been created without them. I would also like to thank my colleagues in the ISLAB Petros Brimos and Areti Karamanou for their assistance and collaboration.

Abstract

Transfer learning appears to be one of the most influential techniques used in machine learning today with applications in nearly all state of the art models. From natural language machine learning to computer vision and tabular data, transfer learning has reshaped the way machine learning algorithms and models are developed and applied. In this thesis the way transfer learning works is examined and results of its applications on different domains (agricultural-image data, medical-natural language data) are presented. We conclude that this technique could change the landscape of machine learning and artificial intelligence as a whole even more within the next few years, something backed up by its performance and flexibility on a variety of tasks.

Keywords: Artificial Intelligence, Machine Learning, Transfer Learning, Computer Vision, Natural Language Processing

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The fact that the field of Machine learning is constantly evolving is a well known and documented fact. Starting from the age of simple computer aided statistical modeling in the 60's (Fradkov, 2020), and extending all the way to today's state of the art machine learning models, one thing has remained in the focus of its development; ease of access. By the term ease of access to machine learning we refer to the the level of expertise and the machinery needed to create machine learning solutions, evaluate them and deploy them to an environment where they will benefit their users. Easy of access however, tends to be inversely proportional to another factor; model quality.

With the coordinated research efforts taking place in the field of machine learning, innovations have been made that have improved the efficiency of the models' predictive capabilities. Nowadays, models exist that can infer directly or indirectly the cause and effect of processes with astounding precision. Problems such as image recognition, sentiment analysis, development of speech-to-text systems and so many more have been successfully addressed through the use of constantly improving machine learning models.

The combination of the two aforementioned opposing forces -ease of access and model quality- along with a technological hardware plateau that seems to have been reached (Theis and Wong, 2017), has lead to a developmental bottleneck, since the better the models, the less accessible they will be to the user, with little to no chance that further advancements in hardware will be enough to mitigate the loss of accessibility due to the scarcity of computational resources.

Transfer learning Transfer Learning (TL) is a proposed solution to much of the aforementioned issue. By employing the mechanisms of pre-training and fine-tuning, described in detail in the chapters that follow, transfer learning can increase the accessibility of the model while maintaining its high quality. It essentially separates the model training process into two parts, one being an extremely computationally demanding task that concerns the general fit of the model to the data (pre-training), while the other is a less demanding reiteration of the training procedure, that further fits the already created model to more specific data of a different but similar task (fine-tuning). This way, models can be pre-trained initially by parties with access to massive hardware resources and data (e.g. Google, Meta) and then distributed to users with fewer, so that they can, too, use them by fine-tuning the initial models for their own purposes.

With the constant need for more data and hardware, transfer learning seems to be the future of AI applications, where innovation will be led by a select few organizations and the individual machine learning engineers or small companies will simply add the necessary components to adapt the models to their needs. This will very likely also lead to the domination of the field by no code platforms that rely on the former to offer AI services to customers with no technical background, comparable to those that are today available mostly to professionals.

For this reason it was deemed important that this thesis was dedicated to this technology, not only in the form of a simple literature review, but as a demonstration of its applications. The applications chosen, in chapters 3 and 4, concern two of the most frequently appearing fields in transfer learning: natural language processing and computer vision. In this way, we hope to present a comprehensive analysis of transfer learning and the state of its current practical applications.

1.2 Goal

The goal of the thesis is to provide background theoretical information on the technologies of machine learning, transfer learning, some of the state of the art algorithms that are currently used in the fields of natural language processing and computer vision, as well as a thorough view into the capabilities of the technologies mentioned, showcased in real world applications.

1.3 Thesis Outline

The thesis presented consists of five chapters: in the first and current chapter the introduction to the rest of the thesis is made as a way both to help guide the reader as well as inform of the thesis's goal and motivation. In the second chapter, background theoretical information is presented for machine learning, transfer learning and the algorithms that have been used in the case studies of the next chapters. Algorithms considered the conceptual predecessors of the used algorithms are also presented in that chapter. Chapters 3 and 4 are the case studies, where in chapter 3 a computer vision study is examined, namely the use of transfer learning computer vision models to predict the existence of verticillium fungus on olive trees with the usage of image data captured by unmanned aerial vehicles, while in chapter 4 the case concerns the classification of surgery outcome regarding the rate of cytoreduction in patients, with the use of textual data. Chapter 5 is the discussion and the colcusions derived from the thesis.

Chapter 2

Background

2.1 Machine Learning Theory

Throughout history, the mechanisms of approach of problems has been the focus of research. Until very recently, the process of problem solving required careful examination of the problem's details and a strict enough definition of the solution of the problem that would allow for the development of a solution that would be considered reproducibly satisfactory. Though this approach had served humans very well in a plethora of tasks, these tasks were mainly simplified versions of real-world problems, that could be perfectly modelled mathematically. However, there was an array of problems that simply could not be addressed in this way.

That was due to two main reasons that could already be seen in the description of the nature of the problems that could be solved through the previous approach: the problems needed to be both strictly defined and their solutions to be completely rigid, in the sense that every part of the solution should be able to be named and explained.

In real-world problems, it is often the case that the person required to solve the problem does not know the exact specifics of the problem and thus cannot, alone or with the help of others, model a solution. This is especially true in the case of highly complex problems that involve components that are not fully understood. One such case is prevalent in the medical sector, where often, through conventional means, it is not possible to determine the reason for a patient's reaction to a disease, since that can be affected by health conditions not yet discovered or understood. Approaching this problem traditionally has a high chance of eventually leading to no, or worse, erroneous conclusions, depending on the doctor's experience and general aptitude. Another, not so obvious reason for which the traditional approach could fail to deliver results could be an indirect version of the cause mentioned above. All the variables could be understood one by one, but lack of knowledge about their interactions could cost the doctors the solution.

The second family of cases where the traditional approaches fail, concern the solution of the problem itself. In the real world, problems can be strictly defined but their solution can still not be strict enough. One such example is the recognition of images. A task that humans are naturally able to do and one that can be considered as well defined, has proven to be nearly impossible for a traditional approach to model and solve. That is due to the fact that even though the goal sounds straightforward, image recognition involves so many mechanisms that are impossible to map accurately, leading to solutions that can be applied only locally without generalization capacities, and often their ability derives from human knowledge. An example for that task is the recognition of dog and cat faces. The two differ greatly but there is no single metric that could capture that difference perfectly. A traditional approach could rely on human knowledge to insert that metric into the equation. For example, knowing that cat faces on average are smaller that those of dogs could indeed prove to be a way for the difference to be modelled. As mentioned previously though, this solution would work on only a subset of the real problem of image recognition, since every item in the world would require a distinct knowledge-based rule to separate it from other items; the task would become too large to model, and would become effectively impossible.

For these reasons, a new approach, or better yet, a new field of approaches was needed to address those tasks that it was impossible to frame in a traditional way. Artificial Intelligence (AI), whose beginnings were marked by Alan Turing's mention of intelligent systems (Turing, 2009), in their effort to approximate human intelligence, have revolutionized the way in which problems are approached and solved. The inherent noise in the problems' formulations that could be attributed to the way the natural world behaves, has been successfully overcome by humans and all living organisms through evolution. The success of Artificial Intelligence in mimicking human thought results indirectly in creating an approach to solve those problems that fell into one of the two cases mentioned in the above paragraphs.

Machine Learning (ML), considered a subset of AI, is the implementation of AI in problem solving tasks, and has been the primary way of addressing problems that conventional programming cannot as easily solve. In other words, ”Machine learning describes the capacity of systems to learn from problem-specific training data to automate the process of analytical model building and solve associated tasks” (Janiesch et al., 2021a). In contrast to classical programming, that combined input data with classical human-made algorithms to lead to certain solutions and results, ML leverages data, usually annotated but not exclusively, with expected results to produce a program that can solve the problem at hand (Figure 2.1).

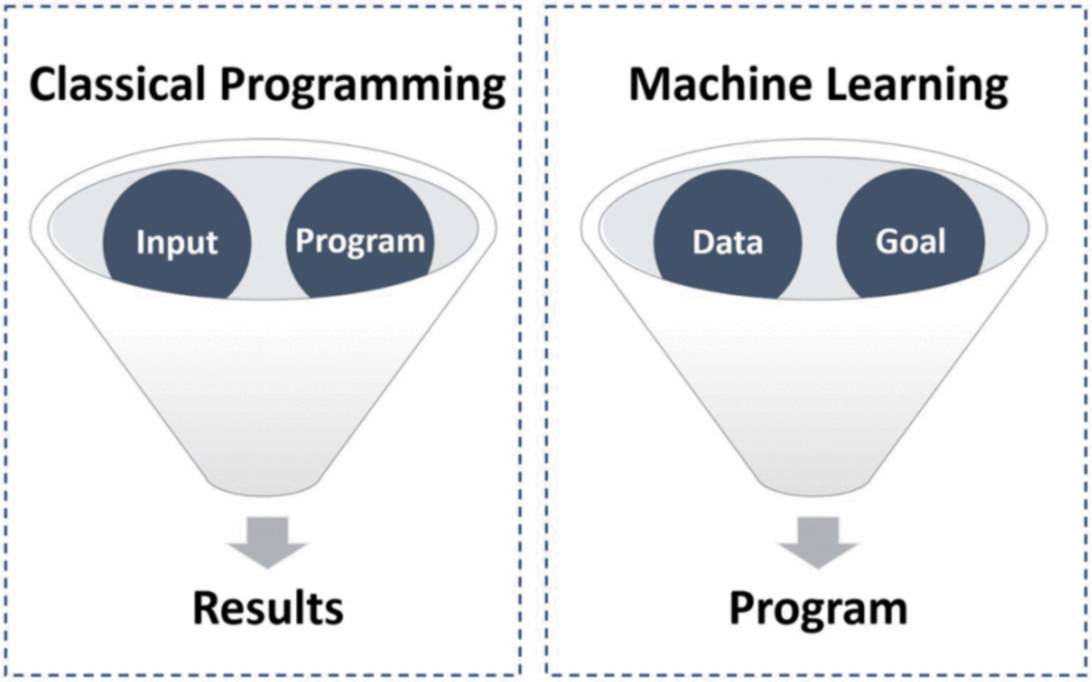


Figure 2.1: The difference between classical programming and machine learning (Krichen et al., 2022)

This way of approaching the problem solving task entails the use of large amounts of data. Though not always necessary, especially in problems of smaller complexity where a solution is usually easy to create, there are more than a few instances where, depending on the machine learning algorithm used, in order for a good solution to be reached, vast amounts of data are needed. In previous eras of information, the tasks of collecting and preprocessing data to prepare them for machine learning algorithms was extremely time consuming if not simply impossible. With the advent of the internet and the technologies that ensued to cover the needs that were created in the process, the creation, collection and preparation of data have become tasks that can be completed with relative ease. This has enabled the creation of extremely complicated machine learning models -a term coined to describe the results of the application of the machine learning algorithms- that have the capacity to perform tasks that rival humans or even in some cases, surpass their abilities.

Machine learning consists of three main branches, whose usage depends on the availability of data and the task at hand. The first branch is the one mentioned previously, where by the combination of input data and results, a machine learning model is created that can approximate a solution to the problem. This branch of machine learning is called Supervised Machine Learning (SML). Supervised machine learning has the advantage of being able to reach high performance in most tasks, and the ability to output results that can be evaluated; something that is of critical importance in cases where uncertainty of model performance can be costly to the model's users. Supervised machine learning depends highly on the given data as well as the annotation that accompanies them (the term annotation is used to signify the task's expected results). This entails several requirements that need to be fulfilled if the model is expected to function correctly. But firstly, the quality of the data is of the utmost importance.

The quality of data can be both task-independent and task-dependent (Pipino et al., 2002). By the term task-dependent we refer to the fitness of the data, related to the task at hand: it is often the case that one dataset is of poor quality for one task but of excellent quality for another. Quality in this sense in used to translate the notion of predictive strength. For example, assuming that the problem that needed to be solved was the prediction of the value of the NASDAQ index tomorrow, a dataset of cheese production in Italy per region in 2001 is not a good quality dataset for that purpose. That is due to the fact that relatively few insights can be drawn from the data source that could guide the model's prediction towards the right direction. However, if we wanted to predict cheese consumption in Italy in 2001, the same dataset would be almost perfect. The structure and content of the data didn't change, quality in this case was affected solely by the goal. On the other hand, task-independent data quality is the product of a few objective criteria. The most important of which is veracity. The term veracity is used to denote truthfulness. Data can be untruthful due to a variety of reasons, like human data collection and entry error, inherent noise in the data etc. High veracity data is much more valuable regardless of the task that needs to be solved. The second largest task-independent data quality measure would certainly be completeness. Completeness is the extent to which data is not missing (Pipino et al., 2002); something extremely important for every analysis.

Apart from data quality, another contributing factor to the supervised model's ability to approximate solutions to problems is the volume of the data. As the problems that require a solution increase in complexity, so does the necessary volume of data. High volume is necessary to enable the machine learning model to capture complex relationships between features. This second requirement has become more and more easy to satisfy, since data is becoming increasingly available, not just to enterprises and companies but also to the common user.

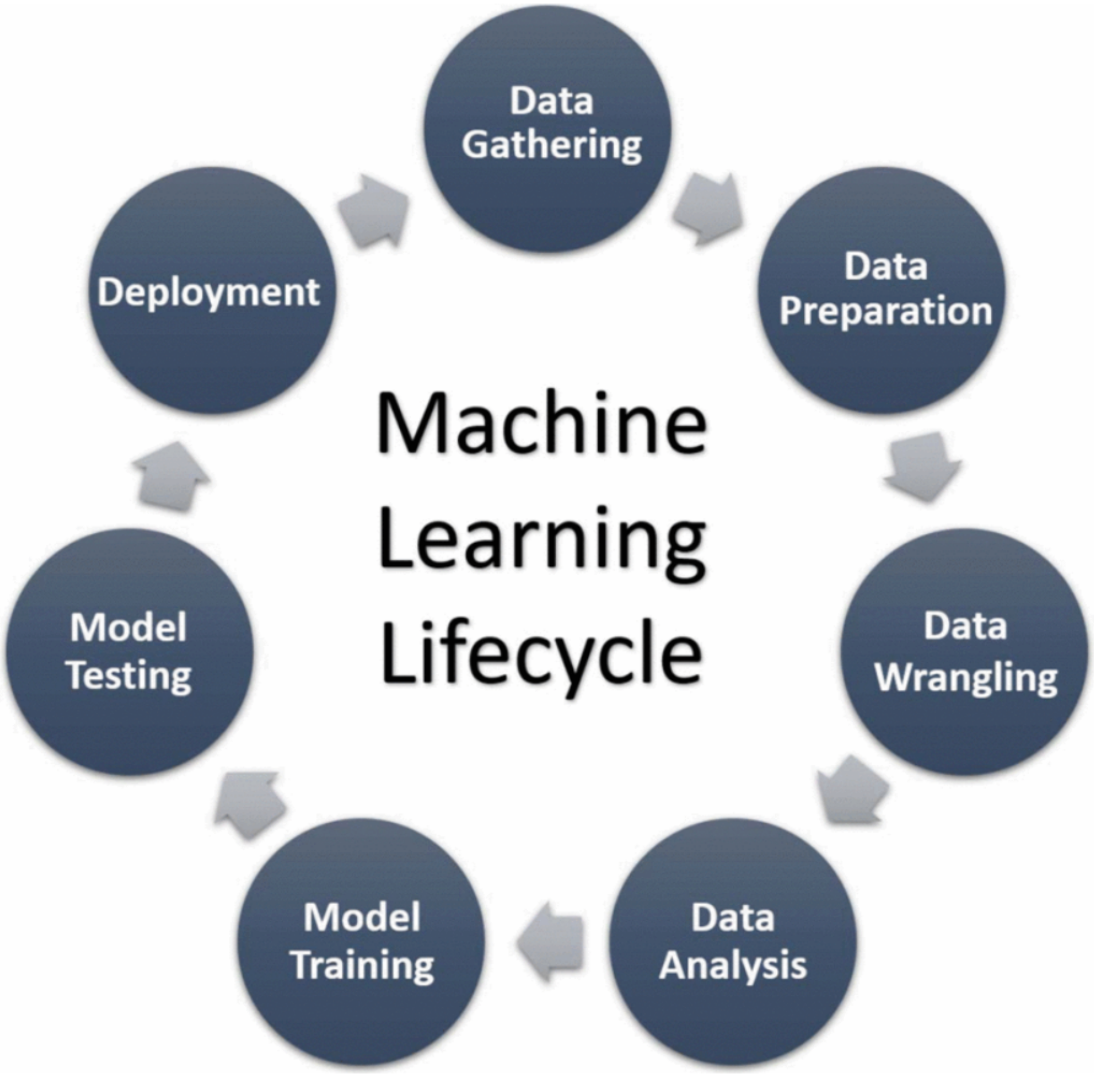


Figure 2.2: Machine learning lifecycle (Krichen et al., 2022)

Supervised Machine Learning is split into two main categories of problems, depending on the variable that is to be predicted in each case. These two main tasks are regression and classification. Regression is used when the dependent variable is a real number, while classification is used in the case that the dependent variable is categorical (Figure 2.4). These two categories of problems cover a massive field of possible applications, since almost any predictive problem can be generalized as a form of one of those two. Examples of real-world classification problems can be the prediction of loan default (Madaan et al., 2021; Jiang et al., 2018a), customer churn (Tsai and Lu, 2009; Huang et al., 2012), prediction of discrete ratings of a service (Shah et al., 2020), and text classification (Ciarelli et al., 2009). The applications of regression are also equally numerous compared to those of classification and concerns a variety of fields such as medicine (Taloba et al., 2022; Steyerberg et al., 2014), economics (Ghoddusi et al., 2019) and even music (Yang et al., 2008).

Classification is further divided into three typical problem scenarios. The first and probably most common scenario is binary classification (Kumari and Srivastava, 2017). In binary classification, the predicted discrete variable is dichotomous. Examples of such dichotomous dependent variable problems can be the prediction of boolean outcomes (yes-no, true-false). The case studies in the present thesis, are also binary classifications, as shown later, in chapters 3 and 4.

In is important to note another classification scenario: one-class classification. One-class classification can also be considered, surprisingly, a special form of binary classification, since the pool of possible outcomes is also dichotomous. However the difference between the two lies in the fact that one-class classifiers have been trained on datasets that only indicate the belonging to one class, while binary classifiers have used training data that contained information about both classes. One-class classification is used widely in outlier detection tasks such as fraud detection (Kamaruddin and Ravi, 2016; Zheng et al., 2019) and anomaly detection (Wei et al., 2018).

The second large category of classification models is multi-class classification. In problems of that nature, the model is called to predict the outcome from a pool of three or more possible outcomes. In such classifications the dependent variable can have a wide range of cardinality. Examples of low cardinality dependent variables can be the diagnosis of disease in a patient (Siddiqui et al., 2017) from a small pool of possible diseases, while examples of high cardinality classifications can be the classification of a person's identity through a biometrics system such as iris recognition (Rehman, 2021) and fingerprint recognition (Hammad and Wang, 2017). In these later cases the classifier is called to predict the person's identity from a pool of thousands, or even millions of people.

Lastly, multi-label classification (Herrera et al., 2016) is the branch of classification that concerns cases where the class of the dependent variable can be more than one. Multi-label classification is especially important in cases where the observation for which the prediction is made doesn't have strictly one class. An example of such cases can be the prediction of a movie's genre (Wehrmann and Barros, 2017), since a movie can simultaneously have more than one genres.

The implementation of each individual classification algorithm can lead to sensitivity to imbalanced datasets that will have to be mitigated accordingly to lead to classifications of high quality. The techniques used to fix class imbalance can be over and under sampling as well as synthetic data augmentation (Fernández et al., 2018; Tang et al., 2008; Burnaev et al., 2015; Zheng et al., 2015; Rayhan et al., 2017).

The usual procedure followed by machine learning engineers in supervised machine learning is to split the dataset into three partitions (Reitermanova et al., 2010); training, validation, and testing. The training partition is used to train the algorithm and create a model, however alongside the initial training, the validation dataset is usually used to provide an unbiased way for the model to be evaluated and its hyperparameters tuned, where by the term hyperparameters we denote the algorithm specific parameters. Lastly, the testing set is used to evaluate the created model's performance on completely unseen data; essentially a simulation with the goal of seeing the real-world level of success that the model is expected to have if the data remains the same. The are many variations of the setup that can be used ranging from small differences such as using only a testing and training split (Tan et al., 2021), or changing the relative percentages of the dataset that are to be allocated to the split datasets (Rácz et al., 2021), to the implementation of constant evaluation techniques such as cross validation (Berrar et al., 2019; Stone, 1978), where the data is iteratively split N times in 2 non-overlapping chunks of set percentage where in every iteration of the method, the newly created chunks will be used as train and test partitions. This provides a way of constant monitoring of the model's performance without losing the information present in the validation dataset in contrast to the approach of setting a partition exclusively for validation.

Supervised machine learning has the added advantage of being able to use evaluation metrics to, as the name implies, assess model performance as mentioned previously. Evaluation methods differ between regression and classification. In regression the most commonly used evaluation metrics rely on the calculation of the difference between real and predicted values (Botchkarev, 2018). Some of those widely used metrics follow below:

- Mean Absolute Error:
$$MAE = \frac{\sum_{i=0}^{N-1} |y_i - \hat{y}_i|}{N} \tag{2.1}$$

- Mean Squared Error:
$$MSE = \frac{\sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2}{N} \tag{2.2}$$

- Root Mean Square Error:
$$RMSE = \sqrt{\frac{\sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2}{N}} \tag{2.3}$$

- Mean Percentage Error:
$$MPE = \frac{100\%}{N} \sum_{i=0}^{N-1} \frac{y_i - \hat{y}_i}{y_i} \tag{2.4}$$

- Mean Absolute Percentage Error:

$$MAPE = \frac{100\%}{N} \sum_{i=0}^{N-1} \frac{|y_i - \hat{y}_i|}{|y_i|} \tag{2.5}$$

- R^2 - Coefficient of Determination:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{2.6}$$

, where in all of the above, y_i is the true value, $\hat{y}_i$ is the predicted one, and N denotes the number of observations.

In classification tasks, given that the prediction is mostly in the form of a boolean value, the calculation of the evaluation metrics is performed on the basis of comparing the amount of correctly and wrongly classified observations (Hossin and Sulaiman, 2015; Fatourechi et al., 2008; Vujović et al., 2021). Correctly classified observations, also known simply as 'True', are those were the predicted category agrees with the real one, whereas incorrectly classified observations, or 'False', are the opposite.

Each type of classification, from the ones mentioned previously, has its own way of evaluating the performance of the models, with all of them being generalizations of the metrics used in binary classification. The most common metrics in binary classification are the following:

- Accuracy:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.7}$$

- Precision:

$$PREC = \frac{TP}{TP + FP} \tag{2.8}$$

- Recall or Sensitivity:

$$RECALL = \frac{TP}{TP + FN} \tag{2.9}$$

- Specificity:

$$SPEC = \frac{TN}{FP + TN} \tag{2.10}$$

- False Positive Rate:

$$FPR = \frac{FP}{FP + TN} \tag{2.11}$$

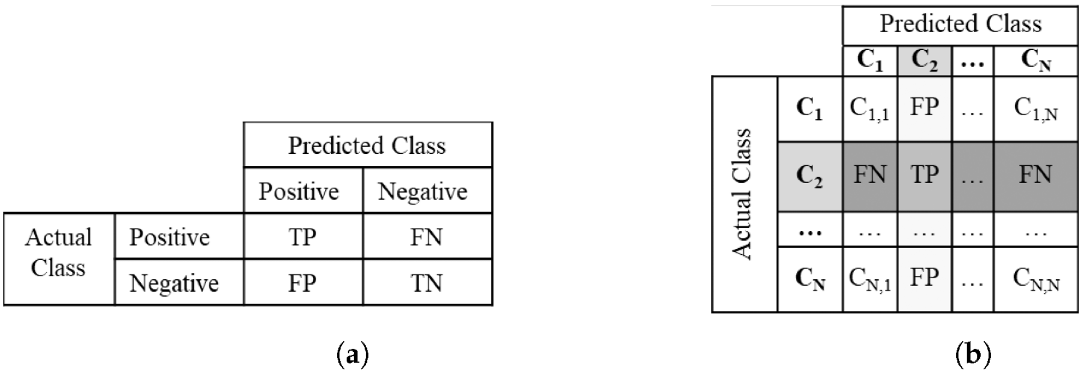
- F1 Score:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{2.12}$$

- Area under the Receiver Operating Characteristic curve (AUROC), quantified as the area under the curve formed by plotting the Sensitivity against the False Positive Rate at incremental threshold values

- Area under the Precision-Recall curve (AUPRC), quantified as the area under the curve formed by plotting the Recall against the precision at incremental threshold values

, with TN, TP, FN and FP being true negatives, true positives, false negatives and false positives respectively. The terms positive and negative serve as a way to distinguish between the two classes, with the appointment of the classes being instantiated in the problem definition.



		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

(a)

		Predicted Class			
		C_1	C_2	…	C_N
Actual Class	C_1	$C_{1,1}$	FP	…	$C_{1,N}$
	C_2	FN	TP	…	FN
	…	…	…	…	…
	C_N	$C_{N,1}$	FP	…	$C_{N,N}$

(b)

Figure 2.3: Example of a confusion matrix in a binary classification (a) and a multi class classification (b), a diagram widely used to show true and false positives and negatives in a classification. (Markoulidakis et al., 2021)

The choice of the right evaluation metric is a product of the problem itself and the approach adopted (Hauser and Katz, 1998; Taha et al., 2014; Mackie et al., 2014; Schröder et al., 2011).

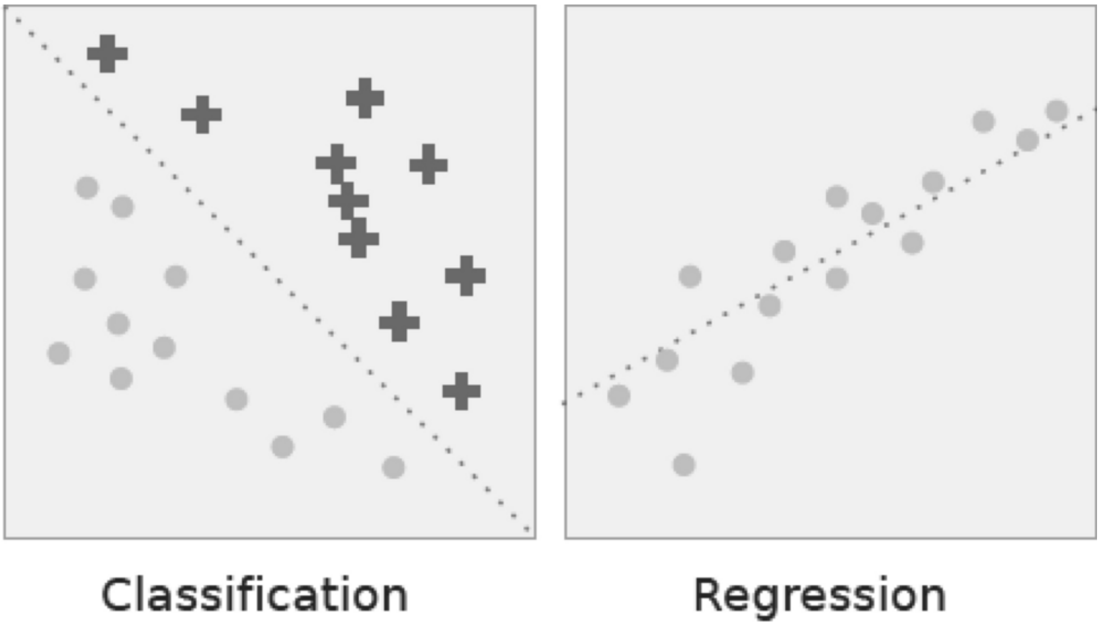


Figure 2.4: Classification vs. regression. In classification the dotted line represents a linear boundary that separates the two classes; in regression, the dotted line models the linear relationship between the two variables (Sarker, 2021)

Even though the aforementioned supervised learning approach has a wide variety of uses and is the machine learning branch used in this thesis, it is important to also mention briefly the two other branches. Supervised machine learning is not the only branch, due to a simple fact: annotated data are not always available. This need gave birth to the second branch of machine learning, Unsupervised Machine Learning (UML). Unsupervised machine learning consists of algorithms that don't rely on results to approximate solutions to problems, but rather use the knowledge hidden within the dataset to create patterns. These pattern can either be groups within the data (clustering), association rules etc. Unsupervised machine learning has the advantage of not needing expected result data to operate, but this comes at the cost of missing a way to evaluate the performance of the algorithm.

Lastly, Reinforcement Learning (RL) is the third and last branch of Machine Learning that mainly concerns problems of decision making. As mentioned in Wiering and van Otterlo (2012), ”situated in between supervised learning and unsupervised learning, the paradigm of reinforcement learning deals with learning in sequential decision making problems in which there is limited feedback”. It has found applications in robotics and optimization problems and while widely used and useful, it is beyond the scope of this thesis to further delve into the mechanics of the methods contained in reinforcement learning.

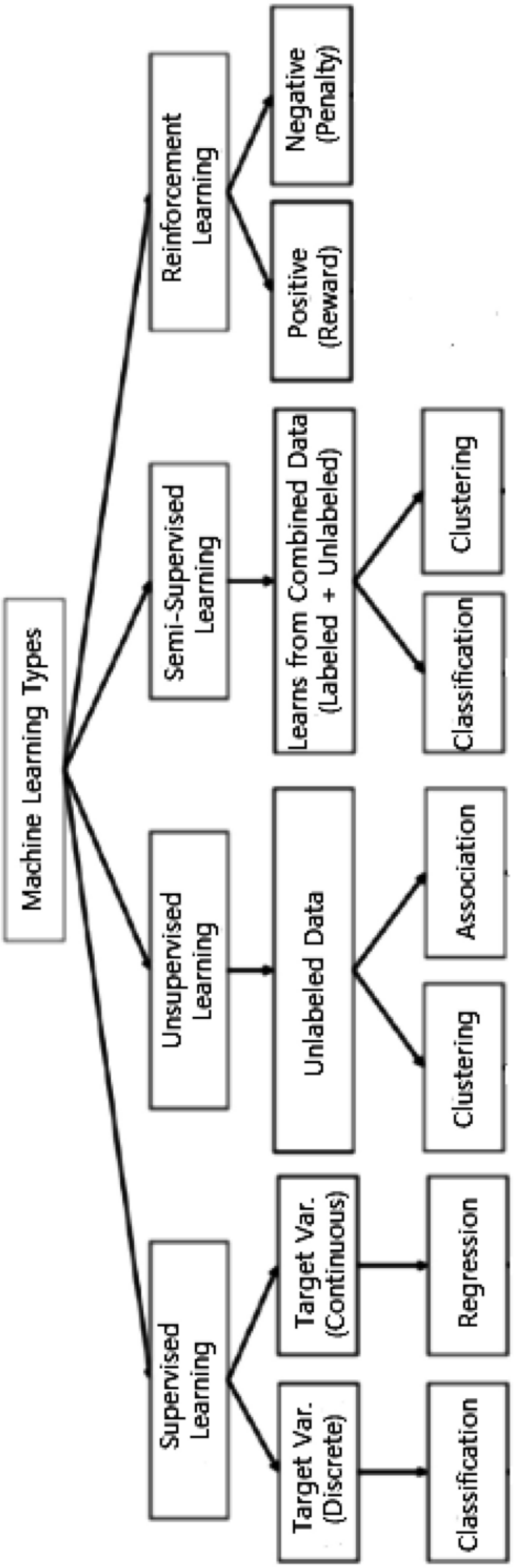


Figure 2.5: Various types of machine learning techniques (Sarker, 2021)

2.1.1 Deep Learning

Deep learning is considered a subset of machine learning and consequently, of artificial intelligence as a whole (Figure 2.7). More specifically, deep learning refers to a special subset of a family of algorithms in machine learning, Deep Neural Networks (DNNs), that are Artificial Neural Networks (ANNs) with more than one hidden layer (Janiesch et al., 2021b). In order to fully understand what that means it is important to firstly analyze what ANNs are and how they work.

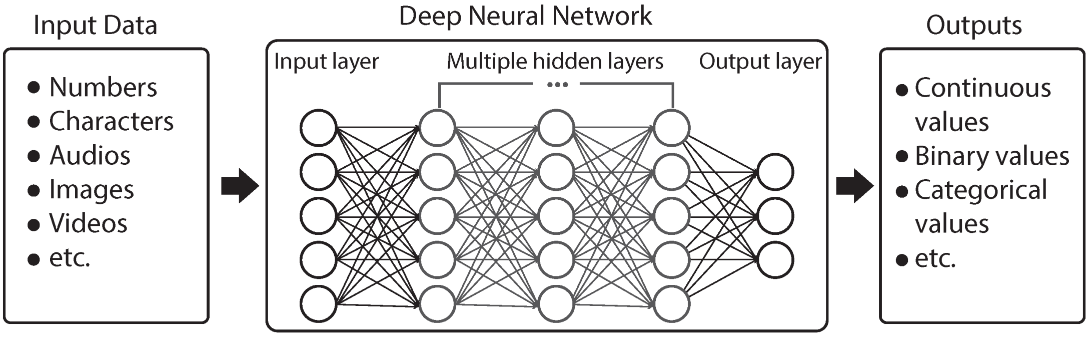


Figure 2.6: Diagram of a deep neural network (Zhu et al., 2023)

Artificial neural networks are a family of machine learning algorithms that mimic the way brain neurons work to solve problems (Krogh, 2008). Essentially, as in biological life forms the neurons are activated due to certain stimuli (visual, haptic etc.) that lead to certain responses, so are artificial neurons, also known as nodes, who accept input, and produce an output. The input can be of any form of data encoded as bytes. It is not uncommon for neural networks to accept as input data derived from videos, text or images. The output can be integers or floats, depending on the task. The activation of the artificial neurons is performed by the activation function, that aggregates the input to produce the output of the neuron. In this way, neurons can be ”chained” so that the output of one can be passed as the input to the next. This property of the neural networks creates an architecture of layers of nodes. The layers can be of three kinds: either input layers, that accept the initial input, output layers, or hidden layers that are responsible for aggregating the input through activation functions before passing it as the output to the next layer (Figure 2.6). Layers of nodes often include a bias node, that has the value of one, to aid the network through adding a constant in the calculation of the outputs. This chaining of hidden layers, enables the network to capture complex patterns in the data, thus becoming a universal function approximator (Csáji et al., 2001). As stated in LeCun et al. (2015), ”Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction”. As also seen in later sections, architectures can vary, with some being very effective in a selection of tasks but ineffective in others.

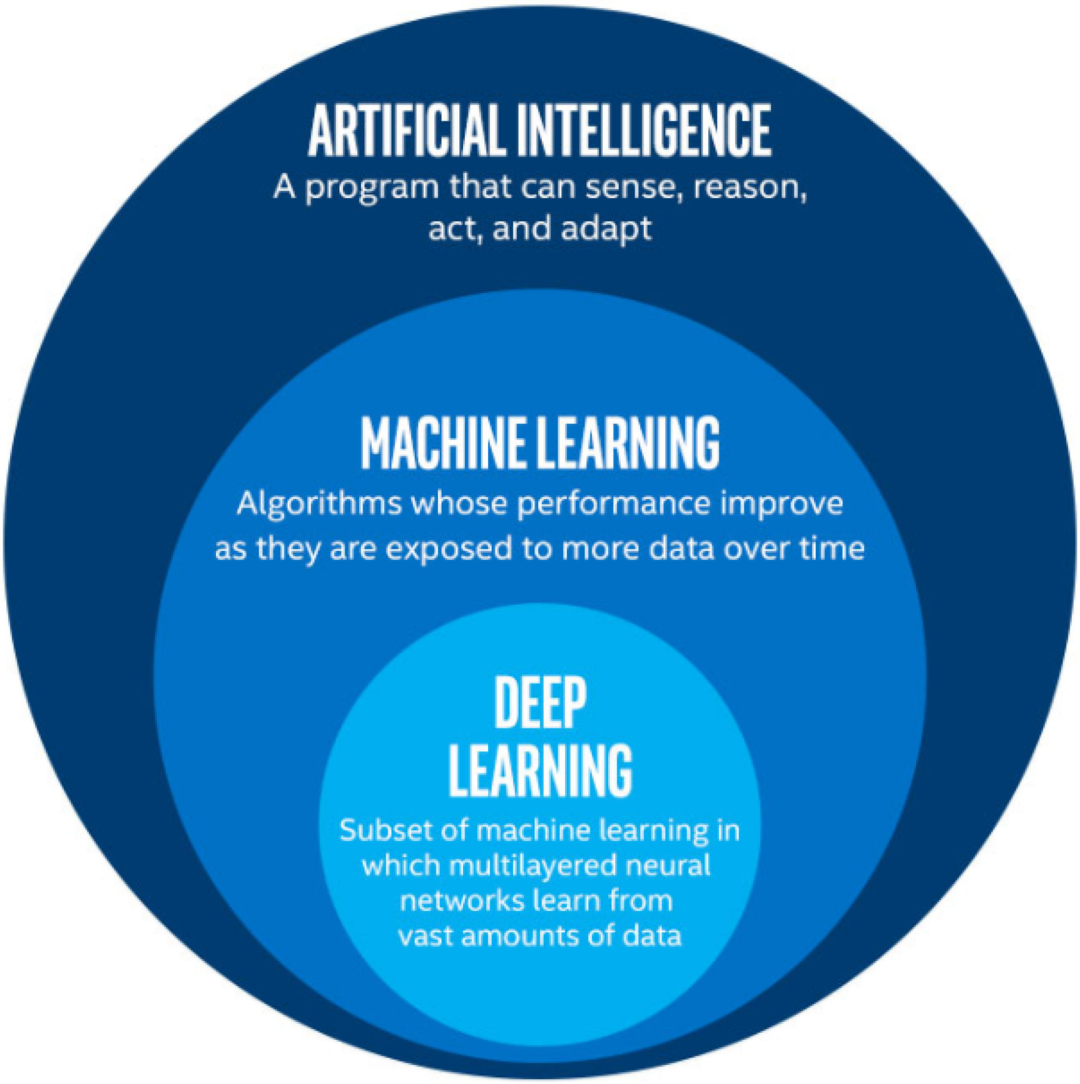

ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

Figure 2.7: The relationship between Artificial Intelligence, Machine Learning and Deep Learning (Arooj et al., 2022)

Delving more deeply into the mechanics of artificial neural networks, the activation function is considered by some to be the single most important point of the network's architectural choices (Ding et al., 2018). The choice of the function dictates how different nodes interact with one another and consequently what feature interactions are created and passed to the next hidden layer, or the output layer of the network. The activation functions are also of critical importance to the model's output, as different functions are suited best for different problems. For example, the sigmoid function is widely used in binary classification problems, while the softmax function is used in multi-class classification ones. It is also not uncommon to not use an activation function at all during certain parts of the network, or to be more precise, to use the identity function ($f(x) = x$), called the linear activation. This function is used as the output node's activation function in regression tasks. Below follow some of the most common activation functions used:

- Sigmoid:
$$s(x) = \frac{1}{1 + e^{-x}} \tag{2.13}$$
- Tanh:
$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{2.14}$$
- ReLU:
$$relu(x) = max(0, x) \tag{2.15}$$
- LeakyReLU:
$$lrelu(x) = max(0.1x, x) \tag{2.16}$$
- Parametric ReLU:
$$prelu(x) = max(ax, x) \tag{2.17}$$
- ELU:
$$elu(x) = \begin{cases} x, & \text{if } x \geq 0 \\ a(e^x - 1), & \text{if } x < 0 \end{cases} \tag{2.18}$$
- Softmax:
$$softmax(x) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}, \quad for\ i = 1, 2, \ldots, K \tag{2.19}$$

The next most important detail is the way the network ”learns”. For each connection of nodes, the network appoints a weight that is initialized either randomly or through an algorithmic procedure. Weight initialization is an extremely active research topic (Kumar, 2017; Narkhede et al., 2022; de Sousa, 2016), and as seen in the next chapters, it plays a very important role in transfer learning. For every instance of training data, meaning every individual observation, the model passes the variables of the observation to the input layer of the network, and following the activation functions and the network's connections, results in the output node that is responsible for the predicted variable's value. The predicted value is then compared with the true value, through a loss function and the model's loss is calculated. Then the change in model weights that would be more beneficial to the model's performance is calculated through an optimization algorithm, usually gradient descent, and the weights are updated for the next iteration. The mechanism of calculating the updates needed to be done to the model's weights for every previous layer of the network is done with the process of backpropagation where the error is ”propagated” to the previous layer. The loss functions and the optimizers used are numerous and are also a vast research field that constantly evolves (Wang et al., 2020a; Janocha and Czarnecki, 2017; Steinwart, 2007; Bera and Shrivastava, 2020; Vani and Rao, 2019).

2.1.2 Transfer Learning

Transfer learning is considered to be the future of machine learning. It has become the proposed solution to the problem of creation of high performance models when data are few or when computational resources are limited. Given that larger models tend to perform better but need a bigger volume of data to train on, while at the same time computational efficiency has seemingly reached a plateau, transfer learning seems to be the only viable way to make state of the art machine learning solutions available to the general public.

Transfer learning can roughly be defined as the usage of knowledge learned in one task applied to a related task in order to boost performance. A mathematical definition is given by (Weiss et al., 2016): let D be a domain defined by a feature space χ and a marginal probability distribution $P(X)$ where $X = \{x_1, x_2, \ldots, x_n\} \in \chi$ such that $D = \{\chi, P(X)\}$. Also, let T be a task defined by a label space Y and a predictive function $f()$ learned from corresponding pairs belonging to the feature space and the label space $\{x_i, y_i\}$, where $x_i \in X, y_i \in Y$ such that $T = \{Y, f()\}$. If D_s is a source domain with a corresponding source task T_s, and D_t the target domain with a corresponding source task T_t, transfer learning is the process of improving the target predictive function f_t by using the related information from D_s and T_s where $D_s \neq D_t$ and/or $T_s \neq T_t$.

Building on that definition we can further define homogeneous and heterogeneous transfer learning. If $D_s \neq D_t$ and since $D_s = \{\chi_s, P(X_s)\}, D_t = \{\chi_t, P(X_t)\} \therefore \chi_s \neq \chi_t$ and/or $P(X_s) \neq P(X_t)$. If $\chi_s \neq \chi_t$ then this will be heterogeneous transfer learning, while $\chi_s = \chi_t$ is referred to as homogeneous. In simpler terms, homogeneous transfer learning is when the source feature space is the same as the target's, with the opposite being the case of heterogeneous transfer learning. Regarding the distributions P_s and P_t, it is noted that the more they differ the less transferable the knowledge becomes (Shimodaira, 2000).

Another case of dissimilarity between the source and the target is if the tasks differ ($T_s \neq T_t$). This case is called inductive transfer learning (Pan and Yang, 2010). Following the same logic as before, if $T_s \neq T_t$ and since $T_s = \{Y_s, f()_s\}, T_t = \{Y_t, f()_t\} \therefore Y_s \neq Y_t$ and/or $f()_s \neq f()_t$. If $f()_s \neq f()_t$ then the conditional probability distributions between the source and target domains are different (feature context bias) (Weiss et al., 2016). This is also a major factor in the ”transferability” of knowledge between source and target. Similar is the case with the inequality of Y_s and Y_t. Different label spaces are a major barrier in transfer learning that tends to be mitigated with transformations in the source or target data. In contrast to the above, in traditional machine learning, both $D_s = D_t$ and $T_s = T_t$.

Special settings of transfer learning can also exist. Such is the case with transductive transfer learning where $T_s = T_t$ and $D_s \neq D_t$. Another unique situation is unsupervised transfer learning where T_t is different but related to T_s and both tasks are related to unsupervised problems.

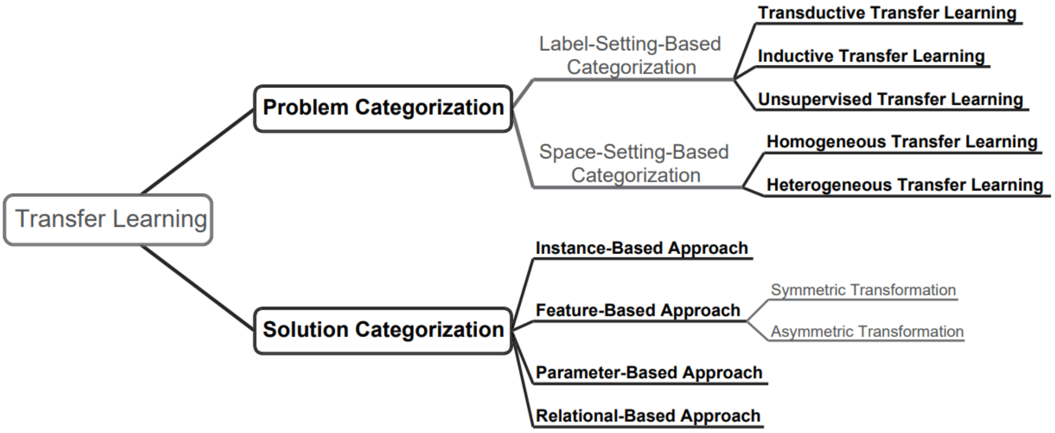

Transfer Learning
Problem Categorization
Label-Setting-Based Categorization
Transductive Transfer Learning
Inductive Transfer Learning
Unsupervised Transfer Learning
Space-Setting-Based Categorization
Homogeneous Transfer Learning
Heterogeneous Transfer Learning
Solution Categorization
Instance-Based Approach
Feature-Based Approach
Symmetric Transformation
Asymmetric Transformation
Parameter-Based Approach
Relational-Based Approach

Figure 2.8: Categorizations of transfer learning (Zhuang et al., 2020)

The application of transfer learning is plagued by many other issues such as a difference in label imbalance between source and target (frequency feature bias), but generally any difference between the distributions of features or labels tend to affect the transfer of knowledge negatively. In some cases, even negative transfer can occur. This will result in the $f()_t$ becoming weaker if trained with a transfer learning approach than it would be without.

The ways to mitigate those barriers vary depending on the definition of the problem but plenty of methods have been created that are successful in this effort. More specifically, seven main approaches have been created, whose descriptions follow.

- **Instance-based transfer learning** aims to use data points selected from the source domain that are similar to those in the target domain to train the target predictive function. This approach is extremely useful in cases where data are few in the target domain.

- **Feature-based transfer learning**, where features are transformed to be similar between source and target domains. This approach is split in two sub cases:

  - **Asymmetric**, where either domain's feature space is transformed to the other's.

  - **Symmetric**, where both feature spaces are used to determine a joint feature space.

- **Parameter-based transfer learning**, where the parameters learned through the training of the source predictive function are used to train the target predictive function. This is a very common scenario in deep learning transfer learning where pre-trained layers are ”frozen”, in the sense that training will not affect them, and the target predictor uses the features that are contained within to create its own prediction using additional layers that are not ”frozen”.

- **Relational-based transfer learning** aims to use knowledge that is common between the two domains to reach a point that transferability of knowledge increases.

- **Hybrid-based transfer learning** that utilizes a combination of the above.

From the above approaches all can be used in homogeneous transfer learning with the exception of feature-based transfer learning that can be utilized both in homogeneous and heterogeneous transfer learning.

Special cases that are considered part of transfer learning are the Few-Shot Learning (FSL), One-Shot Learning (OSL), and Zero-Shot Learning (ZSL) paradigms. In the cases of few-shot and one-shot learning, models are trained such that they can rapidly generalize to new tasks containing only a few samples of supervised information (Wang et al., 2020b). In zero-shot learning models, no supervised instances are needed for the model to perform the task on the new dataset. In this sense it is safe to say that zero shot models have reached such a point of generalization that can now perform the task that they were trained to do, regardless of the target dataset (Xian et al., 2019).

In practical terms, transfer learning techniques are applied mostly in deep learning. The main approaches adopted are the aforementioned ”freezing” of the pre-trained network up to a certain point, and using the learned parameters (weights) to train a second network with the target data, or the fine-tuning of the entire network with the new data. Both approaches are used widely with many state of the art algorithms utilizing them.

Applications of transfer learning have been made in a variety of fields, including, but not limited to, medical image recognition (Kim et al., 2022), atmospheric particle classification (Ma et al., 2015), marketing applications (Perlich et al., 2014), and many more.

2.2 Machine Learning Applications

2.2.1 Machine Learning in Tabular Data Tasks

Until recently the majority of machine learning applications had relied on the use of tabular data (Shwartz-Ziv and Armon, 2022). In this context we use the term tabular data to denote data structured in a set, two dimensional, way. Typically, this structuring follows the design of column-variable row-observation. However there are many formats that are used, depending on the purpose. Thanks to the already structured nature of the data, approaches have been created that can harness this property to output machine learning models of high predictive strength, that manage to outperform traditional statistical models that were until very recently the norm.

The added benefit of tabular data is their homogeneity. Analyses can take place on well structured tabular data without the need for clever data manipulations and transformations that are somewhat necessary with other types of data (e.g. textual data, audio data). Machine and deep learning can analyze tabular data of considerable volume, in a variety of settings such as finance (Clements et al., 2020), online marketing (Guo et al., 2017), medicine (Ulmer et al., 2020; Somani et al., 2021; Borisov et al., 2021) and anomaly detection tasks (Pang et al., 2022; Wang et al., 2022; Škvára et al., 2021), with many programming libraries written for that exact purpose (Pedregosa et al., 2011a).

Many different machine learning algorithms have been written and tested on prediction tasks in tabular data, both for supervised and unsupervised learning. Such examples are the Support Vector Machines (Hearst et al., 1998), the nearest neighbors algorithm (Taunk et al., 2019), the decision trees (Quinlan, 1996) and so on and so forth, with the algorithms being so numerous that listing them all would be redundant. However, there is one algorithm that stands out enough to be mentioned separately: the XGBoost algorithm.

2.2.1.A The Extreme Gradient Boosting Algorithm

Amongst the most commonly used, if not the most common, algorithm in prediction tasks that rely on tabular data is the Extreme Gradient Boosting Algorithm, or XGBoost for short (Chen and Guestrin, 2016). XGBoost comes as an improvement to already existing Classification and Regression Tree (CART) based machine learning algorithms, by adding a process of special ensembling (Figure 2.9); an aggregation of ”weak” models to produce a ”strong” final model. Thanks to the fact that XGBoost inherits many of the properties of classification and regression trees, it can be used for both tasks. The ensembling added to the already existing CART algorithm is a form of gradient boosting with regularization, where CARTs are iteratively added to the model and the errors of the previous ”weak” CART are used to train the next with the final prediction being a result of this procedure. Thanks to its unique structure, XGBoost has managed to outperform many of its predecessors, and even some of the newer algorithms, in a plethora of tasks (Liang et al., 2020; Hancock and Khoshgoftaar, 2020; Zamani Joharestani et al., 2019).

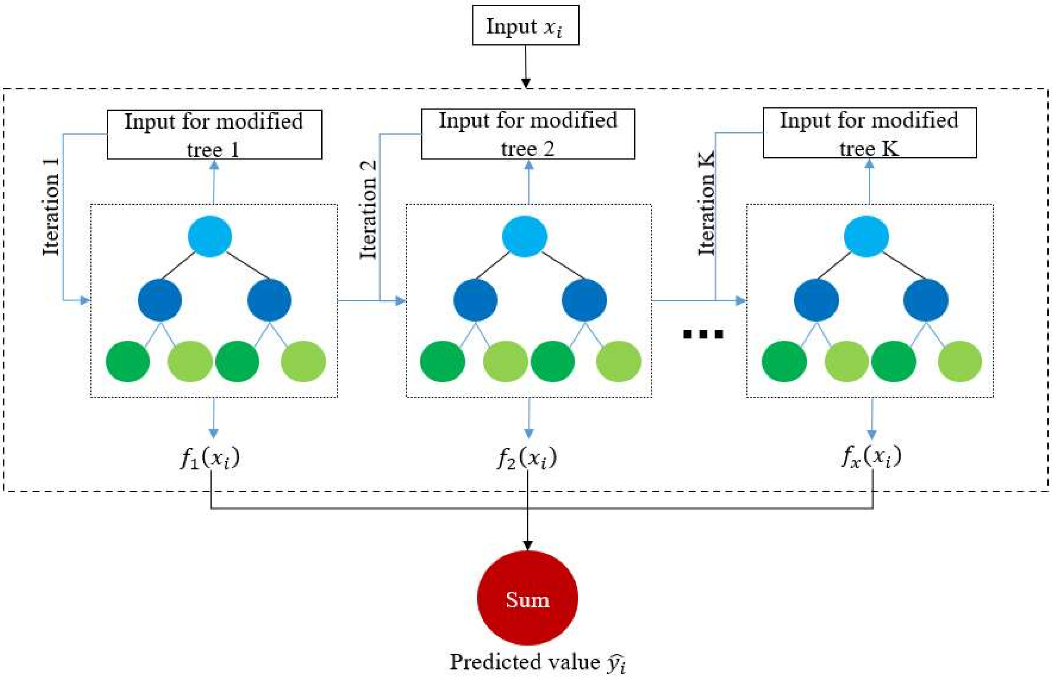

Input x_i
Iteration 1
Input for modified tree 1
Iteration 2
Input for modified tree 2
Iteration K
Input for modified tree K
...
$f_1(x_i)$
$f_2(x_i)$
$f_x(x_i)$
Sum
Predicted value $\hat{y}_i$

Figure 2.9: Schematic diagram of the XGBoost regression tree model (Zou et al., 2022)

2.2.2 Machine Learning in Image Tasks

One of the greatest barriers that classical machine learning has had to overcome has been the analysis of image data. The reason for this has been the fact that image data is unstructured, with images varying both in dimensions, and homogeneity of features. In tabular data the features usually occupy one column each, while the definition and location of the features in image data is a question with a more complex answer. To address such problems, a new field of artificial intelligence was created, computer vision (Forsyth and Ponce, 2002; Voulodimos et al., 2018). Computer vision encompasses all the tasks that have come to be, related to machine learning approaches in image data.

Computer vision techniques rely on deep learning, thanks to deep learning’s ability to perform automatic feature extraction and engineering. With the images passed as matrices containing color value information for every pixel, and through the implementation of deep learning algorithms, the most important of which will be covered in the next sections, computer vision has managed to outperform all classical algorithms in image tasks.

The tasks in computer vision are numerous and vary in complexity (Figure 2.10). Perhaps the most common, and simplest of all tasks is image classification. In image classification the task is to predict, usually from a limited pool of possible answers what the item portrayed in an image is (Lu and Weng, 2007). Due to the definition of the problem, such data are often images where only one item of interest is included. Image classification is considered a simple problem, always relative to other computer vision tasks, since the goal is straightforward and the prediction output either just one value (binary and multi-class classifications), or multiple values that are calculated in the same way as one value would (multi-label classification). As seen later, multi-object problems have had to adopt more complex approaches in order to be solved.

As a second step, and an advancement to simple image classification, localization and object detection have been incorporated in classification tasks. In object localization and detection the images are not only classified as a class depending on their contents, but in cases where it is applicable, the items are located within the image. The difference between the two methods is simply the number of items localized. In case only the item that is the focus of the classification is located, this is a task of localization. In contrast, when multiple objects are classified and located separately, then the task is that of object detection. In this way, object detection can be considered a specific case of object localization. In both tasks the algorithm not only has to classify the objects but also to predict the coordinates of their bounding boxes. Hence this problem is both a classification (object class), and a regression problem (coordinate prediction).

Lastly, in this same vein, instance segmentation is another task of computer vision where items are both classified and localized. Instance segmentation is a generalization of object detection since object coordinates are not simple bounding boxes that can be defined by 4 points, but polygons defined by a variable amount of points. Semantic segmentation is another form of instance segmentation where objects of the same class are segmented as a unified entity.

The tasks mentioned have become the building block upon which several advanced deep learning tasks are based. By incorporating textual data and language understanding models, applications such as image captioning (Hossain et al., 2019; Vinyals et al., 2016) and visual relationship detection (Lu et al., 2016) have been created. Also generative AI models trained on image data have made it possible to generate images of high quality (Teterwak et al., 2019), or augment already existing ones of lower quality (super-resolution) (You et al., 2019; Wang et al., 2020c).

The majority of the aforementioned applications are part of the supervised learning paradigm, that requires both data and expected output. However, approaches have been made that employ unsupervised learning algorithms, that often under perform when compared to their supervised counterparts. The advent of transfer learning has led to zero-shot classification solutions that overcome both previous paradigms' weaknesses.

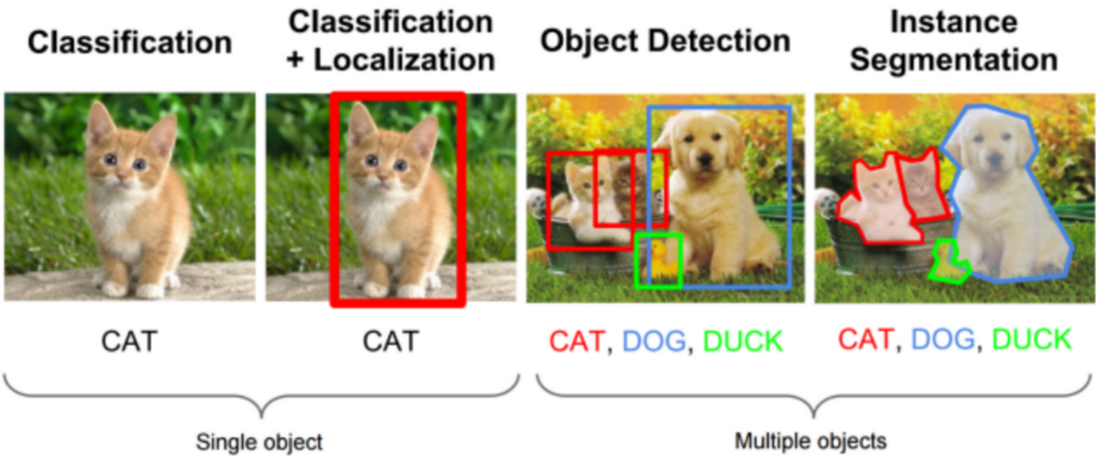


Figure 2.10: Image classification, localization, detection, and segmentation as tasks in computer vision (Jaiswal et al., 2021)

2.2.2.A The Convolutional Neural Network

The first major innovation for machine learning in image data tasks has been the introduction of the Convolutional Neural Network (CNN) (O'Shea and Nash, 2015). The convolutional neural network is a type of deep learning neural network architecture that is comprised of nodes that can capture hierarchically structured information by taking into account the proximity of the input nodes. In other words, by considering the proximity of pixels within an image, the CNN can capture information by firstly constructing simple features and by repeating the process of convolution within the network, it can gradually build more and more complex features based one the previously created ones. The result of multiple convolutions taking place in the network is reaching a point were very detailed features will be created and the classification of very complex images will be possible, while also staying relatively unaffected by the shift of the item within the image.

Convolutional neural networks are based on the convolution layer, a type of neural network layer that uses filter nodes that detect the presence of low level features throughout the image. The filters, or most commonly known as kernels, are essentially matrices usually of small sizes (3x3, 4x4, 5x5) that represent simple features, like lines, edges or points. The kernels perform the convolution over the image; an operation of ”sliding” the first matrix (kernel) over the second (image) and outputting a result matrix that is the dot product of the two at any given point that the kernel has slid. The result is a feature map of the image. The feature map is a representation of the presence of this particular feature on the original image. The next feature engineering operation happens at a later convolutional layer, that taking as input the results of the first, creates features of higher complexity by using the previously created features as building blocks. An example of such a procedure is the creation of the feature of crosses in the second convolutional layer, when the first convolution had created feature maps of vertical and horizontal lines, since a cross derives from the combination of two.

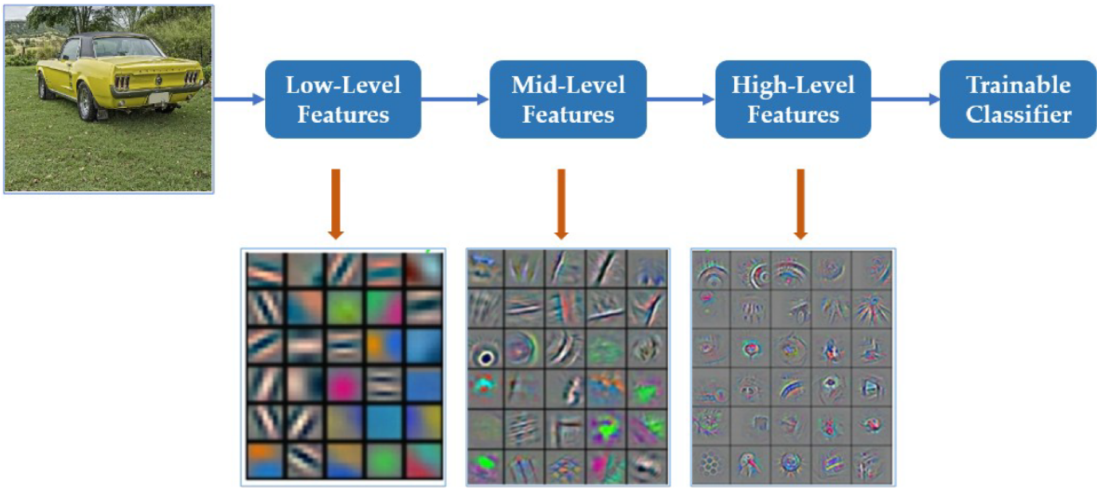


Figure 2.11: Visualization of features learned in different layers of a convolutional neural network. The network gradually learns more and more complex pattern based on the previously learned features, before finally proceeding to the classification. (Meng et al., 2019)

Given that the convolutional layers tend to be computationally expensive, the addition of pooling layers in the network is common practice. Pooling layers serve as a way to downsample an image. More specifically, the pooling operation is the aggregation of windows of the image or feature map matrices through a function (usually the average, max or min) to recreate a smaller sized matrix. Pooling layers have the benefit of reducing the need for many convolutional layers and thus also lower computational times. Another commonly added layer is a layer that contains a non-linear activation function. The purpose for that is to also add features that are not linear combinations of others, and can lead the network to create features such as curves from simple lines. Lastly, the created high level features are passed to the final section of the network, usually a fully connected layer, that performs the necessary task (e.g. classification).

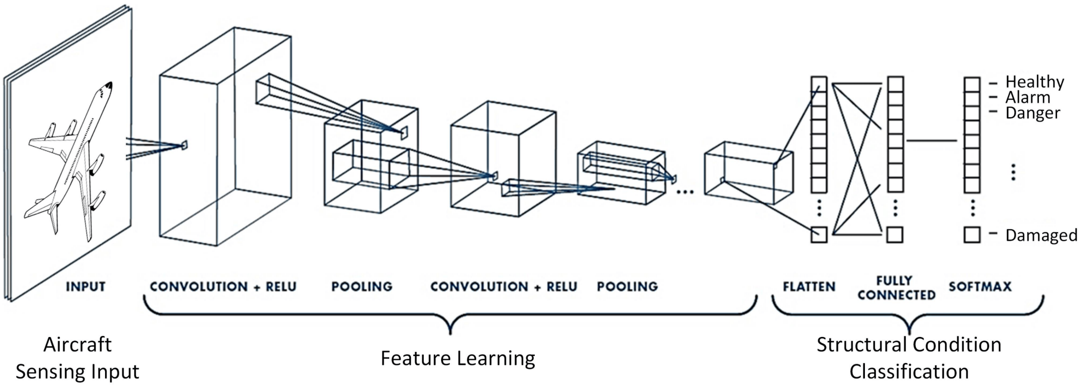


Figure 2.12: Example of a typical convolution neural network architecture (Tabian et al., 2019)

2.2.2.B The You Only Look Once Algorithm

The You Only Look Once (YOLO) algorithm proposed in 2015 adopts an approach of framing object detection as a regression problem to spatially separated bounding boxes and associated class probabilities (Redmon et al., 2015). Essentially, classification and bounding box location calculation happen simultaneously.

The algorithm begins by partitioning the image in an N x N grid and then for each cell of the grid, predicts bounding box locations, confidence of detection of an object, and class probability for every class, thus having the entire object detection and classification process happen in a single pass over the image.

YOLO, as a project, is actively developed, with newer and more enhanced versions coming out in the form of versions steadily throughout the last few years, with improvements made in the algorithm's speed and performance (Jiang et al., 2022) through the change of the algorithm's architecture or the addition of new capabilities. Each new release receives a new version number; we are currently in the 8th official release of YOLO, YOLOv8, however in the case presented in this effort, YOLOv5(Jocher, 2020) will be used. The reasoning behind the choice is that while newer versions of the YOLO algorithm have been developed, YOLOv5 has been the most recent version preferred by researchers in related works in the literature, as also shown in the next chapter.

The YOLO algorithm is especially powerful in the task of detecting olive tree infections by the Verticillium wilt from UAV images, since as shown in the literature (Jiang et al., 2022), it has both high inference speeds that allow for the classification of images in near real-time and can achieve high classification performance metrics. Actually, it is not an overstatement to say that it is currently the only algorithm that offers such a balanced solution, in terms of speed and performance, for the simultaneous object detection and classification task.

YOLOv5's network design is comprised of a Backbone, a Neck and a Head, as described by its authors. All of these, are essentially sub-networks that when assembled together in the right order, mentioned above, create the YOLOv5 model. The architecture of YOLOv5 that was used in the current effort, uses the New CSP-Darknet53 as the backbone, the SPPF and the New CSP-PAN as the neck and the YOLO head layer as the Head, as shown in Figure 2.13.

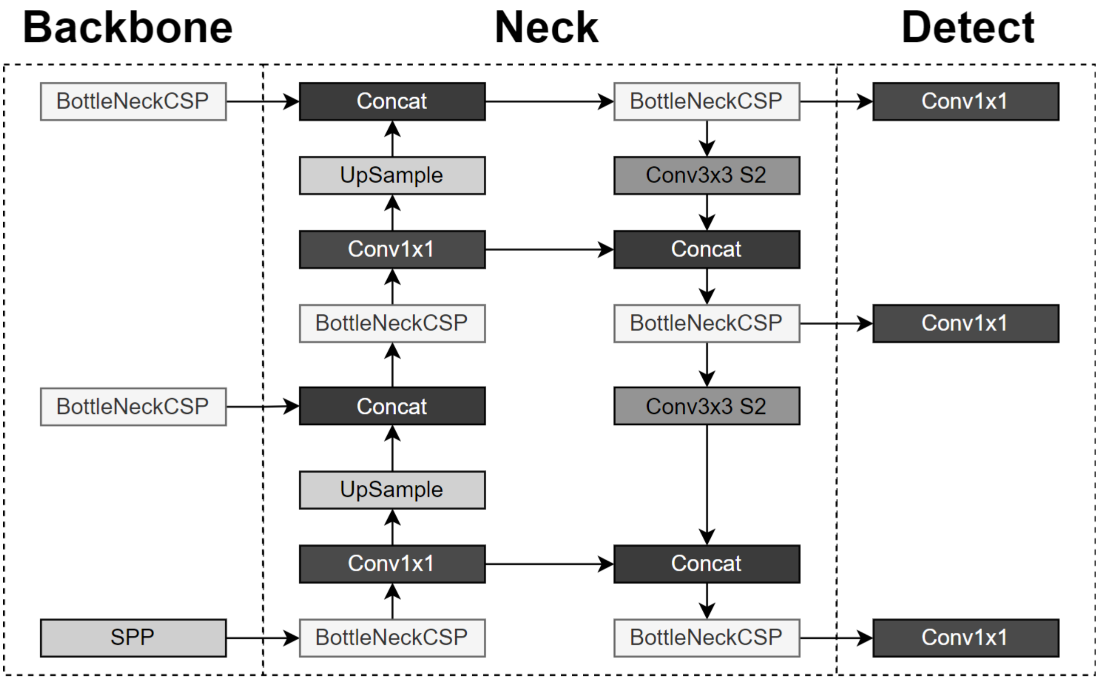


Figure 2.13: YOLO version 5 architecture

The differences between the medium, small and nano architectures of YOLOv5 concern the size of the network according to depth and width multipliers, whose values are shown in Table 2.1.

Table 2.1: Specifications of the different YOLOv5 architectures

YOLOv5 architecture	depth_multiple	width_multiple	ResNet in CSPNet	Convolution kernel
Medium	0.67	0.75	24	768
Small	0.33	0.50	12	512
Nano	0.33	0.25	12	256

The model can be used as-is, without initialized weights, or with weights that have been created through prior training on a given dataset.

2.2.2.C The Segment Anything Model

Until recently, object segmentation was considered a supervised learning problem that required careful data selection and time intensive labeling since unsupervised object segmentation algorithms existed, but yielded lower performance scores compared to their supervised counterparts in most real-world cases. Supervised approaches required large amounts both of training data and manually provided object masks, resulting in being a high-cost solution. Recently Meta introduced the Segment Anything Model, a self-supervised model trained on a massive dataset of 1 billion object masks and 11 million images to preform generalized object segmentation (Kirillov et al., 2023). The model is both able to be applied without prior supervised training or pre-training on data and still perform at a level high enough to compete with domain specific supervised segmentation algorithms.

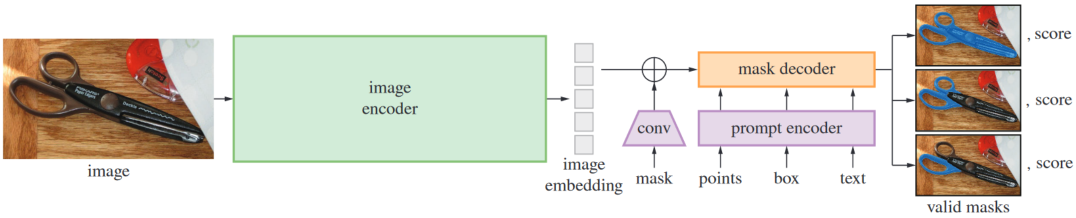


Figure 2.14: Segment Anything Model overview. A heavyweight image encoder outputs an image embedding that can then be queried by input prompts to produce object masks. For ambiguous prompts corresponding to more than one object, SAM can output multiple valid masks and associated confidence scores. (Kirillov et al., 2023)

More specifically, the Segment Anything Model, or SAM for short, is a newly proposed, zero-shot, promptable segmentation model. The model is described as able to perform ”zero-shot” object segmentation. Zero-shot object segmentation is the ability of the model to segment instances of an object without training or fine-tuning on any examples of that class; in simple terms, the model can simply segment any object on command. The command is given in the form of a prompt to the model, with the prompt being either a bounding box, mask, text or a point. Prompts can also be negative, to help refine the selection of the example object that is to be segmented. The result of the model is the segmentation in the form of a mask and a confidence score representing the confidence of the model for the quality of the segmentation (Figure 2.14). This quality of segmentation is the internally calculated expected Intersection over Union. In this case, the IoU is defined as the intersection area of the real bounding mask and the predicted bounding mask divided by the union area of the two masks.

The architecture of the model consists of 3 components: the image encoder, the prompt encoder and the mask decoder. The image encoder is a MAE (He et al., 2022) pre-trained, adapted (Li et al., 2022) Vision Transformer (ViT) (Dosovitskiy et al., 2020). The task of the image encoder is to create the embeddings for the entire image. The prompt encoder follows a different embedding strategy for each type of prompt. Text prompts are encoded through a text encoder from CLIP (Radford et al., 2021), boxes and points are encoded with positional encodings summed with learned embeddings, while mask prompts are considered as "dense" in contrast to the other prompt types and thus are passed to a convolutional neural network whose outputs are summed element-wise with the image embeddings. All methods result in the embeddings of the prompt that, along with the embeddings of the image are passed to the mask decoder. The last component of the network, uses attention mechanisms to update all previously created embeddings and output the mask along with its estimated IoU in regard to the area that the real segmentation would occupy.

2.2.3 Machine Learning in Text Tasks

Textual data are one of the most commonly found formats of unstructured data today. Given the fact that the majority of data is unstructured, this puts the necessity of using texts into perspective. As with all types of unstructured data, analyzing them and making use of the predictive strength that lies inside them is not a task that can be easily completed through classical machine learning approaches.

Machine learning, both classical and deep, relies on the transformation of text into numerical features. Many algorithms have been developed for this purpose, from the simple bag-of-words approach that leads to each document being encoded as a sparse array of counts of occurrences of its words (Qader et al., 2019), or the words' TF-IDF scores (Salton and Buckley, 1988) (see equation 4.1), to the Word2Vec (Goldberg and Levy, 2014) and the Global Vectors for Word Representation (GloVe) models (Pennington et al., 2014) that adopt an embedding approach to the encoding of textual data. In classical machine learning where the numerical features are created manually without embeddings on preprocessed texts (replaced abbreviations, removed stopwords, stemmed and lemmatized) and passed directly into classical machine learning algorithms, the models perform well enough at simple tasks but fail once the goal becomes reliant on the inherent complexity of textual data, such as broader context. For this reason deep learning has dominated the field of NLP, as will be shown later on.

$$similarity = cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| \cdot ||\vec{b}||} \tag{2.20}$$

The tasks that have to do with natural language texts are at least as many as the uses for language itself. For this reason we will be focusing on the tasks that are the most common and concluding on the most influential in the development of NLP AI.

One of the most common tasks and perhaps the first to be attempted is text classification (Kowsari et al., 2019). In text classification, the task is to use the information in the text to predict to which class a part of the document (e.g. sentence, paragraph etc.) or even the entire document belongs best from a pool of available classes. Examples of such problems are many and in many different fields (Jiang et al., 2018b; Kowsari et al., 2017; McCallum et al., 1998; Kowsari et al., 2018). Text classification is done by using a text encoder and applying either deep or classical machine learning algorithms as they would usually be used on the encoded data. Encodings resulting from deep learning algorithms perform better than their classical counterparts. The next big problem is text similarity (Wang and Dong, 2020). Similarity and its downstream tasks such as keyword extraction and topic modeling are tasks that focus on finding ways to measure the level of semantic similarity between documents, words, or documents and words simultaneously. Given the intuitive properties of embeddings (Figure 2.15) text similarity tasks are performed with ease mainly by relying mostly on the method for embedding creation and metrics such as the cosine similarity between vectors (equation 2.20). Lastly, tasks such as Named Entity Recognition (NER) (Marrero et al., 2013; Mohit, 2014) and text normalization (Sproat and Jaitly, 2016), seek to standardize the information of the text by locating areas of interest and annotating them as such.

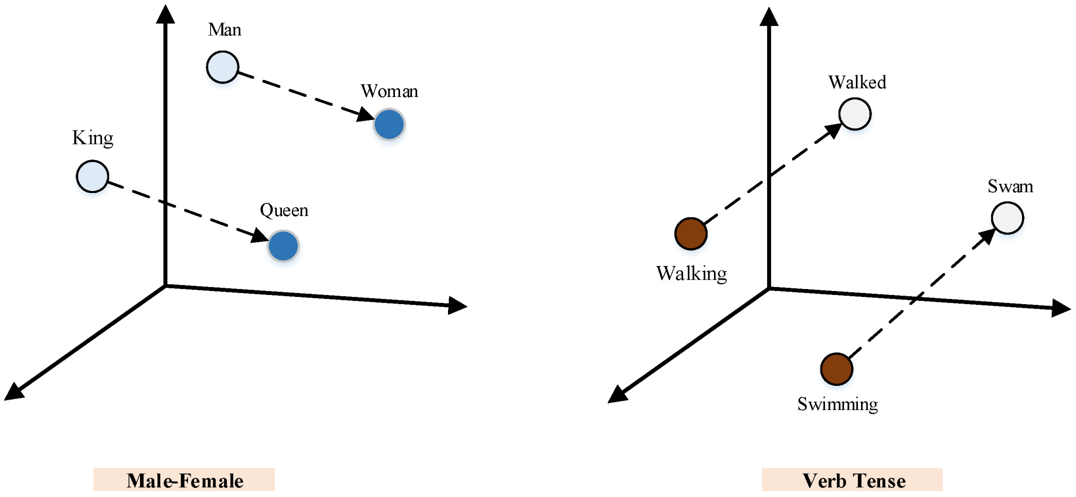
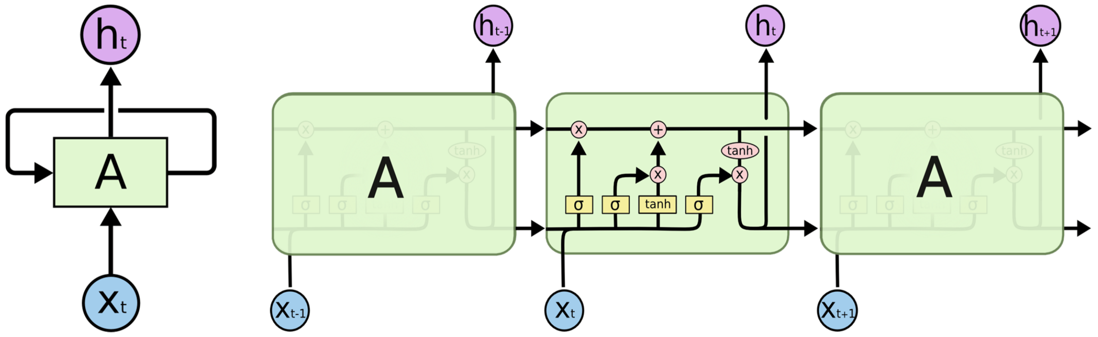


Figure 2.15: Word embedding properties. An intuitive property of word embeddings is that vector translations correspond to semantic concepts. In the left example, translation of the point corresponding to ”man” by a vector $\vec{A}$, results to the point for ”woman”. This means that this translation signifies the notion of change of sex. Hence, when the same vector translates the point for ”king” the resulting point will be very close to ”queen”. (Obiedat et al., 2021)

The tasks mentioned above, while common, could be performed by classical machine learning approaches to a certain degree. However, tasks such as text generation, question answering, text summarization and dialogue understanding needed a way for the text to be represented such that meaning could be completely encoded. The most influential task that required such an approach, was the problem of translation. Translation is defined as a sequence prediction (translated text) task given an original sequence (text to be translated). In translation, especially between languages that follow vastly different syntax rules, it is very common to encounter cases where the text in the first language differs in size compared to the text in the second one. Deep learning came as a solution for this task as seen in the next section and this also paved the way for the solving of many other demanding, related tasks.

2.2.3.A The Transformer

Deep learning has become the norm in complex NLP tasks, with many deep learning algorithms being constantly invented and used. The numerical features created to represent text are not a product of manual work, but of the deep neural networks themselves, allowing for information to be reached in new ways. The first model that was widely adopted for the purpose of translation and overcame the limit reached by classical machine learning algorithms was the Recurrent Neural Network (RNN) and later a special type of RNN, the Long Short Term Memory (LSTM). Recurrent neural networks are used in time series analysis as well as NLP for their ability to capture sequential information. By employing hidden states, as they are called, RNNs can encode sequences by passing information not strictly from the input, but from the previous hidden state to create the output. The hidden states are also calculated by taking into account the input and the previous hidden state. This creates a mechanism analogous to memory, and can result in the network having better understanding of sequential data; something extremely valuable in text related tasks.



(a) (b)

Figure 2.16: Unit of an LSTM cell, folded (left) and unfolded (right) (Ku et al., 2023)

However, the simple RNNs that employed a simple strategy for hidden state calculation resulted in poor performance for tasks where the sequence was too long. This happened due to simple RNNs' problems with exploding and vanishing gradients (Manaswi, 2018). LSTMs came as a solution by proposing an architecture that included "gates". The gates are used to control when information enters memory, resulting in a retrieval of the most relevant information (Figure 2.16).

As a further improvement on sequence specific neural networks, the transformer architecture was proposed by Google (Vaswani et al., 2023) in 2017 (Figure 2.17). The transformer network is a text to text network that includes an encoder-decoder combination that relies on attention mechanisms to preform both of those actions. The network first passes the input text into the encoder to create the embeddings and then into the decoder to give the target output text. Attention, simply put, is a way for the model to map the relevant hidden states in the network's memory to draw the information from, in order to extract it optimally. Attention is calculated by taking into account not only the hidden states but also the tokens' positional encoding used to retrieve the most important hidden states in memory. The result is an output that takes into consideration global semantic information thus overcoming the barriers of different syntax rules and broad context.

2.2.3.B The Bidirectional Encoder Representations from Transformers and the Robustly Optimized BERT Approach

The Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018b) came as an extension to transformer's encoding block. The purpose of BERT is to be a strong encoder whose produced embeddings can be used for a variety of tasks different that text to text. Essentially, BERT's goal is to offer generalized word embeddings. The architecture of BERT is multiple chained transformer's encoding blocks, the number of which ranges depending on what version of BERT we are referring to. Two sizes of BERT were originally created; one was $BERT_{BASE}$ while the other was $BERT_{LARGE}$, the details of whom are presented in table 2.2.

Table 2.2: Architectures of BERT models

Model size	Number of transformer block layers	Hidden size	Number of self-attention heads	Total Parameters
$BERT_{BASE}$	12	768	12	110M
$BERT_{LARGE}$	24	1024	16	340M

BERT was trained initially with unlabeled data and then fine-tuned on labeled data for several downstream tasks. The unlabeled initial training was done through two

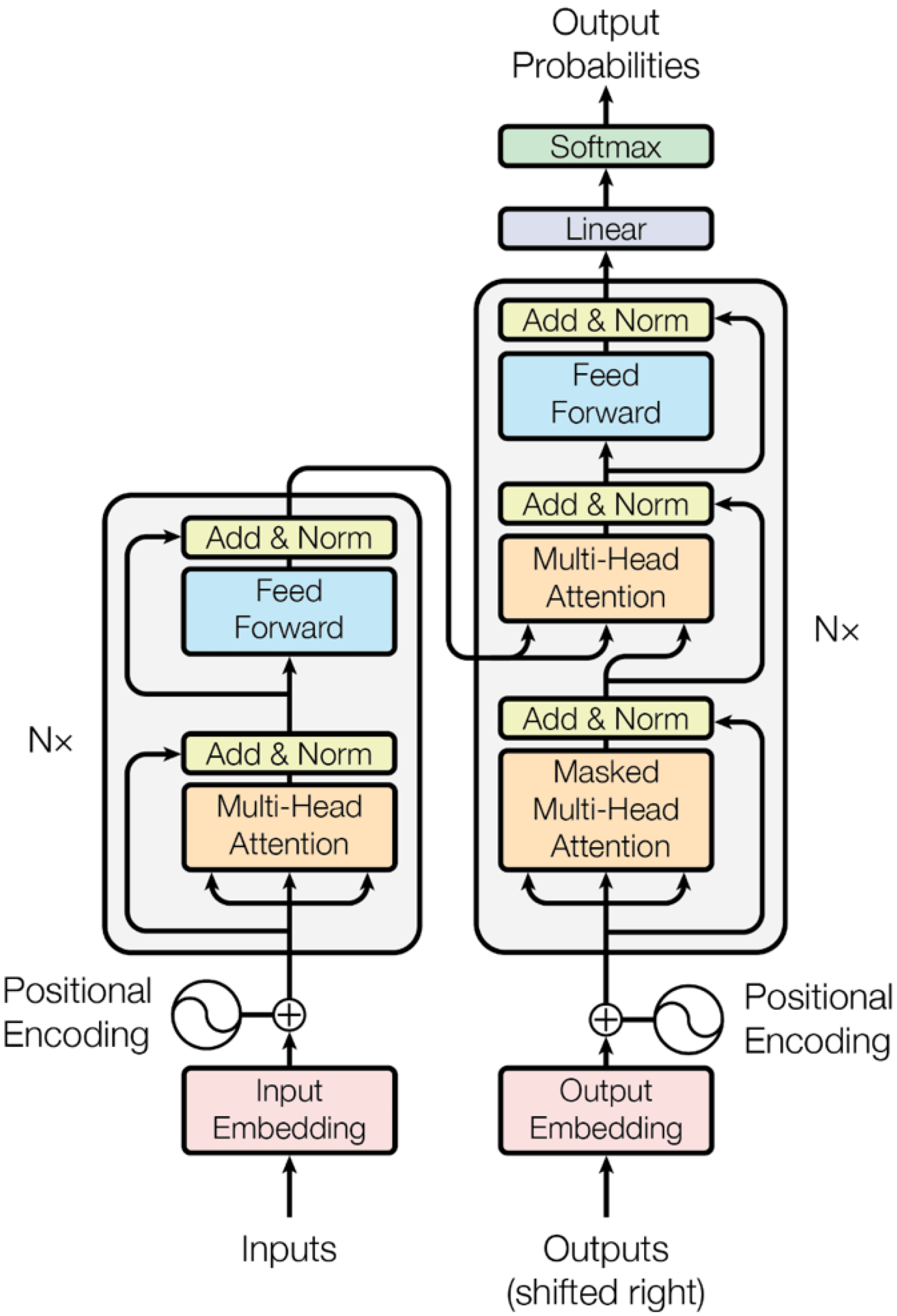


Figure 2.17: Model architecture of Transformer (Vaswani et al., 2023)

tasks. Firstly, it was trained by using Masked Language Modeling (MLM), where a percentage of words in a sentence are masked randomly and the model is called to predict the masked values, and then using Next Sentence Prediction (NSP). In NSP, BERT received two sentences and was asked to predict whether those two were consecutive or not in the original text. This was possible since BERT is capable of receiving two sentences simultaneously as a single token sequence, allowing for tasks such as question answering, through the use of special separation tokens. The pre-training allowed BERT to learn a good representation of the textual data, and build on top of the initially learned text representations through fine-tuning on downstream tasks. Fine-tuning was performed on the tasks of classification, question answering and named entity recognition.

In 2019, Meta Research published their findings regarding BERT; more specifically, they stated that the original BERT was under trained, and with proper training could outperform State of the Art (SOTA) models that came afterwards. In their paper named ”RoBERTa: A Robustly Optimized BERT pre-training Approach” (Liu et al., 2019) they proposed a novel way of pre-training the BERT model. RoBERTa included an increase of training data, training the data for longer with bigger batches, removing the next sentence prediction objective, training on longer sequences and finally dynamically changing the masking pattern applied to the training data (Liu et al., 2019). After the above changes in the initial training of BERT, it managed to once again reach SOTA results in multiple benchmarks (GLUE, RACE, SQuAD), thus returning to the forefront of research.

Chapter 3

Case: Deep Learning for Detecting Verticillium Fungus in Olive Trees: Using YOLO in UAV Imagery

The Verticillium fungus(Ruggieri, 1946) is one of the largest and most widespread causes of destruction in olive trees around the world. The fungus survives in soil and can transmit through water (Pérez-Rodríguez et al., 2016; López-Escudero and Mercado-Blanco, 2010). Controlling it is extremely challenging, since it has a wide selection of alternative hosts and is able to initially manifest asymptomatically (Alstrom, 2001). In order to protect the crops, it is of the utmost importance to accurately detect and assess tree health at scale.

Recently image data collection has been facilitated thanks to advances in technology, such as the improvement of cameras' resolutions and Unmanned Aerial Vehicle (UAV) technologies that enable automatic collection of data. Machine learning and especially deep learning has made it possible to analyse and classify such data accurately (Fichtel et al., 2021). YOLO is one of the most used algorithms for such tasks. It has been used effectively in tasks of tree damage detection from UAV imagery both for parasite infestations(Safonova et al., 2022), and for environmental damage detection (Puliti and Astrup, 2022).

The timely detection of Verticillium infections is a complex and time intensive task, due to the nature of the disease, that as previously stated, initially manifests asymptomatically. Until recently, skilled labor was needed to determine whether infection has occurred. Still, in cases where the wilt had manifested in areas not visible to the examiner, like the top of the tree, infection could go undetected and thus lead to disease spreading. In this context the objective of this work is to explore the potential of the application of the YOLO algorithm and the SAM algorithm paired with convolutional neural networks to detect Verticillium infections in olive trees. We have used Red, Green, Blue (RGB) images captured with UAVs from three fields in northern Greece during October and November, when the symptoms of the disease are more pronounced. The images were used to train three different architectures of the YOLO version 5 algorithm with promising results and one combination of the SAM-CNN algorithm. By employing UAVs, we attempted to eliminate cases where infections were not visible to an examiner, and by using the machine learning methods we introduced an objective way of determining tree infection with high accuracy and without the need for time intensive human labor.

The structure of this effort is organized as follows. In the beginning background information on the Verticillium fungus and the Unmanned Aerial Vehicles is presented. Next, we list the work done by other researchers on similar tasks, focusing on the application of YOLO on UAV tree imagery data. Then, we describe the materials and methods used to conduct the experiment as well as information on the evaluation processes that took place. The results of this research effort are presented afterwards, both regarding the produced dataset as well as the machine learning models' reached performances. Lastly, discussion of the results and future work that is to be done as an extension of this present work ensues.

3.1 Background

3.1.1 The Verticillium Wilt

Verticillium wilt of olive tree is caused by the soil-borne fungus Verticillium dahliae Kleb. It is currently considered the main soilborne disease threatening olive production worldwide. This disease was first described in Italy in 1946 (Ruggieri, 1946), followed by California (Snyder et al., 1950) and Greece (Zachos, 1963). Descriptions of disease occurrence have been reported from 1970 and onwards in Turkey, France, Spain, Syria, Morocco, Jordan, Algeria, Israel, Iran, Malta, and Australia (Geiger et al., 2000; Jiménez-Díaz et al., 1998; Levin et al., 2003; Naser et al., 1998; Porta-Puglia et al., 2005; Sanei et al., 2004; Saydam and Copcu, 1972; Sergeeva et al., 2009) practically covering all olive production zones. This disease is one of the most significant diseases of olives causing every year big economic damages not only in terms of yields but also in terms of trees that die decreasing thus permanently production potential. (López-Escudero and Mercado-Blanco, 2010). One more factor affecting financial sustainability is the fact that fruits of V. dahliae-infected trees have poor organoleptic properties (Báidez et al., 2007). The fungus survives in soil by means of microsclerotia which serve as primary inoculation means. Hyphae generated by microsclerotia germination penetrate the roots and grow toward the xylem vessels, producing mycelium and conidia (Pegg and Brady, 2002).

The typical symptoms of infestation includes early drop of asymptomatic, green leaves from individual twigs and branches that eventually end to complete defoliation However there are cases that apoplexy is rapidly developing (acute form of the disease) (Blanco-López et al., 1984; Thanassoulopoulos CC, 1979; Zachos, 1963). The symptoms are more evident from late fall to late spring. The blocking of xylem by fungus mycelia reduces the water flow and leads to water stress (Ayres, 1978; Trapero et al., 2018) affecting amongst others, plant transpiration rates. Cultivation techniques have contributed to fast dispersal of the disease worldwide. Infested plants that are transported to new areas the means of new infestations, increased water levels in soil help microsclerotia migrate to new areas infecting new olive trees (Pérez-Rodríguez et al., 2016; López-Escudero and Mercado-Blanco, 2010). V. dahlia biology includes specific traits that make control very difficult. The most important are the numerous alternative hosts and the asymptomatic appearance of infested olive trees at initiative stages of infection (Alstrom, 2001).

3.1.2 Unmanned Aerial Vehicles

Until recently, certain procedures required the presence of humans, either for process monitoring purposes, or simply for carrying out tasks that needed human decision making or dexterity to be involved. Due to the above, automation of such tasks was considered practically impossible. However, the technological advancements made in recent years have changed the landscape of applied technology and process automation along with it. Amongst those new technologies, one stands out for its flexibility and performance in a wide enough array of tasks for it to be considered revolutionary: unmanned vehicles.

Unmanned vehicles have made it possible for tasks to be carried out without human involvement. They are separated into two main classes: ground, and aerial. Both have their unique uses, and while there is a considerable overlap in some cases, in others, they can be employed symbiotically (Lazna et al., 2018; Zoto et al., 2020; Vasudevan et al., 2016). Cooperative applications of the two types of vehicles have resulted in the automation of elaborate tasks, and have shown that these technologies can reduce if not completely eliminate the need for human labor.

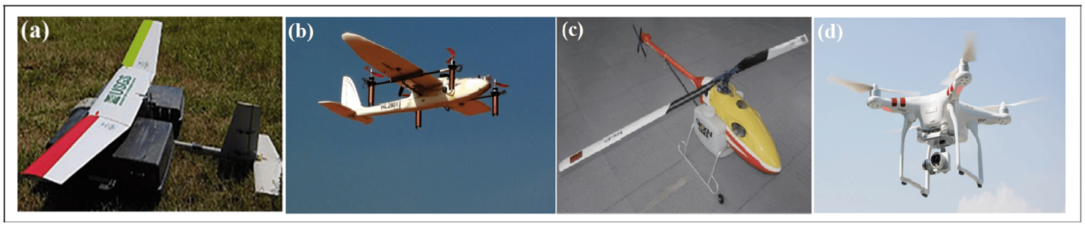


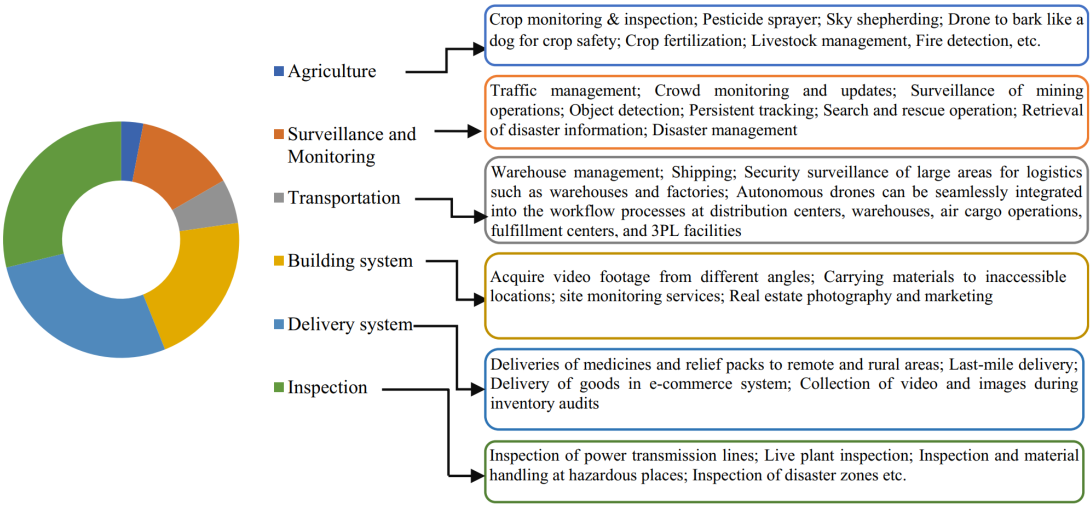
Figure 3.1: Categories of Unmanned Aerial Vehicles. (a) fixed-wing, (b) fixed-wing hybrid, (c) single rotor, and (d) multirotor UAV. (Mohsan et al., 2022)

Unmanned aerial vehicles, also known as drones or UAV for short and henceforth, are flying vehicles that do not require a human pilot to be present onboard for them to be operated. However, in the majority of UAV operation cases, a pilot is still required to drive the vehicle, remotely, from a Ground Control Station (GCS). In other cases, drones can have their operation procedure completely automated through the use of smart systems that utilize artificial intelligence technologies to handle tasks such as path planning and landing (Politi et al., 2022). Their size can range from a few centimetres to a few meters. There are different categories depending on the usage of the drone, with the two largest distinctions being the existence of wings and the number of rotors (Figure 3.1). Unmanned aerial vehicles can have only one rotor or as many as eight. Usually, multi-rotor vehicles are preferred for precision tasks, as they have the ability to hover and stabilize themselves better, compared to single or double rotor vehicles (Mohsan et al., 2022). UAVs can both collect data and analyze them in real-time through equipment embedded in the drone, or simply collect the data and send it to the ground control station for analysis; the first option requires on-board processors with high processing power, while the second option requires a reliable connection and a large enough bandwidth for the data to be transferred timely and accurately to the station responsible for their analysis.

UAVs have seen a rise in popularity in the last few years due to their ability to perform various tasks in a relatively short amount of time compared to conventional means used for the same tasks in the past. More specifically, UAVs are known to offer both an inexpensive (Gaffey and Bhardwaj, 2020; Pérez et al., 2013) and also a highly flexible medium for real time monitoring tasks (Lee et al., 2017), making them a staple in research efforts focused in showing results in a low budget setting. Their ability to carry lightweight sensor equipment such as cameras, microphones, temperature, proximity and acceleration sensors have widened the field of their potential uses (Pajares, 2015). Another factor that has contributed positively towards their usability is the advancement of the networking technologies responsible for the transfer of data between drone and the ground control station.

For this reason unmanned aerial vehicles have dominated the field of research and have been responsible for many recent breakthroughs in multiple fields of science. In 2022, researchers used UAVs equipped with low-cost sensors to monitor air quality (Arroyo et al., 2022). Facilitated by the aforementioned flexibility and inexpensiveness of the drone technology, the researchers managed to achieve their goal without employing costly specialized equipment, that was used until recently to perform this task. In the same year, another team of researchers employed drones, this time with camera equipment, to develop a car-free street mapping model (Lee et al., 2022). They showed that it is possible to map a road network easily without cars and the subsequent expenses.

The applications of UAVs are too many to list, but efforts have been made to map the fields that are to become the main beneficiaries of this technology (Ahmed et al., 2022). Namely, and as shown in Figure 3.2, those are agriculture, surveillance and monitoring, transportation, building, delivery and inspection.



Crop monitoring & inspection; Pesticide sprayer; Sky shepherding; Drone to bark like a dog for crop safety; Crop fertilization; Livestock management, Fire detection, etc.

Traffic management; Crowd monitoring and updates; Surveillance of mining operations; Object detection; Persistent tracking; Search and rescue operation; Retrieval of disaster information; Disaster management

Warehouse management; Shipping; Security surveillance of large areas for logistics such as warehouses and factories; Autonomous drones can be seamlessly integrated into the workflow processes at distribution centers, warehouses, air cargo operations, fulfillment centers, and 3PL facilities

Acquire video footage from different angles; Carrying materials to inaccessible locations; site monitoring services; Real estate photography and marketing

Deliveries of medicines and relief packs to remote and rural areas; Last-mile delivery; Delivery of goods in e-commerce system; Collection of video and images during inventory audits

Inspection of power transmission lines; Live plant inspection; Inspection and material handling at hazardous places; Inspection of disaster zones etc.

Figure 3.2: Expected reach of UAVs in various applications (Ahmed et al., 2022)

3.2 Related Works

Unmanned aerial vehicles and machine learning have been a game changer in tree image data collection and analysis, that used to rely solely on satellites until recently. More specifically, in the case of tree image collection and analysis, it is shown (Fichtel et al., 2021) that the two technologies when paired together, could provide a means to analyze data with significantly lowered costs and at a much greater speed thus enabling the accurate and timely detection of damages, both caused by weather conditions or pests and disease.

The YOLO algorithm is ideal for tree image data detection and classification tasks since it provides a unified solution for those two most common agricultural problems at scale, with the added advantage of fast inference time; something that makes it suitable for real time applications.

The combination of YOLO on UAV image data has been tested on multiple scenarios, with one of the most general of them being the simple detection of fruit trees regardless of species in an orchard. In that research effort, the researchers deployed an improved version of the YOLOv4 algorithm, and managed to successfully build a fast and accurate model with 1380 RGB UAV images of varying resolutions that were augmented through standard augmentation methods (change in orientation, brightness and by adding noise) to a total of 3000 images. Their model achieved 98.21% Mean Average Precision (mAP), and 0.936 f1-score for canopy detection, showing that a model like that can be an effective way to address tree detection tasks (Zhu et al., 2022b).

Moving to more specific situations, the implementation of YOLO on UAV tree image data has been used to detect only trees of a specific genus (Tian et al., 2022) or species (Özer et al., 2022; Jintasuttisak et al., 2022; Aburasain et al., 2021; Chowdhury et al., 2022; Wibowo et al., 2022). In all of those cases, the detection was not anymore a matter of just detecting trees but the task had to be narrowed down to the correct type of tree; something that was done with success, indicated by high classification scores with multiple metrics (precision, recall, f1-score) in the above cases. The parameters of the experiments is shown below, on Table 3.1.

Table 3.1: Research parameters of detecting specific trees with YOLO on UAV images

Reference	YOLO Version Tested	Number of Images	UAV Flight Altitude
(Tian et al., 2022)	YOLOv5	-	50m
(Özer et al., 2022)	YOLOv5 (s, m, x)	889	-
(Jintasuttisak et al., 2022)	YOLOv5	125	122m
(Aburasain et al., 2021)	YOLOv3	221	-
(Chowdhury et al., 2022)	Improved version of YOLOv5	1,558	-
(Wibowo et al., 2022)	YOLOv3,v4,v5m	17,343	200m

As seen from the table, the research efforts had a high level of experimental variance, with the number of images used ranging from 125(Jintasuttisak et al., 2022) to 17,343(Wibowo et al., 2022), the UAV flight altitude ranging from 50 meters(Tian et al., 2022) to 200(Wibowo et al., 2022), while the versions of the YOLO algorithm used were v3(Aburasain et al., 2021; Wibowo et al., 2022), v4(Wibowo et al., 2022), and v5 (Tian et al., 2022; Özer et al., 2022; Jintasuttisak et al., 2022; Chowdhury et al., 2022; Wibowo et al., 2022), with the last version having many different architectural variants. Thus, it can be concluded that YOLO is more than capable to tackle problems of detecting very specific objects from UAV images, thus allowing for finer detection tasks to be undertaken.

Still, the two applications of the YOLO-UAV combination described above only show the algorithm's capacity to detect one class, whether it is trees in general or a specific species. Research has been conducted to highlight the ability of YOLO to perform simultaneous detection of multiple classes. In (Safonova et al., 2022), researchers deployed the YOLO algorithm on UAV images to detect predominantly spruce trees damaged by the bark beetle. In that case, the classes that the algorithm was called to classify the trees into were 4: green attack, yellow attack, red attack and grey attack; essentially different levels of tree damage from the bark beetle. YOLO versions 2,3 and 4 were tested on 400 images taken 120 meters from ground level. There was significant class imbalance on the training set (green: 312, yellow: 622, red: 76, grey:188) and the validation set (green: 202, yellow: 400, red: 20, grey:61) (Safonova et al., 2022), with the two sets having different class imbalances. Despite that, YOLOv4 along with the author's proposed method of image preprocessing, achieved impressive results with

0.95 precision, 0.76 recall and a mAP of 0.94.

Another case study was that of (Puliti and Astrup, 2022), where the researchers applied the YOLOv5 algorithm on a vast dataset of tree UAV image data, derived from 40,697 individual trees photographed during different times of the day, month and year to detect snow damage on trees. The classes were healthy, damaged and dead, while there was a great class imbalance; only 16% of the instances were damaged or dead. Nevertheless, the classification evaluation metrics proved to be high, with the small caveat that the different classes had significantly different results, with precision ranging from 0.759 to 0.546 and recall from 0.78 to 0.40 (Puliti and Astrup, 2022) - almost double.

Finally, in (Sun et al., 2022), research was conducted to spot trees affected by pine wilt nematode, a fast spreading disease affecting forest areas. The disease starts from the top of the affected tree and spreads to the bottom, making the use of UAVs ideal since they provide a top view, allowing for an easy and early detection. In this research effort, 116,012 images were taken at different heights ranging from 50m to 300m, and the YOLOv4 algorithm was applied again with success (precision: 1, recall: 0.8969).

Based on the results of the research efforts mentioned above we can determine that the YOLO algorithm can be used effectively in damage or disease detection on trees that belong to a particular species. It was also shown that the algorithm was capable of handling class imbalance but at the same time it would be possible for one class to be more easily detectable than another; sometimes with large differences.

3.3 Methodology

Our areas of study were three olive fields, Field A, E and K, located in Northern Greece (Figure 3.3), photographed at midday between the 5th of October and the 4th of November of 2022. The UAV used to gather the images was an Air Surveyor 4 equipped with a SONY ILCE-6000 camera. The camera was running the 3.21 version of its software, and captured images at 8×10^{-4} seconds exposure time, with the use of a SAMYANG AF 24mm F2.8 lens. The produced RGB images where 6000x4000 pixels.

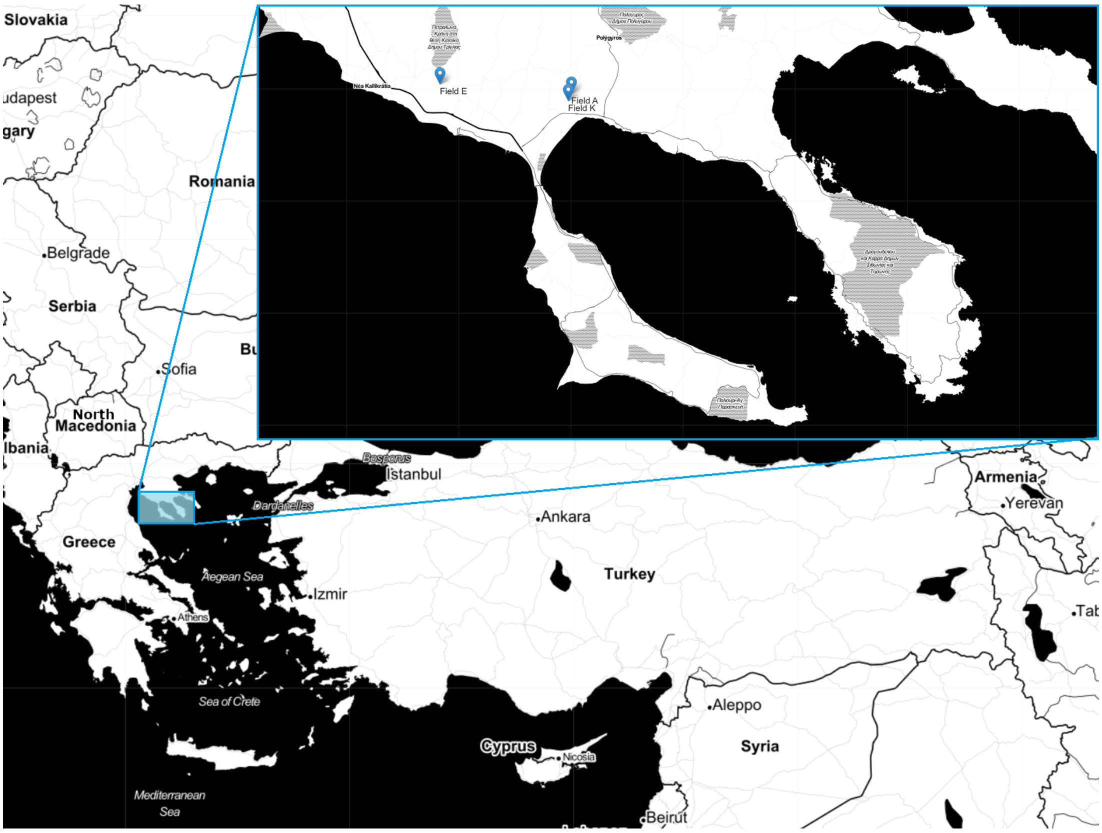


Figure 3.3: Locations of the olive fields

After the images were collected by the UAV, they were separated depending on what field they originated from and images from the same field were stitched together to create an orthomosaic. Every created field orthomosaic was then cut in tiles of size 3000x2000 (pixels) and was annotated with the use of the Labelimg package (Figure 3.4) to create the data that was provided to the YOLO algorithm.

The annotations were produced in the YOLO format, where for each image a text file with the same name storing the results of the annotation was created, and in every text file, each instance (tree) was represented as a line of text. The annotations had the form of bounding rectangles, and were represented in text as follows: the first number in the line conveyed the class to which the instance belonged, encoded as an integer, while the next four numbers represented the coordinates of the bounding boxes, with the first two being the centers of the bounding box in the X and Y axis divided by the height and width of the image respectively, and the last two being the width and height of the bounding box divided by the height and width of the image respectively.

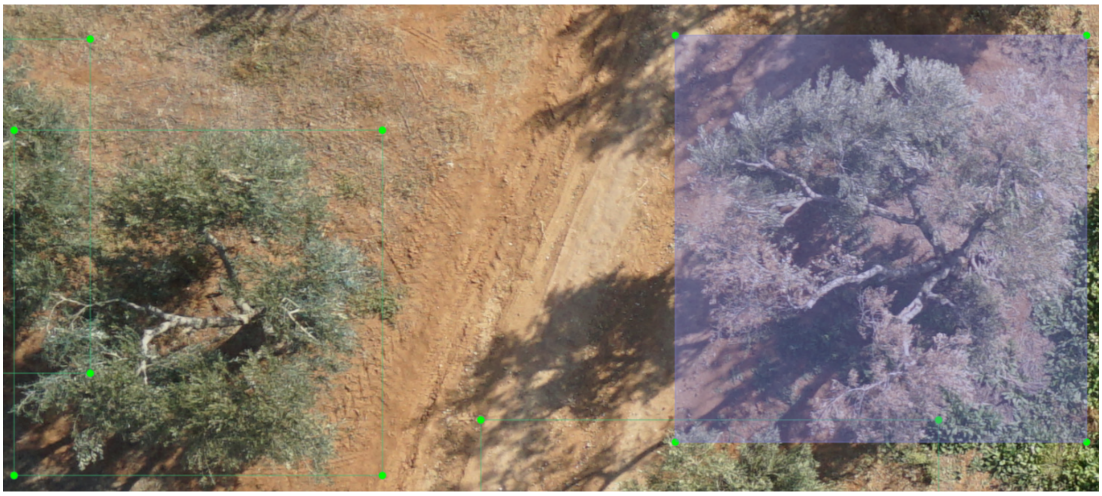


Figure 3.4: Annotating trees with bounding boxes through the Labelimg graphical interface

The classes that every tree was classified into were two: either damaged or healthy. The annotation process had two parts: initially, the bounding boxes were created by visually determining the borders of each tree, while the second part - the class annotation - was performed by experts based on what constitutes as damage caused by the verticillium wilt - the damaged trees had visibly altered color, a result of the withering effect that the wilt has on olive tree leaves (Figure 3.5).

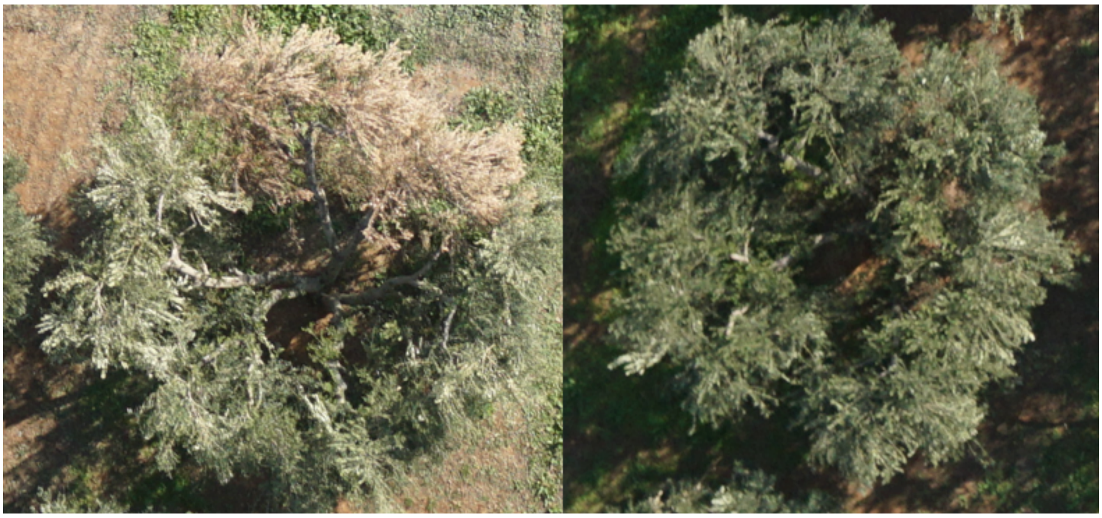


Figure 3.5: Comparison of a damaged tree (left) and a healthy one (right).

The annotated images were randomly split in train, validation and test datasets. The split happened so that the training dataset contained 60% of the data while validation and testing contained 20% each. To avoid providing the algorithm with datasets of different percentages of damaged trees, and thus, to ensure that the three fields were equally represented in the splits, the splitting of the data was stratified to account for the field from which the tiles originated.

The YOLOv5n, YOLOv5s and YOLOv5m models were applied on the dataset at two different image scales, utilizing network weights created from pre-training on the COCO128 dataset. The models were trained on a desktop computer equipped with an Intel(R) Core(TM) i7-10700 Central Processing Unit (CPU) running at 2.90Gigahertz (GHz) and 16 Gigabyte (GB) of Random-Access Memory (RAM). The results were evaluated with the metrics commonly used in evaluating YOLO model performance, that are described in detail below:

- Precision:
$$\frac{TP}{TP+FP} \tag{3.1}$$
- Recall:
$$\frac{TP}{TP+FN} \tag{3.2}$$
where TP = true positives, FP = false positives, FN = false negatives.
- Mean Average Precision or mAP, which is the mean of Average Precision (AP) values calculated for a certain threshold (e.g. mAP[0.5]: mAP for threshold value of 0.5) or range of thresholds (e.g. mAP[0.5:0.95]: mAP for threshold values of 0.5 up to 0.95 with a step of 0.05) for all classes:
$$mAP = \frac{1}{n}\sum_{n} AP_n \tag{3.3}$$
with
$$AP_n = \frac{1}{101}\sum_{r\in\{0.0,\ldots,1.0\}} \max_{\widetilde{r}\geq r} PRC(\widetilde{r}) \tag{3.4}$$
, where PRC is the precision-recall curve, $\widetilde{r}$ is the 101-point interpolated recall value and n is the class.

The thresholds above are referring to the minimum value of Intersection over Union (IoU) over which a classification is considered correct. The IoU is defined as the intersection area of the real bounding box and the predicted bounding box divided by the union area of the two boxes.

As a further step towards the direction of creation of high performant image classifiers, we compared the results of the best YOLOv5 model with a convolution neural network classifier trained with resized images that were the result of segmentation. More specifically, the images that corresponded to each of the three datasets (training, validation, testing) were passed to the Segment Anything Model prompted by the same bounding boxes that the YOLO algorithm was trained on, to create high precision object masks and consequently segmentations of individual olive trees.

3.4 Results

3.4.1 Data Collection and Dataset Processing

The three fields that were the subject of our research were photographed at different times between the 5th of October and the 4th of November of 2022, shown in detail below, in Figure 3.6.

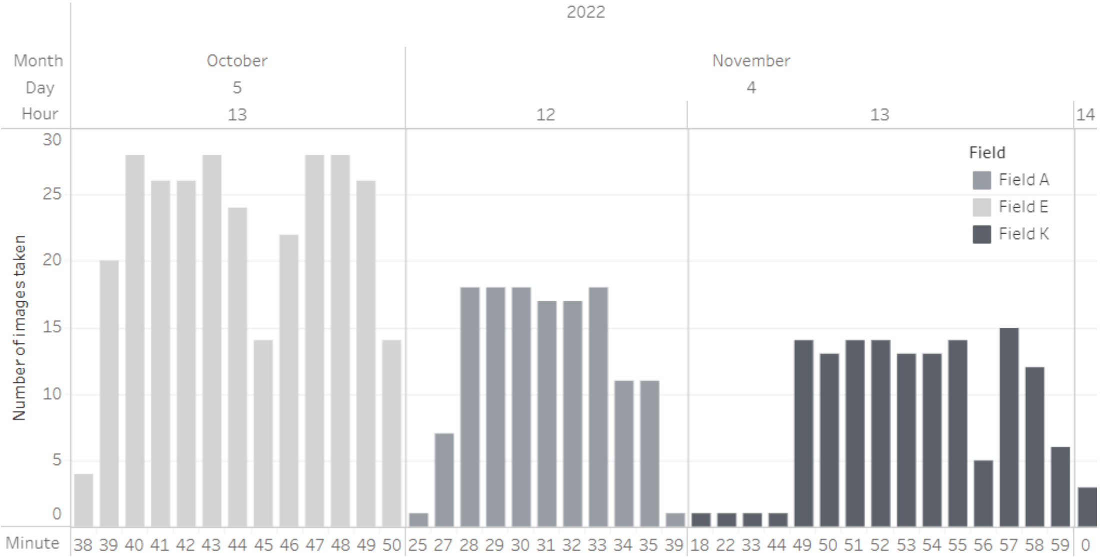


Figure 3.6: Date and time when images where captured

In total, 429 images were captured out of which 137 were in Field A, 152 were in Field E, and 140 where in Field K. The images had a high level of overlap and were captured in sequence. These images created three orthomosaics, one for each field, and then cut into tiles. The tiles created were 160 with almost every field having the same number of them (Field A: 50, Field E: 55, Field K: 55).

The total number of annotation instances (trees or parts of trees) that were created originally was 3038. However, the three fields varied greatly in terms of percentage of damage done by the verticillium wilt (Table 3.2).

Table 3.2: Number and percentage of damaged trees for every field

	Field A	**Field E**	**Field K**
Number and percentage of healthy instances	804 (92.94%)	927 (98.82%)	1102 (89.23%)
Number and percentage of damaged instances	61 (7.05%)	11 (1.17%)	133 (10.76%)

To tackle this problem, stratified shuffling, available through the Scikit-Learn Python library (Pedregosa et al., 2011b), was employed during the train-test-validation split and its use resulted in datasets of almost equal percentage of damaged trees (Table 3.3). The stratified shuffling method creates datasets by sampling without replacement from the original dataset, while also taking into account the class of the samples, so that every resulting dataset contains the same distribution of classes. The created train, validation and test datasets contained 60%, 20% and 20% of the total images (Train: 96 images, Validation: 32 images, Test: 32 images), respectively.

Table 3.3: Number and percentage of damaged trees for every dataset after stratified splitting of data

	Train	**Validation**	**Test**
Number and percentage of healthy instances	1610 (92.36%)	591 (93.95%)	631 (94.74%)
Number and percentage of damaged instances	133 (7.63%)	38 (6.04%)	35 (5.25%)

3.4.2 Application of the YOLOv5 Algorithm

The training, validation and testing datasets whose creation was described above, were provided to three architectures of the YOLOv5 algorithm to train and make predictions on. The architectures were YOLOv5n, YOLOv5s and YOLOv5m with the last letter of the model name referring to the size of the model architecture (nano, small, medium). Given that the dataset had high class imbalance (see Table 3.3), in order to ensure that the frequency of occurrence of the dominant class does not cause the model to optimize only for that class, we weight the penalty for false predictions by multiplying the loss of each class by the inverse frequency of that class. This technique is often adopted in classification tasks with high class imbalance to ensure equal performance metrics between classes(Huang et al., 2016; Mahajan et al., 2018; Mikolov et al., 2013; Wang et al., 2017). Additionally, the build-in method of `--image-weights` was employed, that sampled images by taking into account the proportion of each class's instances present in each image, thus, downsampling images with high proportional content of the dominant class and vice versa.

The models were trained for 300 epochs with early stopping enabled with a patience of 100 epochs. The patience mechanism would restore the best weights of the models if for 100 epochs no advancement was made in the model fitness metric. The model fitness metric in this case was calculated as the weighted sum of the mAP[0.5] and mAP[0.5:0.95] metrics, a combination widely used in the literature (Chen et al., 2023; Bjerge et al., 2023; Kubera et al., 2022), as shown below:

$$model fitness = \begin{bmatrix} 0.0 & 0.0 & 0.1 & 0.9 \end{bmatrix} \begin{bmatrix} Precision \\ Recall \\ mAP@0.5 \\ mAP@0.5:0.95 \end{bmatrix} \tag{3.5}$$

The model batch size used was 16, and the three models were trained with two different model input sizes each, with one size being 1216x1216 and the other 640x640. The model input translates to size of input image in pixels.

Table 3.4: Model training fitness statistics

Architecture	Model input size	Max fitness epoch	Max model fitness reached
Nano	640x640	282	0.577714
Nano	1216x1216	300	0.569750
Small	640x640	208	0.587946
Small	1216x1216	269	0.616960
Medium	640x640	263	0.640254
Medium	1216x1216	262	0.652587

The evaluation of the models was done on the testing set that was held aside for that purpose. The models' performances are shown in Figure 3.7, where model YOLOv5m with model input of size 640x640 managed to outperform all other models in every metric and for every class.

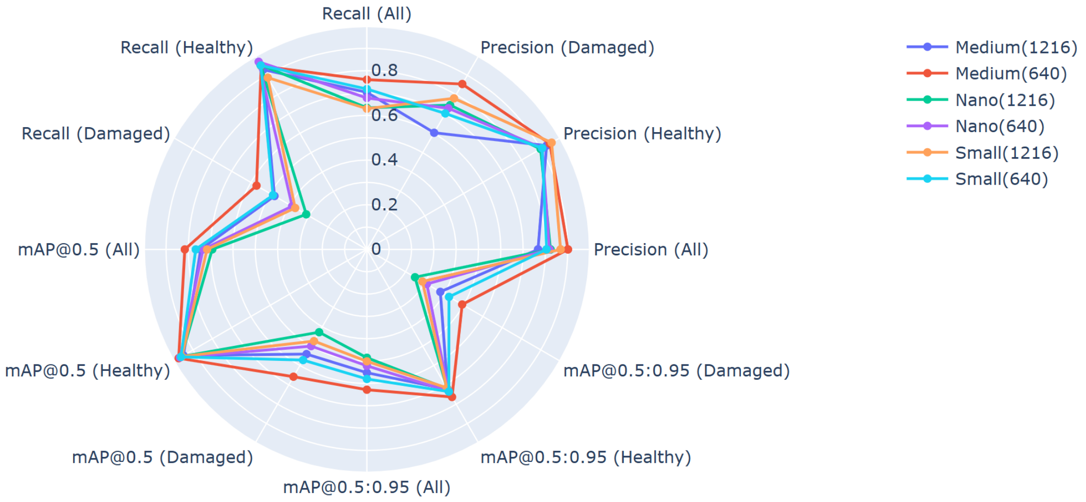


Figure 3.7: Model performances on testing data

The speed statistics of the model application pipeline are shown below, in table 3.5. The three columns displayed are the preprocessing speed, that refers to the time it took for the model to transform the input image into a pytorch array and rescale its values, the inference speed, that is the time the model spent detecting bounding boxes within the image and predicting the corresponding class, and lastly the Non-Maximum Suppression (NMS) speed, showing the time Non Max Suppression needed to be performed on the predicted bounding boxes of the previous step to exclude overlapping boxes.

Table 3.5: Model application on the testing set speed statistics (in milliseconds, per image)

Architecture	Model input size	Preprocessing speed	Inference speed	NMS speed
Nano	640x640	1.0	94.7	5.9
Nano	1216x1216	1.0	77.5	2.5
Small	640x640	1.0	178.9	2.4
Small	1216x1216	1.0	168.0	1.5
Medium	640x640	1.3	328.8	1.5
Medium	1216x1216	1.0	318.0	2.0

Since depending on the hardware these speeds are bound to change, we should focus on the relative difference between each model's speed. The table showcases an interesting detail: models using the reduced input size of 640x640 required longer to infer the contents of the images. Also we can see that as the model architecture increases in size, inference time also increases while NMS time drops. This is an expected outcome as with larger models more calculations must be performed but since these calculations tend to lead to better results, NMS time is decreased as it is not needed as much to sort out the erroneous predicted bounding boxes.

3.4.3 Application of the Segment Anything Model paired with Convolutional Neural Networks

The Segment Anything Model was applied to each image of every dataset. The datasets used were those created after the stratification procedure (table 3.3) as a way to standardize the experiment parameters between applications of the different algorithms. Each of the images in the datasets was passed to SAM to create the object segmentation masks. SAM was prompted with the bounding boxes produced during the labeling procedure that took place in the original data annotation phase. This resulted in 3047 individual tree image masks. The resulting segmentation masks were used to create images where only the tree was visible with the rest of the image being black (Figure 3.8).

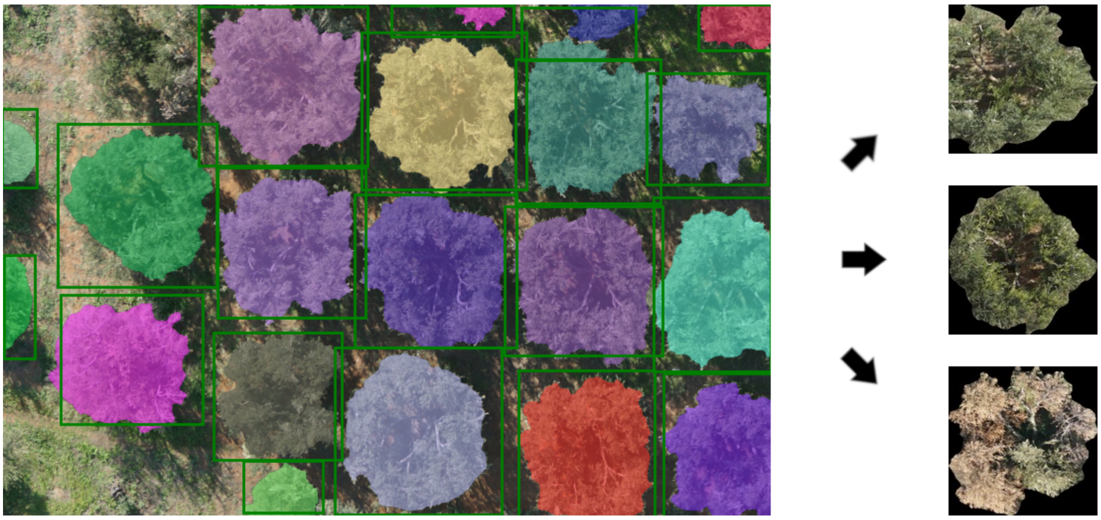


Figure 3.8: Application of the Segment Anything Model on the aerial images. The green box outlines are the manually created tree annotations used as prompts for the SAM model.

The images were resized to the size of 150x150 pixels and passed to a convolutional neural network whose architecture consisted of three consecutive pairs of convolutional and max pooling layers, followed by a dense layer of 1024 nodes that finally led to a single node responsible for the classification. The activation functions of the network were ReLu for all layers with the exception of the last, single node layer, that used a sigmoid activation function.

The neural network was set to train for 50 epochs but with early stopping enabled and a patience of 5 epochs, trained for only 8. This was not surprising considering the simple nature of the task, and the fact that the data were now clear of noise. Still, the results reached were phenomenal. More specifically, the combination of SAM-CNN managed to reach a precision of 94.66% a recall of 100% and an f1 score of 97.25%, thus surpassing even the best YOLOv5 model regarding the task of classification.

3.4.4 Result Evaluation and Discussion

All YOLO models tested were close in terms of performance, with model m640 outperforming them all by a small margin in most cases. A possible explanation as to why model m640 managed to do that, could be due to a pattern shown in the results. We can generally see that as input size decreases and model capacity increases, the performance increases. This pattern indicates that for the task at hand, models with high capacity are needed to capture the necessary information for a successful classification. This also works in reverse: input of smaller size reduces the needed model capacity. Few things should be noted relative to the models' performances: The first general pattern observed is that every YOLO model consistently achieved lesser scores when it came to predicting the damaged class. That could be something that can be attributed to the relatively few instances of that class present in the training set. This is a known problem, that affected previous research efforts; even those that have had datasets of considerable size. In (Puliti and Astrup, 2022), even though a total of 40,697 trees were used to train the model, the class that was the most underrepresented, consistently achieved the lowest scores. More precisely, instances of that class achieved precision, recall, mAP@0.5 and mAP@0.5:0.95 of 0.546, 0.40, 0.45 and 0.24, respectively, with the instances themselves making up 4% of the total cases. That was in contrast to the metrics of the other classes that outperformed the metrics of the underrepresented one by 12% to 38%. Nevertheless, the metrics reached in our case, were satisfactory, especially considering that the algorithm was trained with so few training images, and showed that YOLOv5 models of varying architectures can be utilized to detect trees in UAV imagery and classify their health status effectively. This was an indication that depending on the specific dataset used, the same version of the YOLO algorithm can have quite different results.

The second pattern of the models' performances that should be noted is that the metrics of the damaged class were the most sensitive to the choice of model architecture, having the highest variability in precision, recall, mAP@0.5 and mAP@0.5:0.95. This could be due to the nature of the appearance of the damaged class, that is more dependent on model architecture to be detected correctly.

YOLO and YOLO-based models have shown that they can be effectively used for real-time agricultural applications. In (Liu et al., 2022), researchers came up with a model based on the lightweight YOLO v4-tiny model, that could detect seedling maize weeds in real time, with the detection speed reaching 57.33 f/s. It is necessary to note that real time applications also depend on the available and used hardware, as inference times are highly dependent on processing power. Especially in UAVs, the ability of the YOLO model to be lightweight enough but at the same time capable of high quality inference is of critical importance, since typically, hardware of higher processing power are of larger size and weight, requiring larger and thus more expensive UAVs. With our model's inference speed (77.5ms-328.8ms), we can deduce that the trained model can be used in real time or near real time applications in precision agriculture tasks.

Regarding the application of SAM-CNN, given the difference in tasks of the two algorithms, a direct comparison between YOLO and SAM-CNN is not possible, however we can confidently note that in a real world scenario where instant object detection and classification is needed, YOLO is far more functional, but in a scenario where high performance classification is more important and object bounding boxes are available, the SAM-CNN algorithm has the upper hand. Hypothetically, a pipeline consisting of a YOLO model whose outputs of bounding boxes could be passed to SAM to create object segmentations and classify them with a CNN at a much higher quality would be ideal.

As an extension of the current research effort, future work will include application of the presented machine learning object detection and classification pipeline on thermal and near infrared images, with the final aim of early detection of the existence of Verticillium wilt on olive trees.

Chapter 4

Case: RoBERTa-Assisted Outcome Prediction in Ovarian Cancer Cytoreductive Surgery using Operative Notes

Contemporary efforts to predict surgical outcomes and postoperative complications usually focus on the associations between traditional surgical risk factors, including age or preoperative albumin (Uppal et al., 2013; Barber et al., 2015). In addition to risk factors in discrete data fields, we now have access to abundant textual data within the digital medical records. In the era of healthcare digitalization, the increasing implementation of Electronic Health Records (EHR) at UK Hospitals has created valuable data sources for clinical and translational research (Economics, 2019). Although EHRs hold structured data, a large proportion of clinical notes are in narrative text format. It is estimated that unstructured data accounts for more than 80 percent of currently available healthcare data(Martin-Sanchez and Verspoor, 2014). Reading note text and extracting information is resource intensive. Artificial Intelligence (AI) has emerged as a potential solution in harnessing these data.

More specifically, Natural Language Processing (NLP) is the AI discipline that focuses on extracting information form texts by converting narrative clinical notes into a structured format. The NLP methods have been shown to achieve remarkable results in such tasks using hundreds to thousands clinical notes (Spasic and Nenadic, 2020). Their implementation has been promoted and accelerated during the COVID era (Zhu et al., 2022a). Nevertheless, clinical research has been heavily affected by the underutilization of unstructured data from EHRs (Seol et al., 2019).

Amongst the best NLP models employed to date, the Bidirectional Encoder Representations from Transformers (BERT) model was created by Google in 2018. Thanks to its architecture, it can extract information from texts by considering bidirectional contextual information (Devlin et al., 2018a). BERT's advanced information extraction capacities when combined with the fact that traditional NLP methods such as Word2Vec have shown promising results in classification tasks in clinical settings (Barber et al., 2021b), can lead to the reasonable expectation that a BERT-based classification model would outperform previously used methodologies.

Since 2018, several augmentations occurred, with Facebook publishing the Robustly Optimized BERT Approach (RoBERTa) language model in 2019 (Liu et al., 2019), surpassing previously set records. The RoBERTa is a late, robust, unsupervised pre-trained language model that can be used in the context of supervised tasks with outstanding results (Wang et al., 2018).

Undoubtedly, the abundance of clinical information is locked in clinical narratives. Documentation of EHRs is now developing into standard practice. For instance, the surgeons spend significant time documenting and reading, amongst other tasks, narrative descriptions of operative reports and findings (Rosenbloom et al., 2011). Developing tools to facilitate clinical review of these unstructured data can derive clinical meaningful insights for advanced Epithelial Ovarian Cancer (EOC), a heterogeneous disease. Compared to standard approaches, they can potentiate condensation of results from several tasks and optimize analysis time. One aspiration could be the prediction of no residual disease (R0 resection) following cytoreductive surgery for EOC. Such task of confirming macroscopic clearance remains subjective (Laios et al., 2022b), to the point that photographic ”mapping” has been recommended that allows for an assessment of the surgical effort at primary surgery or provides a baseline for determining the effect of neo-adjuvant chemotherapy at delayed surgery (Jones and Mohamed, 2020). As a result, most of the quantitative intra-operative assessment tools have mainly focused on their predictive value for suboptimal surgery (Hosoya et al., 2022). To improve modern care, application of NLP tools could be useful to determine whether processing of unstructured full text documents improves the ability to forecast outcomes in clinical conditions with significant heterogeneity such as EOC.

In this work, we utilized the pre-trained RoBERTa-base language model to predict whether residual disease persists in EOC patients following their cytoreductive surgery. We hypothesized that operative notes contain valuable information associated with surgical outcomes. We aimed to develop an NLP methodology that would address the objectiveness of R0 resection through information hidden in unstructured operative notes.

4.1 Methodology

4.1.1 Dataset Collection

Electronic Health Records EHR were queried to identify women with advanced EOC who underwent cytoreductive surgery at St James’s University Hospital, Leeds from January 2014 to December 2019. The modern EHR dataset included the following clinical features; diagnosis codes (ICD-10 codes), procedure codes (OPCS-4 codes), age at diagnosis, grade, stage and operative notes with findings. An internally developed advanced EOC clinical database was integrated with the EHR system

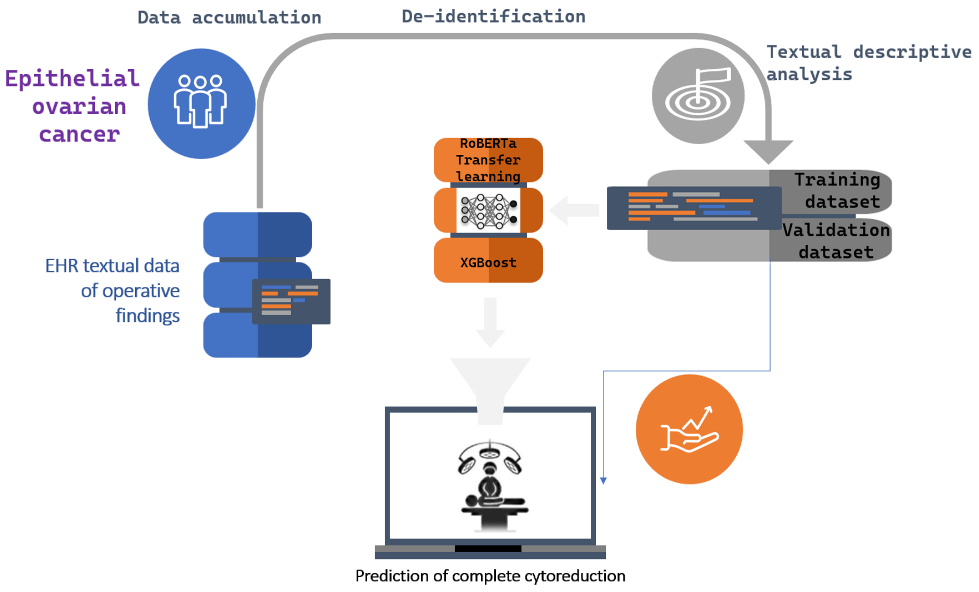

Data accumulation
De-identification
Epithelial ovarian cancer
Textual descriptive analysis
RoBERTa Transfer learning
XGBoost
Training dataset
Validation dataset
EHR textual data of operative findings
Prediction of complete cytoreduction

Figure 4.1: The components and the flow of the machine learning pipeline applied in our case

(Newsham et al., 2011) to provide availability of discrete and engineered data. Institutional research ethics board approval was obtained through the Leeds Teaching Hospitals Trust (MO20/133163/18.06.20), and informed written consent was obtained. The study was added to the UMIN/CTR Trial Registry (UMIN000049480). Treatment was pre-operatively planned at the weekly central gynecological oncology MultiDisciplinary Team (MDT) meeting prior to patient review. The cohort details, hospital setting, indications for surgery and surgical procedures have been described in previous studies (Laios et al., 2022b,c). Comprehensive visual assessment of all the areas of the abdomen and pelvis was routinely performed, and no visible residual disease was documented as R0 resection. The analysis took place in three steps: Firstly, words and combinations of words were analyzed based on their frequency and the case they concerned. Following the initial descriptive text analysis, the RoBERTa classifier was employed to predict case outcomes based on operative notes. Lastly, an XGBoost classification model was tasked with predicting the same outcome, this time using tabular discrete data, but also the probabilities that were derived from the RoBERTa classifier of the second step. A Flowchart of our approach is shown in Figure 4.1.

4.1.2 Textual descriptive analysis

For the analysis of the text, word frequencies were calculated, and tables were created using the most common words and n-grams. N-grams are continuous word sequences of words, as they could be found in the text. The length of the n-grams can be as small as one, meaning one word, or as large as the entirety of the text. N-grams are important because they carry contextual information more than simple words do. To find the n-grams that best discriminated between the two cases, we performed an analysis based on the Term Frequency – Inverse Document Frequency (TF-IDF). The TF-IDF is a metric used to quantify n-gram importance in a particular document (Tang and Ng, 2006). The score, as implied by its name, is a function of the number of times the n-gram appears in the document adjusted for the number of times it appears in the rest of the documents, as shown also in equation 4.1.

$$tfidf_{i,j} = tf_{i,j} * \log \frac{N}{df_i} \tag{4.1}$$

Where:

- $tf_{i,j}$ is the number of occurences of n-gram i in document j
- df_i is the number of documents containing i
- N is the total number of documents

For each of the two possible outcomes (R0 resection vs non-R0 resection) we compiled a document consisting of the concatenation of all the individual notes that concerned this outcome. The words inside the documents were reduced to their lemmas, to make the analysis more representative of the real n-gram frequency without accounting for word conjugation. The two resulting documents were inputted to Sklearn's TfidfVectorizer, that was tasked with assigning scores per n-gram, per document. A high TF-IDF score for an n-gram in a document signifies n-gram importance to this document. The TF-IDF n-gram scores for the documents reporting non-R0 resection were then subtracted from the TF-IDF scores for the documents reporting R0 resection. This way, the higher the absolute difference in scores, the higher the ability of the n-gram to discriminate between the two cases. Positive difference scores show that the n-gram belongs to R0 resection case notes, while negative scores show the opposite.

4.1.3 Natural language classification with RoBERTa

We utilized the pre-trained RoBERTa-base language model to extract information from the unstructured surgical notes through transfer learning. Transfer learning is the process of re-training part of a pre-trained model on specific data to fine-tune its performance for a specific task. The initial training often uses vast datasets that hold most of the information relevant to the task at hand. Re-training allows for finer details to be captured by the model. The main advantage of transfer learning is that the resulting model can reach high performance without needing to use large amounts of data. The RoBERTa-base language model is a RoBERTa language model pre-trained on a large corpus of English data using the BERT-base architecture and has 125 million parameters (Liu et al., 2019). The surgical data was used to train and test the model at a ratio 4:1. The model was trained for 40 epochs. Discrimination was measured using the most common performance metrics for classification tasks, namely with:

- Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$
- Precision = $\frac{TP}{TP+FP}$
- Recall = $\frac{TP}{TP+FN}$
- F1-score = $\frac{2*Precision*Recall}{Precision+Recall}$
- Area under the Receiver Operating Characteristic curve (AUROC)
- Area under the Precision-Recall curve (AUPRC)

Understanding how the RoBERTa reached the conclusions is essential in evaluating its performance. For such an extremely complicated model, tracing the individual impact of the text tokens on the final prediction is a task requiring advanced decompositional methods. In this effort, we employed the transformers-interpret Python library (Pierse, 2021) to explain and visualize the factors that contributed to the model's prediction accuracy. In turn, the library employs the captum model interpretability and understanding library (Kokhlikyan et al., 2020). Using integrated gradients, the library evaluates the contribution of each input feature to the model output of the model. The net result is an attribution score for each token; that is positive when the token contributes towards class prediction, and negative in the reverse scenario.

As a final step, we employed a surrogate model in order to augment the explainability effort. In this context, a surrogate model denotes a simpler model compared to the powerful original one (in our case RoBERTa), whose outputs are interpretable. The surrogate model trains on the outputs of the original, and through its interpretable coefficients, offers a way to access the decision process of the complex, original model. The surrogate model used was a simple logistic regression. The dependent variable was the RoBERTa predictions in the form of binary integer values, created by setting a threshold of 0.5 on the original probabilities, while the independent variables were the TF-IDF sentence vectors created through the method of TF-IDF vectorization. In this way, the aim was two-fold: The surrogate model would serve both as an explainer and as a conceptual link between the RoBERTa outputs and the TF-IDF scores created in the descriptive analysis.

4.1.4 XGBoost classification model

Subsequently, we trained an XGBoost model (Chen and Guestrin, 2016) to predict R0 resection using a combination of structured and unstructured data sources. The independent variables included the Aletti Surgical Complexity Score (SCS), the size of the largest bulk of the disease in centimetres, the age of the patient, the Pre-Surgery CA125, the IntraOperative Mapping of ovarian cancer (IMO) score, the operative time in minutes, the Estimated Blood Loss (EBL), the Pre-treatment CA125, the tumour grade encoded as a binary variable, the Peritoneal Carcinomatosis Index (PCI), the timing of surgery (encoded as a binary variable where primary debulking equalled 0 and interval debulking surgery equalled 1), the ANAtomic FIngerprints (ANAFI) score, and the probabilities that the RoBERTa classifier outputted when solely tasked to predict R0 resection (real number in the interval of 0 to 1). The PCI and IMO scores were calculated at the beginning of surgery to describe the intra-operative location of the disease (Jacquet and Sugarbaker, 1996; Sehouli et al., 2003). The Aletti SCS was assigned to describe the surgical effort (Aletti et al., 2007). The ANAFI score is an AI-derived novel intra-operative score that assigns specific weights to the EOC dissemination patterns (Laios et al., 2023). It appears to be more predictive of R0 resection than the entire PCI and IMO scores whilst it retains its prognostic power. Most of these discrete and engineered data predictors have been interrogated in previous studies (Laios et al., 2022b,c, 2023, 2021, 2022a).

The hyperparameters of the XGBoost model were selected by using an exhaustive grid hyperparameter search. The grid search also implemented cross validation. The hyperparameter grid is shown in Table 4.1. The feature importance was determined using the SHAPley Additive Explanations (SHAP) framework to interpret the model's predictions based on the Shapley values (Lundberg et al., 2020).

4.2 Results

4.2.1 The Dataset

Using the ICD-10 code for EOC, we identified 555 cases of EOC cytoreduction performed by eight surgeons between January 2014 and December 2019. This cohort has been previously described (Laios et al., 2022b,c). Some basic descriptive statistics are shown in Table 1. The rate of complete cytoreduction was 65.4%.

4.2.2 Textual descriptive analysis

Discrete word clouds weighted by n-gram TF-IDF score difference (Table 2) between R0 and non-R0 resection were identified (Figure 4.2).

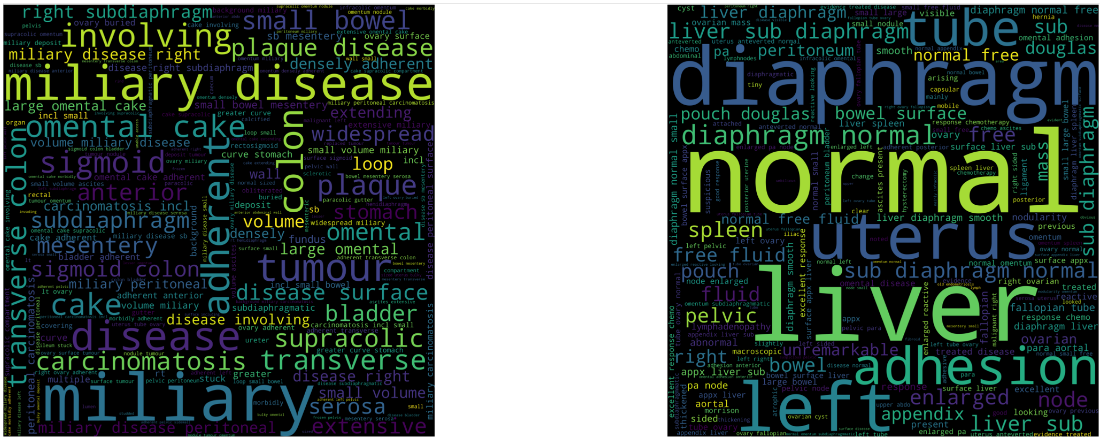


Figure 4.2: N-gram word clouds for findings notes where residual disease is non zero (left) and zero (right)

The words ”normal” and ”miliary” best discriminated between both groups. For non-R0 resection prediction, these included n-grams related to the EOC dissemination, such as ”omental cake”. The appearance of the cancer was best described by the predictive n-gram ”miliary disease”. The average word count was 320±98 vs 292±105 in case notes with non-R0 vs R0 resection, respectively. The average stop word count was 11.22±6.27 vs 9.86±5.89 in case notes with non-R0 vs R0 resection, respectively.

4.2.3 Natural language classification with RoBERTa

The model reached high evaluation metrics (Area under ROC 0.86; area under precision-recall curve 0.87 , precision, recall and F1 score of 0.77 and accuracy of 0.81 (Figures 4.3, 4.4 and 4.5)), surpassing even specialized BERT and DistilBERT models tested (BioBERT (DMIS-Lab): R 0.6, P 0.84, F1 0.7, ACC 0.79, AUROC 0.84, AUPRC 0.84, ClinicalBERT (Medicalai): R 0.68, P 0.72, F1 0.7, ACC 0.76, AUROC 0.82, AUPRC 0.82 and BioClinicalBERT (Alsentzer et al., 2019): R 0.64, P 0.76, F1 0.69, ACC 0.77, AUROC 0.81, AUPRC 0.79).. The true positives, true negatives, false positives and false negatives were 35, 56, 10, 10 respectively.

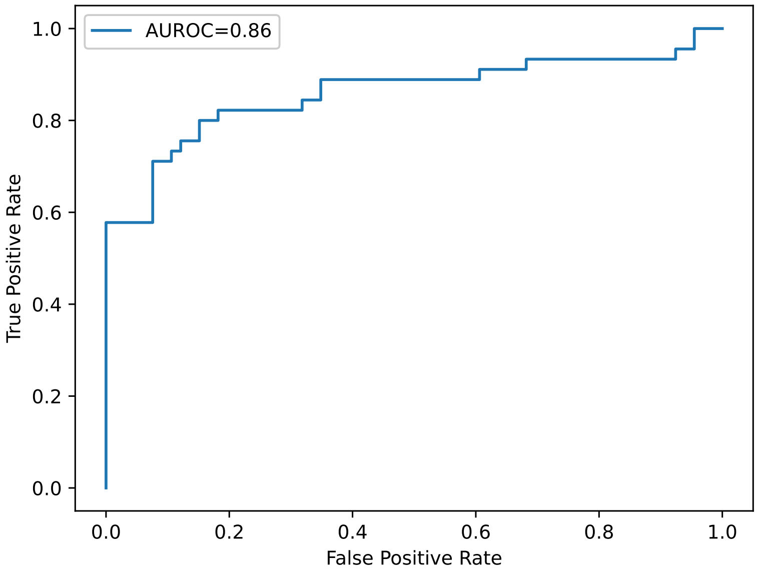


Figure 4.3: Receiver operating characteristic curve and area under the curve for the RoBERTa classifier

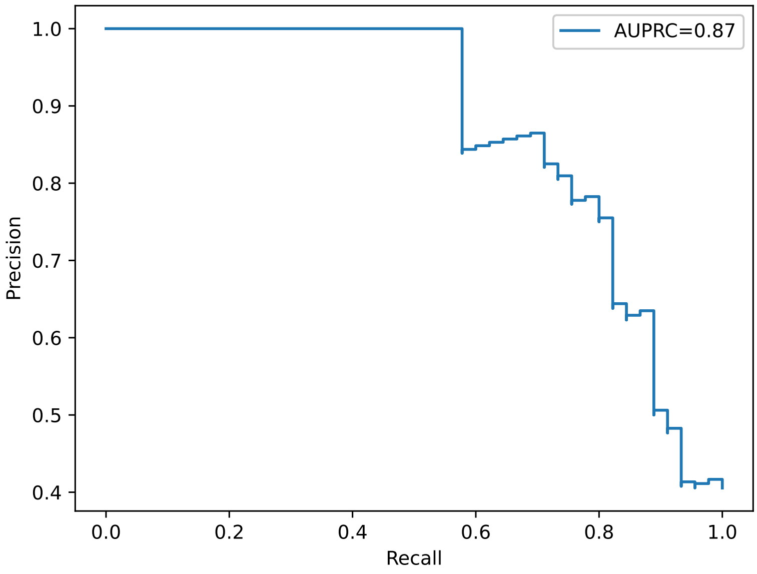


Figure 4.4: Precision-recall curve and area under the curve for the RoBERTa classifier

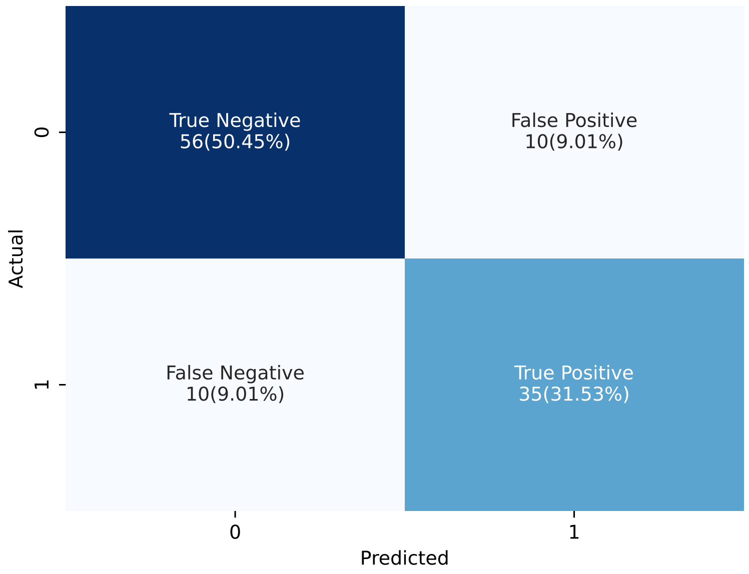


Figure 4.5: Confusion matrix for the RoBERTa classifier

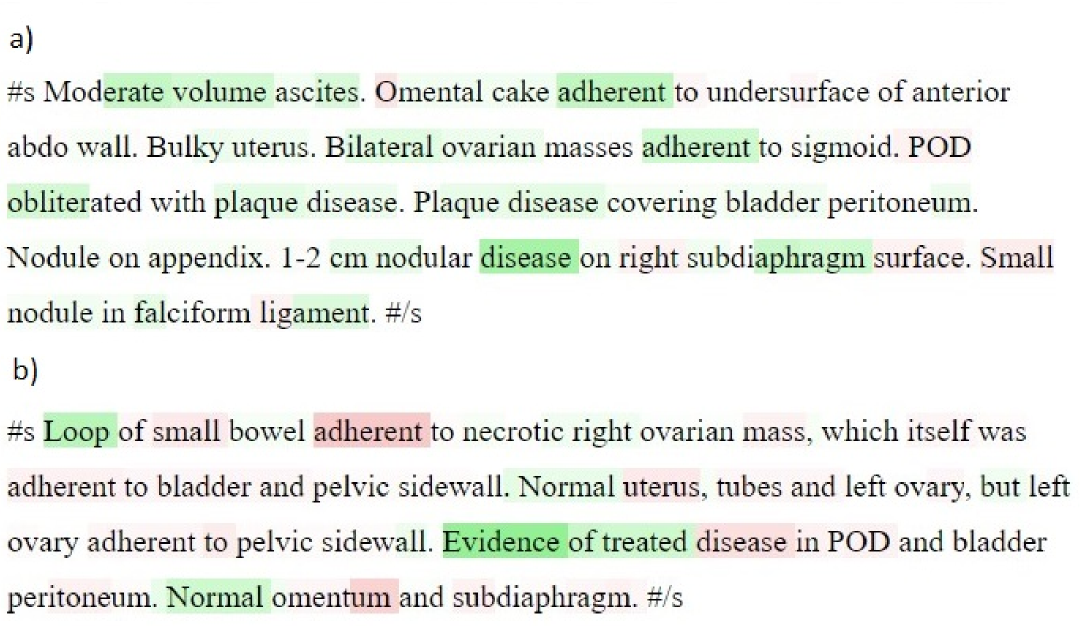
The explanations of the model's predictions were visualized in a highlighted text plot, where negatively contributing tokens are coloured as red, and positively contributing tokens are coloured as green, whereas the colour intensity is translated to an attribution score strength (Figure 4.6).

As shown in the figure, it is easy to discern the fact that there is a correlation between word contribution to the prediction and TF-IDF score difference. This makes sense, as n-grams with high TF-IDF score difference tend to discriminate better between the two cases. However, it is equally important to see that not all words that have high prediction contribution score appear as entries in table 2. That is due to the fact that since RoBERTa is able to capture contextual meaning spanning several words that could also be non sequential, it is possible that local information that wasn't apparent though simple TF-IDF analysis was now deemed as important to the prediction.

The results of the surrogate logistic regression model employed, further reinforce the results acquired from the RoBERTa model. Specifically, n-grams that reached high TF-IDF difference scores (Table 2), appeared as top coefficients in the logistic regression, in either direction (Figure 4.7).

4.2.4 XGBoost classification model

The XGBoost model that employed both discrete features and the probabilities from the RoBERTa classifier was then trained on the same training data set as the RoBERTa. The grid hyperparameter search resulted in 180 model evaluations with the best combination of hyperparameters shown on table 4.1 alongside the search spaces.



a)

#s Moderate volume ascites. Omental cake adherent to undersurface of anterior abdo wall. Bulky uterus. Bilateral ovarian masses adherent to sigmoid. POD obliterated with plaque disease. Plaque disease covering bladder peritoneum. Nodule on appendix. 1-2 cm nodular disease on right subdiaphragm surface. Small nodule in falciform ligament. #/s

b)

#s Loop of small bowel adherent to necrotic right ovarian mass, which itself was adherent to bladder and pelvic sidewall. Normal uterus, tubes and left ovary, but left ovary adherent to pelvic sidewall. Evidence of treated disease in POD and bladder peritoneum. Normal omentum and subdiaphragm. #/s

Figure 4.6: Explainability on the RoBERTa inference on textual data. The green highlighting indicates the section of text that contributed positively in the classification of the note as belonging to the assigned class, while the red highlighting indicates the opposite. Examples are text instances correctly classified as describing cases where a) residual disease persisted and b) no residual disease persisted after surgery

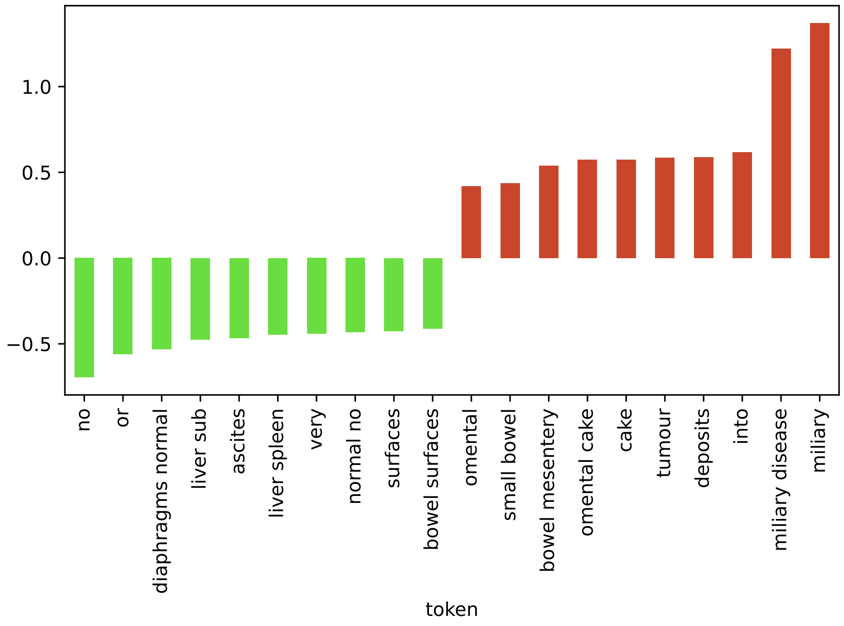


Figure 4.7: The top 10 n-grams with the lowest (green) and highest (red) coefficients of the logistic regression model. The negative sign denotes non-existence of residual disease and vice versa.

Table 4.1: Hyperparameter search space of grid search and chosen parameters

Hyperparameter	**Search space**	**Chosen hyperparameter value**
max_depth	[4, 6, 8, 10]	4
n_estimators	[500, 800, 1200]	500
learning_rate	[0.01, 0.03, 0.08]	0.01
colsample_bytree	None	0.5

For the XGBoost model, while precision, recall, f1 score and accuracy remained static, a marginal performance improvement was demonstrated as shown by the AUPRC and AUROC reaching 0.91. The RoBERTa probabilities, when used as a prediction feature performed significantly better than discrete features, but also engineered features such as the ANAFI score (Figure 4.8).

4.2.5 Result Evaluation and Discussion

In this proof-of-principle study, we demonstrated the capability of the RoBERTA classifier to extract and process information from unstructured operative note formats that can enable important clinical tasks, such as R0 resection prediction following EOC surgical cytoreduction. We showcased how EHRs can be a helpful data source for supporting surgeon's activities by automated data coding for quality assessment, while reducing the burden for chart review. As an estimated 70% of clinicians report EHR-related, specialty-specific burnout (Gardner et al., 2019), this information may guide healthcare organisations how to remediate burnout amongst their staff. Equally, we surmise this effort can help establishing interoperability standards of surgical narration to ensure objectivity, when it comes to reporting residual disease. Working with EHR data is relatively challenging due to data heterogeneity. Being able to quickly retrieve important information stored in surgical narratives carries the potential to improve understanding of patient journeys and identify subgroups of patients for research purposes. For those reasons, the design and application of a system that could offer the NLP AI-derived insights directly to the surgeon in real time would be extremely beneficial. The system could offer an objective feedback on written notes. A study on the effects of such a system should be investigated.

The driving motivation behind this effort was to explore the potential of using the RoBERTa algorithm in the EOC domain. This transformer architecture has been recently used to extract adverse drug events from biomedical text to monitor drug-safety (Jain et al., 2021). Barber et al initially developed an NLP-augmented algorithm that improved the ability to predict postoperative complications and hospital readmissions

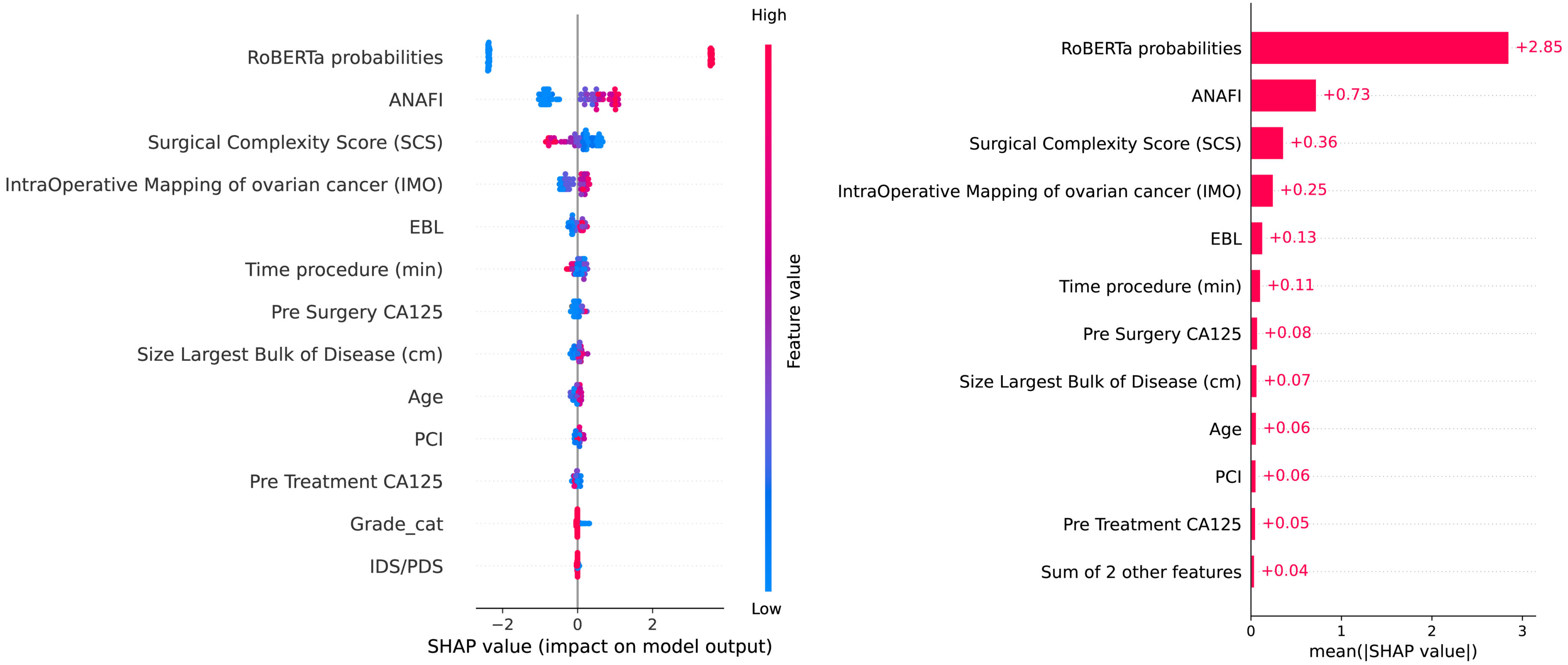


Figure 4.8: Explainability plots for the XGBoost classification model. The beeswarm feature impact plot (left) visualizes the relationship between direction of model prediction and value of feature. The bar feature impact plot (right), shows the absolute impact of each feature on model prediction

among women with EOC undergoing surgical cytoreduction (Barber et al., 2021a). They compiled discrete data with different types of NLP features from unstructured clinical notes, and sequentially employed machine learning to build new sets of features. Herein, we purely used a novel NLP tool that recognizes the specific local textual context thus enabling a recommendation concerning the prediction of residual disease. Preprocessing steps contributed to the rather high AUROC of 0.86, which shows how surgeons tend to capture more of the predictive information in their words. This "hunch" critically layered upon situational awareness and human factors has been addressed in Laios et al. (2022b). The model specificity was higher than its sensitivity, which is critical, should this be used as a cancer screening tool for quality control. Reports of surgical findings are less restrictive in vocabulary than other EHRs but their efficiency at scale has never been previously examined. They do not usually contain highly complex sentence structures, so they are not incorrectly abstracted as a result. By avoiding the model to make assumptions, this advantage would potentially explain the high-performance accuracy.

More importantly, we demonstrated a distinct pattern of word differential expression between R0 resection and non-R0 resection operative notes from 555 surgical events. While survival is the ultimate treatment outcome, prediction of residual disease is a key issue in the advanced EOC trajectory. This disease quantification can valuably complement work using AI to predict EOC-specific surgical outcomes (Laios et al., 2022b,c, 2023, 2021, 2022a) and validate the paradigm shift towards complete clearance to improve the survival outcomes of these patients (Jong et al., 2022; Laios et al., 2022b). The use of language in medicine is often underestimated not that all Gynaecologic Oncology Surgeons speak the same language (Brennan and Moran, 2021). While addressing the need to improve standardization and reproducibility of surgical outcomes, we made some interesting observations. Despite several words or n-grams being commonly shared between examined surgical outcomes, several descriptive words were found to be predictive of residual disease. For instance, the words "stuck" and "adherent" tend to describe a more complex and morbid surgery; dissemination leading to residual disease was best described by "(small volume) miliary disease" or "miliary in all peritoneal surfaces". "Excellent response to chemo" was clearly an obvious indication to achieve R0 resection. Notably, the "completion of cytoreduction" (CC) scoring system was developed to evaluate the extent of resection for peritoneal malignancies (Jacquet and Sugarbaker, 1996). We clearly showed that the word "miliary" if quantified, rather refers to CC1 (residual disease nodules up to 2.5mm in size) of the perhaps outdated Sugarbaker classification (PCI) (Jacquet and Sugarbaker, 1996). We provide valid language evidence that the CC score is more likely to give a convincing and reproducible description for residual disease in EOC. In addition, subtle performance superiority of textual data when compared with discrete surgical data can be also invoked. Going forward, data integration between structured and unstructured formats can promote innovative thinking to perfect the prediction of surgical outcomes.

The challenges of Machine Intelligence in healthcare have been consistently addressed (NIoBIa., 2019). We anticipate portability of our RoBERTa algorithm across similar practice settings by conducting original studies albeit we acknowledge the heterogeneous nature of the clinical language. Historically, linguistic models are evaluated by perplexity, i.e., the probability of predicting the word in its context. Our study used retrospective data from a single institution. On that note, our data size was small-to-moderate, which entertains the general wisdom that the less data we feed the model, the lower the expected quality. We reiterate our strong preference for explainable NLP methods (Chen et al., 2019), which has been showcased in this work. Understanding the features that drive a model prediction can potentially support decision-making in the healthcare domain. As NLP is moving to deep learning, it is becoming increasingly challenging for these complex non-linear data transformations to satisfy transparency (Shickel et al., 2018).

The latest hype from the technological advancements in large language models has been embraced with some cautious excitement. Undoubtedly, the AI-based chatbots engage in a capacity to understand multiple languages and possess knowledge of various topics. They can generate fabricated information in healthcare settings word by word (Sallam, 2023). In ovarian cancer research, most efforts focus on addressing the disease heterogeneity (Hu et al., 2020). It is likely that this heterogeneity contains “special grammars” that cannot be distilled from simply vast amounts of pre-trained textual data resources. Our work highlights the need for a bespoke, proprietary ovarian cancer-specific natural language that can pay attention to detail and learn beyond human knowledge.

Chapter 5

Conclusions

The level of quality in the models that utilize transfer learning has been demonstrated throughout this thesis. From the first case, that of detection of existence of disease in olive trees, we noted that fine-tuning pre-trained weights worked well, especially considering the fact that in this problem both domain and task differed between source and target. The result was the creation of a model that performed great in object detection and classification. Furthermore, the application of the zero-shot segmentation model, SAM, produced equally great results in its own task. Given the fact that in the case of SAM the source domain was more similar to the target domain, since the initial model was trained on nearly a billion image segmentations of different objects, the difference in performance between YOLO and SAM in their respective tasks was expected. In the second case, of prediction of cytoreduction from medical notes with the pre-trained RoBERTa language model, we noticed that the the model was more than able to capture the information well, especially considering the number of instances used for the fine-tuning. Both cases highlighted the ability of transfer learning models to overcome common machine learning problems, such as scarcity of data and/or computational resources, in two fields that differed greatly; computer vision and natural language processing.

As seen throughout this thesis, transfer learning is a technique whose reach, performance and flexibility in application is unparalleled. With applications both in structured and unstructured data, with proven performance, transfer learning is in the process of transforming the way machine learning algorithms and models are developed. The effect that it is expected to have both on the academic and the business worlds is hard to quantify, and even harder to predict, however it is certain that it is here to stay.

Bibliography

R. Y. Aburasain, E. A. Edirisinghe, and A. Albatay. Palm tree detection in drone images using deep convolutional neural networks: Investigating the effective use of yolo v3. In C. Biele, J. Kacprzyk, J. W. Owsiński, A. Romanowski, and M. Sikorski, editors, *Digital Interaction and Machine Intelligence*, pages 21–36, Cham, 2021. Springer International Publishing. ISBN 978-3-030-74728-2.

F. Ahmed, J. C. Mohanta, A. Keshari, and P. S. Yadav. Recent advances in unmanned aerial vehicles: A review. *Arabian Journal for Science and Engineering*, 47(7): 7963–7984, Apr. 2022. doi: 10.1007/s13369-022-06738-0. URL https://doi.org/10.1007/s13369-022-06738-0.

G. D. Aletti, S. C. Dowdy, K. C. Podratz, and W. A. Cliby. Relationship among surgical complexity, short-term morbidity, and overall survival in primary surgery for advanced ovarian cancer. *American Journal of Obstetrics and Gynecology*, 197(6):676.e1–676.e7, Dec. 2007. doi: 10.1016/j.ajog.2007.10.495. URL https://doi.org/10.1016/j.ajog.2007.10.495.

E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. B. A. McDermott. Publicly available clinical bert embeddings, 2019.

S. Alstrom. Characteristics of bacteria from oilseed rape in relation to their biocontrol activity against verticillium dahliae. *Journal of Phytopathology*, 149(2):57–64, Feb. 2001. doi: 10.1046/j.1439-0434.2001.00585.x.

S. Arooj, S. u. Rehman, A. Imran, A. Almuhaimeed, A. K. Alzahrani, and A. Alzahrani. A deep convolutional neural network for the early detection of heart disease. *Biomedicines*, 10(11), 2022. ISSN 2227-9059. doi: 10.3390/biomedicines10112796. URL https://www.mdpi.com/2227-9059/10/11/2796.

P. Arroyo, J. Gómez-Suárez, J. L. Herrero, and J. Lozano. Electrochemical gas sensing module combined with unmanned aerial vehicles for air quality monitoring. *Sensors and Actuators B: Chemical*, 364:131815, 2022. ISSN 0925-4005. doi: https://doi.org/10.1016/j.snb.2022.131815. URL https://www.sciencedirect.com/science/article/pii/S0925400522004579.

P. Ayres. Water relations of diseased plants. in “water deficits and plant growth”(tt kozlowski, ed.), 1978.

A. G. Báidez, P. Gómez, J. A. D. Río, and A. Ortuño. Dysfunctionality of the xylem in olea europaea l. plants associated with the infection process by verticillium dahliae kleb. role of phenolic compounds in plant defense mechanism. *Journal of Agricultural and Food Chemistry*, 55(9):3373–3377, 2007. doi: 10.1021/jf063166d.

E. L. Barber, S. E. Rutstein, W. C. Miller, and P. A. Gehrig. A preoperative personalized risk assessment calculator for elderly ovarian cancer patients undergoing primary cytoreductive surgery. *Gynecologic Oncology*, 139(3):401–406, 2015. ISSN 0090-8258. doi: https://doi.org/10.1016/j.ygyno.2015.09.080. URL https://www.sciencedirect.com/science/article/pii/S0090825815301414.

E. L. Barber, R. Garg, C. Persenaire, and M. Simon. Natural language processing with machine learning to predict outcomes after ovarian cancer surgery. *Gynecol. Oncol.*, 160(1):182–186, jan 2021a.

E. L. Barber, R. Garg, C. Persenaire, and M. Simon. Natural language processing with machine learning to predict outcomes after ovarian cancer surgery. *Gynecologic Oncology*, 160(1):182–186, Jan. 2021b. doi: 10.1016/j.ygyno.2020.10.004. URL https://doi.org/10.1016/j.ygyno.2020.10.004.

S. Bera and V. K. Shrivastava. Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification. *International Journal of Remote Sensing*, 41(7):2664–2683, 2020.

D. Berrar et al. Cross-validation., 2019.

K. Bjerge, J. Alison, M. Dyrmann, C. E. Frigaard, H. M. R. Mann, and T. T. Høye. Accurate detection and identification of insects from camera trap images with deep learning. *PLOS Sustainability and Transformation*, 2(3):e0000051, Mar. 2023. doi: 10.1371/journal.pstr.0000051. URL https://doi.org/10.1371/journal.pstr.0000051.

M. A. Blanco-López, R. M. Jiménez-Díaz, and J. M. Caballero. Symptomatology, incidence and distribution of verticillium wilt of olive trees in andalucía. *Phytopathologia Mediterranea*, 23(1):1–8, 1984. ISSN 00319465, 15932095.

V. Borisov, E. Kasneci, and G. Kasneci. Robust cognitive load detection from wristband sensors. *Computers in Human Behavior Reports*, 4:100116, 2021.

A. Botchkarev. Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. *arXiv preprint arXiv:1809.03006*, 2018.

D. J. Brennan and B. J. Moran. Time to evolve terminology from “debulking” to cytoreductive surgery (CRS) in ovarian cancer. *Annals of Surgical Oncology*, 28 (11):5805–5807, July 2021. doi: 10.1245/s10434-021-10490-4. URL https://doi.org/10.1245/s10434-021-10490-4.

E. Burnaev, P. Erofeev, and A. Papanov. Influence of resampling on accuracy of imbalanced classification. In *Eighth international conference on machine vision (ICMV 2015)*, volume 9875, pages 423–427. SPIE, 2015.

L. Chen, Y. Gu, X. Ji, C. Lou, Z. Sun, H. Li, Y. Gao, and Y. Huang. Clinical trial cohort selection based on multi-level rule-based natural language processing system. *J. Am. Med. Inform. Assoc.*, 26(11):1218–1226, nov 2019.

S. Chen, D. Yang, J. Liu, Q. Tian, and F. Zhou. Automatic weld type classification, tacked spot recognition and weld roi determination for robotic welding based on modified yolov5. *Robotics and Computer-Integrated Manufacturing*, 81:102490, 2023. ISSN 0736-5845. doi: https://doi.org/10.1016/j.rcim.2022.102490. URL https://www.sciencedirect.com/science/article/pii/S0736584522001727.

T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, pages 785–794, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785. URL http://doi.acm.org/10.1145/2939672.2939785.

P. N. Chowdhury, P. Shivakumara, L. Nandanwar, F. Samiron, U. Pal, and T. Lu. Oil palm tree counting in drone images. *Pattern Recognition Letters*, 153:1–9, 2022. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2021.11.016.

P. M. Ciarelli, E. Oliveira, C. Badue, and A. F. De Souza. Multi-label text categorization using a probabilistic neural network. *International Journal of Computer Information Systems and Industrial Management Applications*, 1(133-144):51, 2009.

J. M. Clements, D. Xu, N. Yousefi, and D. Efimov. Sequential deep learning for credit risk monitoring with tabular financial data. *arXiv preprint arXiv:2012.15330*, 2020.

B. C. Csáji et al. Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24(48):7, 2001.

C. A. de Sousa. An overview on weight initialization methods for feedforward neural networks. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 52–59. IEEE, 2016.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018a. URL https://arxiv.org/abs/1810.04805.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018b.

B. Ding, H. Qian, and J. Zhou. Activation functions and their characteristics in deep neural networks. In *2018 Chinese control and decision conference (CCDC)*, pages 1836–1841. IEEE, 2018.

DMIS-Lab. Dmis-lab/biobert-v1.1 · hugging face. URL https://huggingface.co/dmis-lab/biobert-v1.1.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

M. Economics. Ehrs should be a tool, not a task, 2019. URL https://www.medicaleconomics.com/article/ehrs-should-be-tool-not-task.

M. Fatourechi, R. K. Ward, S. G. Mason, J. Huggins, A. Schlögl, and G. E. Birch. Comparison of evaluation metrics in classification applications with imbalanced datasets. In *2008 seventh international conference on machine learning and applications*, pages 777–782. IEEE, 2008.

A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, 61:863–905, 2018.

L. Fichtel, A. M. Frühwald, L. Hösch, V. Schreibmann, C. Bachmeir, and F. Bohlander. Tree localization and monitoring on autonomous drones employing deep learning. In *2021 29th Conference of Open Innovations Association (FRUCT)*, pages 132–140, 2021. doi: 10.23919/FRUCT52173.2021.9435549.

D. A. Forsyth and J. Ponce. *Computer vision: a modern approach*. prentice hall professional technical reference, 2002.

A. L. Fradkov. Early history of machine learning. *IFAC-PapersOnLine*, 53(2):1385–1390, 2020. ISSN 2405-8963. doi: https://doi.org/10.1016/j.ifacol.2020.12.1888. URL https://www.sciencedirect.com/science/article/pii/S2405896320325027. 21st IFAC World Congress.

C. Gaffey and A. Bhardwaj. Applications of unmanned aerial vehicles in cryosphere: Latest advances and prospects. *Remote Sensing*, 12(6), 2020. ISSN 2072-4292. doi: 10.3390/rs12060948. URL https://www.mdpi.com/2072-4292/12/6/948.

R. L. Gardner, E. Cooper, J. Haskell, D. A. Harris, S. Poplau, P. J. Kroth, and M. Linzer. Physician stress and burnout: the impact of health information technology. *J. Am. Med. Inform. Assoc.*, 26(2):106–114, feb 2019.

J. Geiger, M. Bellahcene, Z. Fortas, A. Matallah-Boutiba, and D. Henni. Verticillium wilt in olive in algeria: geographical distribution and extent of the disease. *Olivae*, 82:41–43, 2000.

H. Ghoddusi, G. G. Creamer, and N. Rafizadeh. Machine learning in energy economics and finance: A review. *Energy Economics*, 81:709–727, 2019.

Y. Goldberg and O. Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

H. Guo, R. Tang, Y. Ye, Z. Li, and X. He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

M. Hammad and K. Wang. Fingerprint classification based on a q-gaussian multiclass support vector machine. In *Proceedings of the 2017 International Conference on biometrics engineering and application*, pages 39–44, 2017.

J. Hancock and T. M. Khoshgoftaar. Performance of catboost and xgboost in medicare fraud detection. In *2020 19th IEEE international conference on machine learning and applications (ICMLA)*, pages 572–579. IEEE, 2020.

J. Hauser and G. Katz. Metrics: you are what you measure! *European Management Journal*, 16(5):517–528, 1998.

K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

F. Herrera, F. Charte, A. J. Rivera, M. J. Del Jesus, F. Herrera, F. Charte, A. J. Rivera, and M. J. del Jesus. *Multilabel classification*. Springer, 2016.

S. Hosoya, K. Ueda, S. Odajima, K. Ogawa, H. Komazaki, T. Seki, M. Takenaka, M. Saito, H. Tanabe, K. Yamada, H. Takano, Y. Iida, N. Yanaihara, and A. Okamoto.

Scoring systems of peritoneal dissemination for the prediction of operative completeness in advanced ovarian cancer. *Anticancer Res.*, 42(1):115–124, jan 2022.

M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CsUR)*, 51(6):1–36, 2019.

M. Hossin and M. N. Sulaiman. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2):1, 2015.

Z. Hu, M. Artibani, A. Alsaadi, N. Wietek, M. Morotti, T. Shi, Z. Zhong, L. S. Gonzalez, S. El-Sahhar, E. M. Carrami, G. Mallett, Y. Feng, K. Masuda, Y. Zheng, K. Chong, S. Damato, S. Dhar, L. Campo, R. G. Campanile, H. S. majd, V. Rai, D. Maldonado-Perez, S. Jones, V. Cerundolo, T. Sauka-Spengler, C. Yau, and A. A. Ahmed. The repertoire of serous ovarian cancer non-genetic heterogeneity revealed by single-cell sequencing of normal fallopian tube epithelial cells. *Cancer Cell*, 37(2):226–242.e7, Feb. 2020. doi: 10.1016/j.ccell.2020.01.003. URL https://doi.org/10.1016/j.ccell.2020.01.003.

B. Huang, M. T. Kechadi, and B. Buckley. Customer churn prediction in telecommunications. *Expert Systems with Applications*, 39(1):1414–1425, 2012.

C. Huang, Y. Li, C. C. Loy, and X. Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.

P. Jacquet and P. H. Sugarbaker. Clinical research methodologies in diagnosis and staging of patients with peritoneal carcinomatosis. In *Cancer Treatment and Research*, pages 359–374. Springer US, 1996. doi: 10.1007/978-1-4613-1247-5_23. URL https://doi.org/10.1007/978-1-4613-1247-5_23.

H. Jain, N. Raj, and S. Mishra. A sui generis QA approach using RoBERTa for adverse drug event identification. *BMC Bioinformatics*, 22(Suppl 11):330, oct 2021.

A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1), 2021. ISSN 2227-7080. doi: 10.3390/technologies9010002. URL https://www.mdpi.com/2227-7080/9/1/2.

C. Janiesch, P. Zschech, and K. Heinrich. Machine learning and deep learning. *Electronic Markets*, 31(3):685–695, Apr. 2021a. doi: 10.1007/s12525-021-00475-2. URL https://doi.org/10.1007/s12525-021-00475-2.

C. Janiesch, P. Zschech, and K. Heinrich. Machine learning and deep learning. *Electronic Markets*, 31(3):685–695, 2021b.

K. Janocha and W. M. Czarnecki. On loss functions for deep neural networks in classification. *arXiv preprint arXiv:1702.05659*, 2017.

C. Jiang, Z. Wang, R. Wang, and Y. Ding. Loan default prediction by combining soft information extracted from descriptive text in online peer-to-peer lending. *Annals of Operations Research*, 266(1-2):511–529, 2018a.

M. Jiang, Y. Liang, X. Feng, X. Fan, Z. Pei, Y. Xue, and R. Guan. Text classification based on deep belief network and softmax regression. *Neural Computing and Applications*, 29:61–70, 2018b.

P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma. A review of yolo algorithm developments. *Procedia Computer Science*, 199:1066–1073, 2022. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2022.01.135.

R. Jiménez-Díaz, E. Tjamos, and M. Cirulli. Verticillium wilt of major tree hosts: olive. *A compendium of Verticillium wilts in tree species*, pages 13–16, 1998.

T. Jintasuttisak, E. Edirisinghe, and A. Elbattay. Deep neural network based date palm tree detection in drone imagery. *Computers and Electronics in Agriculture*, 192:106560, 2022. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2021.106560.

G. Jocher. YOLOv5 by Ultralytics, 5 2020.

M. Jones and F. Mohamed. Photography transillumination techniques: Multicystic peritoneal mesothelioma. *J. Biocommun.*, 44(1):e3, jun 2020.

D. D. Jong, M. Otify, I. Chen, D. Jackson, K. Jayasinghe, D. Nugent, A. Thangavelu, G. Theophilou, and A. Laios. Survival and chemosensitivity in advanced high grade serous epithelial ovarian cancer patients with and without a BRCA germline mutation: More evidence for shifting the paradigm towards complete surgical cytoreduction. *Medicina*, 58(11):1611, Nov. 2022. doi: 10.3390/medicina58111611. URL https://doi.org/10.3390/medicina58111611.

S. Kamaruddin and V. Ravi. Credit card fraud detection using big data analytics: use of psoaann based one-class classification. In *Proceedings of the international conference on informatics and analytics*, pages 1–8, 2016.

H. E. Kim, A. Cosa-Linan, N. Santhanam, M. Jannesari, M. E. Maros, and T. Ganslandt. Transfer learning for medical image classification: a literature review. *BMC medical imaging*, 22(1):69, 2022.

A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything, 2023.

N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson. Captum: A unified and generic model interpretability library for pytorch, 2020.

K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes. Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)*, pages 364–371. IEEE, 2017.

K. Kowsari, M. Heidarysafa, D. E. Brown, K. J. Meimandi, and L. E. Barnes. Rmdl: Random multimodel deep learning for classification. In *Proceedings of the 2nd international conference on information system and data mining*, pages 19–28, 2018.

K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.

M. Krichen, A. Mihoub, M. Y. Alzahrani, W. Y. H. Adoni, and T. Nahhal. Are formal methods applicable to machine learning and artificial intelligence? In *2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, pages 48–53, 2022. doi: 10.1109/SMARTTECH54121.2022.00025.

A. Krogh. What are artificial neural networks? *Nature biotechnology*, 26(2):195–197, 2008.

C. S. Ku, J. Xiong, Y.-L. Chen, S. D. Cheah, H. C. Soong, and L. Y. Por. Improving stock market predictions: An equity forecasting scanner using long short-term memory method with dynamic indicators for malaysia stock market. *Mathematics*, 11(11), 2023. ISSN 2227-7390. doi: 10.3390/math11112470. URL https://www.mdpi.com/2227-7390/11/11/2470.

E. Kubera, A. Kubik-Komar, P. Kurasiński, K. Piotrowska-Weryszko, and M. Skrzypiec. Detection and recognition of pollen grains in multilabel microscopic images. *Sensors*, 22(7), 2022. ISSN 1424-8220. doi: 10.3390/s22072690. URL https://www.mdpi.com/1424-8220/22/7/2690.

S. K. Kumar. On weight initialization in deep neural networks. *arXiv preprint arXiv:1704.08863*, 2017.

R. Kumari and S. K. Srivastava. Machine learning: A review on binary classification. *International Journal of Computer Applications*, 160(7), 2017.

A. Laios, A. Katsenou, Y. S. Tan, R. Johnson, M. Otify, A. Kaufmann, S. Munot, A. Thangavelu, R. Hutson, T. Broadhead, G. Theophilou, D. Nugent, and D. De Jong. Feature selection is critical for 2-year prognosis in advanced stage high grade serous ovarian cancer by using machine learning. *Cancer Control*, 28: 10732748211044678, jan 2021.

A. Laios, D. L. D. D. Freitas, G. Saalmink, Y. S. Tan, R. Johnson, A. Zubayraeva, S. Munot, R. Hutson, A. Thangavelu, T. Broadhead, D. Nugent, E. Kalampokis, K. M. G. de Lima, G. Theophilou, and D. D. Jong. Stratification of length of stay prediction following surgical cytoreduction in advanced high-grade serous ovarian cancer patients using artificial intelligence; the leeds l-AI-OS score. *Current Oncology*, 29(12):9088–9104, Nov. 2022a. doi: 10.3390/curroncol29120711. URL https://doi.org/10.3390/curroncol29120711.

A. Laios, E. Kalampokis, R. Johnson, S. Munot, A. Thangavelu, R. Hutson, T. Broadhead, G. Theophilou, C. Leach, D. Nugent, and D. De Jong. Factors predicting surgical effort using explainable artificial intelligence in advanced stage epithelial ovarian cancer. *Cancers (Basel)*, 14(14):3447, jul 2022b.

A. Laios, E. Kalampokis, R. Johnson, A. Thangavelu, C. Tarabanis, D. Nugent, and D. De Jong. Explainable artificial intelligence for prediction of complete surgical cytoreduction in advanced-stage epithelial ovarian cancer. *J. Pers. Med.*, 12(4):607, apr 2022c.

A. Laios, E. Kalampokis, R. Johnson, S. Munot, A. Thangavelu, R. Hutson, T. Broadhead, G. Theophilou, D. Nugent, and D. D. Jong. Development of a novel intra-operative score to record diseases' anatomic fingerprints (ANAFI score) for the prediction of complete cytoreduction in advanced-stage ovarian cancer by using machine learning and explainable artificial intelligence. *Cancers*, 15(3):966, Feb. 2023. doi: 10.3390/cancers15030966. URL https://doi.org/10.3390/cancers15030966.

T. Lazna, P. Gabrlik, T. Jilek, and L. Zalud. Cooperation between an unmanned aerial vehicle and an unmanned ground vehicle in highly accurate localization of gamma radiation hotspots. *International Journal of Advanced Robotic Systems*, 15(1):172988141775078, Jan. 2018. doi: 10.1177/1729881417750787. URL https://doi.org/10.1177/1729881417750787.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

J. Lee, J. Wang, D. Crandall, S. Šabanović, and G. Fox. Real-time, cloud-based object detection for unmanned aerial vehicles. In *2017 First IEEE International Conference on Robotic Computing (IRC)*, pages 36–43, 2017. doi: 10.1109/IRC.2017.77.

S. Lee, S. Kim, and S. Moon. Development of a car-free street mapping model using an integrated system with unmanned aerial vehicles, aerial mapping cameras, and a deep learning algorithm. *Journal of Computing in Civil Engineering*, 36(3):04022003, 2022. doi: 10.1061/(ASCE)CP.1943-5487.0001013. URL https://ascelibrary.org/doi/abs/10.1061/(ASCE)CP.1943-5487.0001013.

A. Levin, S. Lavee, and L. Tsror. Epidemiology of verticillium dahliae on olive (cv. picual) and its effect on yield under saline conditions. *Plant Pathology*, 52(2):212–218, 2003.

Y. Li, H. Mao, R. Girshick, and K. He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pages 280–296. Springer, 2022.

W. Liang, S. Luo, G. Zhao, and H. Wu. Predicting hard rock pillar stability using gbdt, xgboost, and lightgbm algorithms. *Mathematics*, 8(5):765, 2020.

S. Liu, Y. Jin, Z. Ruan, Z. Ma, R. Gao, and Z. Su. Real-time detection of seedling maize weeds in sustainable agriculture. *Sustainability*, 14(22), 2022. ISSN 2071-1050. doi: 10.3390/su142215088. URL https://www.mdpi.com/2071-1050/14/22/15088.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

F. J. López-Escudero and J. Mercado-Blanco. Verticillium wilt of olive: a case study to implement an integrated strategy to control a soil-borne pathogen. *Plant and Soil*, 344(1-2):1–50, Nov. 2010. doi: 10.1007/s11104-010-0629-2.

C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 852–869. Springer, 2016.

D. Lu and Q. Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28 (5):823–870, 2007.

S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

Y. Ma, W. Gong, and F. Mao. Transfer learning used to analyze the dynamic evolution of the dust aerosol. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 153:119–130, 2015.

S. Mackie, R. McCreadie, C. Macdonald, and I. Ounis. On choosing an effective automatic evaluation metric for microblog summarisation. In *Proceedings of the 5th Information Interaction in Context Symposium*, pages 115–124, 2014.

M. Madaan, A. Kumar, C. Keshri, R. Jain, and P. Nagrath. Loan default prediction using decision trees and random forest: A comparative study. In *IOP Conference Series: Materials Science and Engineering*, volume 1022, page 012042. IOP Publishing, 2021.

D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018.

N. K. Manaswi. *RNN and LSTM*, pages 115–126. Apress, Berkeley, CA, 2018. ISBN 978-1-4842-3516-4. doi: 10.1007/978-1-4842-3516-4_9. URL https://doi.org/10.1007/978-1-4842-3516-4_9.

I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis. Multiclass confusion matrix reduction method and its application on net promoter score classification problem. *Technologies*, 9(4), 2021. ISSN 2227-7080. doi: 10.3390/technologies9040081. URL https://www.mdpi.com/2227-7080/9/4/81.

M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. M. Gómez-Berbís. Named entity recognition: fallacies, challenges and opportunities. *Computer Standards & Interfaces*, 35(5):482–489, 2013.

F. Martin-Sanchez and K. Verspoor. Big data in medicine is driving big changes. *Yearb. Med. Inform.*, 9(01):14–20, aug 2014.

A. McCallum, K. Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Madison, WI, 1998.

Medicalai. Medicalai/clinicalbert · hugging face. URL https://huggingface.co/medicalai/ClinicalBERT.

F. Meng, X. Wang, F. Shao, D. Wang, and X. Hua. Energy-efficient gabor kernels in neural networks with genetic algorithm training method. *Electronics*, 8(1), 2019.

ISSN 2079-9292. doi: 10.3390/electronics8010105. URL https://www.mdpi.com/2079-9292/8/1/105.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

B. Mohit. Named entity recognition. In *Natural language processing of semitic languages*, pages 221–245. Springer, 2014.

S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah, and M. H. Alsharif. Towards the unmanned aerial vehicles (uavs): A comprehensive review. *Drones*, 6(6), 2022. ISSN 2504-446X. doi: 10.3390/drones6060147. URL https://www.mdpi.com/2504-446X/6/6/147.

M. V. Narkhede, P. P. Bartakke, and M. S. Sutaone. A review on weight initialization strategies for neural networks. *Artificial intelligence review*, 55(1):291–322, 2022.

Z. Naser, A. Al-Raddad Al-Momany, et al. Dissemination factors of verticillium wilt of olive in jordan. *Dirasat. Agricultural Sciences*, 25(1):16–21, 1998.

A. C. Newsham, C. Johnston, G. Hall, M. G. Leahy, A. B. Smith, A. Vikram, A. M. Donnelly, G. Velikova, P. J. Selby, and S. E. Fisher. Development of an advanced database for clinical trials integrated with an electronic patient record system. *Computers in Biology and Medicine*, 41(8):575–586, Aug. 2011. doi: 10.1016/j.compbiomed.2011.04.014. URL https://doi.org/10.1016/j.compbiomed.2011.04.014.

B. NIoBIa. Machine intelligence in healthcare: Nih, 2019. URL https://ncats.nih.gov/expertise/machine-intelligence#workshop.

R. Obiedat, L. Al-Qaisi, R. Qaddoura, O. Harfoushi, and A. M. Al-Zoubi. An intelligent hybrid sentiment analyzer for personal protective medical equipments based on word embedding technique: The covid-19 era. *Symmetry*, 13(12), 2021. ISSN 2073-8994. doi: 10.3390/sym13122287. URL https://www.mdpi.com/2073-8994/13/12/2287.

K. O'Shea and R. Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

G. Pajares. Overview and current status of remote sensing applications based on unmanned aerial vehicles (uavs). *Photogrammetric Engineering & Remote Sensing*, 81 (4):281–329, 2015. ISSN 0099-1112. doi: https://doi.org/10.14358/PERS.81.4.281. URL https://www.sciencedirect.com/science/article/pii/S0099111215300793.

S. Pan and Q. Yang. A survey on transfer learning. ieee transaction on knowledge discovery and data engineering, 22 (10), 2010.

G. Pang, C. Aggarwal, C. Shen, and N. Sebe. Editorial deep learning for anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6): 2282–2286, 2022.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011a.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011b.

G. F. Pegg and B. L. Brady. *Verticillium wilts*. CABI Publishing, UK, 2002.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

M. Pérez, F. Agüera, and F. Carvajal. LOW COST SURVEYING USING AN UNMANNED AERIAL VEHICLE. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-1/W2:311–315, Aug. 2013. doi: 10.5194/isprsarchives-xl-1-w2-311-2013. URL https://doi.org/10.5194/isprsarchives-xl-1-w2-311-2013.

M. Pérez-Rodríguez, N. Serrano, O. Arquero, F. Orgaz, J. Moral, and F. J. López-Escudero. The effect of short irrigation frequencies on the development of verticillium wilt in the susceptible olive cultivar 'picual' under field conditions. *Plant Disease*, 100(9):1880–1888, Sept. 2016. doi: 10.1094/pdis-09-15-1018-re.

C. Perlich, B. Dalessandro, T. Raeder, O. Stitelman, and F. Provost. Machine learning for targeted display advertising: Transfer learning in action. *Machine learning*, 95 (1):103–127, 2014.

C. Pierse. Transformers Interpret, Feb. 2021. URL https://github.com/cdpierse/transformers-interpret.

L. L. Pipino, Y. W. Lee, and R. Y. Wang. Data quality assessment. *Commun. ACM*, 45(4):211–218, apr 2002. ISSN 0001-0782. doi: 10.1145/505248.506010. URL https://doi.org/10.1145/505248.506010.

E. Politi, A. Garyfallou, I. Panagiotopoulos, I. Varlamis, and G. Dimitrakopoulos. Path planning and landing for unmanned aerial vehicles using AI. In *Lecture Notes in Networks and Systems*, pages 343–357. Springer International Publishing, Oct. 2022. doi: 10.1007/978-3-031-18461-1_23. URL https://doi.org/10.1007/978-3-031-18461-1_23.

A. Porta-Puglia, D. Mifsud, et al. First record of verticillium dahliae on olive in malta. *Journal of Plant Pathology*, 87(2):149, 2005.

S. Puliti and R. Astrup. Automatic detection of snow breakage at single tree level using yolov5 applied to uav imagery. *International Journal of Applied Earth Observation and Geoinformation*, 112:102946, 2022. ISSN 1569-8432. doi: https://doi.org/10.1016/j.jag.2022.102946.

W. A. Qader, M. M. Ameen, and B. I. Ahmed. An overview of bag of words; importance, implementation, applications, and challenges. In *2019 international engineering conference (IEC)*, pages 200–204. IEEE, 2019.

J. R. Quinlan. Learning decision tree classifiers. *ACM Computing Surveys (CSUR)*, 28 (1):71–72, 1996.

A. Rácz, D. Bajusz, and K. Héberger. Effect of dataset size and train/test split ratios in qsar/qspr multiclass classification. *Molecules*, 26(4):1111, 2021.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

F. Rayhan, S. Ahmed, A. Mahbub, R. Jani, S. Shatabda, and D. M. Farid. Cusboost: Cluster-based under-sampling with boosting for imbalanced classification. In *2017 2nd international conference on computational systems and information technology for sustainable solution (csitss)*, pages 1–5. IEEE, 2017.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection, 2015.

A. Rehman. Light microscopic iris classification using ensemble multi-class support vector machine. *Microscopy Research and Technique*, 84(5):982–991, 2021.

Z. Reitermanova et al. Data splitting. In *WDS*, volume 10, pages 31–36. Matfyzpress Prague, 2010.

S. T. Rosenbloom, J. C. Denny, H. Xu, N. Lorenzi, W. W. Stead, and K. B. Johnson. Data from clinical notes: a perspective on the tension between structure and flexible documentation. *Journal of the American Medical Informatics Association*, 18(2): 181–186, 01 2011. ISSN 1067-5027. doi: 10.1136/jamia.2010.007237. URL https://doi.org/10.1136/jamia.2010.007237.

G. Ruggieri. Una nuova malatia dell'olivo. *L'Italia Agricola*, 83:369–372, 1946.

A. Safonova, Y. Hamad, A. Alekhina, and D. Kaplun. Detection of norway spruce trees (picea abies) infested by bark beetle in UAV images using YOLOs architectures. *IEEE Access*, 10:10384–10392, 2022. doi: 10.1109/access.2022.3144433.

M. Sallam. ChatGPT utility in healthcare education, research, and practice: Systematic review on the promising perspectives and valid concerns. *Healthcare*, 11(6): 887, Mar. 2023. doi: 10.3390/healthcare11060887. URL https://doi.org/10.3390/healthcare11060887.

G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

S. Sanei, S. Okhoavat, G. A. Hedjaroude, H. Saremi, and M. Javan-Nikkhah. Olive verticillium wilt or dieback of olive in iran. *Communications in Agricultural and Applied Biological Sciences*, 69(4):433–442, 2004.

I. H. Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3), Mar. 2021. doi: 10.1007/s42979-021-00592-x. URL https://doi.org/10.1007/s42979-021-00592-x.

C. Saydam and M. Copcu. Verticillium wilt of olives in turkey. *Journal of Turkish Phytopathology*, 1(2):45–49, 1972.

G. Schröder, M. Thiele, and W. Lehner. Setting goals and choosing metrics for recommender system evaluations. In *UCERSTI2 workshop at the 5th ACM conference on recommender systems, Chicago, USA*, volume 23, page 53, 2011.

Sehouli, J. Könsgen, D. Mustea, A. Oskay-Özcelik, G. Katsares, I. Weidemann, H. Lichtenegger, and W. , , IMO" - intraoperatives mapping des ovarialkarzinoms. *Zentralblatt für Gynäkologie*, 125(3/4):129–135, 2003. doi: 10.1055/s-2003-41864. URL https://doi.org/10.1055/s-2003-41864.

H. Y. Seol, S. Sohn, H. Liu, C.-I. Wi, E. Ryu, M. A. Park, and Y. J. Juhn. Early identification of childhood asthma: The role of informatics in an era of electronic health records. *Front. Pediatr.*, 7:113, apr 2019.

V. Sergeeva, R. Spooner-Hart, et al. Olive diseases and disorders in australia. *4th Eur. Meeting of the IOBC working Group Integrated Protection of Olive Crops.*, 59, 2009.

A. Shah, K. Kothari, U. Thakkar, and S. Khara. User review classification and star rating prediction by sentimental analysis and machine learning classifiers. In *Information and Communication Technology for Sustainable Development: Proceedings of ICT4SD 2018*, pages 279–288. Springer, 2020.

B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi. Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *IEEE Journal of Biomedical and Health Informatics*, 22(5):1589–1604, Sept. 2018. doi: 10.1109/jbhi.2017.2767063. URL https://doi.org/10.1109/jbhi.2017.2767063.

H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

R. Shwartz-Ziv and A. Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

M. F. Siddiqui, G. Mujtaba, A. W. Reza, and L. Shuib. Multi-class disease classification in brain mris using a computer-aided diagnostic system. *Symmetry*, 9(3):37, 2017.

V. Škvára, J. Francå, M. Zorek, T. Pevnỳ, and V. Šmídl. Comparison of anomaly detectors: context matters. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2494–2507, 2021.

W. Snyder, H. Hansen, S. Wilhelm, et al. New hosts of verticillium alboatrum. *Plant Disease Reporter*, 34(1):26–27, 1950.

S. Somani, A. J. Russak, F. Richter, S. Zhao, A. Vaid, F. Chaudhry, J. K. De Freitas, N. Naik, R. Miotto, G. N. Nadkarni, et al. Deep learning and the electrocardiogram: review of the current state-of-the-art. *EP Europace*, 23(8):1179–1191, 2021.

I. Spasic and G. Nenadic. Clinical text data in machine learning: Systematic review. *JMIR Med Inform*, 8(3):e17984, Mar 2020. ISSN 2291-9694. doi: 10.2196/17984. URL http://www.ncbi.nlm.nih.gov/pubmed/32229465.

R. Sproat and N. Jaitly. Rnn approaches to text normalization: A challenge. *arXiv preprint arXiv:1611.00068*, 2016.

I. Steinwart. How to compare different loss functions and their risks. *Constructive Approximation*, 26:225–287, 2007.

E. W. Steyerberg, T. van der Ploeg, and B. Van Calster. Risk prediction with machine learning and regression methods. *Biometrical Journal*, 56(4):601–606, 2014.

M. Stone. Cross-validation: A review. *Statistics: A Journal of Theoretical and Applied Statistics*, 9(1):127–139, 1978.

Z. Sun, M. Ibrayim, and A. Hamdulla. Detection of pine wilt nematode from drone images using uav. *Sensors*, 22(13), 2022. ISSN 1424-8220. doi: 10.3390/s22134704.

I. Tabian, H. Fu, and Z. Sharif Khodaei. A convolutional neural network for impact detection and characterization of complex composite structures. *Sensors*, 19(22), 2019. ISSN 1424-8220. doi: 10.3390/s19224933. URL https://www.mdpi.com/1424-8220/19/22/4933.

A. A. Taha, A. Hanbury, and O. A. J. del Toro. A formal method for selecting evaluation metrics for image segmentation. In *2014 IEEE international conference on image processing (ICIP)*, pages 932–936. IEEE, 2014.

A. I. Taloba, A. El-Aziz, M. Rasha, H. M. Alshanbari, A.-A. H. El-Bagoury, et al. Estimation and prediction of hospitalization and medical care costs using regression in machine learning. *Journal of Healthcare Engineering*, 2022, 2022.

J. Tan, J. Yang, S. Wu, G. Chen, and J. Zhao. A critical look at the current train/test split in machine learning. *arXiv preprint arXiv:2106.04525*, 2021.

H. Tang and J. H. K. Ng. Googling for a diagnosis—use of google as a diagnostic aid: internet based study. *BMJ*, 333(7579):1143–1145, Nov. 2006. doi: 10.1136/bmj.39003.640567.ae. URL https://doi.org/10.1136/bmj.39003.640567.ae.

Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser. Svms modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):281–288, 2008.

K. Taunk, S. De, S. Verma, and A. Swetapadma. A brief review of nearest neighbor algorithm for learning and classification. In *2019 international conference on intelligent computing and control systems (ICCS)*, pages 1255–1260. IEEE, 2019.

P. Teterwak, A. Sarna, D. Krishnan, A. Maschinot, D. Belanger, C. Liu, and W. T. Freeman. Boundless: Generative adversarial networks for image extension. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10521–10530, 2019.

T. E. Thanassoulopoulos CC, Biris DA. Survey of verticillium wilt of olive trees in greece. *plant dis. reporter*, 63(11):936–940, 1979.

T. N. Theis and H.-S. P. Wong. The end of moore's law: A new beginning for information technology. *Computing in Science & Engineering*, 19(2):41–50, 2017. doi: 10.1109/MCSE.2017.29.

H. Tian, X. Fang, Y. Lan, C. Ma, H. Huang, X. Lu, D. Zhao, H. Liu, and Y. Zhang. Extraction of citrus trees from uav remote sensing imagery using yolov5s and coordinate transformation. *Remote Sensing*, 14(17), 2022. ISSN 2072-4292. doi: 10.3390/rs14174208.

C. Trapero, E. Alcántara, J. Jiménez, M. C. Amaro-Ventura, J. Romero, B. Koopmann, P. Karlovsky, A. von Tiedemann, M. Pérez-Rodríguez, and F. J. López-Escudero. Starch hydrolysis and vessel occlusion related to wilt symptoms in olive stems of susceptible cultivars infected by verticillium dahliae. *Frontiers in Plant Science*, 9, Jan. 2018. doi: 10.3389/fpls.2018.00072.

C.-F. Tsai and Y.-H. Lu. Customer churn prediction by hybrid neural networks. *Expert Systems with Applications*, 36(10):12547–12553, 2009.

A. M. Turing. *Computing Machinery and Intelligence*, pages 23–65. Springer Netherlands, Dordrecht, 2009. ISBN 978-1-4020-6710-5. doi: 10.1007/978-1-4020-6710-5_3. URL https://doi.org/10.1007/978-1-4020-6710-5_3.

D. Ulmer, L. Meijerink, and G. Cinà. Trust issues: Uncertainty estimation does not enable reliable ood detection on medical tabular data. In *Machine Learning for Health*, pages 341–354. PMLR, 2020.

S. Uppal, A. Al-Niaimi, L. W. Rice, S. L. Rose, D. M. Kushner, R. J. Spencer, and E. Hartenbach. Preoperative hypoalbuminemia is an independent predictor of poor perioperative outcomes in women undergoing open surgery for gynecologic malignancies. *Gynecologic Oncology*, 131(2):416–422, 2013. ISSN 0090-8258. doi: https://doi.org/10.1016/j.ygyno.2013.08.011. URL https://www.sciencedirect.com/science/article/pii/S0090825813010962.

S. Vani and T. M. Rao. An experimental approach towards the performance assessment of various optimizers on convolutional neural network. In *2019 3rd international conference on trends in electronics and informatics (ICOEI)*, pages 331–336. IEEE, 2019.

A. Vasudevan, D. A. Kumar, and N. S. Bhuvaneswari. Precision farming using unmanned aerial and ground vehicles. In *2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*, pages 146–150, 2016. doi: 10.1109/TIAR.2016.7801229.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.

O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663, 2016.

A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, et al. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

Ž. Vujović et al. Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications*, 12(6):599–606, 2021.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018. doi: 10.18653/v1/w18-5446.

J. Wang and Y. Dong. Measurement of text similarity: a survey. *Information*, 11(9): 421, 2020.

Q. Wang, Y. Ma, K. Zhao, and Y. Tian. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, pages 1–26, 2020a.

S. Wang, J. Liu, G. Yu, X. Liu, S. Zhou, E. Zhu, Y. Yang, J. Yin, and W. Yang. Multiview deep anomaly detection: A systematic exploration. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3), jun 2020b. ISSN 0360-0300. doi: 10.1145/3386252. URL https://doi.org/10.1145/3386252.

Y.-X. Wang, D. Ramanan, and M. Hebert. Learning to model the tail. *Advances in neural information processing systems*, 30, 2017.

Z. Wang, K. Jiang, P. Yi, Z. Han, and Z. He. Ultra-dense gan for satellite imagery super-resolution. *Neurocomputing*, 398:328–337, 2020c.

J. Wehrmann and R. C. Barros. Movie genre classification: A multi-label approach based on convolutions through time. *Applied Soft Computing*, 61:973–982, 2017.

Q. Wei, Y. Ren, R. Hou, B. Shi, J. Y. Lo, and L. Carin. Anomaly detection for medical images based on a one-class classification. In *Medical Imaging 2018: Computer-Aided Diagnosis*, volume 10575, pages 375–380. SPIE, 2018.

K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big Data*, 3(1), May 2016. doi: 10.1186/s40537-016-0043-6. URL https://doi.org/10.1186/s40537-016-0043-6.

H. Wibowo, I. Sitanggang, M. Mushthofa, and H. Adrianto. Large-scale oil palm trees detection from high-resolution remote sensing images using deep learning. *Big Data and Cognitive Computing*, 6:89, 08 2022. doi: 10.3390/bdcc6030089.

M. Wiering and M. van Otterlo, editors. *Reinforcement Learning*. Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-27645-3. URL https://doi.org/10.1007/978-3-642-27645-3.

Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, 2019. doi: 10.1109/TPAMI.2018.2857768.

Y.-H. Yang, Y.-C. Lin, Y.-F. Su, and H. H. Chen. A regression approach to music emotion recognition. *IEEE Transactions on audio, speech, and language processing*, 16 (2):448–457, 2008.

C. You, G. Li, Y. Zhang, X. Zhang, H. Shan, M. Li, S. Ju, Z. Zhao, Z. Zhang, W. Cong, et al. Ct super-resolution gan constrained by the identical, residual, and cycle learning ensemble (gan-circle). *IEEE transactions on medical imaging*, 39(1):188–203, 2019.

G. D. Zachos. La verticilliose de l'olivier en greece. *Benaki Phytopathological Institute*, 5:105–107, 1963.

M. Zamani Joharestani, C. Cao, X. Ni, B. Bashir, and S. Talebiesfandarani. Pm2. 5 prediction based on random forest, xgboost, and deep learning using multisource remote sensing data. *Atmosphere*, 10(7):373, 2019.

P. Zheng, S. Yuan, X. Wu, J. Li, and A. Lu. One-class adversarial nets for fraud detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1286–1293, 2019.

Z. Zheng, Y. Cai, and Y. Li. Oversampling method for imbalanced classification. *Computing and Informatics*, 34(5):1017–1037, 2015.

Y. Zhu, A. Mahale, K. Peters, L. Mathew, F. Giuste, B. Anderson, and M. D. Wang. Using natural language processing on free-text clinical notes to identify patients with long-term covid effects. In *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, number 46 in BCB '22, page 9, New York, NY, USA, 2022a. Association for Computing Machinery. ISBN 9781450393867. doi: 10.1145/3535508.3545555. URL https://doi.org/10.1145/3535508.3545555.

Y. Zhu, J. Zhou, Y. Yang, L. Liu, F. Liu, and W. Kong. Rapid target detection of fruit trees using uav imaging and improved light yolov4 algorithm. *Remote Sensing*, 14 (17), 2022b. ISSN 2072-4292. doi: 10.3390/rs14174324.

Y. Zhu, M. Wang, X. Yin, J. Zhang, E. Meijering, and J. Hu. Deep learning in diverse intelligent sensor based systems. *Sensors*, 23(1), 2023. ISSN 1424-8220. doi: 10.3390/s23010062. URL https://www.mdpi.com/1424-8220/23/1/62.

F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

J. Zoto, M. A. Musci, A. Khaliq, M. Chiaberge, and I. Aicardi. Automatic path planning for unmanned ground vehicle using uav imagery. In K. Berns and D. Görges, editors, *Advances in Service and Industrial Robotics*, pages 223–230, Cham, 2020. Springer International Publishing. ISBN 978-3-030-19648-6.

M. Zou, W.-G. Jiang, Q.-H. Qin, Y.-C. Liu, and M.-L. Li. Optimized xgboost model with small dataset for predicting relative density of ti-6al-4v parts manufactured by selective laser melting. *Materials*, 15(15), 2022. ISSN 1996-1944. doi: 10.3390/ma15155298. URL https://www.mdpi.com/1996-1944/15/15/5298.

T. Özer, C. Akdoğan, E. Cengız, M. M. Kelek, K. Yildirim, Y. Oğuz, and H. Akkoç. Cherry tree detection with deep learning. In *2022 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 1–4, 2022. doi: 10.1109/ASYU56188.2022.9925332.

Appendices

Table 1: Cohort statistics

Variable	Overall (n=555)	Training Set (n=444)	Testing Set (n=111)	pvalue (training)	Zero residual (n=363)	Non-zero residual (n=192)	pvalue (R0 vs non R0)
Grade_cat	502 (90.45)	403 (90.77)	99 (89.19)	0.745	332 (91.46)	170 (88.54)	0.337
IDS/PDS	385 (69.37)	306 (68.92)	79 (71.17)	0.730	247 (68.04)	138 (71.88)	0.404
Surgical Complexity Score (SCS)	3.77 ± 2.07	3.8 ± 2.04	3.66 ± 2.2	0.544	4.13 ± 2.27	3.08 ± 1.4	<0.001
Size Largest Bulk of Disease (cm)	8.83 ± 5.57	8.71 ± 5.62	9.3 ± 5.37	0.306	8.31 ± 5.67	9.79 ± 5.25	0.002
Age	63.57 ± 11.23	63.88 ± 10.96	62.32 ± 12.25	0.221	62.45 ± 11.65	65.69 ± 10.1	0.001
Pre Surgery CA125	412.99 ± 1180.36	406.44 ± 1229.0	439.45 ± 963.76	0.762	399.52 ± 1297.67	438.6 ± 919.64	0.682
IntraOperative Mapping of ovarian cancer (IMO)	4.9 ± 1.95	4.83 ± 1.91	5.2 ± 2.11	0.096	4.36 ± 1.88	5.93 ± 1.65	<0.001
Time procedure (min)	168.82 ± 75.13	169.5 ± 72.33	166.08 ± 85.71	0.699	172.77 ± 80.26	161.35 ± 63.84	0.068
EBL	521.84 ± 386.84	523.06 ± 400.16	516.95 ± 329.82	0.868	512.1 ± 417.74	540.26 ± 320.62	0.377
Pre Treatment CA125	1525.33 ± 2719.94	1421.98 ± 2573.98	1938.7 ± 3218.95	0.118	1499.46 ± 2911.22	1574.24 ± 2322.0	0.742
PCI	7.3 ± 4.39	7.16 ± 4.26	7.89 ± 4.85	0.145	6.52 ± 4.3	8.79 ± 4.17	<0.001
ANAFI	5.02 ± 5.45	4.72 ± 5.27	6.23 ± 5.98	0.016	2.85 ± 4.4	9.13 ± 4.86	<0.001

Table 2: Top 10 n-grams with the highest TF-IDF difference scores per case outcome. The two top-level header columns indicate the case according to which the score difference was sorted. R0 n-grams had high positive score difference while non R0 n-grams had high negative score difference

R0				**non R0**			
word	**TF-IDF score for word in R0 document**	**TF-IDF score for word in non-R0 document**	**TF-IDF Score difference**	**word**	**TF-IDF score for word in R0 document**	**TF-IDF score for word in non-R0 document**	**TF-IDF Score difference**
normal	0.261	0.154	0.106	miliary	0.026	0.170	-0.143
liver	0.151	0.068	0.082	miliary disease	0.018	0.127	-0.109
diaphragm	0.141	0.071	0.070	disease	0.269	0.346	-0.077
left	0.213	0.164	0.048	tumour	0.108	0.178	-0.069
uterus	0.213	0.168	0.044	adherent	0.170	0.237	-0.066
adhesion	0.108	0.064	0.044	colon	0.058	0.116	-0.058
tube	0.094	0.052	0.042	involving	0.012	0.059	-0.046
diaphragm normal	0.046	0.006	0.039	omental cake	0.031	0.073	-0.042
sub diaphragm normal	0.045	0.006	0.038	cake	0.032	0.073	-0.040
liver sub	0.046	0.007	0.038	sigmoid	0.090	0.131	-0.040

Acronyms

AI Artificial Intelligence.

ANAFI ANAtomic FIngerprints.

ANN Artificial Neural Network.

AUPRC Area under the Precision-Recall curve.

AUROC Area under the Receiver Operating Characteristic curve.

BERT Bidirectional Encoder Representations from Transformers.

CART Classification and Regression Tree.

CNN Convolutional Neural Network.

CPU Central Processing Unit.

DNN Deep Neural Network.

EBL Estimated Blood Loss.

EHR Electronic Health Records.

EOC Epithelial Ovarian Cancer.

FSL Few-Shot Learning.

GB Gigabyte.

GCS Ground Control Station.

GHz Gigahertz.

GloVe Global Vectors for Word Representation.

IMO IntraOperative Mapping of ovarian cancer.

IoU Intersection over Union.

LSTM Long Short Term Memory.

mAP Mean Average Precision.

MDT MultiDisciplinary Team.

ML Machine Learning.

MLM Masked Language Modeling.

NER Named Entity Recognition.

NLP Natural Language Processing.

NMS Non-Maximum Suppression.

NSP Next Sentence Prediction.

OSL One-Shot Learning.

PCI Peritoneal Carcinomatosis Index.

RAM Random-Access Memory.

RGB Red, Green, Blue.

RL Reinforcement Learning.

RNN Recurrent Neural Network.

RoBERTa Robustly Optimized BERT Approach.

SCS Surgical Complexity Score.

SHAP SHAPley Additive Explanations.

SML Supervised Machine Learning.

SOTA State of the Art.

TF-IDF Term Frequency – Inverse Document Frequency.

TL Transfer Learning.

UAV Unmanned Aerial Vehicle.

UML Unsupervised Machine Learning.

XGBoost Extreme Gradient Boosting Algorithm.

YOLO You Only Look Once.

ZSL Zero-Shot Learning.