Processing Image Data from Unstructured Environments

by

Spiridon Kasapis

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Naval Architecture and Marine Engineering)
in The University of Michigan
2023

Doctoral Committee:

Professor Nickolas Vlahopoulos
Assistant Professor Yang Chen
Assistant Professor Maani Ghaffari Jadidi
Professor Matthew Kai Johnson-Roberson
Dr. Jonathon Smereka

Spiridon Kasapis

skasapis@umich.edu

ORCID iD: 0000-0002-0972-8642

Similarly to Dr. Stephen Jay Gould, I too am, somehow, at the time of writing this thesis, less interested in how machines can be trained to perform as intelligent beings but rather the near certainty that humans of extreme intellectual potential, working in cotton fields and sweatshops, were never given the chance to develop their talent. This work is dedicated to them.

ACKNOWLEDGEMENTS

I wish to start this dissertation in a rather simple way: by thanking my parents, sister and uncle. It is quite self-explanatory how without their support I would never have been where I am.

This thesis, although written by me, is a product of exceptional educators who I was lucky to cross paths with. I am indebted to my high school teachers, who laid down the fertile soil of literature, science and art in which I grew as a scientist.

"*Love danger. What is most difficult? That is what I want! Which road should you take? The most craggy ascent! It is the one I also take: follow me!*" wrote Nikos Kazantzakis in *Ascesis: Salvatores dei*. Dr. Cowlagi, Dr. Gatsonis and Dr. Demetriou through their teaching and mentoring during my undergraduate years inspired me to pursue this craggy ascent of academia. The least I could do is extend my gratitude to them.

The University of Michigan in Ann Arbor has been the place where I grew up the most, both as a person and as a scientist. I need to thank all the people I was fortunate enough to work and live with during this time. It goes without saying, that my PhD advisor, Dr. Nickolas Vlahopoulos, has been a great mentor and has guided me through this entire process. Academic giants like Dr. Tamas Gombosi and truly inspirational women such as Dr. Chen and Dr. Barbara Thompson have been role models and convinced me to not settle for mediocrity. To them all I owe my career in research.

Lastly, I want to thank all those people who, throughout the years, I was fortunate to call friends. To all those who need not be mentioned, this is for you- my journey in education would be unbearable without you.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

LIST OF ABBREVIATIONS

Abbreviations	Meaning	Page
ADNN	Adaptive Deconvolutional Neural Networks	3
AIC	Aikaike Information Criterion	8
ATR	Automatic Target Recognition	18
AUC	Area Under the Curve	30
BIC	Bayesian Information Criterion	8
CNN	Convolutional Neural Network	4
CPU	Central Processing Unit	53
DB	Davies - Bouldin index	9
E-VRC	Extended Variance Ratio Criterion	6
FN	False Negative	31
FP	False Positive	31
FPR	False Positive Rate	31
FR	False Relevant	31
FRR	False Relevant Rate	31
GPU	Graphics Processing Unit	53
GHz	Gigahertz	53
Gb	Gigabytes	53
IDLS	Instance-Discrimination Low-Shot	6
IR	Infra-Red	18
ISTA	Iterative Shrinkage-Thresholding Algorithm	26
LFH	Low-frequency Fourier Harmonics	72
NN	Neural Network	74
N-PID	Non-Parametric Instance Discrimination	3
OSLS	Open-Set Low-Shot	6
PCA	Principal Component Analysis	71
PEELER	oPen sEt mEta LEaRning	13
ROC	Receiver Operating Characteristic	13

Abbreviations	Meaning	Page
ResNet	Residual Network	14
SVD	Singular Value Decomposition	23
SW	Silhouette Width	9
SVM	Support Vector Machine	26
SGM	Score-based Generative Modeling	35
SwAV	Swapping Assignment between Views	49
SimCLR	Simple framework for Contrastive Learning of visual Representations	22
TL	Transfer Learning	35
TN	True Negative	31
TP	True Positive	31
TR	True Relevant	31
TPR	True Positive Ratio	31
TRR	True Relevant Ratio	31
VRC	Variance Ratio Criterion	6

ABSTRACT

Advanced mobility research centers capture large amounts of data from ground vehicle systems during development and experimentation in both manned and autonomous operations. This exponential growth of digital image data has given rise to the need of understanding the content of image datasets by clustering and classifying them without the use of manual labor. Currently, there is a lack of tools which -through processing raw data- can provide a semantic understanding of an environment or dataset and can be used in place of a human to provide context to situations that threaten the uninterrupted operation of an autonomous vehicle.

In search, exploration, and reconnaissance tasks performed with autonomous ground vehicles, an image classification capability is needed for specifically identifying targeted objects (relevant classes) and at the same time recognize when a candidate image does not belong to anyone of the relevant classes (irrelevant images). An open-set low-shot (OSLS) classifier was developed for addressing this need. During its training, it uses a modest number (less than 40) of labeled images for each relevant class, and unlabeled irrelevant images that are randomly selected at each epoch of the training process. The new OSLS classifier is capable of identifying images from the relevant classes, determining when a candidate image is irrelevant, and it can further recognize categories of irrelevant images that were not included in the training (unseen).

The OSLS was integrated with an unsupervised learning feature extraction

framework based on the instance discrimination method for creating an instance discrimination low shot (IDLS) module. The IDLS can identify targeted objects while at the same time recognize when candidate images do not belong to any one of the target classes, both in a very data-inexpensive way. The IDLS is dynamic, adapts to new environments during operation and is resilient to adversaries. The OSLS and IDLS algorithms were compared to a variety of alternative supervised methods showing comparable and often times better results in performing classification tasks, while requiring very few labeled images for training (i.e. less that 0.3% of labeled data compared to a supervised CNN for comparable levels of accuracy).

This work also developed a soft-labeling capability for grouping collected images into categories using a new formulation that is based on an extended variance ratio criterion (E-VRC). The E-VRC comprises an unsupervised clustering capability since it does not require any initializations or prior knowledge about how many clusters will be encountered. As it is done with the previous two modules (OSLS and IDLS), the E-VRC too is being tested on several different datasets, demonstrating that it is useful not only for autonomous exploration and reconnaissance operations but also for the efficient content management and retrieval tasks. Additionally, the E-VRC algorithm developed by this research was compared to other available unsupervised clustering methods yielding superior results.

CHAPTER I

Introduction

1.1 Motivation

During development and experimentation in both manned and autonomous operations, advanced mobility research centers capture -through sensors mounted on ground vehicle systems- large amounts of data in the form of images, GPS coordinates, gear changes, user interventions, RPMs, tractive effort and others. Before entering operational situations that threaten their uninterrupted operation, autonomous vehicles need an understanding of the environment which can be used in place of a human to provide context to situations where a user intervention is needed. Currently, there is a lack of such tools for processing unlabeled data in a semantic manner. This missing capability would allow engineers to create a clear understanding of the information collected, in the form of meta-data, from uncommon and unique events captured during experiments. Engineers could use this information for planning and implementing control strategies for responding and navigating similar conditions in an operational setting, with the ones that caused failure during experimentation. Such capability could be used to significantly enhance the ability of autonomous vehicles to perform their mission without interruption and failure in both civilian and military applications by helping with the efficient image content understanding,

management and retrieval tasks.

Civilian autonomous vehicle programs have spread through the continents, with pilot projects taking places in numerous cities around the world, one of them being Ann Arbor too. In the not too distant future, the US Army expects to field autonomous vehicles such as the Squad Multipurpose Equipment Transport for carrying equipment alongside soldiers in an urban terrain. Therefore, on one hand, the safety of passengers riding civilian autonomous vehicles in off-road situations will depend on the successful understanding of the unmapped operation environment, while on the other hand the soldiers' safety along with a military mission success will heavily rely on the uninterrupted autonomous vehicle performance. Therefore, Michigan College of Engineering [1] research in collaboration with the Automotive Research Center[2] (ARC) is developing capabilities for increasing low-shot classification accuracy and for soft labeling (i.e., clustering in groups with similar statistical characteristics) images and video frames that are collected but not currently labeled.

More specifically, a capability to increase accuracy when a small number of labeled images are used for training a classifier is of interest in reconnaissance operations of autonomous, off-road vehicles. Additionally, a capability for unsupervised soft labeling (i.e. clustering in groups with similar statistical characteristics without knowing ahead of time the number of clusters) images and video frames that are collected but not currently labeled is presented. The autonomous vehicles are expected to collect information about specific targeted objects, ignore temporarily the presence of any other unrelated objects in order to achieve a task, but then understand the surrounding environment using these unrelated objects. Automated annotation would allow for the cross-correlation of the image features with other relevant data to identify significant events and

[1] https://www.engin.umich.edu/
[2] https://arc.engin.umich.edu/

plan for the appropriate action through the control algorithms embedded in the vehicle. For example, a lane change maneuver can be more than just a path plan update or obstacle avoidance, but can infer an indirect effect of a manned-unmanned teaming algorithm influencing the path planner when taken into context with what the vehicle sees. This way the algorithm helps understand what received information from a teammate was "useful".

This doctoral dissertation research aims to enhance data analysis capabilities of autonomous ground vehicles that perform in unstructured environments. To achieve this, an open-set low-shot (OSLS) classification capability is developed and demonstrated (*Kasapis et al.*, 2020). The OSLS classifier includes some unique characteristics such as a target matrix which incorporates values for unlabeled images, a differentiable exponential loss function which converges quickly to the local minima and a euclidean normalization. Additionally, to maximize the cumulative accuracy (labeled and unlabeled images), the classifier uses the Receiver Operating Characteristic (ROC) Method for the ROC Threshold Criterion which also allows the user to discriminate for or against labeled images.

Additionally, an unsupervised feature extractor based on a simple framework for constructive learning of visual representations (*Chen et al.*, 2020) is being integrated with the new open-set low-shot classifier creating the instance discrimination low-show (IDLS) classifier (*Kasapis et al.*, 2022). The main advantages of IDLS module are that the feature extractor is trained on data related to the specific operation scenario, even though no labels are being used, the module is dynamic (i.e. training can take place during operation and therefore adapt to new unstructured environments and improve over time) and lastly adversaries do not have access to the data that a transfer earning approach with a pre-trained feature extractor relies on. It is important to also note that

the IDLS module uses only 0.27% of the image annotations (2,000 instead of 727,913) that an equivalent CNN uses.

Finally, an automated annotation clustering schema that does not require a priori knowledge of the number of clusters is developed. This new capability extends the Variance Ratio Criterion (VRC) method to make it applicable for computer vision applications by normalizing the image features using the Euclidean Normalization and by adding an exponent term to the VRC equation. By normalizing the image features, we help the Extended Variance Ratio Criterion (E-VRC) method solve multi-dimensional problems as the direction of the feature vectors are necessary compared to the magnitude (dimension) which is not. Lastly, the exponent p restores the VRC equation balance by diminishing the effect of a large K value.

This three-stage apparatus aims to understand the environment based on unlabeled data collected from operating off-road ground vehicles. The main contributions of this work are the OSLS, the IDLS and the E-VRC algorithm. The OSLS algorithm was developed which can identify targeted objects and recognize unrelated images, using a minimal amount of labeled data and a lot of unlabeled data. The IDLS (i.e. the OSLS connected to an instance discrimination method) uses raw data and an unsupervised training feature extractor to create a module which is dynamic, adapts to new environments during operation and is resilient to adversaries. Finally, the E-VRC clustering approach is completely unsupervised and is not dependent to any initializations or prior knowledge of the number of clusters, making it useful for a variety of image processing tasks such as automatic image search, labeling and retrieval.

1.2 Background

Computer vision is the field of machine learning that tries to understand and imitate tasks that the human visual system can do, by processing and analyzing digital imagery (*Shapiro et al.*, 2001). In the recent years, the exponential growth of such data has given rise to the need of efficiently managing and retrieving images based on their content. Evidently, because of the volume and diversity of the digital datasets, methods that can achieve such tasks without human supervision are required.

An image recognition capability consists of two distinct processes. The first is the training of a multi-layer and complex in structure feature extractor. It processes the input data (i.e., images, time domain signals, etc.) for filtering and reducing the dimension of the original entities into a smaller set of descriptors. The descriptors -referred to as feature vectors- are then used by the second, decision-making part for generating determinations about classification, clustering, etc. The quality of the feature extractor is associated with the ability to meaningfully reduce the size of the information and produce robust descriptors. The training of the feature extractor depends on the expected utilization of the descriptors that it generates. Therefore, it is often done simultaneously with the training of the second, decision making process.

A long-standing problem in image recognition is learning effective visual representations without human effort. The majority of machine learning algorithms require a large amount of labeled data for training both processes simultaneously by making them recognize correctly what each annotated training entity is (*Russakovsky et al.*, 2015). Deep learning algorithms that learn patterns by processing data which is not annotated comprise unsupervised learning methods (*Sanakoyeu et al.*, 2018; *Bansal*, 2020). Using such methods (*Caron et al.*, 2018; *Wu et al.*, 2018), the computer vision community has progressively closed

the performance gap with supervised algorithms.

The reason for pursuing the unsupervised training of the feature extractor originates from ground combat vehicles operating in battlefields that have no resemblance to the type of images used for training readily available feature extractors. Due to the unavailability of army-specific labeled data, the images used in the original training of the feature extractor are different to the ones we aim to recognise, leading to a deterioration in the quality of the descriptors created for the latter (*Zhang et al.*, 2019). Furthermore, for defence applications (e.g. reconnaissance operations) it is highly desirable to avoid using publicly available feature extractors since such knowledge can be used by adversaries for deceiving the machine learning application (*Xie and Li*, 2019).

Unsupervised approaches have three additional benefits. Annotations are not needed for training the feature extractor thereby eliminating the burdensome task of data labeling (*Zhang et al.*, 2008). Furthermore, the usnupervised methods are agnostic to the neural network architecture, which means that they can be implemented in any network design (*Chen et al.*, 2020). Finally, in a supervised network, when testing images that are distorted compared to the labeled images used for training, due to the change in environmental conditions, the performance of the network deteriorates (*Dodge and Karam*, 2016). The latter can be re-trained without manual effort using an unsupervised approach and a new set of unlabeled images reflecting the environmental conditions of interest. In this manner, more meaningful feature vectors are produced.

Unsupervised clustering is another interdisciplinary area of data science that studies the groups of methods which can achieve image content management tasks (*Berkhin*, 2006). The boundary where a clustering method has to be characterized as supervised or unsupervised might not be clear, but one can claim that algorithms such as K-means are not ideal for image data separation of

highly variable -in terms of content- data. This is because supervised algorithms like K-means require information about the datasets they cluster that, until lately, only humans were able to provide.

Such information shows up in clustering applications in the form of parameter initializations or the need to prescribe the number of clusters a priori. In order to overcome the latter shortcoming, a number of validity indices, independent of the clustering algorithms applied to, have been used throughout the years to find the optimal number of clusters the data should be separated in. Some of them are the Variance Ratio Criterion *Caliński and Harabasz* (1974), Bayesian Information Criterion (*Kass and Raftery*, 1995), Akaike Information Criterion (*Bozdogan*, 1987), Dunn's index (*Dunn*, 1973), Davies-Bouldin index (*Davies and Bouldin*, 1979), Silhouette Width (*Rousseeuw*, 1987), Gap statistic (*Tibshirani et al.*, 2001) and other.

Conventional methods that aim to train a decision making processes have been primarily focused on the scenario where large amounts of training data are available. The contributions of this dissertation are two-fold as we push the limits of unsupervised learning in classification and clustering. To make the classification process label-inexpensive while at the same time constructing an approach which could be applied to recognize a broader set of data, we first present a method which couples an unsupervised feature extractor, based on an instance discrimination method (*Chen et al.*, 2020), with an OSLS Classifier (*Kasapis et al.*, 2020). By using a limited amount of labeled data for the type of images which need to be specifically classified (relevant classes) along with unlabeled data for all other types of images, this new algorithm is able to increase identification accuracy in both data categories and further recognize types of images that were not included in the training as irrelevant (open-set classification capability (*Jain et al.*, 2014; *Scheirer et al.*, 2012, 2014)).

Second, we propose an Extension to their Variance Ratio Criterion (E-VRC) method, which helps K-means with clustering image data of high content variance, without the need to input any information, like the number of true image classes. Comparisons with other unsupervised methods are being performed showing the superiority of the proposed method. Lastly, we show that E-VRC is a robust method that is not dependent on initializations, does not care about the data dimensionality nor the content randomness, and therefore is a great tool for efficiently estimating the number of clusters and performing the clustering of image data. The contributions of this work to the field of autonomy through improvements in unsupervised clustering and classification will be thoroughly discussed in the next sections.

CHAPTER II

Open-Set Low-Shot Classifier

2.1 Related Work

Extensive research in the field of machine learning has been progressively improving the performance of object recognition algorithms which achieve impressive results on a variety of multi-class classification tasks (*He et al.*, 2016a, 2017; *Krizhevsky et al.*, 2012). Especially in search, exploration, and reconnaissance applications, object recognition methods have been concentrated on a closed-set setting where all testing samples belong to one of the classes that the classifier has been trained by *Pham and Polasek* (2014). The limited, finite number of classes that are the target of inspection need to be detected out of the infinite object classes that are encountered in an unconstrained environment.

Efforts have been made to endow Convolutional Neural Networks (CNNs) with the innate human brain capability to identify objects they are trained on while deliberately discarding objects of no interest. Lately, the introduction of open-set classification (*Jain et al.*, 2014; *Scheirer et al.*, 2012, 2014) has introduced an ability to correctly identify images as unknown test objects that do not belong to any known classes, as opposed to falsely classify them in one of the known classes (i.e., classes that the model has been trained on). More specifically, *Pernici et al.* (2018) and *Ge et al.* (2017) define open-set classification as the

problem of balancing the known space (specialization) and unknown open space (generalization) of the model. Examples such as out-of-distribution detection (*Hendrycks and Gimpel*, 2016) and realistic classification (*Lu et al.*, 2017) show the interest in the concept of open-set recognition (*Bendale and Boult*, 2016) while showing that CNNs can be trained to reject examples that have not been seen during training or are too hard to classify.

Recently, works on video object discovery (*Wang et al.*, 2014) go against the closed-set assumption that each image during inference belongs to one of the fixed number of relevant classes. In the work of *Wang et al.* (2014), the terminology of "relevant" and "irrelevant" is introduced and is used in this dissertation. In most real-life applications, this closed-set assumption is uncommon and ideal. Therefore, recently proposed methods (*Bendale and Boult*, 2016) are subject to an open-set condition where images not seen during training should be classified into irrelevant or unseen classes. Consequentially, in this Chapter, we introduce the splitting of testing samples in three categories: (a) relevant; labeled samples used during training, (b) irrelevant; unlabeled samples used during training and (c) irrelevant but also unseen; for categories of images that are not seen during training and should be identified as irrelevant.

Another challenge the visual recognition community faces is the absence of labeled examples. Especially in military applications having large labeled datasets is an unreal expectation as needs and mission tools used for search and reconnaissance evolve. An open-set recognizer will face limitations such as the absence of large amounts of training samples. Thus, an open-set recognition technique that simultaneously supports the few-shot setting is needed. Therefore, in this Chapter, we propose a low-shot solution to the problem of open-set recognition, which considers the classification layer of a CNN exclusively.

Specifically, we present an approach on significantly improving the perfor-

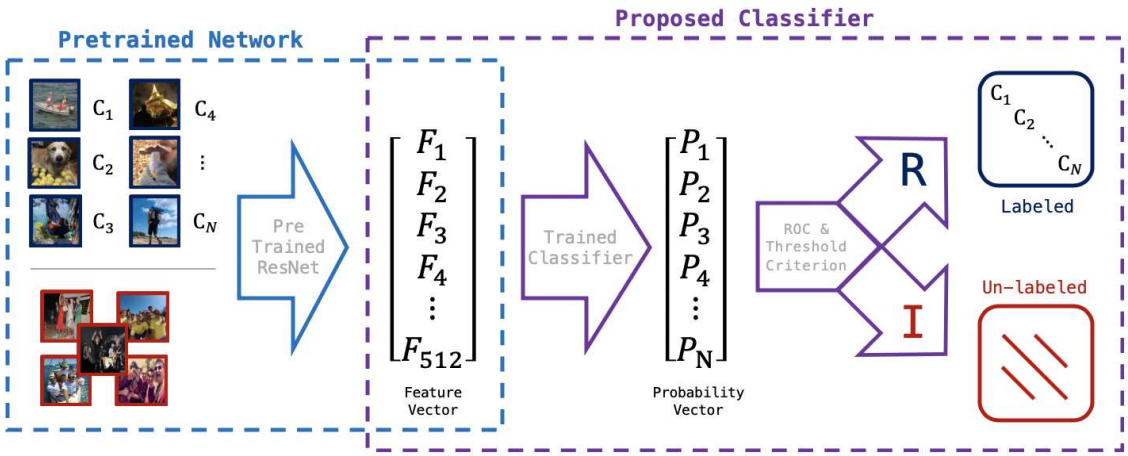

Figure 2.1: Schematic of the two parts of the OSLS network.

mance of a simple, time-efficient, one-layer classifier on recognizing labeled (relevant) images along with non-labeled (irrelevant and unseen as mentioned above) images. The ability to specifically recognize a number (of the order of 50) of relevant classes and also identify when an image does not belong to any of them in a label-inexpensive way is one of the main motivations for low-shot open-set (OSLS) recognition. Figure 2.1 is a visualization of the two-step process that comprises the open-set low-shot classifier algorithm. The OSLS classifier accepts as inputs from a pre-trained Network labeled Relevant images and unlabeled Irrelevant images. For each image, the proposed classifier produces class score vectors that get classified using a threshold criterion and Receiver Operating Characteristic (ROC), with accuracy much greater than already existing techniques, especially for the irrelevant dataset.

Efforts with similar goals have been concentrated on the training of the entire CNN. For example, the PEELER algorithm (*Liu et al.*, 2020) combines the random selection of a set of novel classes per episode, a loss that maximizes the posterior entropy for examples of those classes, and a new metric learning formulation in order to train the weights of a CNN in such a manner that it

can recognize images of a limited amount of classes (≤ 20) that are unseen during training. *Dhamija et al.* (2018) propose the introduction of two loss functions that are designed to maximize entropy for unknown inputs while increasing separation in deep feature space by modifying magnitudes of known and unknown samples. Although the work of Dhamija et al. introduces the concept of unknown sample recognition like we do, the number of recognizable classes is still very limited compared to the testing done in our method.

Both of the aforementioned algorithms train the entirety of the CNN, unlike the methods proposed by *Kozerawski and Turk* (2021), which can augment any few-shot learning method without requiring retraining in order to work in a few-shot multiclass open-set setting. Although not concerned with one-class classification, a similar approach is followed in our work too, where we utilize a pre-trained feature extractor (such as the ones publicly available by PyTorch[1]) and propose an independent open-set low-shot classification method which can augment any existing feature extractor.

To explore the open-set low-shot problem in a holistic, non-specific and easily applicable way, this work is concentrated only on the training of the classification matrix (matrix used to turn the feature vector to a probability vector in Figure 2.1), using the pre-trained ResNet feature extractors (*He et al.*, 2016a) discussed in Chapter 2.2. The image matrices are reduced to feature vectors (*Donahue et al.*; *Azizpour et al.*, 2015) which are then used in Chapter 2.4 to train the classifier with the help of the analytic derivative of our loss function and a unique, partially labeled, target matrix. In Chapter 2.5, the classification variability statistics and a Receiver Operating Characteristic (ROC) curve are used to calculate threshold scores for each relevant class. An approach that uses random selections of unlabeled irrelevant images during each epoch of the

[1]https://pytorch.org/vision/stable/models.html

classifier training is introduced. Testing datasets are used in Chapter 2.3 for determining the ability to classify all Relevant, Irrelevant and Unseen datasets effectively.

In summary, there are three main contributions of the OSLS algorithm, which set the base for the development of the algorithms discussed later in this dissertation. A novel, open-set low-shot (OSLS) classification method is developed, which can be added as the top layer to any pre-trained feature extractor (like the one that will be discussed in Chapter III) in order to create a CNN that can classify images in relevant classes and also determine if an image does not belong to any of the relevant classes. The OSLS Classifier yields improved classification performance compared to classifiers that either do not use unlabeled images during training or assume all unlabeled samples to belong in the same class. Lastly, The number of image classes the OSLS can classify is greater than the ones used in the open-set classification literature (*Ge et al.*, 2017; *Dhamija et al.*, 2018; *Liu et al.*, 2020), making it a great candidate for the scientific questions this thesis is trying to tackle.

2.2 Feature Extractor

Deep Residual Networks have been proven to be a very effective in mapping images to a meaningful feature space, especially when trained from large datasets (*Simonyan and Zisserman*, 2014). In this part of the dissertation we use ResNet18 and ResNet34 (*He et al.*, 2016b) to map the sample images to the vector space. Both architectures produce 512-long feature vectors, which compared to deeper network feature products, lead to a shorter algorithm running time. The different types of ResNets we used, although not significantly different, will be discussed in Chapter 2.3. The weights were trained using the ImageNet1k dataset (*Deng et al.*, 2009), which involves a large-scale ontology of

images. This simple yet powerful feature extractor was selected in order to create generic results that could be used in a variety of applications. The development of the feature extractor itself is out of the scope of the OSLS classifier development and will concern us on the next Chapter. Therefore, pre-trained feature extractors available by PyTorch are used. It is anticipated (and will be proven in Chapter III) that potential targeted changes on the entire net and the use of a more application specific datasets would increase the classification accuracy even more, as using filters trained on pictures with similar characteristics to the ones we are classifying would produce more robust feature representations.

Before providing the training images feature vectors to the OSLS classifier, we normalize them using the following Equation:

$$F_{norm} = \frac{F - F_{min}}{F_{max} - F_{min}}. \tag{2.1}$$

Where F is a 512-long vector feature map and F_{max}, F_{min} are its respective maximum and minimum values in vector form. This type of basic normalization constrains the F values from 0 to 1. We apply the normalization to prevent the exponential loss and its derivative in Equation 2.9 and Equation 2.11, respectively, from gaining extremely high values. Additionally, we demonstrate in Chapter 2.6 that this type of normalization significantly increases our method's classification accuracy compared to the more popular softmax normalization. It can be argued that the use of softmax normalization yields poor solutions for open-set recognition as it tends to over-fit on the training classes.

2.3 Datasets

The proposed OSLS classification method is used on a variety of training and testing examples, each using different sample arrangements. To explore the

capabilities of the proposed method in various settings, two different datasets are being used: the Caltech256 (*Griffin et al.*, 2007) and a custom Mixed dataset. Note that the same datasets described here are also used for training and testing the algorithms presented in Chapters III and IV, therefore Appendix B outlines within tables the contents of these datasets.

To explain with more clarity our method and results, we describe the way the Caltech256 dataset is split into two groups. Caltech256 is an open-source dataset which consists of 256 different image classes and has been recently used a lot as a benchmark for various machine learning applications (*Basha et al.*, 2021; *Ge and Yu*, 2017).

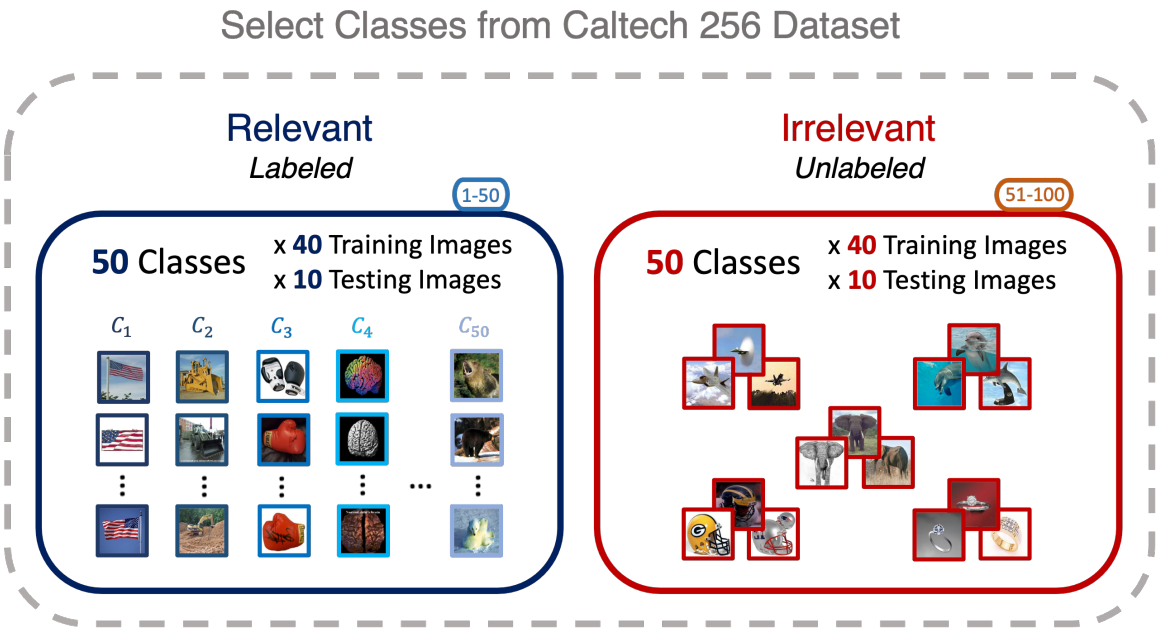

Figure 2.2: The two groups the select classes of Caltech256 are split in: the Relevant (Labeled) and the Irrelevant (Unlabeled).

Similar to the selection of ResNet as a feature extractor, a popular open-source dataset is used to make the case study and results as general and less task-specific as possible. As it is intended to produce work that is going to be used in the future for specific applications, to give a hint on how the method

can be geared towards recognition in unique environments, the eight infrared classes (Figure 2.3), which are available from the Military Sensing Information Analysis Center (SENSIAC) Automatic Target Recognition (ATR[2]) database (*Yu et al.*, 2015) are also used in a number of tests in this Chapter. The ATR database consists of eight classes, seven of them are civilian and combat vehicles and the last one is a human class.

The OSLS classifier is trained on both labeled and unlabeled pictures, therefore the main dataset consists of what we call Relevant and Irrelevant pictures. The relevant (blue) group is consisted of the first 50 classes of Caltech256 and the irrelevant (red) group contains images from the next 50 classes, as shown in Figure 2.2. During training, every class (labeled and unlabeled) contains 40 images, for a total of 2000 labeled and 2000 unlabeled images. The evaluation is performed using 10 images for each class, different than the ones used during training.

To explore the dependency between the classifier's visual recognition accuracy and the number of unlabeled images, two more versions of the Caltech256 dataset are created with an expanded number of Irrelevant images, one has 100 classes of unlabeled images (+50 Irrelevant) and the other has 200 classes of unlabeled images (+150 Irrelevant), both with the same number (40) of pictures per class.

Finally, in order to explore the behavior of our method on unique and very different environments from the ones present in the Caltech256 dataset, we created our own infrared (IR) combat vehicle image group by taking snapshots from the publicly available IR videos provided in the ATR database (examples displayed in Figure 2.3). The new data product is composed of the same amount of pictures as the one in Figure 2.2, with the exception that the first eight relevant

[2]https://dsiac.org/technical-inquiries/notable/infrared-imagery-of-tactical-vehicles-personnel/

Figure 2.3: The manually created infrared (IR) dataset using video snapshots from the ATR database.

image classes are infrared instead of Caltech 256 pictures. Although only eight infrared classes are available in the ATR dataset, the term Infrared Dataset is used to indicate that infrared classes are included within the 50 relevant classes. In Chapter 2.4 the way the dataset images described above are treated during the classifier training process is discussed.

It is important for autonomous off-road vehicles (military, reconnaissance, etc.) to recognise predefined objects of interest during operation and classify them as targets, while also understanding the operation environment and background scenery in a dynamic way as a distinct entity. To demonstrate the usefulness of the OSLS method on such applications we create a custom dataset. This dataset, which we refer to as Mixed, is used to train and test the OSLS classification method. The relevant group is composed of 50 select classes from the ImageNet (*Deng et al.*, 2009) dataset and includes pictures of vehicles, aircraft, humans and weapons. This group will serve as the target images that are expected to be recognized. On the other hand, the irrelevant group is composed of 50 classes from the MIT Places (*Zhou et al.*, 2017) dataset, which includes a variety of outdoor scenery pictures such as buildings and natural environment. The way the Mixed dataset is utilized for training and testing the

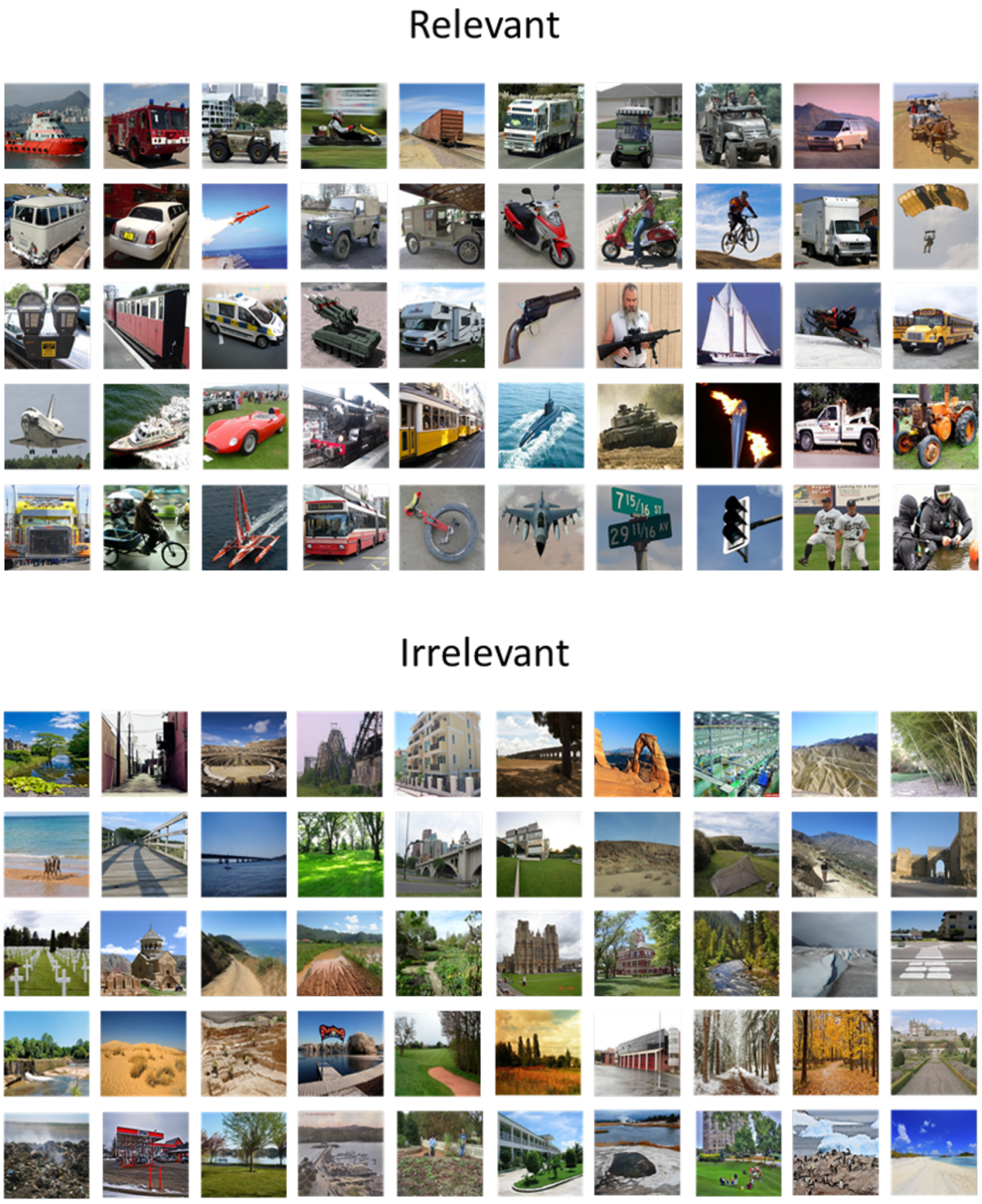

Figure 2.4: Example images from each one of the 100 classes that comprise the Mixed dataset.

low-shot classifier is discussed in Chapter 2.4.

The choice of these two datasets is deliberate, as in our application we are trying to recognize objects in a scene and push away scenes that have no relevant objects. Each class on the relevant part of the mixed dataset is comprised of 1,300 images, while each irrelevant class has available 13,000 pictures on average. The imbalance between relevant and irrelevant images is representative of the imbalance in the unlabeled data captured in the field which will contain many more irrelevant objects compared to targeted classes. From every class in both groups, 10 images are reserved for testing the accuracy of the various methods, which are compared after the training of the classifier has been completed. When using the Mixed dataset in this work, a part of the irrelevant pictures will be reserved and used as unseen samples (Figure 2.9), images that have not been seen during training but have to be recognized by the classifier in the same way as the irrelevant.

The performance of deep visual recognition algorithms severely degrades when objects from classes and scenarios not seen during training are encountered (*Mancini et al.*, 2020). In order to test the performance of our method in correctly placing objects from classes not encountered during the training into the irrelevant category, we introduce a third group of image classes: the Unseen dataset. To address this problem, we introduce a third group of image classes: the Unseen dataset. More specifically, 10 irrelevant classes are reserved and not used during the training of the OSLS classifier so that during testing these images comprise the unseen samples. For example, in Figure 2.2, out of the 50 classes that comprise the irrelevant dataset, 10 classes were assigned as unseen and were not used during training but only for testing. To match the number of irrelevant images, only 40 out of the 50 relevant classes were used to train and evaluate the method. More information about how the Mixed dataset

is utilised for training and testing the unsupervised feature extractor and the low-shot classifier is discussed in Chapter 2.6.

Chapter III presents the IDLS algorithm for image recognition and addresses the need of reconnaissance operations of autonomous ground vehicles to specifically identify a set of target classes and also determine when a candidate image does not belong to a target class, similarly to the OSLS. The IDLS provides this capability using a minimal amount of labeled data for the overall training. In order to demonstrate the value of the IDLS and test its accuracy the Mixed dataset is used in Chapter III too.

Examples of the mixed dataset images are presented on Figure 2.4 while a complete list of the classes in alphabetical order is presented in Appendix B. It is important to note that all images contain exclusively the object of interest. For example, it is guaranteed that there is no relevant object (such as cars or airplanes) included in a building picture of the irrelevant dataset. The "cleanliness" of the dataset plays a key role as this work is concentrated only on pure image classification rather than image segmentation. To train the IDLS feature extractor in Chapter III the way specified by the SimCLR method (*Chen et al.*, 2020), the entire dataset is utilized, but without making use of the labels. On the other hand, the training of the classifier uses only a modest amount of the same relevant images from each class (up to 40) and an equal total amount of irrelevant pictures randomly selected in each epoch from the irrelevant part of the Mixed dataset.

The Mixed dataset is one of the common denominators that tie the three main Chapters of this thesis together. The clustering algorithm proposed in Chapter IV also uses the Mixed dataset in order to test the quality of the Extended Variance Ratio Criterion (E-VRC) method. There, the Mixed dataset is presented as the most difficult to cluster dataset as not only the feature

extractor used to test the E-VRC is trained on different types of imagery, but also because the class contents are much more different from each other compared to the ImageNet and Caltech datasets.

2.4 Low-Shot Classifier Training

The two integral parts of the classifier training process are the target matrix and the loss function. The training goal is to tweak the initially randomized weight matrix so that when multiplying it with a testing feature map, it produces a scoring matrix whose largest value is the desired class element.

In machine learning, a fully connected layer performs the following calculation:

$$\hat{y} = \sigma(WF + \beta) \tag{2.2}$$

where W is the weighting matrix of the classifier, F is the feature map matrix, β is the bias vector and σ is the activation function. A Singular Value Decomposition (SVD) method solves our matrix equations (*Jia et al.*, 2017; *Huang et al.*, 2018). The pseudo-inverse method calculation results as:

$$\hat{y} = \sigma WF \tag{2.3}$$

Here, F is the feature maps, W is the weight matrix we desire to train, and $\hat{y}$ is the target, the ideal outcome for the score matrix. The MATLAB implementation[3] handles the training one class at the time, therefore F is a $N_{img} \times N_{feat}$ matrix and W is a N_{feat} long vector for each class, where N_{img} is the number of training images in every epoch and $N_{feat} = 512$ is the length of the feature vectors (constant).

[3] https://github.com/skasapis/ROCUnlabeledClassification

With no use of the bias vector, and the reversed order of F and W to account for the row-column switch, in SVD the W matrix is calculated one vector (class) at a time, therefore essentially solving for the least square solution of:

$$Ax = b \tag{2.4}$$

When the exact solution does not exist, which means that A is not a full-rank square matrix, the approximate solutions are:

$$Ax = \hat{b} \tag{2.5}$$

Therefore the approximation error is:

$$d = \hat{b} - b \tag{2.6}$$

and in a least-square approach, the loss function is:

$$L = ||d|| = \sqrt{\sum_{i=1}^{n} d_i^2} = \sqrt{\sum_{i=1}^{n} (\hat{b_i} - b_i)^2} \tag{2.7}$$

Substituting Equation 2.6 results to:

$$L = \sqrt{\sum_{i=1}^{n} (\sum_{j=1}^{m} A_{ij} x_j - b_i)^2} \tag{2.8}$$

An exponential version of the least square solution is introduced in order to explore a new, faster converging loss function based on (*Kim and Vlahopoulos*, 2012). The new squared-exponential loss function is:

$$L = \sum_{i=1}^{n} e^{d_i^2} = \sum_{i=1}^{n} e^{(\sum_{j=1}^{m} A_{ij} x_j - b_i)^2} \tag{2.9}$$

therefore the gradient can be proven analytically to be:

$$\frac{\partial L}{\partial x_j} = \sum_{i=1}^{n} 2e^{d_i^2} d_i A_{ij} \tag{2.10}$$

and the gradient vector is:

$$\frac{\partial L}{\partial x} = A^T(d. * e^{d^2}) \tag{2.11}$$

There are two main reasons for choosing this loss function. A squared-exponential function is easy to differentiate analytically and the differentiation is applied to the linear algebra form implemented in the MATLAB code. Note that the dot operator in Equation 2.11 (i.e., $.*$) used with in MATLAB matrix multiplication, creates element-wise operations. Compared to other differentiable functions tested, the square-exponential was the one to converge faster and in a steady way. A problem encountered, which got solved by normalizing the feature maps as described in Equation 2.1, is that because of the nature of the function, for numbers greater than 1, the loss would result in extremely high values.

The gradient matrix in Equation 2.11 is then multiplied by a learning rate (η) and added to the weight matrix (W), repeating this sequence for every epoch. The steps taken towards training the classifier matrix are therefore all independent from machine learning libraries or functions. Although many different loss functions that get differentiated in a semi-analytic fashion are being used by machine learning libraries, we concentrated our efforts on not using any existing libraries to create a stand-alone method. Therefore the squared-exponential loss function is a good fit.

As in most machine learning applications, the update mechanism used towards convergence is some variation of a normal gradient descent equation. In this specific case the update equation used can be mathematically described as

follows:

$$W_{k+1} = W_k - \frac{1}{2}\eta\frac{\partial L}{\partial x} \tag{2.12}$$

Here, in every epoch k, W gets updated by subtracting the product of the learning rate η and the gradient matrix. The learning rate is obtained using an algorithm inspired by Iterative Shrinkage-Thresholding Algorithm (ISTA) (*Beck and Teboulle*, 2009) which is noted in Appendix A. We begin with calculating a pseudo-loss which is going to be compared with the actual loss to determine whether the learning rate needs to be decreased or kept as specified on the previous epoch. This iterative method progressively decreases the learning rate as the algorithm approaches closer to the desired optimal point.

The last, and most unique part about this classification method is the target utilized. As mentioned in Chapter 2.1, the uniqueness of the proposed approach relies on the fact that unlabeled images are used during the training of the weight matrix. This is done by extending a standard one-hot encoding (*Harris and Harris*, 2010) matrix to also include class score distributions as targets for the Irrelevant images. Labeled images have arrays of zeros and a unit value on the correct class element as targets.

Irrelevant pictures belong to none of the classes therefore the score for each class should be zero. By experimentation we concluded that the irrelevant target that works best should be a slight negative value, such as -0.2. This intuition matches some of the binary classification work that has been done on Support Vector Machines' (SVM) correlation filters, where 0 and 1 were not as separable as a negative value (-0.1, -1) and 1 (*Zuo et al.*, 2018; *Boddeti and Kumar*, 2014). As an example, if a training dataset was consisted of six pictures, half of them labeled and half of them unlabeled, and the labeled ones were members of three different classes, the target matrix would look as follows:

$$\hat{y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.2 & -0.2 & -0.2 \\ -0.2 & -0.2 & -0.2 \\ -0.2 & -0.2 & -0.2 \end{bmatrix} \tag{2.13}$$

We train the weight matrix in such a way that during evaluation, the irrelevant images class score values are spread equally between the classes and acquire values as close to zero as possible. This helps the Irrelevant pictures to score less than the respective class threshold.

Along with using this target-oriented training procedure, we also increase our recognition accuracy by calculating our threshold scores using the ROC method as explained in Chapter 2.5. In the case it is not possible to obtain a significant amount of statistical knowledge about the unlabeled data, a method similar to Transductive SVMs could be used to leverage information from irrelevant pictures (*Joachims et al.*, 1999; *Sindhwani and Keerthi*, 2006). The equations outlined in this Chapter are utilized by the first two algorithms of Appendix A.

2.5 Low-Shot Classifier and ROC Threshold Calculation

The multiplication between a feature vector and a weight matrix yields a class score vector. Neural Network classification theory uses the highest score (Top 1, Top 3, or Top 5 have been used too) to group the images into classes. We extend this criterion to make it applicable when unlabeled images are present by introducing a Threshold Score (T_S) value for each class.

The T_S value serves as a binary discriminating test in order to group pictures

in the Relevant and Irrelevant bins. As mentioned above, we not only need to divide pictures into two groups, we also want the relevant group pictures to be normally classified in their respective class.

It is important to note that the T_S is calculated during the training of the OSLS. We need the T_S to be pre-calculated before we start evaluating our testing dataset. Once the training has been completed and the T_S is known, the classification process runs as follows: a) The testing image runs through the classifier and scores, which denote the likelihood of the image belonging to each class, are calculated. b) The image is assigned to the class with the highest score. c) The score of the assigned image is compared to the T_S of the class where it was assigned. If it is higher, then it is considered as a member of the class. If lower, it is determined to be an irrelevant image.

Within Figure 2.5, we present an example where class score values for 10 testing images from two classes are plotted. In blue and red, we see the 50 different class score values of the ten different Chess Board and Grapes images, respectively. In green is the ROC threshold calculated for each one of the 50 classes. During the training of the classifier matrix we treated the Chess Board pictures as labeled (Relevant), with label 45 attributed to them, and the Grapes pictures as unlabeled (Irrelevant). On the graph, we can see that during testing, most of the Chess Board pictures have high class score values on the correct class (45). The correct scores are also above the T_S of this class, therefore they will be classified correctly as members of the chess board class.

On the other hand, the Grapes pictures are treated as Irrelevant during training resulting in lower score values compared to the T_S (green line) of all relevant values. Our target matrix, along with our classification method, achieves to push the Irrelevant score values lower than the Relevant (red lines below the green line), while keeping the score values for the relevant class above

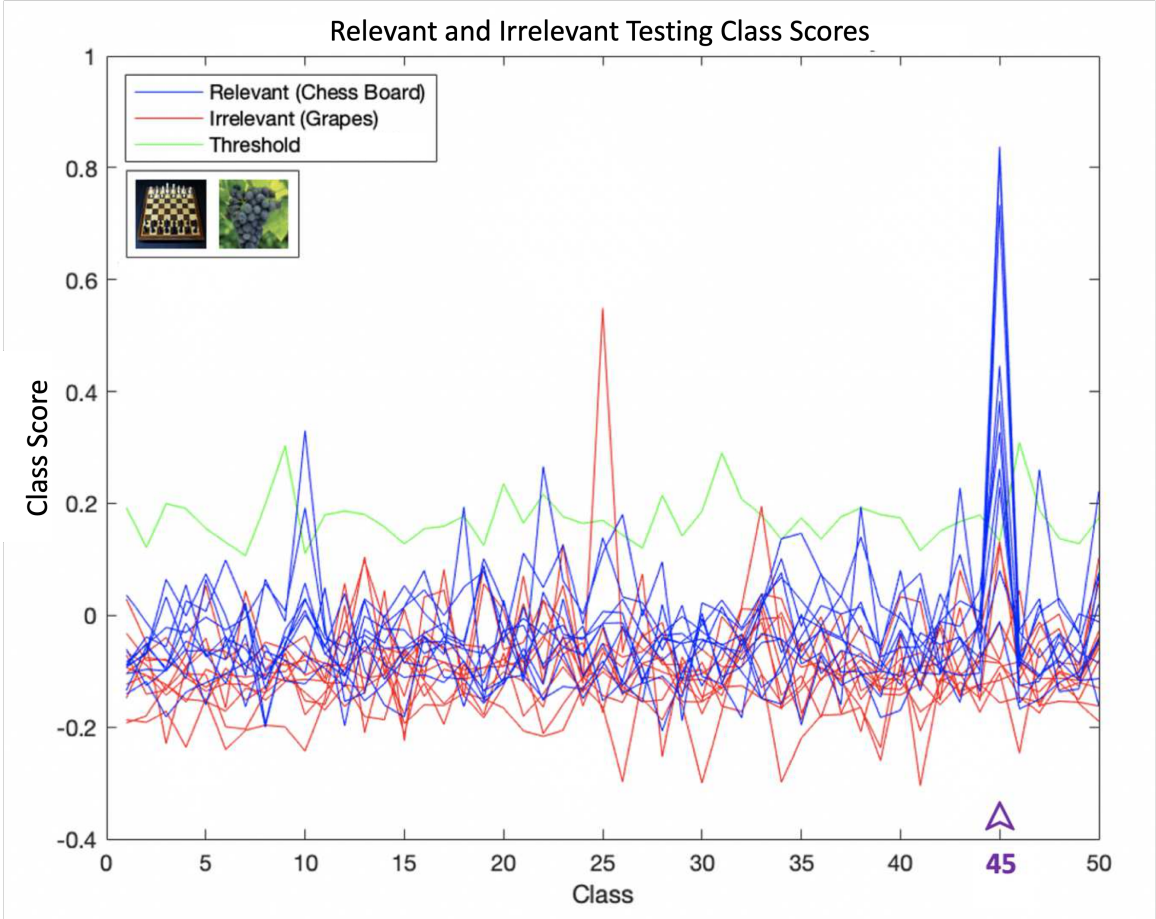

Figure 2.5: Line plot of class score values for 10 testing images belonging in two different classes.

the corresponding T_S (blue lines above green in class 45). Intuitively, we can set the T_S to be the lowest relevant value encountered during the training (Normal Threshold). If a picture is not classified higher than the worst correctly classified training picture, then it should be Irrelevant. As seen in Figure 2.5, although this discriminatory rule will give us the best possible Relevant accuracy, it will strongly discriminate against Irrelevant pictures.

Figure 2.5 shows in blue the normal distribution of Relevant scores within class X, while in red, we see the distribution of the Irrelevant pictures that got classified as class X. Using a normal threshold would classify all Relevant pictures as True Positive, but would hurt the True Negative and total accuracy

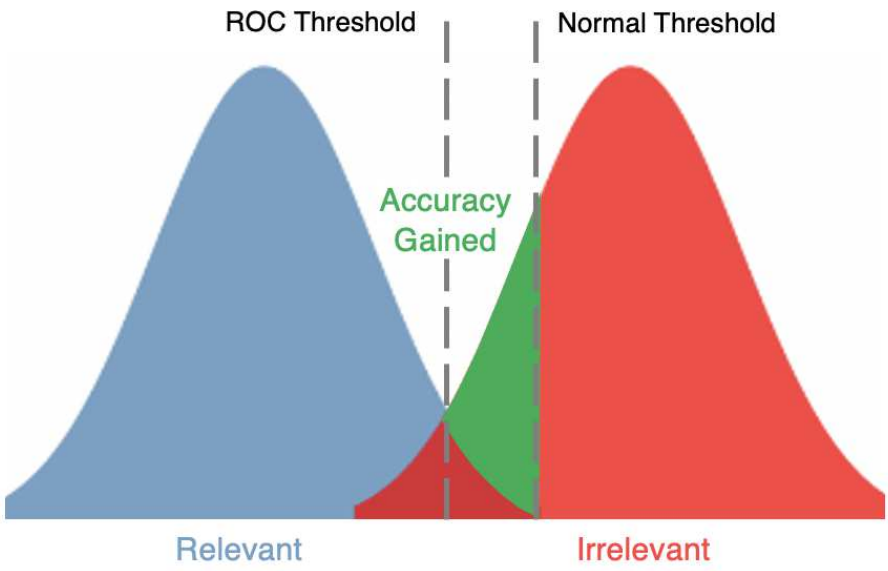

Figure 2.6: Threshold candidates in relation to the relevant and irrelevant distributions.

by a value represented by the green area. To maximize the combined accuracy, we use the ROC curve (*Peres and Cancelliere*, 2014; *Hand and Till*, 2001; *Hand*, 2009) to choose our threshold values. The same way we would do with a Normal Threshold, the ROC Threshold is going to be calculated right after training and before testing, using the training data explicitly.

To demonstrate the need of using the ROC T_S, we graph the ROC curves of five, unique compared to each other, classes of pictures that we trained our classifier on. All five classes were part of the labeled dataset (relevant classes). As seen in Figure 2.5, every different class of pictures has a different response to the ROC implementation. The different Areas Under the Curves (AUC) represent how well our ROC method is able to classify the data, but also underlines the need for such an implementation.

In our analysis, we focus on the classification of Relevant pictures, which we assume to be the Positive statistical case, therefore we use the terms True Relevant Rate (TRR) and False Relevant Rate (FRR). TRR and FRR are no

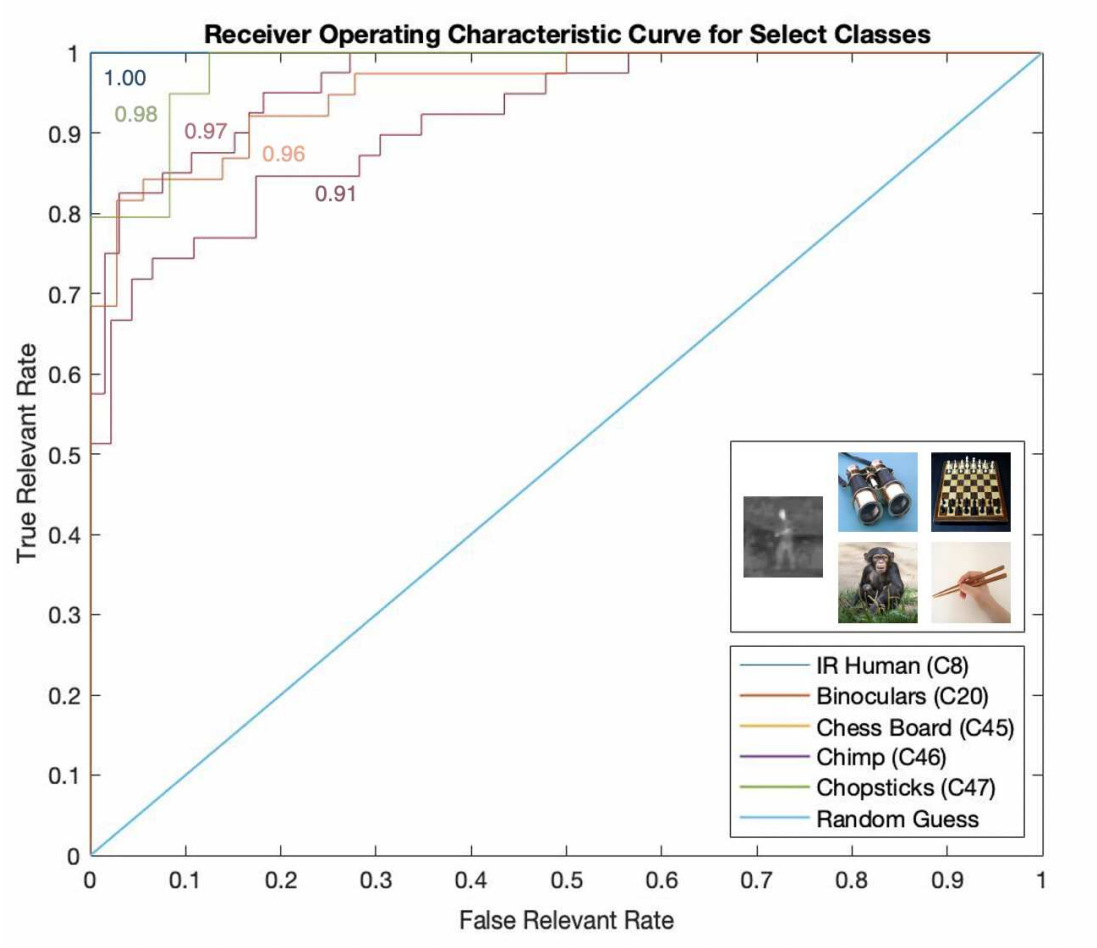

Figure 2.7: ROC curves of five unique classes of pictures used in the training process of the proposed classification method.

different than the True Positive Rate (or Sensitivity) and False Positive Rate (or Fall-Out) respectively, used in statistical analysis. True Relevant Rate is the number of relevant pictures that our classifier recognized as such, over the total number of correctly classified pictures. False Relevant is the ratio of pictures that were irrelevant but were predicted as relevant, over the total number of incorrectly classified pictures. Therefore for this analysis we define:

$$TRR = \frac{TP}{TP + FN} \quad \text{and} \quad FRR = \frac{FP}{FP + TN} \tag{2.14}$$

In Figure 2.7, it is obvious that the IR Human class is so unique that the

classifier does not have any trouble distinguishing it from the rest of the dataset, therefore as seen in the figure above, its AUC equals to 1 and the ROC does not have much effect on its cumulative accuracy. Different classes present different levels of difficulty for our classifier. The Chimp class, as seen above, has an AUC of 0.91, which means that the ROC can significantly improve its cumulative accuracy if a T_S is picked wisely.

The Normal Threshold would pick the point on the graph where the False Relevant Rate is minimum for a True Relevant Rate of 1, hence, for the Chimp graph, the $(0.57, 1.00)$ point. Using our ROC algorithm, any other point on the graph could be selected, such as the $(0.20, 0.84)$ point, which is the one further away from the blue line that represents a random guess. By doing this, although there is a slight decrease in TRR, a great increase in FRR can be gained, which results to a significantly higher cumulative accuracy.

Tables 2.1-2.3 demonstrate the usefulness of the ROC method in classifying Relevant images and rejecting an image if it is Irrelevant. In order to highlight the ROC capabilities, comparisons of these results to the two baseline methods, the "No Irrelevant" and the "+1 Class", are being presented.

The first baseline result (No Irrelevant), was produced by training the classifier only on labeled images of the relevant classes. This is the case where although unlabeled images are available for the irrelevant classes, they are not utilized, expecting the labeled images to have enough meaningful features to accommodate recognizing the irrelevant ones. To evaluate this method, Normal Threshold (in Figure 2.6) discussed above is used, where the lowest correct relevant training score is set as the threshold for each class. During testing, if the image's highest class score is larger than the respective threshold, then its classified as Relevant, if not, as irrelevant.

The second baseline result, called "+1 Class", was generated by training the

classifier to recognize the relevant classes along with one extra class which encapsulates all irrelevant images. During the training of the classifier, all unlabeled images of the Irrelevant classes were assigned to an extra class. The evaluation is being done by simply comparing the highest-scoring index of every image with the correct target.

Table 2.1 shows how the baseline methods scored for both relevant and irrelevant images compared to the Low-Shot Classifier, with and without applying the ROC optimization for the Top-1 selections.

Table 2.1: Low-Shot Classifier Compared to Baseline Examples (Top-1 Accuracy)

	Normal Dataset		Infrared Dataset	
Classifier	R	I	R	I
Low-Shot Classifier	70.8 %	74.8 %	78.2 %	75.4 %
Low-Shot Classifier w/ ROC	64.8 %	87.8 %	71.8 %	89.8 %
+1 Class	49.2 %	91.4 %	56.2 %	92.2 %
No Irrelevant	72.4 %	47.8 %	78.6 %	52.4 %

As "Normal" is described the dataset consisting of 50 relevant and 50 irrelevant Caltech256 classes and as "Infrared" the dataset where 8 of the relevant classes were substituted with IR ones. The contents of both datasets are thoroughly discussed in Chapter 2.3 and outlined in Appendix B. The Low-Shot Classifier results are obtained by running the algorithm noted in Appendix A and the Low-Shot Classifier with ROC by adding the ROC extension. "R" and "I" are the relevant and irrelevant classification accuracy respectively. The numbers shown in the tables are the Top-1 percentages of images that got classified correctly during evaluation.

It can be observed that the baseline methods are unable to classify both groups of images decently. The +1 Class method seems to over-train the classifier on recognizing the unlabeled images failing to put the labeled ones in the correct classes. This happens most likely due to the unbalanced training data, as the

51st class has as many images as the rest 50 together. On the other hand, by using only labeled images, the classifier is trained to specifically recognize the labeled group, failing to filter out the unlabeled images "noise".

In the first row of Table 2.1, the results of the classifier proposed without the ROC extension show that the loss function in Equation 2.9 combined with the unique target matrix in Equation 2.13 and the threshold score criterion can recognize equally well both labeled and unlabeled images. It is notable that for the relevant group, the proposed method loses a small amount of accuracy compared to the label-specific baseline method but does substantially better in identifying irrelevant images.

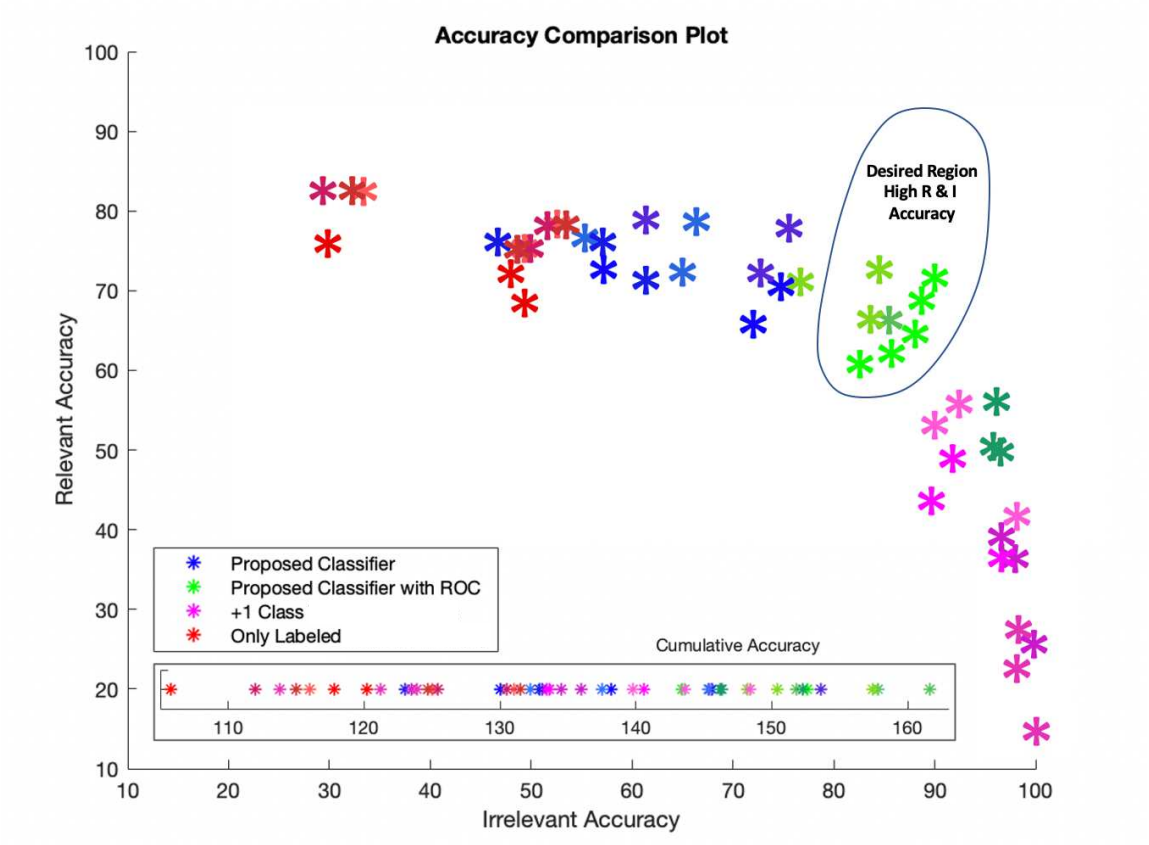

Figure 2.8: Scatter plot used for visualization of the four different methods compared in Table 2.1.

The ROC method greatly increases the unlabeled images recognition, to the

modest expense of the labeled images. Table 2.1 shows the importance of using the ROC to increase the cumulative accuracy. The ROC classifier increases by 12% the cumulative recognition scores compared to the +1 Class method and by 25.4% compared to the label exclusive transfer learning method.

The results presented in Table 2.1 are also depicted in the Accuracy Comparison Graph in Figure 2.8. For every method discussed,three different ResNet10 feature extractors are used (BatchSGM, SGM, L2) in order to show the consistency of the proposed classifier results. The legend follows the color-coding of the four methods in Table 2.1. For each of the four methods, the graph has four different sub-color groups (shades of each respective color) with three data points each. The four different sub-groups represent the different datasets discussed in Chapter 2.3 (Mixed, IR, Caltech +50 and Caltech +150) and the data points are the three different pre-trained feature extractors noted above.

With a few exceptions, no matter the feature extractor or the nature of the dataset used (including infrared or more unlabeled images), the proposed method (green data points) not only provides a higher cumulative accuracy but also eliminates the bias between labeled and unlabeled images by classifying both equally well when compared to the baseline approaches. The results of the two extended datasets are introduced in Figure 2.8, therefore Table 2.2 offers a closer look to the comparison of the two datasets which include more unlabeled samples.

Table 2.2: Extended Datasets Comparisons (Top-1 Accuracy)

	+ 50 Irrelevant		+ 150 Irrelevant	
Classifier	R	I	R	I
Low-Shot Classifier	79.0 %	66.3 %	76.4 %	56.7 %
Low-Shot Classifier w\ ROC	73.0 %	84.4 %	59.0 %	93.0 %
+1 Class	36.8 %	97.7 %	22.0 %	99.2 %
No Irrelevant	78.6 %	51.9 %	78.6 %	52.9 %

We follow the same notation in Table 2.2 as used in Table 2.1, with the only difference being that the "+50" and "+150" Irrelevant datasets are the two expanded datasets noted in Chapter 2.3. Although it is clear in both Table 2.2 and the plot in Figure 2.8, that the proposed method still scores better in a cumulative perspective, one can also observe that biases against the relevant (in Low-Shot Classifier with ROC algorithm) or irrelevant (in Low-Shot Classifier algorithm) group begin to occur when increasing the number of irrelevant images.

The more the irrelevant to relevant ratio increases, the worse the Low-Shot Classifier scores on the irrelevant part. This might seem counter-intuitive as we would expect that the more unlabeled images are seen during training, the better the algorithm would be able to recognize them. In reality, many more feature elements are introduced on the irrelevant part, which leads to consequently eliminating their uniqueness. This observation suggests how well the proposed classifier might work on recognising background noise from actual pictures of interest in the case of Semantic Segmentation (*Arbeláez et al.*, 2012).

When introducing the proposed ROC approach on the second row of Table 2.2, no bias is being introduced because the ROC threshold has been adjusted in such a way that it is non-discriminating against any group (Optimal ROC). Table 2.3 presents the results of the adjusted ROC Classifier when it is being used on the +150 Irrelevant dataset. The same behavior is observed when testing the rest datasets.

The "Optimal ROC" and "No Irrelevant" entries correspond to the second and fourth rows of Table 2.2. Putting a constraint on how much we are willing to shift the T_S to limit the loss in relevant, affects negatively the irrelevant. We desire to find a percentage that during testing gives a decent cumulative accuracy without significant losses on the Relevant part. This could be imagined

Table 2.3: ROC Adjustment for the +150 Irrelevant dataset (Top-1 Accuracy)

	R	I	
No Irrelevant	78.6 %	52.9 %	Increase
Optimal ROC	59.0 %	93.0 %	+ 20.5 %
80% Constraint	61.4 %	90.2 %	+ 20.1 %
90% Constraint	68.2 %	82.0 %	+ 18.7 %
92.5% Constraint	72.0 %	77.0 %	+ 17.5 %

as turning a knob to tune the ROC implementation. This can be specific in every application, therefore an open-ended approach is adopted.

A 100% constraint would be the Low-Shot Classifier without the ROC implementation, as we set the Threshold Scores to be the lowest correctly classified irrelevant picture in every class. In Table 2.3, a 90% Constraint means that the ROC algorithm is asked to keep the thresholds to a value that will not hurt the correct relevant guesses more than 10% during the calculation of the T_S. Therefore these constraints are applied when using the training images and they differ from the percentages encountered in the testing (Table 2.3). As it can be seen, for this specific case, a compromise of a 18.7% total increase is acceptable, instead of the 20.5% of the optimal case, in order to get an equal recognition accuracy.

2.6 OSLS Classifier Results

The Low-Shot Classifier is able to recognize images from the relevant classes and also identify irrelevant images from the classes it has seen during training. Ideally, during operation we desire to recognize objects that are not seen at all during training, which is the main objective of open-set recognition. To achieve this, the capabilities of the Low-Shot Classifier described in Chapter 2.3 are extended to recognising unseen images, resulting in the Open-Set Low-Shot (OSLS) Classifier.

The unseen samples are the sub-group of the irrelevant classes that do not get involved in training, however, it is still expected that the OSLS Classifier recognizes them as irrelevant. This is accomplished by randomizing the selection of the irrelevant samples in every training epoch of the OSLS classifier. More specifically, during the training of the Low-Shot Classifier, there is c number of classes, each of which contains n number of training images for both relevant and irrelevant datasets (only the images for the relevant dataset are labeled). This set of $n \times c$ images is the same in each epoch.

When training the Open-Set Low-Shot Classifier, the irrelevant images are different in each epoch and selected randomly from the pool of unlabeled irrelevant images, while still keeping the total number of irrelevant training images in each epoch the same with the relevant part ($n \times c$). By introducing this imbalance and by not repeating the same irrelevant samples in each epoch, the OSLS classifier is able to generalize better on the irrelevant part, yielding better classification accuracy for the irrelevant and unseen testing samples. In all results presented in this dissertation, the testing images are always different than the images used during training.

In summary, and as seen in Figure 2.9, the two differences of the OSLS method compared to the Low-Shot Classifier discussed in the previous Chapters are that a) we extend our testing dataset to include ten classes of unseen images (each containing ten testing samples) and b) we extend the irrelevant part of the training dataset by introducing randomness and imbalance between epochs and classes respectively. During training, a different selection of irrelevant samples and the same selection of relevant samples is used in every epoch.

A comparison between the traditional (Low-Shot Classifier) and the randomized irrelevant training of the Low-Shot Open-Set Classifier is presented in Figure 2.10. For this comparison, the Low-Shot Classifier training uses the

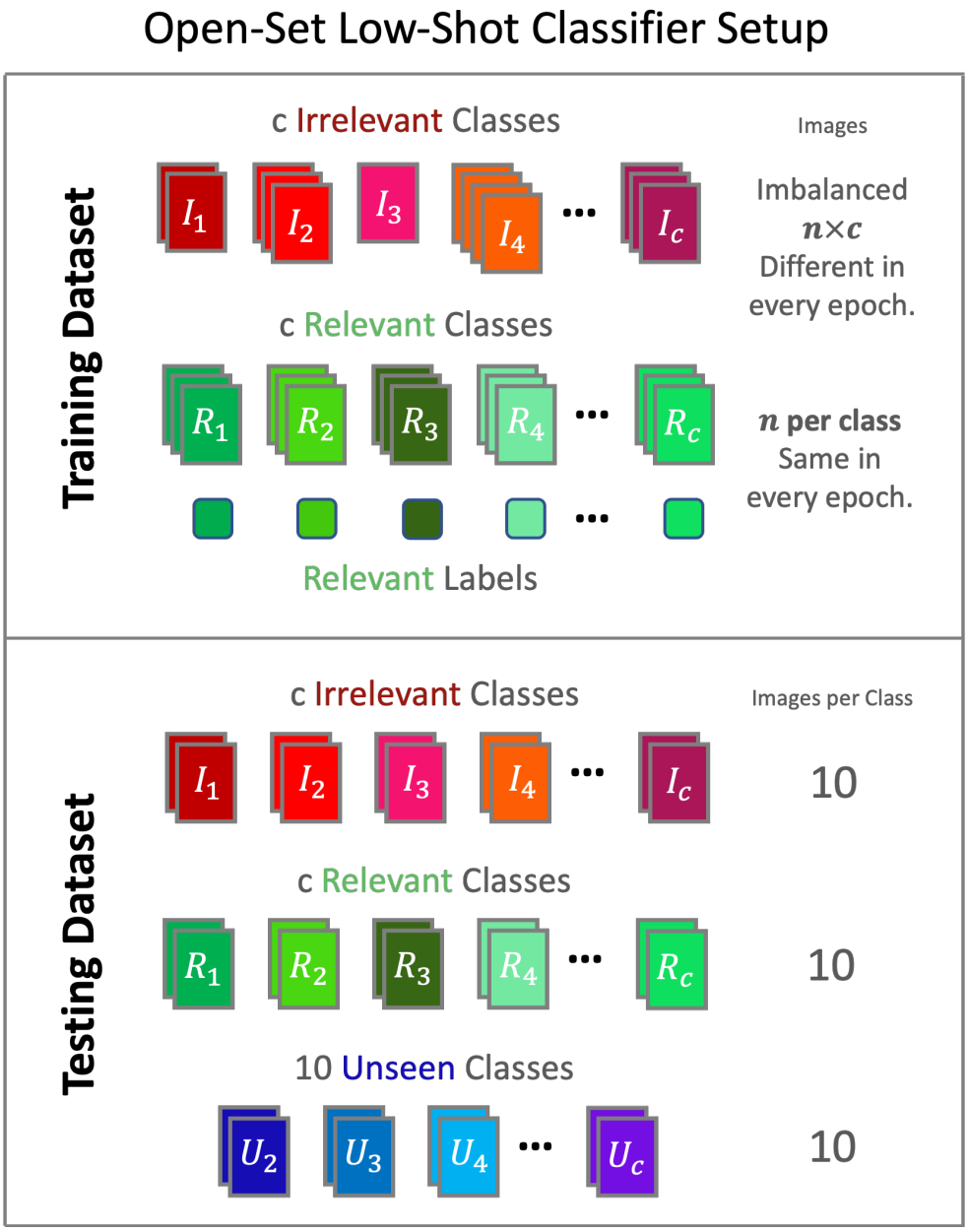

Figure 2.9: Schematic of the image dataset training and testing setup for the OSLS classifier.

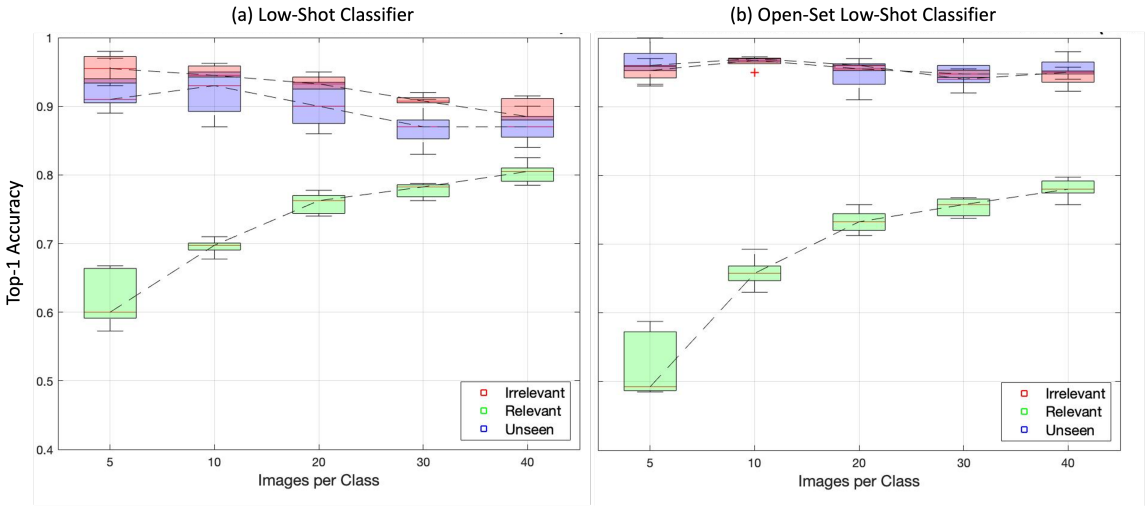

Figure 2.10: Accuracy box plots of the three different testing groups for a) the low-shot classifier and b) the OSLS.

same relevant and irrelevant pictures and classes (40) in every training epoch, whereas the OSLS uses the same set of relevant classes (40) and pictures but samples randomly a different group of irrelevant training images in every epoch. For instance, in the 40 images per class case, the Low-Shot Classifier is trained on the same 40 relevant and irrelevant classes, which all include the same 40 pictures for each class. For the OSLS Classifier case, although the 1,600 relevant images (from 40 different classes) are kept the same throughout training, the 1,600 irrelevant images in each epoch are picked randomly from a pool of 20,000 samples (500 samples for each one of the 1940 classes), introducing not only randomness but also an imbalance between class samples. Here, it is also demonstrated that by introducing randomness and class imbalance during training, for every Images per Class case, there is a slight decrease in the relevant accuracy, but a substantial increase in the testing performance of both irrelevant and unseen. All the results below use the ROC threshold that produces the highest combined relevant and irrelevant score.

In Figure 2.10 the accuracy box plots for the three different testing groups

are presented: in red the Irrelevant, in blue the Unseen and in green the Relevant results. For every Images per Class case, ten different random tests are performed in order to quantify the uncertainty of each case study. The box plot sides represent the median of the lower and upper half of the different results set respectively. The lines extending from the boxes (whiskers) indicate the variability outside the upper and lower quartiles, while the red line within the boxes represents the median of the entire spread. Lastly, the red crosses represent the accuracy of the outlier runs and the dashed line connects the median accuracy values of all the different cases.

Table 2.4: OSLS Results (Top-1 Accuracy) for different numbers of classes and images.

C \ P	5	10	20	30	40
	Relevant				
5	0.56 ± 0.12	0.61 ± 0.07	0.75 ± 0.09	0.78 ± 0.03	0.83 ± 0.08
10	0.53 ± 0.07	0.66 ± 0.07	0.83 ± 0.03	0.84 ± 0.03	0.89 ± 0.04
20	0.53 ± 0.07	0.66 ± 0.02	0.78 ± 0.02	0.83 ± 0.01	0.84 ± 0.05
30	0.51 ± 0.05	0.66 ± 0.03	0.74 ± 0.02	0.77 ± 0.02	0.78 ± 0.01
40	0.51 ± 0.05	0.66 ± 0.02	0.73 ± 0.02	0.75 ± 0.01	0.76 ± 0.02
	Irrelevant				
5	0.95 ± 0.08	0.97 ± 0.03	0.98 ± 0.02	0.98 ± 0.02	0.99 ± 0.01
10	0.93 ± 0.04	0.97 ± 0.04	0.97 ± 0.01	0.98 ± 0.01	0.98 ± 0.02
20	0.95 ± 0.04	0.98 ± 0.01	0.97 ± 0.01	0.96 ± 0.01	0.97 ± 0.02
30	0.97 ± 0.02	0.98 ± 0.01	0.97 ± 0.01	0.96 ± 0.01	0.96 ± 0.01
40	0.96 ± 0.02	0.95 ± 0.01	0.95 ± 0.01	0.95 ± 0.01	0.94 ± 0.01
	Unseen				
5	0.93 ± 0.06	0.96 ± 0.02	0.97 ± 0.03	0.98 ± 0.02	0.98 ± 0.01
10	0.95 ± 0.02	0.95 ± 0.04	0.97 ± 0.02	0.96 ± 0.02	0.96 ± 0.02
20	0.95 ± 0.05	0.98 ± 0.01	0.97 ± 0.01	0.95 ± 0.03	0.96 ± 0.01
30	0.96 ± 0.02	0.97 ± 0.02	0.97 ± 0.01	0.95 ± 0.02	0.93 ± 0.01
40	0.96 ± 0.02	0.96 ± 0.02	0.95 ± 0.02	0.95 ± 0.02	0.94 ± 0.01

Table 2.4 presents the complete set of results for the OSLS method for a variable number of classes and images per class. Horizontally are presented the results for a variable number of pictures per class (P). Vertically are presented

the results for a variable number of classes (C) for each one of the three testing sample categories. For each different example, the mean and standard deviation of 10 different random tests is presented for the Top-1 accuracy.

All the results presented in the box plots of this text are for an OSLS classifier that is trained on 40 relevant and 40 irrelevant classes, both of which have the number of relevant images per class specified in the x-axis. Similar works in the open-set literature (*Ge et al.*, 2017; *Dhamija et al.*, 2018; *Liu et al.*, 2020) use a lower number of classes during training and testing (10 to 95 classes compared to the total number of classes used in this work ranging between 90 and 250). To show how the OSLS Classifier performs in tests where the same order of classes are used, we vary the number of relevant and irrelevant classes used during training. Although it is of interest to recognize samples of as many classes as possible (a maximum of 40 as presented in Figure 2.10), by observing Table 2.4 it is evident that the OSLS Classifier achieves very high Top-1 accuracy scores in situations where the relevant and irrelevant classes we are trying to detect are limited.

Take as an example the case (in bold) where the classifier is trained on 10 Relevant and 10 Irrelevant classes each one of which includes 40 training samples- a total of 400 labeled and 400 unlabeled images. By testing using 10 samples per class from 10 relevant, 10 irrelevant and 10 unseen classes the classifier achieves Top-1 accuracy scores of 0.89 ± 0.04, 0.98 ± 0.02 and 0.96 ± 0.02 respectively, with a very low variance between the random runs ($\sigma \leq 0.04$).

The OSLS classifier is meant to be used as the final layer of any CNN that is expected to recognize samples that belong to the training classes while identifying as irrelevant images that are not relevant regardless if they originate from seen or unseen during training datasets. In order to demonstrate the versatility of the proposed method, we attach the classifier to deeper feature

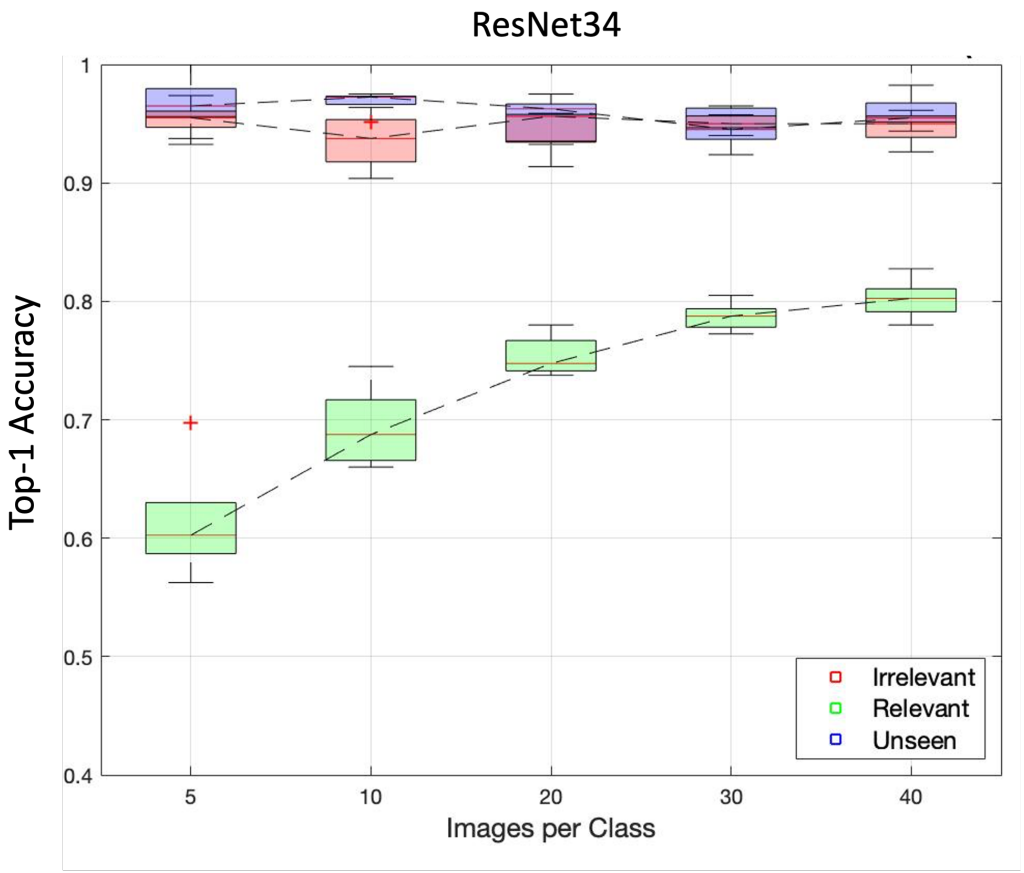

Figure 2.11: Box plots for the OSLS classifier presented in Figure 2.10b if the deeper ResNet34 is used to reduce the images to feature vectors.

extractors. Throughout this Chapter, the feature extractor used to test any classifier was a pre-trained ResNet18 provided by PyTorch[4].

In Figure 2.11, results for a classifier similar to the one in Figure 2.10 are presented, with the only difference being that the feature vectors are produced using the deeper ResNet34. Improvements in accuracy ranging from ≤ 0.11 to ≥ 0.02 (for the 5 and 40 Images per Class cases respectively), compared to those of Figure 2.10b, can be observed for the relevant testing samples while virtually no improvement is observed for the irrelevant and the unseen samples. Similar results are expected if the OSLS Classifier is used as a head for deeper networks that produce feature vectors of higher quality. The improvements can

[4]https://pytorch.org/vision/stable/models.html

be attributed to the fact that a deeper network has the ability to produce better quality feature representations.

The image feature representations used in this study are obtained raw, before any normalization is applied to them. As mentioned in Chapter 2.2, Equation 2.1 is used to normalize the input feature vectors. Figure 2.12 exhibits a decrease in accuracy if the features are normalized using the popular Softmax normalization commonly used in classification layers.

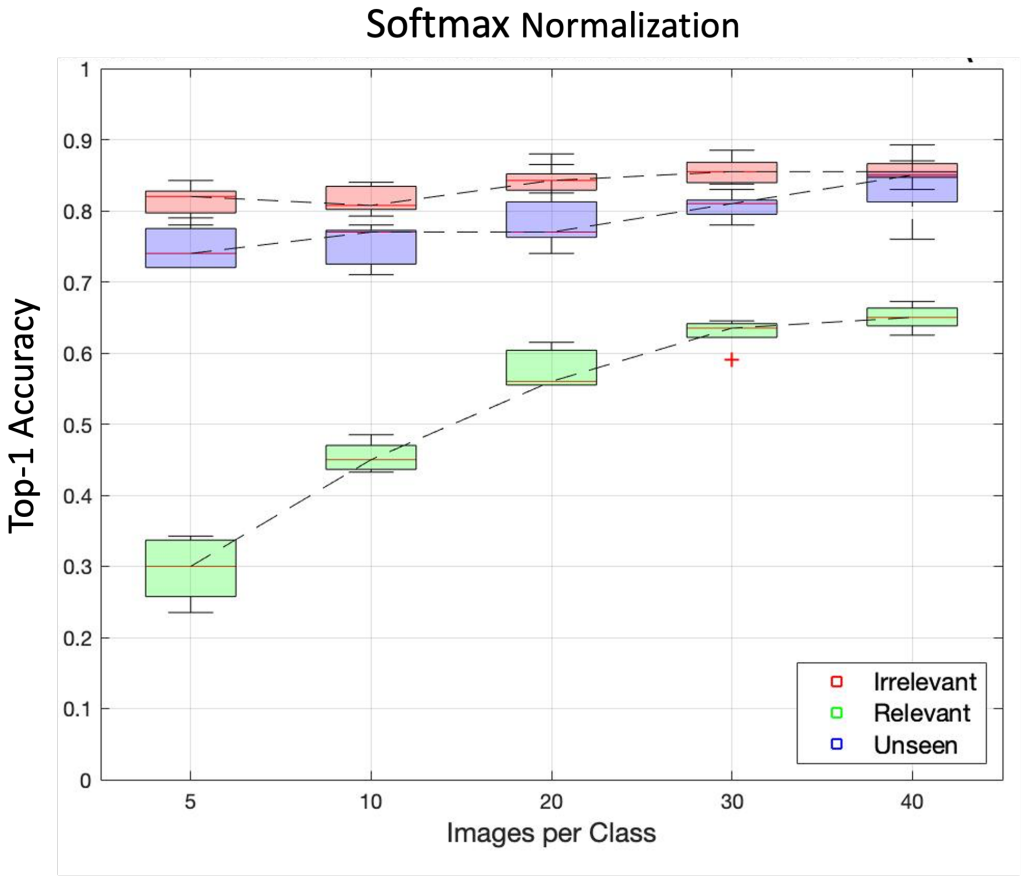

Figure 2.12: Box plots for the OSLS Classifier presented in Figure 2.10b if Softmax was used to normalize the training and testing samples.

More specifically, the OSLS results in Figure 2.10b show an improvement compared to Figure 2.12 that ranges from ≤ 0.19 to ≥ 0.15 for the relevant, ≤ 0.17 to ≥ 0.08 for the irrelevant and ≤ 0.26 to ≥ 0.08 for the unseen testing

samples (for the 5 and 40 Images per Class cases respectively).

Finally, to demonstrate the value of the OSLS classifier, we compare it to the two baseline examples mentioned in Table 2.1, namely, the "+1 Class" and the "No Irrelevant". The "+1 Class" method seen in Figure 2.13a groups all the irrelevant samples in one class during training and expects the irrelevant and unseen testing samples to be classified like they belong to the extra class. The "No Irrelevant" method seen in Figure 2.13b is a normal classification layer that is trained only on relevant images although expected to recognize irrelevant and unseen images too.

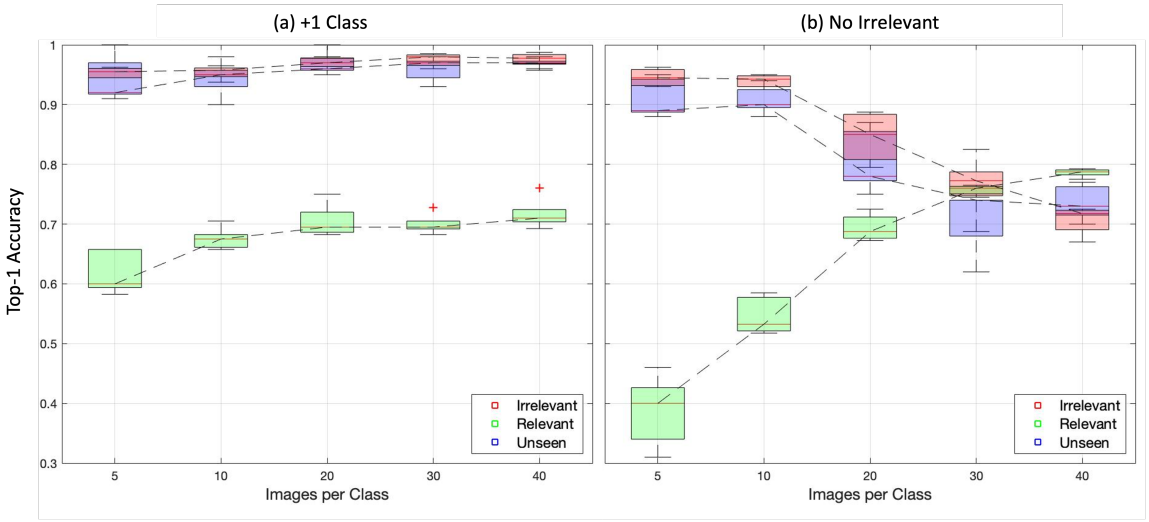

Figure 2.13: Results for the two baseline examples mentioned in Table 2.5.

By comparing Figure 2.10b to Figure 2.13a, for a low number of samples per class, the "+1 Class" method performs equally well or in cases even better in all three categories compared to OSLS, with relevant accuracy scores ranging from ≥ 0.6 to ≤ 0.7 for the 5, 10 and 20 Images per Class cases while unseen and irrelevant recognition reaching accuracies ≤ 0.96. When enough data samples per class are available though, the OSLS method improves the relevant accuracy by 0.05 and 0.1 for the 30 and 40 images per class cases respectively. The

improvement in relevant image classification that the OSLS classification (Figure 2.10b) achieves is significant compared to the minor (≈ 0.01) decrease in relevant and unseen accuracy scores.

The benefits of introducing irrelevant images during training are evident when comparing the results of Figure 2.13b (No Irrelevant) to the OSLS Classifier results in Figure 2.10b. If no irrelevant images are available during training and the number of relevant training images per class is small (5 and 10), a normal classification layer tends to over-fit on the latter. Due to this over-fitting, the ROC Threshold rejects most of the samples during testing resulting to very high (≥ 0.9) irrelevant and unseen and very low (≤ 0.53) relevant accuracy scores. When there are more training images per class, a significant increase in relevant accuracy can be observed, which is followed by a decrease in irrelevant and unseen accuracy.

More specifically, assuming similar specifications (normalization, loss function, etc.), if a single layer classifier is trained on 40 classes, each one including 40 images, the mean relevant, irrelevant and unseen accuracy scores during testing are 0.78, 0.72 and 0.73, respectively. If an equal number of unlabeled images are used during training, in the manner specified by the OSLS method, the mean relevant accuracy decreases by 0.02 while the irrelevant and unseen accuracy scores increase by 0.22 and 0.21, respectively. The trade-off between a very small decrease in relevant accuracy and a ten times larger increase in both irrelevant and unseen classification performance is the best demonstration of how the OSLS Classifier can be utilized in real-life applications.

The proposed OSLS Classifier using the ROC Threshold Score criterion not only makes the resulting model more flexible and easy to customize depending on the needs of the datasets, but also makes the method flexible for any application. This is a specifically interesting feature of this work, as we prove that the

classifier can be used as an extension to any image recognition algorithm which desires to filter out irrelevant and unseen images without the expense of labeling. Chapter III discusses how the OSLS classification method can be combined with an unsupervised feature extractor to produce the Instance Discrimination Low-Shot Classifier (IDLS).

CHAPTER III

Instance Discrimination Low-Shot Classification Framework

3.1 Related Work

The aims of unsupervised learning are to obtain features without requiring manually annotated data. In recent years the performance gap with supervised algorithms has been rapidly closing, increasing the attention to methods like Deepcluster proposed by *Caron et al.* (2018). A relatively old and well known class of unsupervised learning methods is clustering (*Pelleg et al.*, 2000; *Sinaga and Yang*, 2020). Algorithms that aim to cluster data have been used in a wide variety of applications including computer vision. Deepcluster adapts k-means clustering (*Likas et al.*, 2003) to the end-to-end training of visual features on large-scale datasets. The subsequent assignments of a standard clustering algorithm like k-means are used as supervision in order to update the weights of the neural network instead of the typical label-prediction comparison final step.

Like supervised methods (*Girshick et al.*, 2014), Deepcluster tries to capture apparent visual similarity among categories. Whether they come from manual labeling or by cluster assignment processes, these neural networks perform their

outcome comparison and train their filter weights using categorized data.

The effort to detect similarity among categories has recently been extended for generating a good feature representation that captures apparent visual similarity among instances, instead of entire classes or clusters (*Hadsell et al.*, 2006). For example, *Wu et al.* (2018) propose a trainable classification algorithm (N-PID) that asks the output feature to be discriminative of individual instances instead of groups of pictures. Similarly, Dosovitskiy et al. trains their feature extractor in such a way that it discriminates between a set of surrogate classes that are formed by applying a variety of transformations to a randomly sampled image patch (*Dosovitskiy et al.*, 2014). Both methods formulate the intuition of detecting similarity between two objects as a non-parametric classification problem at the instance-level.

The aforementioned methods introduce learning through contrastive instance discrimination which has been the basic idea behind numerous recently published implementations that have shown great promise, achieving state-of-the-art results (*Bachman et al.*, 2019). Instance discrimination is an approach to self-supervised representation learning which is based on maximizing mutual information between features extracted from multiple views of a shared context. An example is the algorithm proposed by Reite et al. (*Reite et al.*, 2019) which by adding a memory bank to the instance-discrimination task, performs well on both common and rare (few training samples) classes, identifies outliers within a labeled data set and learns a natural class hierarchy automatically.

Another recent achievement of contrastive instant discrimination learning methods is the extension proposed by *Caron et al.* (2020) which combines instance discrimination with clustering. Their algorithm (referred to as Swapping Assignments Between Views, or SwAV) takes advantage of contrastive methods without requiring to compute pairwise comparisons. Specifically, instead

of comparing features directly as in the previous methods, SwAV simultaneously clusters the data while enforcing consistency between cluster assignments produced for different augmentations of the same image.

Similar to the work of Caron et al. is the SimCLR algorithm proposed by *Chen et al.* (2020). SimCLR is a simple framework for contrastive learning of visual representations which simplifies the recently proposed contrastive self-supervised learning algorithms without requiring specialized architectures or a memory bank. Both SwAV and SimCLR are online methods which means that they calculate their loss by comparing each feature vector to another view of the same picture, without the need for iterating through the same images multiple times. Due to its architectural simplicity and information capturing superiority, in this Chapter we use this instance discrimination algorithm of the SimCLR to train the feature extractor which reduces images to the vector space, with the intend to connect it to the classification apparatus (OSLS) outlined in Chapter II.

The good performance of the SimCLR is the result of the data augmentations which play a crucial role in defining effective predictive tasks and the introduction of a learnable nonlinear transformation between the representation and the contrastive loss. Extensive work has been done on understanding not only the learning of representations but also the failure mechanisms of the SimCLR (*Kalibhat et al.*, 2022).

An alternative to our unsupervised approach for training a feature extractor is provided by Transfer Learning (TL). In TL, a CNN is trained first using a readily available labeled dataset. The classifier part of the network is disregarded and only the trained feature extractor is retained for further use. TL is a well studied process (*Petangoda et al.*, 2020) for transferring relevant knowledge from known solutions to related tasks. In Chapter II a pre-trained through TL feature

extractor was used in when training the OSLS classifier. During the training of the OSLS classifier threshold probabilities are determined using the ROC statistical method so that they can specifically recognize each relevant class and when irrelevant images are encountered. A random selection of irrelevant unlabeled images at each training epoch enables the classifier to correctly place images unseen during training in the irrelevant category.

When using TL in ground vehicle applications, information about the pre-trained feature extractor can be available to adversaries since the filter weights of the feature extractor or the labeled images used for its training are publicly available. Such information can provide an operational advantage to adversaries. Additionally, the type of images of interest to the low-shot classifier will be much different than the publicly available data set used for training the feature extractor, leading to loss of discriminating capability when reducing images to the feature space.

In this Chapter the SimCLR method is integrated with the OSLS classifier for developing the presented IDLS approach. A large number of unlabeled images from an unconstrained environment of interest are used for training the feature extractor. Modest number of images for each relevant class with a larger set of unlabeled irrelevant images are used for completing the training of the OSLS classifier. Overall, a small fraction of labeled images ($< 0.27\%$) compared to a fully supervised trained CNN are required by the IDLS without any significant loss in accuracy.

3.2 Unsupervised Training Using Instance Discrimination

Traditional machine learning uses a conventional parametric softmax formulation. For an input image x and a feature $z = f(x)$ produced by a feature extractor f, the probability of it belonging in a class i is

$$P(i|z_j) = \frac{e^{\mathbf{w}_i^T z_j}}{\sum_{j=1}^{n} e^{\mathbf{w}_i^T z_j}} \tag{3.1}$$

where typically $\mathbf{w}$ is a trainable weight matrix used for the probability of every image feature, $\mathbf{w}_i$ is the matrix row that corresponds to class i and n is the total number of images involved in the learning example.

In the problem with the parametric softmax formulation in Equation (3.1) w serves as a multi-class prototype information matrix which prevents explicit comparisons between features. The non-parametric variation that recent works (*Reite et al.*, 2019; *Caron et al.*, 2020) are based on, replaces the $\mathbf{w}$ matrix with different views of the feature instances themselves z_j, z'_j as seen in Equation (3.2).

$$P(i|z_j) = \frac{e^{z_j^T z'_j}}{\sum_{j=1}^{n} e^{z_j^T z'_j}} \tag{3.2}$$

To apply Equation (3.2) we need to normalize the features z in such a way that the condition $||z|| = 1$ holds true. Different variations of the non-parametric equation are seen in a number of applications. *Wu et al.* (2018) includes a temperature term which divides the vector's dot products and *Caron et al.* (2020) uses a memory bank to calculate feature codes which then get compared in order to calculate the probability P. In our work we use the loss function of the SimCLR method derived by *Chen et al.* (2020):

$$L = -\sum_{i=1}^{N} log \frac{e^{h_{2i-1}^T h_{2i}}}{\sum_{k=1(k\neq 2i-1)}^{2N} e^{h_{2i-1}^T h_k}} \tag{3.3}$$

where N are the number of images in each batch, h_{2i-1} and h_{2i} are the two views of the i^{th} image processed through the neural network (Figure 3.1). The loss function L from each batch gets accumulated in an overall loss function

for all batches of an epoch. Operations of normalization and scaling using a temperature coefficient are also performed when determining L but are not discussed here.

SimCLR learns representations by maximizing agreement between differently augmented views of the same image using the contrastive loss Equation (3.3). It is important to note that *Chen et al.* (2020) use augmentation as the terminology to describe the distortion of images to produce different views rather than the addition of information to the image data. During training, SimCLR uses a stochastic data augmentation module that transforms any given input image producing two correlated views of the same example. Chen et al. demonstrate that randomly cropping and augmenting the input images is crucial for the performance of the algorithm. The augmented image views are then passed through a neural network base encoder in order to extract feature representations. Finally, the produced representations are processed by a small non-linear projection head that maps them to the space where the contrastive loss in Equation 3.3 is applied.

A comparison between the three state-of-the-art instance discrimination methods (*Wu et al.*, 2018; *Caron et al.*, 2020; *Chen et al.*, 2020) mentioned in the Chapter 3.1 is performed. For efficiency we use ResNet18 as a feature extractor in all methods presented. ResNet18 is a popular neural network with a limited number of trainable parameters compared to deeper networks and produces 512 dimension image vector representations. The amount of computational resources employed for training are representative of the limited resources ($\sim$55 minutes/epoch with a standard 8 Gb GPU and 2.8 GHz CPU) expected to be available during operation.

After training the three unsupervised methods for feature extraction on the first 500 classes of Imagenet1000 without using the labels (*Deng et al.*, 2009),

Table 3.1: Correctly classified images accuracy for three unsupervised methods.

Method	SimCLR	SwAV	N-PID
Author	*Chen et al.* (2020)	*Caron et al.* (2020)	*Wu et al.* (2018)
Top 1	49.5 %	47.2 %	37.6 %
Top 5	74.2 %	72.4 %	60.4 %

the feature extractors are frozen and a simple classifier is attached to them as a head. The classifier is first trained to recognize images from 500 classes of ImageNet and then tested three different times using the pre-trained feature extractor from the three unsupervised methods each time. For testing we use a different set of images for each class which has not been used during training. The testing results presented in Table 3.1 confirm the performance superiority of the method *Chen et al.* (2020) propose. The features generated by SimCLR provide better discrimination capabilities as exhibited by the performance of the common classifier used in this test.

The SimCLR unsupervised training apparatus is used in this work to determine the weights of the feature extractor that produce the feature vectors used by the OSLS classifier. To learn these weights, ResNet18 uses unlabeled image data from the operational environment of interest.

3.3 Integration of Unsupervised Training with Low-Shot Classifier

The SimCLR method cannot by itself perform the classification task. In Chapter II OSLS Classifier (*Kasapis et al.*, 2020) was proposed which can serve as the top layer of a CNN with any already trained feature extractor. The OSLS classifier uses a specialized target matrix, a column-wise assembled loss function, a threshold probability (P_T), a Receiver Operating Characteristic (ROC) criterion and a randomized unlabeled sample training scheme for classifying

images which belong to a relevant class while distinguishing images that are irrelevant or unseen during training. A modest number of labeled relevant images and unlabeled irrelevant data are needed for training. As demonstrated in Chapter 2.6, the OSLS classifier is able to increase identification accuracy for relevant images, while retaining the ability to identify irrelevant and unseen images when compared to a fully supervised classifier that only uses relevant images during training.

In this Chapter, we attach the OSLS classifier to the end of ResNet18 that was trained using the SimCLR framework to create the IDLS capability. We use the ResNet18 architecture which includes five convolutional stages, because it represents a good trade-off between computational time (depth) and performance. Figure 3.1 outlines the training (blue vertical arrows) and operational (horizontal black arrows) processes in the IDLS framework.

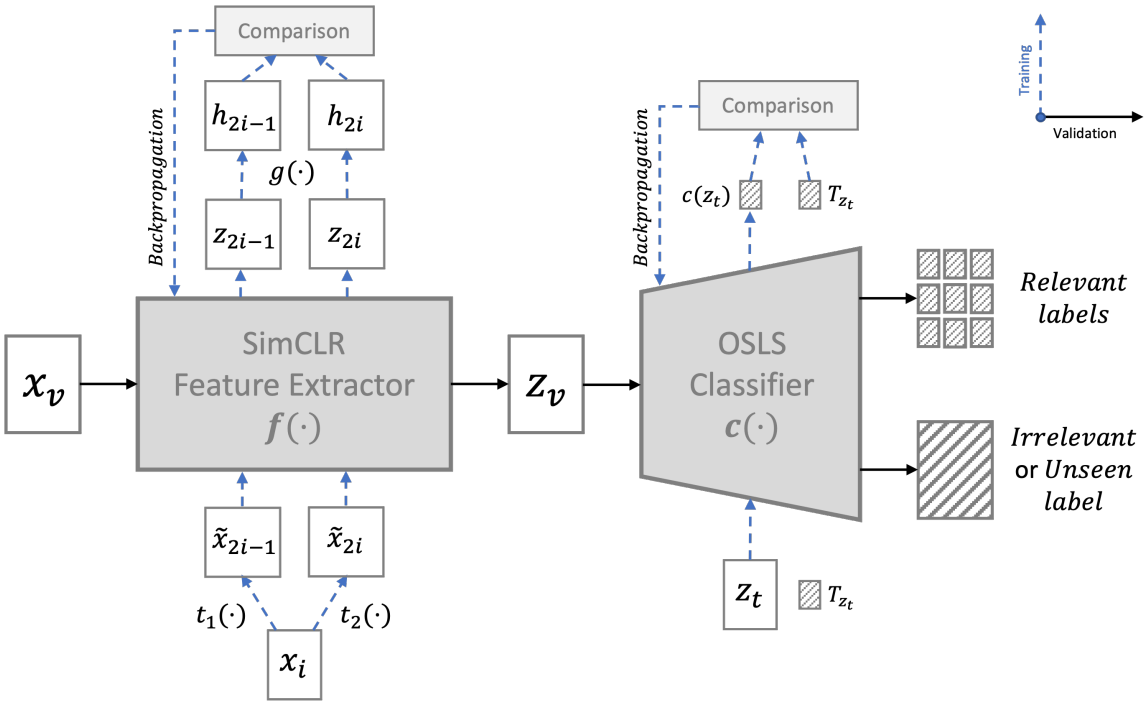

Figure 3.1: Schematic of the Instance Discrimination Low-Shot (IDLS) framework.

Figure 3.1 presents a schematic of the Instance Discrimination Low-Shot (IDLS) framework. We denote as x_t and x_v the training and evaluation images of the mixed dataset and as z_t and z_v the respective feature vectors. The horizontal axis information flow (black arrows) shows the two processes (feature extractor f and low-shot classifier c) that a picture x_v goes through in order to either get assigned a relevant class label or get classified as irrelevant. The vertical information flow diagrams (blue arrows) depict the training processes that f and c go through first.

We begin with training the ResNet18 feature extractor by randomly picking two data augmentation operators $t_1(\cdot)$ and $t_2(\cdot)$ which are applied to the input training image x_i in order to obtain two correlated views $\tilde{x}_{2i-1}$ and $\tilde{x}_{2i}$. The operators are picked from a group of augmentations which includes random cropping followed by resize back to the original size, random color distortions, and random Gaussian blur. The feature extractor that is to be trained then maps the two views to the vector space according to Equation 3.4:

$$z_{2i-1} = f(\tilde{x}_{2i-1}) \quad \text{and} \quad z_{2i} = f(\tilde{x}_{2i}) \tag{3.4}$$

The feature representations z_{2i-1} and z_{2i} which correspond to the two image views are passed through a small non-linear projection head g(·) to produce h_{2i-1} and h_{2i}. These two feature vector projections are used to calculate the contrastive loss using Equation 3.3. The loss will finally be backpropagated in order to update the weights of the Feature Extractor $f(\cdot)$. This process is repeated a defined number of epochs (100 in the results presented here) for each one of the 728,183 unlabeled images in the Mixed dataset using a learning rate schedule which includes warm restarts, linear warmup for the first 10 epochs, and cosine learning rate decay for the remaining 90 epochs (*Loshchilov and Hutter*, 2016). With the above specifications, the running time for every epoch

is approximately 50 minuntes on a machine with a standard 8 Gb GPU and 2.8 GHz CPU.

As soon as the training of the feature extractor is complete, the algorithm continues with training the OSLS classifier. The testing of the IDLS, the results it yields, and comparisons to alternative capabilities are outlined in the next Chapter.

3.4 Testing & Results

To demonstrate the classification capabilities of the IDLS we compare it to a supervised method which is identical to IDLS with the only difference that the feature extractor is trained in a supervised manner using the same total number of images but of different content than the ones we try to classify (first 500 classes of ImageNet1000). This "Supervised" approach represents the case where an autonomous combat vehicle uses a pre-trained feature extractor along with the OSLS for image recognition during operation. For a fair comparison, the same Mixed dataset samples are used for testing both the Supervised and IDLS approaches in Figure 3.2.

The box plot sides in Figure 3.2 represent the median of the lower and upper half of the different results set respectively. The lines extending from the boxes (whiskers) indicate the variability outside the upper and lower quartiles while the red line within the boxes represents the median of the entire spread. Lastly, the red crosses represent the accuracy of the outlier runs. The left panel titled "Supervised Feature Extractor & OSLS Classifier" is a visualization of the performance of the OSLS classifier when using features generated by a supervised method trained on images different than the ones we are trying to classify. On the other hand, the right panel titled "SimCLR Feature Extractor & OSLS Classifier (IDLS)" shows the performance of the OSLS classifier when

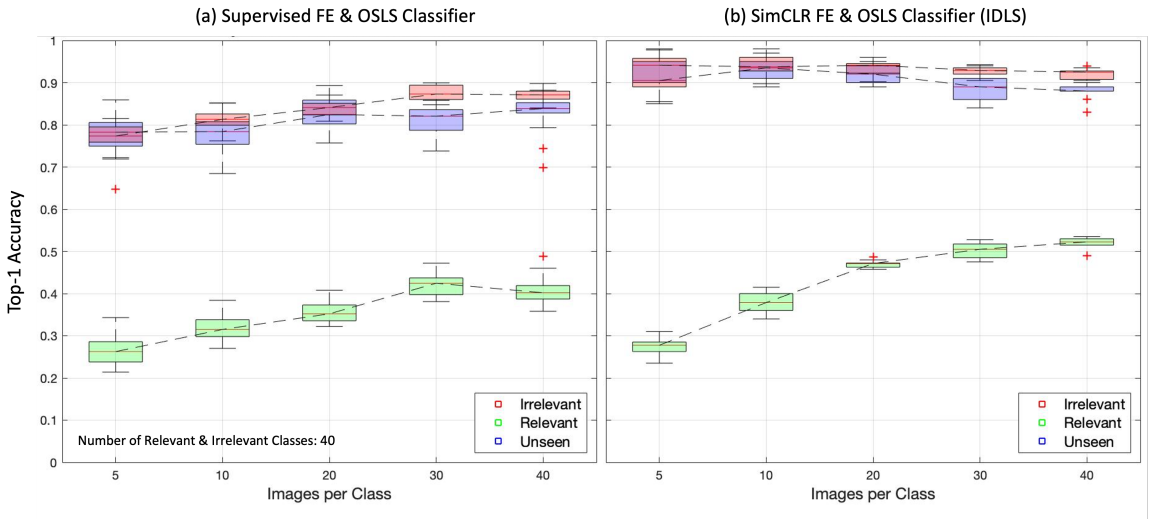

Figure 3.2: Box plot graphs for the cases where the OSLS classifier is connected to a) a supervised feature extractor and b) the SimCLR algorithm.

using features generated by a SimCLR approach trained on the same images we are trying to classify.

In order to demonstrate the statistical robustness of the proposed method, for every set of hyper-parameters (number of images and classes) the training of the OSLS classifier is performed 20 different times, each time with a different set of randomly chosen training and testing images. The four different quartiles of the 20 different runs are presented in the boxplots of Figure 3.2.

In each one of the 20 testing runs, the performance of the IDLS method is demonstrated using 10 images from each relevant class, 10 images from a random selection of irrelevant classes (equal number of classes to the relevant) and 10 images from a random selection of 10 irrelevant classes which are reserved as unseen and were not used during training. There is no overlap between training and testing images in any test. Figure 3.2 shows the Top-1 accuracy achieved by the IDLS method and the "Supervised" method for the relevant, irrelevant and unseen groups. For the relevant group accuracy is defined as the number of testing images that were assigned the correct relevant

label over the total number of relevant testing images. For the irrelevant and unseen groups, accuracy is the sum of all testing images which were correctly assigned in the irrelevant group, over the total number of irrelevant and unseen testing images respectively.

For both the IDLS and the Supervised approach, the feature extractor $f(\cdot)$ is trained only once and then it is used to provide us with the image feature vectors used by the OSLS classifier $c(\cdot)$. The different tests concern the OSLS classifier $c(\cdot)$ which is trained multiple times as seen in Figure 3.2, for a varied number of training images per relevant and irrelevant class. The first experiment uses for training 5 pictures from each one of the 40 classes, the second experiment uses 10 and for every other experiment we increase the number of images by 10 to up to 40. This means that the 10 Images/Class experiment uses 1,000 training images per epoch of which half (irrelevant) are unlabeled and different from the previous epoch, the second uses 2,000 images, the third 3,000 and so on.

As seen in Figure 3.2b, when trained on 5 images per class, the IDLS classifies relevant images with an accuracy of 0.28 ± 0.02 while if we use the features produced by the supervised approach (Figure 3.2a), the accuracy drops to 0.26 ± 0.03. For the same set of tests where only 200 annotated samples are used for training (5 relevant images for each one of the 40 classes), the IDLS method yields a mean irrelevant accuracy of 0.93 ± 0.04 compared to 0.78 ± 0.05 for the supervised method. An improvement in mean accuracy is also observed for the 10 image classes that are not seen during training, as the IDLS improves the unseen score by 0.14 ± 0.04. A similar pattern of improved accuracy is seen for every set of tests that has an increased number of training samples per class (10 to 40 images per class).

The relevant accuracy for both the IDLS increases when training the OSLS

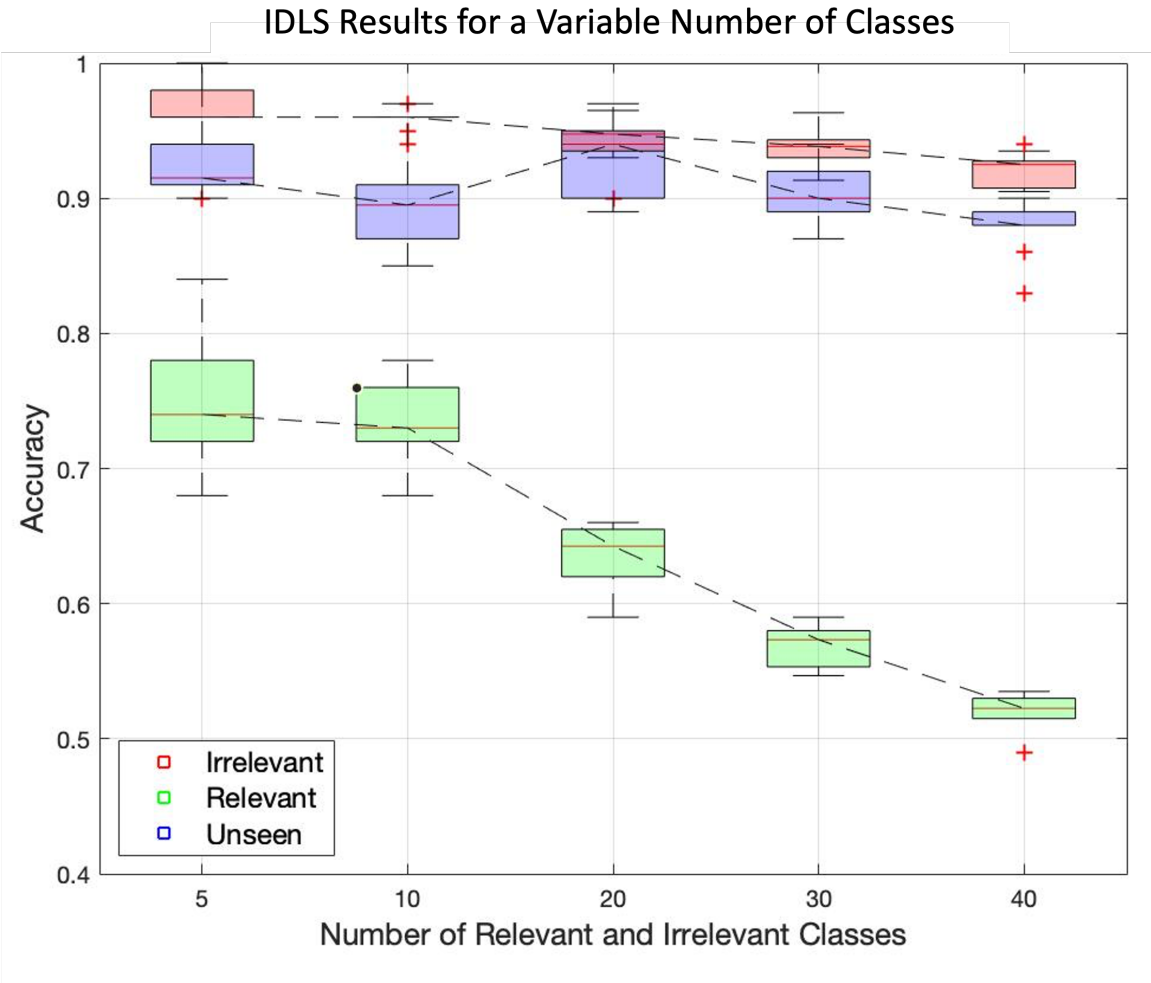

Figure 3.3: Boxplots showing the accuracy scores for the three different image categories when IDLS is trained on 40 images per class and a variable number of classes.

classifier on more images. The IDLS scores for both irrelevant and unseen are superior to the ones produced by the "Supervised" approach. All sets of IDLS results show low variability with the standard deviations varying from 0.01 to 0.09, 0.13 and 0.10 for the relevant, irrelevant and unseen groups respectively. The highest variability results are all observed at the lowest class and images case (5 classes with 5 samples each). This is expected since due to the absence of enough training samples the classifier does not generalize well enough to produce statistically invariable predictions that do not overfit on the training

set.

Although for both the Supervised and SimCLR feature extractors we observe an increasing accuracy trend which agrees with the increase in the numbers of images per class, an unexpectedly high accuracy value can be observed on the 30 images per class case for the Supervised feature extractor (in both Relevant and and Irrelevant accuracies). The mean value of the 20 different runs might be higher, but the bounds of the upper quartiles are not (the two 40 image per class outliers even surpass the accuracy of all the 30 image per class runs), therefore this abnormality is within the statistical error margin. As mentioned before, in every run a different selection of classes for all three groups is chosen, therefore this unusual behavior is attributed to individual runs that happen to select classes that are easy to distinguish from each other. An increase to the number of runs would smoothen out such abnormalities.

The aforementioned results demonstrate that the IDLS provides consistently superior performance compared to the "Supervised" approach. The reason for the classification superiority is that the feature extractor of the IDLS is trained on the same type of images (Mixed dataset) which are of interest to the training and operation of the low-shot classifier. It therefore exhibits better discriminating capabilities compared to the feature extractor trained in a supervised manner but with images which are different (ImageNet1000) compared to the ones of operational interest. It is worth mentioning that the feature extractor in the IDLS is trained using the mixed dataset which contains more than 90% irrelevant images. This imbalance is intended to resemble real life autonomous vehicle operation where the background environment (buildings and outdoors scenery) is more frequent than target objects.

Out of the experiments with 40 relevant classes, the one that interests us the most is the one that uses 40 images per class as it is the least label expensive

Table 3.2: IDLS results for a variable number of classes and images per class.

Top-1 Accuracy	5 Pictures per Class	10 Pictures per Class	20 Pictures per Class	30 Pictures per Class	40 Pictures per Class
			Relevant		
5 Classes	0.38 ± 0.09	0.49 ± 0.08	0.63 ± 0.06	0.70 ± 0.06	0.75 ± 0.05
10 Classes	0.30 ± 0.04	0.45 ± 0.04	0.62 ± 0.05	0.66 ± 0.05	0.73 ± 0.03
20 Classes	0.27 ± 0.02	0.42 ± 0.04	0.54 ± 0.03	0.60 ± 0.03	0.64 ± 0.03
30 Classes	0.29 ± 0.02	0.41 ± 0.03	0.51 ± 0.02	0.54 ± 0.03	0.57 ± 0.02
40 Classes	0.28 ± 0.02	0.38 ± 0.02	0.47 ± 0.01	0.49 ± 0.02	0.52 ± 0.01
			Irrelevant		
5 Classes	0.81 ± 0.13	0.94 ± 0.04	0.94 ± 0.03	0.94 ± 0.02	0.96 ± 0.03
10 Classes	0.91 ± 0.04	0.90 ± 0.08	0.97 ± 0.02	0.97 ± 0.02	0.96 ± 0.01
20 Classes	0.94 ± 0.03	0.96 ± 0.02	0.95 ± 0.02	0.94 ± 0.02	0.94 ± 0.02
30 Classes	0.95 ± 0.01	0.95 ± 0.02	0.95 ± 0.01	0.94 ± 0.01	0.94 ± 0.01
40 Classes	0.93 ± 0.04	0.94 ± 0.02	0.93 ± 0.01	0.93 ± 0.01	0.92 ± 0.01
			Unseen		
5 Classes	0.82 ± 0.10	0.91 ± 0.08	0.92 ± 0.03	0.92 ± 0.03	0.92 ± 0.02
10 Classes	0.91 ± 0.03	0.91 ± 0.04	0.96 ± 0.02	0.94 ± 0.03	0.90 ± 0.04
20 Classes	0.91 ± 0.05	0.95 ± 0.03	0.93 ± 0.03	0.91 ± 0.02	0.93 ± 0.03
30 Classes	0.96 ± 0.02	0.96 ± 0.02	0.93 ± 0.02	0.91 ± 0.03	0.90 ± 0.02
40 Classes	0.91 ± 0.05	0.93 ± 0.03	0.92 ± 0.02	0.89 ± 0.03	0.88 ± 0.03

IDLS example where all runs surpass the 0.5 relevant accuracy threshold. For the 40 images per class case, Figure 3.3 demonstrates how the mean accuracy for all groups changes when varying the number of relevant and irrelevant classes which are considered. As the number of classes increases, the classification problem gets tougher, therefore the behaviour observed in Figure 3.3 for the relevant group is expected. It is worth noting that although the relevant group shows a decrease in accuracy, the IDLS classifier does not lose much of its ability to classify well irrelevant and unseen samples when the number of classes increases.

For completeness, the full set of IDLS method results for all numbers of classes and images per class (variable) are presented in Table 3.2. For each different example, the accuracy mean and standard deviation of 10 different random tests is presented. All results in the table and the box plots are for the

optimal $r = 0.8$.

Table 3.3 presents the median value of the results that the 40 images per class experiment produced using the IDLS and the "Supervised" approach, along with the results of a regularly trained CNN. The images used to train the CNN are the same with the images used for training the feature extractor of the IDLS, but all of their labels are utilized in the supervised training. Therefore as CNN we denote a ResNet18 convolutional neural network trained on the entire Mixed Dataset with a final classification layer that puts every testing sample in one of the Mixed Dataset 100 classes. The OSLS classifier allows us to select the ROC ratio r in a way that it enhances the accuracy of the desired group of images (relevant or irrelevant), therefore the results for $r < 0.8$ and < 1 are presented.

Table 3.3: Accuracy results for different r values for the IDLS (40 classes with 40 images per class) and the two methods we compare it to.

	IDLS		Supervised		CNN
r limit	≤ 0.8	≤ 1	≤ 0.8	≤ 1	
Relevant	0.523	0.543	0.401	0.459	0.698
Irrelevant	0.924	0.790	0.870	0.697	0.705
Unseen	0.880	0.787	0.840	0.673	N/A

Using the training images, the OSLS algorithm learns using the ROC criterion a Threshold value for each class (extensive discussion is available in Chapter II). This Threshold value defines when an image will be categorized as relevant or irrelevant based on a criterion which allows the user to select the limit of the ratio r in Equation 3.5.

$$r = \frac{TR}{TR + FR} \tag{3.5}$$

Here, TR is the True Relevant and FR the False Relevant, the amount of pictures that got correctly and incorrectly classified as relevant (respectively)

within a class during the training of the OSLS. When selecting the limit of the ratio r_{ROC} to be less than one then the user asks the algorithm to recognise as many relevant images as it can at the expense of irrelevant accuracy. Any other limit value would give room for improvement of the irrelevant accuracy in the expense of the relevant.

In all experiments that include the OSLS classifier, one can observe a big difference in accuracy scores between relevant and irrelevant. This can be explained by the fact that the irrelevant images (scenery and buildings) are substantially different from the relevant classes (vehicles, humans and weapons) and it is therefore easier to determine that they do not belong to any of the relevant classes when using the OSLS classifier.

As a summary, the SimCLR is considered to be a state-of-the-art unsupervised training method for image feature extraction therefore combining it with the OSLS classifier yields significantly better results compared to the "Supervised" approach. More specifically, our experiments show that if we consider the 40 classes case in Figure 3.2 for the 5 image per class case the IDLS improves the mean accuracy by 2%, 15% and 14% for the relevant, irrelevant and unseen groups respectively when compared with the "Supervised" approach. For the 40 image per class case the IDLS improves the mean accuracy by 12%, 4% and 3% for the same three groups respectively compared to the "Supervised" approach. The accuracy scores achieved by the IDLS are comparable to the ones of a fully supervised CNN (relevant IDLS 54.3% vs. CNN 69.8% and irrelevant IDLS 79.0% vs. CNN 70.5 %).

In conclusion, it is demonstrated in this Chapter that by integrating an unsupervised learning feature extraction framework based on the Instance Discrimination method with an Open-Set Low-Shot classifier discussed in Chapter II, a new image recognition capability is created for specifically classifying rel-

evant objects and at the same time identifying as irrelevant or unseen, objects that do not belong to any known classes. It is important to underline that the performance of the IDLS is achieved by using only 0.27% of the image annotations (2,000 instead of 727,913) that a normal CNN uses. In the next and final Chapter, an unsupervised clustering algorithm is proposed which complements the IDLS capability in the effort to process image data from unstructured environments.

CHAPTER IV

Extended Variance Ratio Criterion for Unsupervised Clustering

4.1 Related Work

McQueen (1967) proposed an algorithm where datasets having n number of data points are being partitioned in k number of groups (clusters). This method, which was given the name "K-means", has since become the most popular clustering method in literature. Although a number of enhanced variations of K-means have been proposed, such as the one by *Hartigan and Wong* (1979), all of them are subject to major disadvantages: they computationally scale poorly, their membership search is prone to local minima, and most importantly the number of clusters K has to be defined by the user as an input.

Algorithms that tackle this last and very important shortcoming of K-means are called unsupervised. Several different algorithms have been proposed in literature where clustering can be performed without prior knowledge of the true number of clusters. External methods that can independently point to the correct number of clusters have also been devised. One such cluster number prediction method uses the VRC. In this Chapter we present the performance of VRC when applied to image data and the results from two other popular

unsupervised clustering methods: X-means proposed by *Pelleg et al.* (2000) and U-k-means proposed by *Sinaga and Yang* (2020). These two methods will provide baselines for comparing the performance of the new E-VRC method.

In their 2000 paper, *Pelleg et al.* (2000) proposed the X-means method which provides remedies for the three aforementioned shortcomings of K-means. The algorithm they propose -which is fast, statistically founded and outputs both the number of classes and their parameters- is based on algorithmic acceleration work and searches efficiently the cluster numbers and locations space in a way that the BIC and AIC indices are optimized.

X-means is a hierarchical clustering method as it starts with a user-defined lower-bound number of clusters K and continues to add cluster centroids where needed until a prescribed upper bound is reached. The cluster membership is defined by running the original K-means algorithm (operation described as Improve-Params in their work). The additional centroids appear when based on the BIC or AIC metric the algorithm decides that a cluster needs to split (described as Improve-Structure). The algorithm oscillates between the Improve-Params and Improve-Structure operations until the upper bound K is reached. During this process the set of centroids that achieves the best score is recorded and outputted as the final result. Different extensions of the X-means algorithm have been proposed, such as the non-hierarchical version proposed by *Ishioka et al.* (2020).

Pelleg and Moore (1999) test their algorithm in low dimensional datasets (2D and 3D) which include a large number of clusters (ranging between 50 and 250) and points (in the order of tens of thousands). Although X-means performs well compared to K-means in this specific setting, in this dissertation we prove that its performance deteriorates when the dataset dimension increases (image features) or the number of true clusters differs from the prescribed upper bound

of clusters.

A more recent, free of any initializations, unsupervised clustering schema for the k-means algorithm was proposed by *Sinaga and Yang* (2020). The unsupervised k-means (U-k-means) algorithm automatically finds the optimal number of clusters for a given set of data points, without relying in parameter selection by the user. It does this by optimizing a version of the k-means cost function augmented with two entropy terms. The U-k-means algorithm is unique because although it is based on K-means, it does not utilize the algorithm itself at any point. Instead, the algorithm's steps are derived by equating the Lagrangian of the augmented cost function to zero, creating a novel data treatment pipeline completely independent to the k-means algorithm.

Although unique in nature, the U-k-means algorithm performs well in a very strict data domain. The authors test their method only on low-dimensional data (2D and 3D) which are grouped in a relatively low number of true clusters (14 at most). Similar to X-means, the U-k-means method encounters difficulties when the algorithm is presented with high-dimensional data or when the algorithm is expected to predict a large number of clusters.

Intuition says that a high quality clustering model has simultaneously its cluster centroids far apart from each other but the cluster members close to the respective centroids. *Xie and Xu* (2020) applied the concept of well separated clusters in Bayesian analysis of mixture models, aiming to solve the problem of similarity and hence redundancy of components. Similarly, *Petralia et al.* (2012) propose a penalty on components placed close together, generating clusters from a repulsive process.

The idea of modeling in an attractive and simultaneously repulsive manner has been applied not only on the field of clustering but also in engineering applications such as the formation design of multi-agent systems (*Cheng et al.*, 2011).

In computer vision and image recognition, *Kenyon-Dean et al.* (2018) proposed a clustering-oriented representation learning using a attractive-repulsive loss function instead of using a standard categorical cross-entropy loss function. More specifically, the authors train convolutional neural networks (CNNs) using a loss function L comprised of two terms: an attractive term $L_{attract}$ which penalizes members of a cluster that are distant from the cluster centroid and a repulsive term $L_{repulse}$ which penalizes clusters that are close to each other.

A similar approach was followed by *Caliński and Harabasz* (1974) in their paper titled "A dendrite method for cluster analysis". The authors suggest a method for identifying clusters of points in a multidimensional Euclidean space along with devising an indicator for the optimal number of clusters. The Variance Ratio Criterion (VRC) indicator, similarly to the loss function by Kenyon, is comprised by two terms: the attractive term W and the repulsive term B. For every clustering model therefore W is given by:

$$W = \sum_{k=1}^{K} \sum_{i=1}^{n_k} ||X_i - \bar{X}_k||_2^2 \tag{4.1}$$

where K is the total number of clusters, n_k is the total number of members X_i in each cluster k and

$$\bar{X}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} X_i \tag{4.2}$$

is the centroid of each cluster k given the total number of cluster members n_k. Similarly, the repulsive term B is given by:

$$B = \sum_{k=1}^{K} n_K ||\bar{X}_k - \bar{X}||_2^2 \tag{4.3}$$

In theory, a good clustering arrangement means that the distance W is minimized while B is maximized. The two terms can be therefore utilized in a

variety of ways. In the case where we need to minimize a cost function, a good candidate equation would be $L = W - B$. For the VRC indicator, Calinski and Harabaz use an equation analogous to the F-statistic in univariate analysis, previously proposed by *Edwards and Cavalli-Sforza* (1965):

$$VRC = \frac{B(n-K)}{W(K-1)} \tag{4.4}$$

where n is the total number of points within the dataset being used and K is the number of clusters that is being tested in each iteration of the method. *Caliński and Harabasz* (1974) suggest that in order to find the optimal number of clusters, we test clustering arrangements with $K = 2, 3, ..., K_{max}$ and pick the number K where the VRC is maximum. In this manner the distance W is minimized while B is maximized.

In this Chapter we address the difficulties faced when the VRC is used in computer vision and image clustering. *Käster et al.* (2003) use the VRC to test the validity of the resulting clustering methods used for image retrieval techniques. This work is used as one of the three methods we compare the E-VRC results with in this Chapter. *Addagarla and Amalanathan* (2020) use it as part of their similar image recommender system and *Kermani et al.* (2015) as a method to calculate the optimal cluster number for their automatic color segmentation of breast infrared images apparatus.

The VRC is a popular index even in image recognition, but it comes with a very important shortcoming: a) it yields good results only when the data dimensionality is very low and b) it does not scale well when the true cluster number it tries to predict is higher. All of the aforementioned applications use either algorithms that produce features of low dimensionality or they reduce their feature dimensionality using methods such as the Principal Component Analysis. In this Chapter we propose an extension of the VCR that works for

higher dimensionality features, avoiding the loss of information that methods such as the PCA involve. Lastly, the cure we propose is proven to work for datasets that of higher true cluster numbers.

4.2 Extended Variance Ratio Criterion

The VRC index is a quantity describing the degree of inter-cluster separation (Equation 4.3) and intra-cluster homogeneity (Equation 4.1). In this research we have identified two cases where the VRC index loses its property to accurately point to the correct number of clusters: a) when the dimension of the feature space is big (i.e. when image features have hundreds of elements) and b) when the image content diversity is big (i.e. when datasets include more than 15 image classes). We propose the normalization of image features and the addition of an exponent term to Equation 4.4. These two simple but yet powerful changes to the VRC index calculation can remedy both aforementioned problems respectfully.

In his "Survey of Clustering Data Mining Techniques", *Berkhin* (2006) claims that vectors used in clustering algorithms are known to work effectively for dimensions below 16, while for dimensions above 20 their performance degrades to the level of sequential search (with some exceptions). Two solutions to the problem of high dimensionality have been proposed. The first is domain decomposition which is used when there is large data with many clusters and there is no actual dimension reduction (*McCallum et al.*, 2000). The second is attributes transformations which are simple functions of existent attributes with popular methods being the Principal Components Analysis (*Mardia*, 1988; *Vidal et al.*, 2016), Singular Value Decomposition (*Berry and Browne*, 2005; *Berry et al.*, 1995), Low-frequency Fourier Harmonics in conjunction with Parseval's theorem (*Agrawal et al.*, 1993), wavelets and many other kinds of transformations (*Keogh et al.*, 2001).

Both approaches are often problematic since they produce clusters with poor interpretability or lead to loss of information. The results discussed in Chapter 4.2 demonstrate that we do not need either as we can just normalize the image feature vectors using the Euclidean norm ($X_{norm} = X/|X|$) which yields the squared distance between two vectors where their lengths have been scaled to have unit norm. This normalization succeeds with helping the VRC method solve multi-dimensional problems as the direction of the feature vectors are necessary for the solution of the problem but the magnitude (dimension) is not. We therefore solve the first shortcoming of the VRC method by virtually reducing dimensionality when only interested into the direction, but without loosing information.

The second weakness of the VRC method is that it does not scale for datasets with higher number of classes. This is because of the nature of Equation 4.4 which was devised with having low numbers of K in mind. For datasets that do not contain many data points and should only be split in a handful of clusters, the B and W multipliers are relatively balanced. This does not hold true when we increase the desired number of clusters. By introducing an exponent term $p < 1$ in the k term of the B multiplier as seen in Equation 4.5, we restore this balance by diminishing the effect of a large K value on reducing the numerator.

$$VRC_E = \frac{B(n - K^p)}{W(K - 1)} \tag{4.5}$$

The Euclidean normalization of the features in combination with the proposed extended (by adding the exponent) VRC equation are the two key aspects of the new E-VRC algorithm we introduce. The range of robust values for the exponent p is determined and their effects are discussed in Chapter 4.3, but for the tests performed using the E-VRC algorithm, its value remains constant and is $p = 0.3$. For any given image dataset, with $K_{start} = 2$ and K_{stop} any

very large number (cannot be infinity as the loop would never be terminated) we propose Algorithm A.4 in Appendix A.

Given a dataset of images, the E-VRC algorithm determines the number of clusters in the dataset and at the same time it places the images into clusters. The determination of the upper limit of clusters (K_{stop}) is not of high importance like in other hierarchical clustering algorithms, as the performance of the E-VRC parameter in Figure 1 shows that a build in termination criterion (Algorithm A.2) can be used.

We test the E-VRC algorithm using two popular image datasets: the Validation ImageNet1000 (*Deng et al.*, 2009) and the Caltech256 (*Griffin et al.*, 2007). We pick the first 50 classes from the Caltech256 dataset, as they are diverse enough, whereas we select to use every 10 classes from ImageNet1000 as the dataset classes are ordered based on the nature of the image contents (classes list begins with wildlife content and continues to general imagery objects as seen in Table B.1.3 of Appendix B). Additionally, to be consistent with the rest of the chapters too, we test the E-VRC algorithm on the Mixed dataset, acknowledging that it will be the most difficult group of pictures to cluster.

The features X used in Equations 4.1 and 4.3 are produced using a ResNet34 feature extractor which has been trained on the entirety of ImageNet1000 training dataset. ResNet34 is a popular neural network with a limited number of trainable parameters compared to deeper networks and produces 512 dimension image vector representations. For simplicity and replicability of our results, we use the PyTorch[1] pretrained ResNet34 model for this application.

Figure 1 shows the variation of the E-VRC index for values of $K = [2, 100]$ ($K_{start} = 2$ and $K_{stop} = 100$ on Algorithm A.4), although the algorithm is terminated earlier, as soon as no further increase in the E-VRC index is detected.

[1]https://pytorch.org/vision/stable/models.html

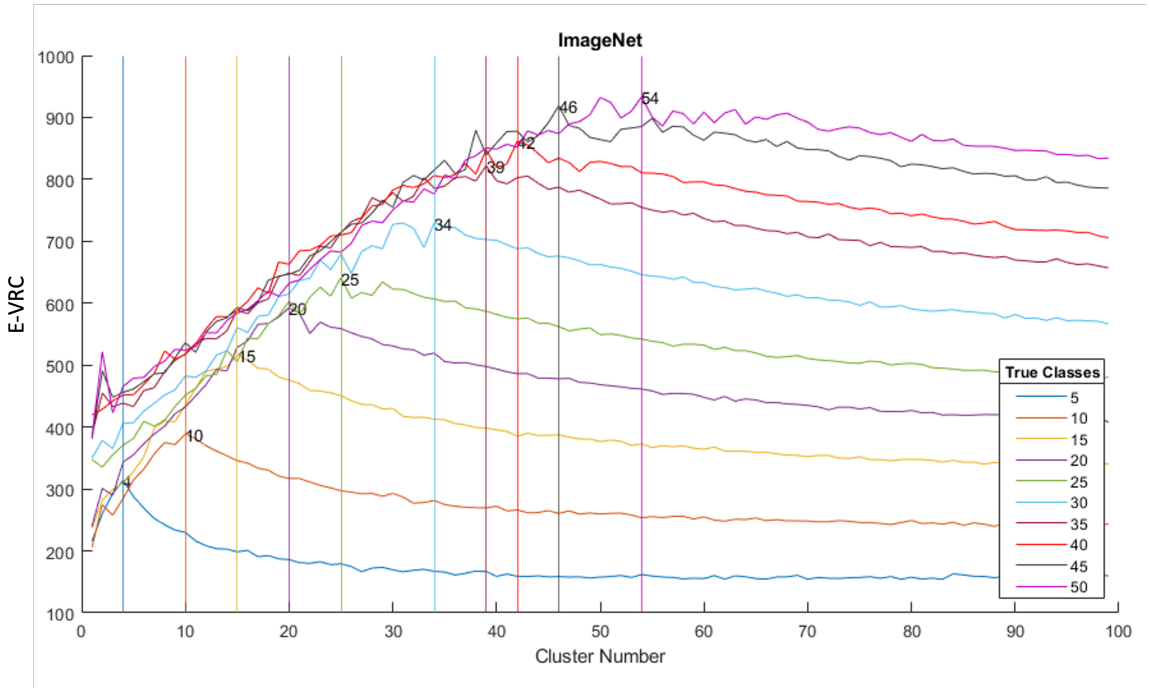

Figure 4.1: Variation of E-VRC index for the ten different true cluster number cases.

The vertical lines signify the maximum E-VRC value for each case, the respective cluster number (x-axis) is the one selected by our algorithm. The plot shows the results for the ImageNet dataset features produced using the PyTorch ResNet34 pretrained model. By plotting the E-VRC index over a much longer number of clusters, the consistent decay of the E-VRC index after reaching its maximum value is demonstrated. Ten different cases are being explored, for ten different true cluster (classes) values: $K_{true} = [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]$. Each class includes 50 images that are randomly selected from a pool of more than 1200 images. The vertical lines indicate the maximum E-VRC value for each K_{true} case.

One can observe that the E-VRC index predicts with very high accuracy the true number of clusters in each case. For the $K_{true} = [10, 15, 20, 25]$ cases, the algorithm predicts the exact number of true clusters, while for the $K_{true} =$

$[5, 30, 35, 40, 45, 50]$ cases it predicts a number of clusters very close to the true one. It is important to note that the results presented in Figure 4.1 present the best possible E-VRC scenario as the quality of the ImageNet (validation) features used is very high due to the fact that the pretrained ResNet34 model used for feature extraction is trained using the same ImageNet (training) dataset.

Throughout this research we have been using the Normalized Mutual Information (NMI) as the measure for determining the quality of clustering. The NMI is a normalization of the Mutual Information score which scales clustering results between 0 for clustering arrangements that present no mutual information and 1 for perfect correlation. We pick this metric as it is normalized, allowing us to compare clustering arrangements of different number of clusters and pictures. The NMI is a popular external measure (*Kvålseth*, 2017; *Estévez et al.*, 2009; *McDaid et al.*, 2011), meaning that the true cluster labels should be known for its calculation, which is information that we do have in our case studies. Along with the comparison between predicted and true number of clusters (ΔK) in each run, the NMI is used for evaluating the performance of the E-VRC.

Figure 4.2 shows in green the NMI values of the clustering generated by the E-VRC method. We compare the quality of the E-VRC method by clustering the same datasets with the three main unsupervised clustering methods discussed in Chapter 4.1, namely: the normal VRC where features have been reduced using the PCA (orange), the X-means clustering algorithm (pink) and the U-k-means method (blue). Each of the respective lines tracks the mean value for each case. The red line presents the NMI accuracy obtained using the K-means method with the true number of clusters as an input. The red line presents the best possible clustering performance. The closest to the red line an unsupervised clustering method is, the better it performs. For each K_{true} case, ten different runs are performed using 50 different images from each class to create the

testing dataset in order to quantify the uncertainty.

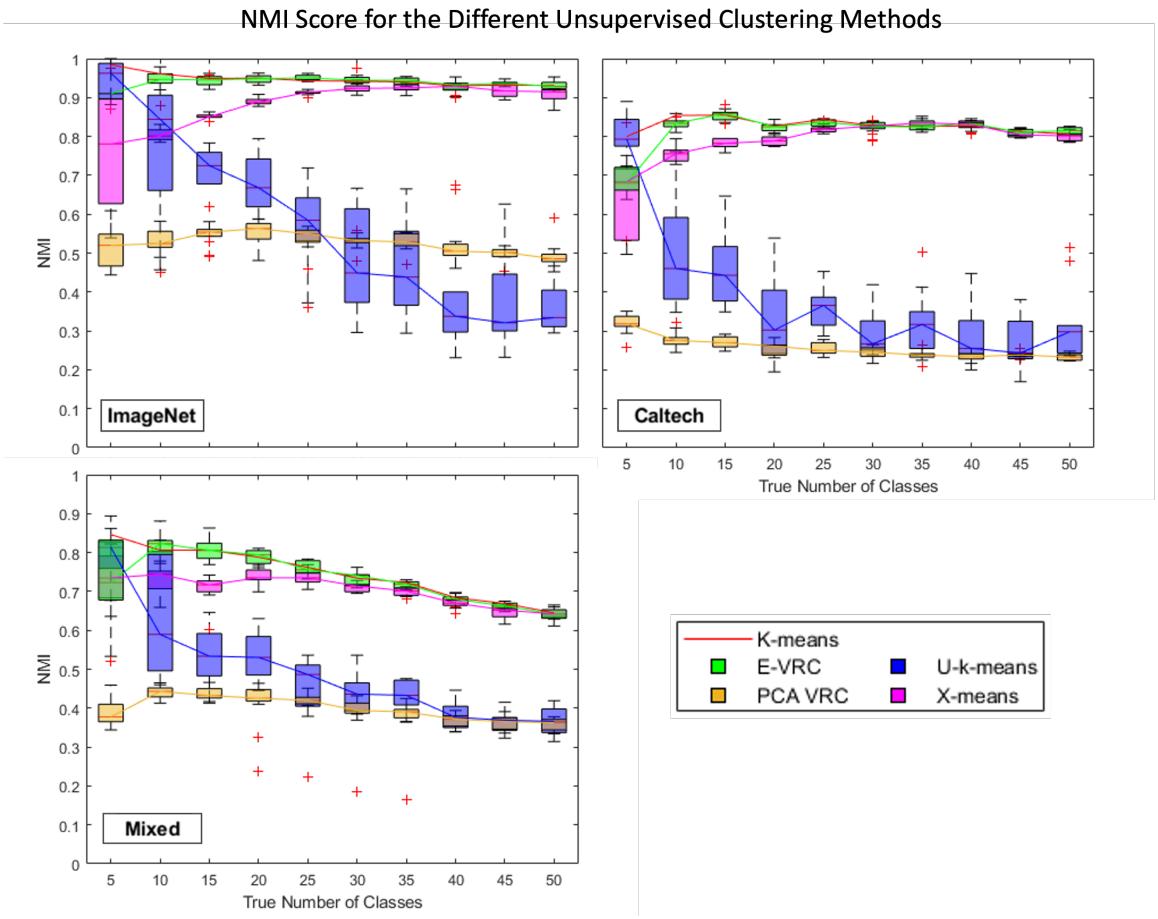

Figure 4.2: Box plots for the 4 different unsupervised methods' NMI results.

As seen in Figure 4.2, the E-VRC method outperforms in almost all cases (except for $K_{true} = 5$) the other three unsupervised clustering methods while also producing results similar to the ones produced by a K-means method that uses the true number of classes as an input. The K-means method is considered to be the best case scenario as the number of classes is not calculated by the algorithm itself, but is given as an input. The ImageNet NMI results for all four methods are superior to the respective Caltech and Mixed dataset results, as all methods are dependent on the quality of the features and the feature extractor used here was trained using the ImageNet dataset.

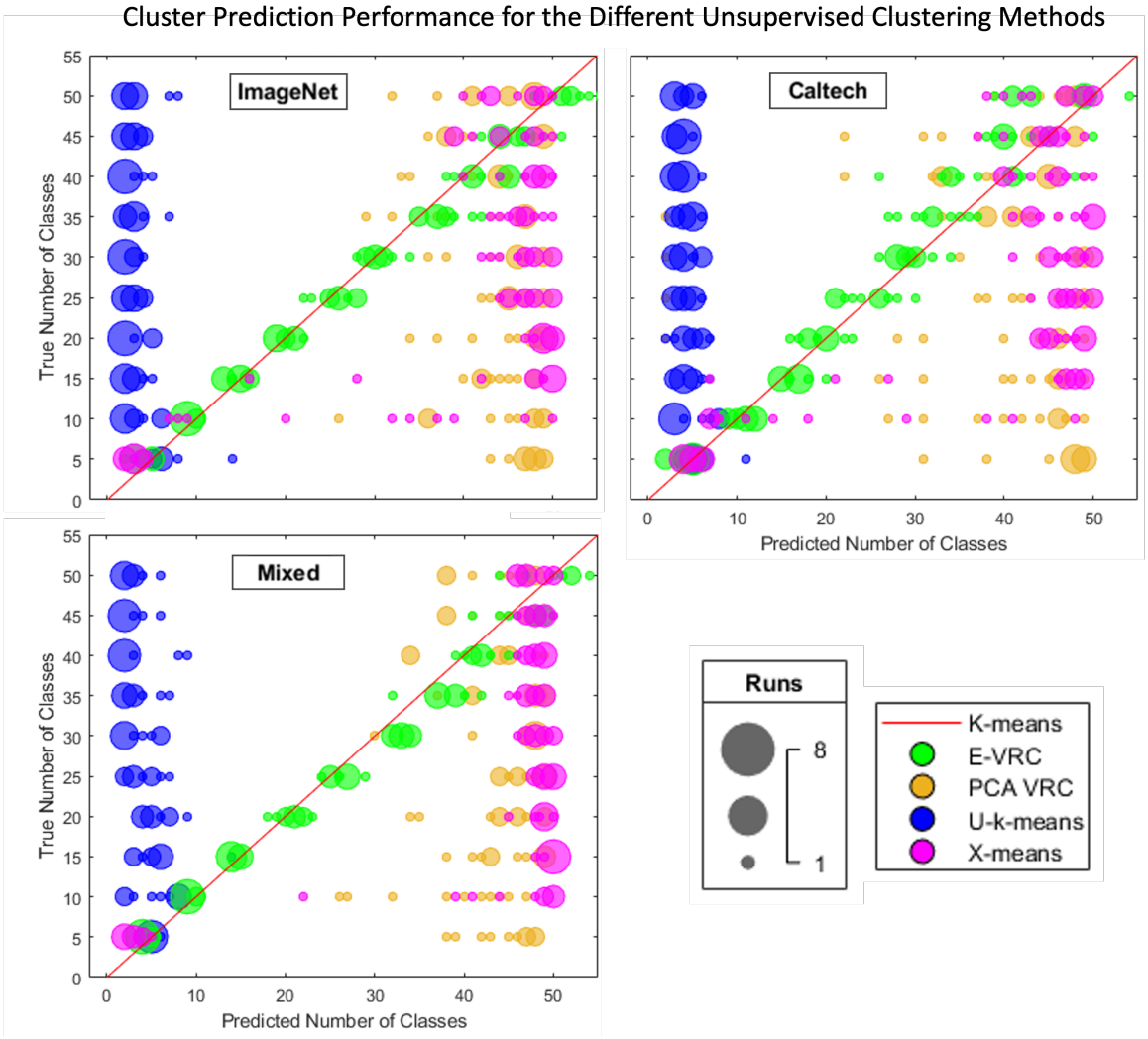

Figure 4.3: Bubble plots for the four different unsupervised methods' cluster prediction.

Each one of the boxes in Figure 4.2 is a depiction of 10 different NMI values as for each clustering method, 10 different datasets are analyzed for $K_{true} = [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]$. Each dataset contains a different set of images for the same classes. The box plot sides represent the median of the lower and upper half of the different results set respectively. The lines extending from the boxes (whiskers) indicate the variability outside the upper and lower quartiles while the red line within the boxes represents the median of the entire

spread. The red crosses represent the accuracy of the outlier runs. The median values of each box are tracked by the respective same color lines. The red line tracks the median value of the K-means approach where the true number of clusters is known a priori and comprises the best possible clustering.

Regardless the true number of clusters, all methods are evaluated without changing the input parameters of the algorithm. Both X-means and U-k-means can potentially yield better results if in each case the algorithm input parameters are adjusted based on the knowledge we have about the dataset (namely the number of classes the algorithm is presented with and the number of pictures in each class). The PCA VRC method performs poorely due to the fact that significant information loss occurs when reducing a 512 long feature map to a 3D vector. The success of the E-VRC method lies on the fact that it scores well without the need to calibrate the algorithm based on the number of classes or images per class. It is important to note that the NMI spread the E-VRC method yields is significantly lower than the rest of the methods (especially U-k-means).

The box plots in Figure 4.2 help us also visualize the robustness of each method by providing us with information about the spread of the different runs' results. It is important to note how the E-VRC results have a very small standard deviation (small boxes), which means that regardless the differences in imagery in every run, the method produces equally good results compared to the U-k-means method where the standard deviation is very large, meaning that the method proposed by *Sinaga and Yang* (2020) yields significantly different results in every run. All different methods (except the U-k-means), regardless the dataset tested on, have a larger spread for the cases where K_{true} is small, and this can be attributed to the fact that when incorrectly estimating the number of clusters -even by a little- in a low K_{true} case, the effect in the cluster quality is larger.

Table 4.1: Mean NMI values of the ten different random runs (ImageNet).

K	**E-VRC**	**PCA VRC**	**U-k-means**	**X-means**	**K-means**
5	0.916 ± 0.035	0.516 ± 0.052	0.948 ± 0.045	0.757 ± 0.152	0.984 ± 0.008
10	0.948 ± 0.017	0.529 ± 0.040	0.773 ± 0.175	0.810 ± 0.028	0.961 ± 0.024
15	0.943 ± 0.013	0.550 ± 0.038	0.715 ± 0.121	0.851 ± 0.009	0.949 ± 0.018
20	0.947 ± 0.010	0.553 ± 0.031	0.684 ± 0.071	0.890 ± 0.009	0.950 ± 0.011
25	0.950 ± 0.007	0.538 ± 0.032	0.565 ± 0.121	0.912 ± 0.006	0.944 ± 0.012
30	0.947 ± 0.012	0.530 ± 0.021	0.481 ± 0.137	0.924 ± 0.012	0.942 ± 0.015
35	0.943 ± 0.009	0.528 ± 0.025	0.457 ± 0.122	0.925 ± 0.010	0.942 ± 0.010
40	0.929 ± 0.015	0.505 ± 0.020	0.392 ± 0.153	0.923 ± 0.012	0.930 ± 0.010
45	0.936 ± 0.006	0.498 ± 0.018	0.364 ± 0.171	0.918 ± 0.018	0.932 ± 0.011
50	0.931 ± 0.011	0.485 ± 0.016	0.374 ± 0.093	0.907 ± 0.018	0.931 ± 0.011

The bubble plots in Figure 4.3 present the respective number of clusters predicted in each run performed in Figure 4.2. In an ideal case (red line), we would desire the true number of clusters to be equal to the predicted one. Every K_{true} case is run ten times, using a different set of images from the same classes each time. Instead of single points (scatter plot) we use bubbles to present the cases where different runs predicted the same K.

As mentioned in Chapter 4.1, U-k-means performs well when presented with low K_{true} datasets, but fails to scale for higher class variability datasets. On the contrary, X-means is a superior method for large datasets that include numerous data points and classes that tends to overestimate K when presented with smaller datasets. This can be remedied by changing algorithmic parameters such as the minimum cluster membership number or the maximum predicted K allowed but parameter optimisation based on prior dataset knowledge defeats the purpose of an unsupervised method. In these results the X-means performs well for a larger number of K_{true} only because the upper bound of clusters was set to be 50. As it can be observed, the X-means prediction for the number of clusters tend to gravitate towards this upper bound. The PCA VRC class prediction seems to be highly random due to the low feature quality in contrast

Table 4.2: Mean NMI values of the ten different random runs (Caltech)

K	**E-VRC**	**PCA VRC**	**U-k-means**	**X-means**	**K-means**
5	0.685 ± 0.078	0.319 ± 0.028	0.803 ± 0.060	0.640 ± 0.090	0.800 ± 0.081
10	0.834 ± 0.014	0.279 ± 0.022	0.508 ± 0.155	0.763 ± 0.036	0.854 ± 0.017
15	0.855 ± 0.012	0.271 ± 0.015	0.456 ± 0.100	0.794 ± 0.073	0.855 ± 0.014
20	0.823 ± 0.012	0.255 ± 0.018	0.332 ± 0.120	0.790 ± 0.013	0.827 ± 0.020
25	0.836 ± 0.009	0.253 ± 0.016	0.362 ± 0.058	0.819 ± 0.010	0.844 ± 0.012
30	0.824 ± 0.015	0.244 ± 0.016	0.291 ± 0.060	0.824 ± 0.015	0.828 ± 0.017
35	0.829 ± 0.014	0.237 ± 0.015	0.327 ± 0.082	0.834 ± 0.009	0.826 ± 0.012
40	0.829 ± 0.014	0.235 ± 0.012	0.288 ± 0.081	0.831 ± 0.009	0.827 ± 0.014
45	0.811 ± 0.009	0.239 ± 0.008	0.275 ± 0.070	0.806 ± 0.008	0.813 ± 0.012
50	0.813 ± 0.012	0.233 ± 0.008	0.318 ± 0.100	0.801 ± 0.012	0.806 ± 0.011

to the E-VRC method which tends to predict correctly -with a small margin of error- the number of clusters.

Tables 4.1, 4.2 and 4.3 summarize the results by presenting the mean and standard deviation of each box in Figure 4.2. The success of a clustering method should be evaluated in two ways: based on the NMI score they yield and based on the cluster number prediction. It is important to highlight this as there is a big disparity between the results yielded by different datasets. For example, the NMI results for all methods when tested on ImageNet are superior to the Caltech and even more to the Mixed dataset ones for the reasons mentioned before (feature extraction trained on ImageNet).

Lets first examine the E-VRC results in Tables 4.1 and 4.3. The proposed E-VRC method when faced with ImageNet pictures yields NMI values that range from a high 0.950 ± 0.007 to a low 0.916 ± 0.035 while when it encounters Mixed dataset images yields NMI values that range from a high 0.822 ± 0.033 to a much lower 0.643 ± 0.012. This difference in clustering quality cannot be attributed to the E-VRC method but rather to the K-means clustering itself. As seen in Figure 4.3, in both cases (ImageNet and Mixed) the E-VRC method predicts equally well the correct number of clusters, or at least gets close to it

Table 4.3: Mean NMI values of the ten different random runs (Mixed)

K	**E-VRC**	**PCA VRC**	**U-k-means**	**X-means**	**K-means**
5	0.728 ± 0.124	0.389 ± 0.038	0.079 ± 0.057	0.720 ± 0.090	0.846 ± 0.011
10	0.822 ± 0.033	0.441 ± 0.017	0.626 ± 0.151	0.732 ± 0.040	0.806 ± 0.014
15	0.805 ± 0.028	0.437 ± 0.018	0.539 ± 0.073	0.706 ± 0.039	0.806 ± 0.013
20	0.790 ± 0.019	0.414 ± 0.064	0.518 ± 0.089	0.738 ± 0.020	0.788 ± 0.020
25	0.760 ± 0.018	0.404 ± 0.065	0.466 ± 0.059	0.737 ± 0.019	0.762 ± 0.009
30	0.734 ± 0.017	0.380 ± 0.071	0.439 ± 0.056	0.711 ± 0.012	0.734 ± 0.019
35	0.715 ± 0.014	0.369 ± 0.074	0.431 ± 0.039	0.700 ± 0.010	0.723 ± 0.017
40	0.679 ± 0.011	0.369 ± 0.018	0.380 ± 0.036	0.670 ± 0.014	0.684 ± 0.022
45	0.663 ± 0.009	0.363 ± 0.019	0.366 ± 0.030	0.648 ± 0.015	0.669 ± 0.022
50	0.643 ± 0.012	0.360 ± 0.016	0.367 ± 0.038	0.640 ± 0.016	0.645 ± 0.043

(green bubbles reside along the red $x = y$ line). A difference in NMI accuracy is observed simply because with lower quality features even K-means clustering (with K known) has a hard time performing a good clustering.

Similar observations can be done for the rest of the unsupervised clustering methods we compare our algorithm's results with. Let's take as an example the second most reliable method, the X-means clustering. As seen in Figure 4.3, for all datasets the cluster prediction that the X-means method yields is pretty much the same regardless the different K_{true}. Now, if we consider the NMI results in Tables 4.1-4.3 we can see that the quality of clustering deteriorates, with best possible scores (upper limits) starting at 0.925 ± 0.010, 0.834 ± 0.009 and 0.738 ± 0.020 for ImageNet, Caltech and Mixed datasets respectively. This doesn't mean that the X-means is worse at predicting the number of clusters in the Mixed dataset case, it simply means that the clustering task itself is much more challenging. We should therefore be careful when evaluating the different clustering methods as the NMI score does not always provide us with the full picture. Once again, one should always look both at the NMI score and the cluster number prediction simultaneously before evaluating the quality of a clustering method.

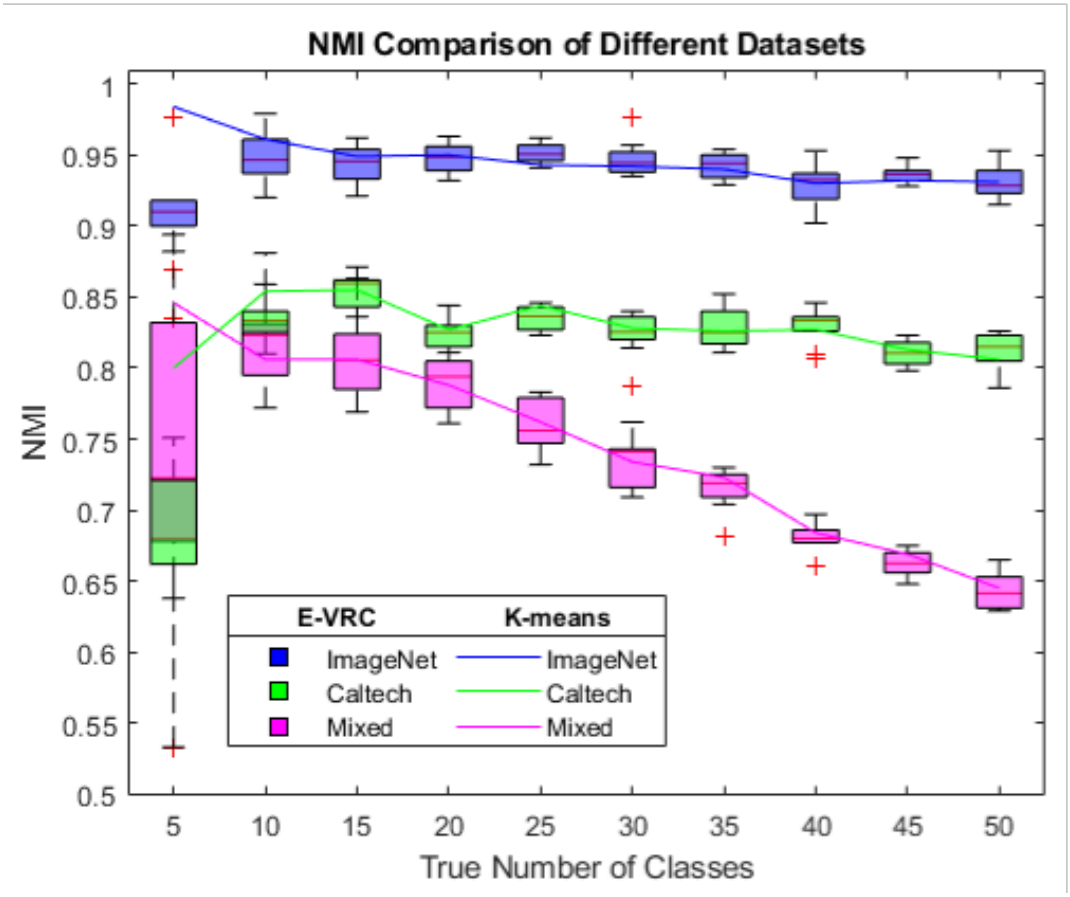

Figure 4.4: Cumulative box plots for the three datasets NMI results.

Despite the difficulty to define what consists of a "good quality" clustering, the superiority of the E-VRC method is made clear by considering both measures of performance. When it comes to cluster number prediction, as seen in Figure 4.3 the E-VRC method for all datasets is the only unsupervised clustering approach that consistently yields results that are close to the true number of clusters K_{true}. On the other hand, Tables 4.1-4.3 show us that even the best possible results for the PCA VRC, the U-k-means and the X-means clustering methods are in most cases inferior to the worst possible E-VRC method result.

Overall, the results that the E-VRC method yields are promising when compared not only to the rest unsupervised methods, but also to the K-means clustering which -because we know K_{true}- we assume it to be the upper limit.

For the ImageNet dataset (easiest task), our new clustering method has a median NMI value of 0.939 ± 0.014 across all ten different K_{true}, which is very comparable to a median value of 0.947 ± 0.013 for the K-means approach. Similarly, for the Caltech dataset (medium difficulty task), E-VRC scores a median NMI of 0.813 ± 0.019 compared to 0.828 ± 0.019 for K-means and for the Mixed dataset (hardest task) 0.734 ± 0.029 compared to 0.746 ± 0.019 respectively. The median differences between E-VRC and K-means are 0.008, 0.015 and 0.003 for ImageNet, Caltech and Mixed respectively, which can be thought to be insignificant if we take in consideration that no information about the true number of clusters K_{true} is utilized by the E-VRC.

Figure 4.4 presents the results in Figure 4.2 but only for the E-VRC clustering. The figure highlights the importance of a well trained feature set when it comes to clustering. Here we observe that the results for ImageNet and Caltech are not only superior in terms of the mean NMI values, but they are also consistent along the true number of classes axis compared to the Mixed dataset where there is a significant deterioration in clustering accuracy for the higher number of classes. Yet again, regardless the deterioration in NMI accuracy for the more difficult to cluster datasets, it is important to note that the K-means results follow the same trend. Once again, this proves that the deterioration in accuracy is not only due to the dysfunction of the E-VRC method but also because there is a limit on how well a high dimensional feature set can be clustered.

Similarly, Figure 4.5 presents the respective cluster prediction for each run depicted in Figure 4.4. Conversely to the plots presented in Figure 4.3 where bubbles occupy almost the entirety of the plots spaces, it can be observed that all the E-VRC method's bubbles are adjacent to the red $y = x$ line. It is noteworthy that for the lower number of clusters, the class prediction is slightly more

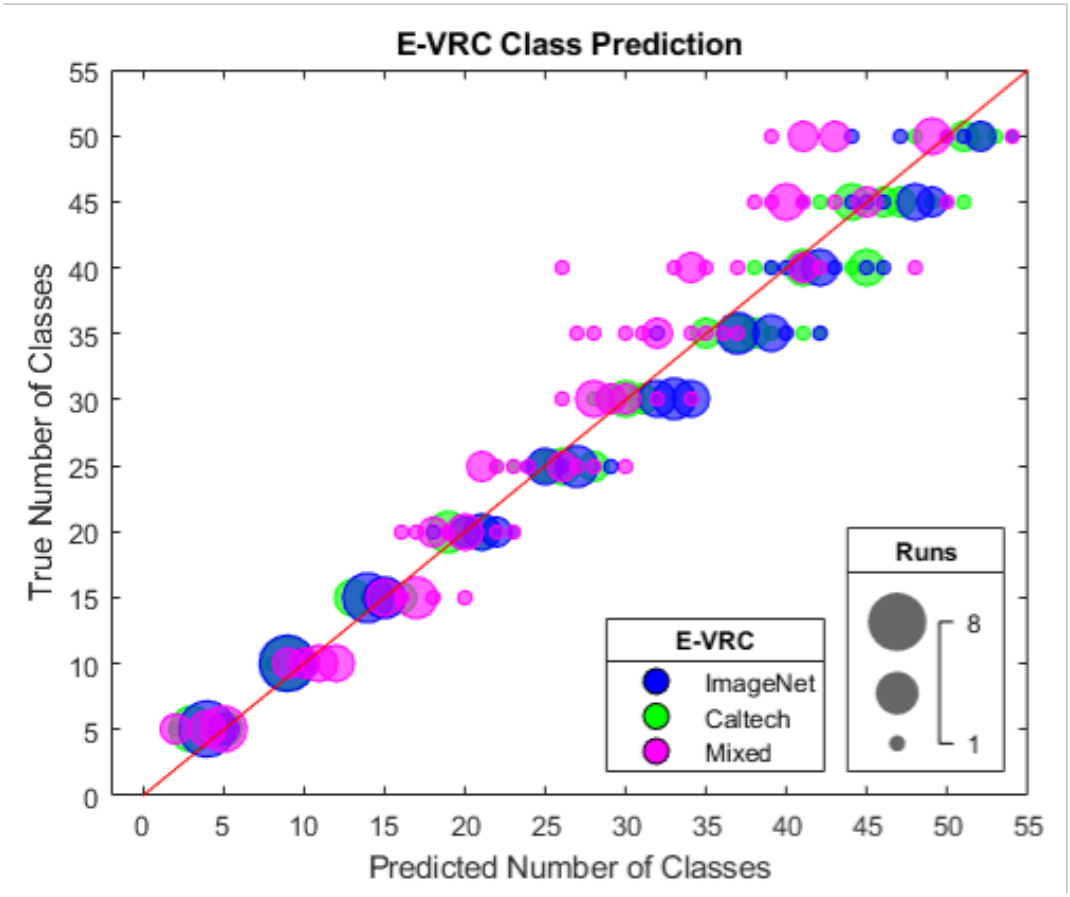

Figure 4.5: Cumulative predicted number of clusters bubble plots for the three datasets.

consistent compared to the higher number of classes where we can observe in Figure 4.5 that the bubbles spread further away from the perfect prediction line, regardless the dataset. The deterioration in accuracy seen for the Mixed datasets' higher number of classes cases (pink) seen in Figure 4.4 is also evident in Figure 4.5: when $K_{true} \geq 35$ the E-VRC algorithm tends to slightly underestimate the number of clusters.

Before moving to the next Sections we need to note that all the results presented in Figures 4.2 and 4.3 and in Tables 4.1-4.3 are products of an experimental algorithm where all classes have the same, balanced amount of images. In the Sections that follow, we extend our analysis to clustering setups where randomness is introduced in the number of images per class, proving the ro-

bustness of the E-VRC method.

4.3 Robustness

As of now we have demonstrated that the E-VRC algorithm can cluster image datasets regardless their size (the tests performed range from 5 to 50 classes with each containing 50 images). The true clusters number predictive capability is shown to be superior compared to a number of other unsupervised clustering methods.

This superiority of the E-VRC method lies on the fact that, regardless the size of the dataset, the algorithm does not rely on any user input at all. This means that both the normalization performed after the feature extraction and the selection of the p exponent are universal and they do not need to be adjusted based on the number of classes or images that it tries to cluster.

To demonstrate the robustness of the method, we prove that the same exponent p can be used for different datasets and that values between 0.1 and 0.5 provide stable results. This shows that the algorithm does not need to be tuned for every different case it is presented with.

In all the case studies performed in Chapter 4.2, the value of the p exponent is set to be equal to 0.3. Figure 4.6 explores the cluster number predictive capability of E-VRC for p values ranging from 0.1 to 1. As observed, for the ImageNet dataset exponent values of $p = [0.1, 0.6]$ and for the Caltech dataset values of $p = [0.1, 0.5]$, the E-VRC method succeeds on predicting the correct number of classes with a relatively small margin of error.

Figure 4.6, similarly to Figure 4.3 shows the number of clusters predicted by the E-VRC method compared to the true classes values (K_{true}). In an ideal case (red line), we would desire the true number of clusters to be equal to the predicted one. Similarly to the Figure 4.3 runs, every K_{true} case is run

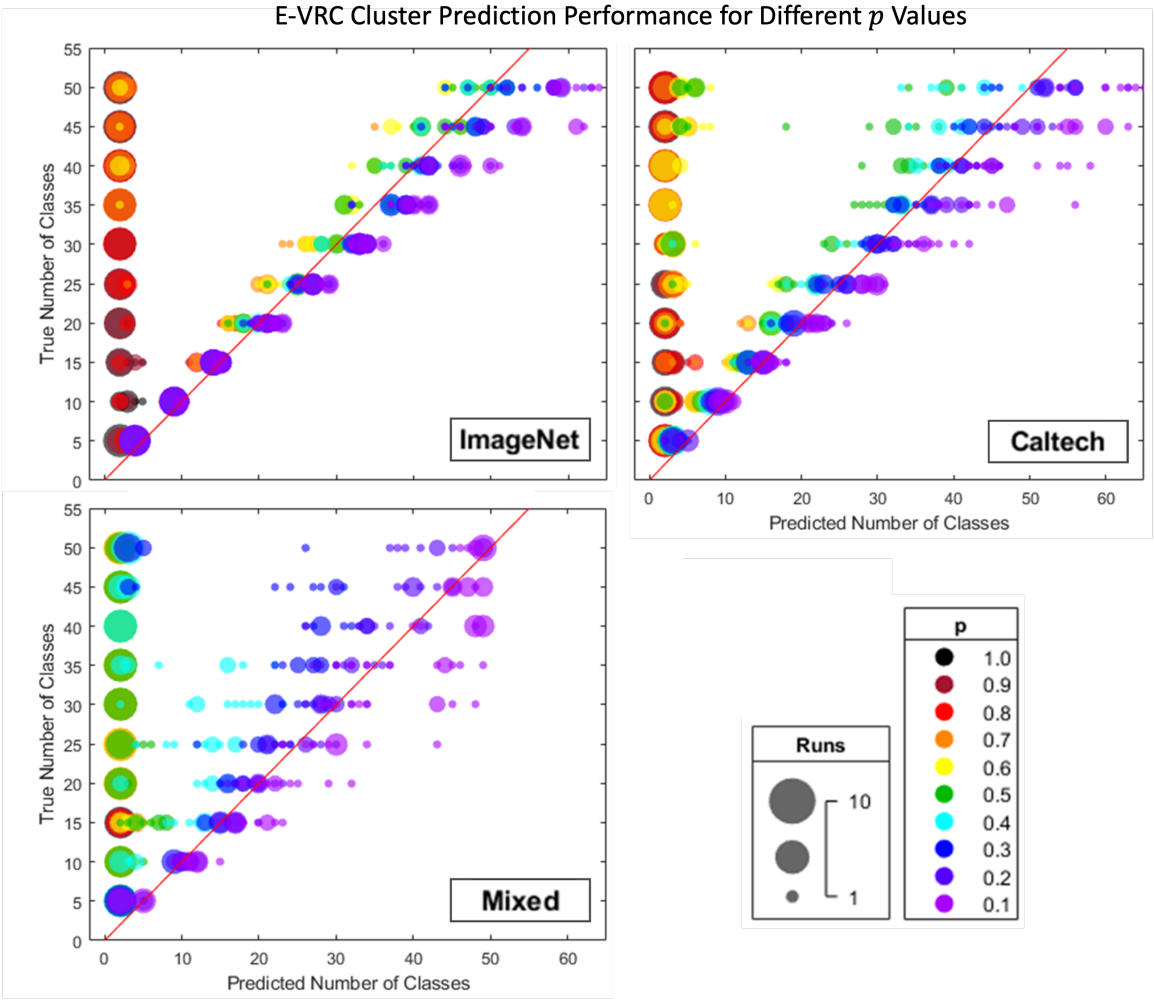

Figure 4.6: Bubble plots presenting the cluster prediction ability of the E-VRC method for ten different exponent p values.

ten times, using a different set of images from the same classes each time. Instead of single points we use bubbles to present the cases where different runs predicted the same K.

The deterioration of clustering quality -where quality here is measured based only on the $K_{pred} - K_{true}$ difference- is evident when testing the exponent variation effect in our algorithm. Note how in Figure 4.6 the cluster predictions of the ImageNet case are much tighter around the red $y = x$ correct prediction line compared to the Caltech and Mixed dataset cases. How loosely the predictions

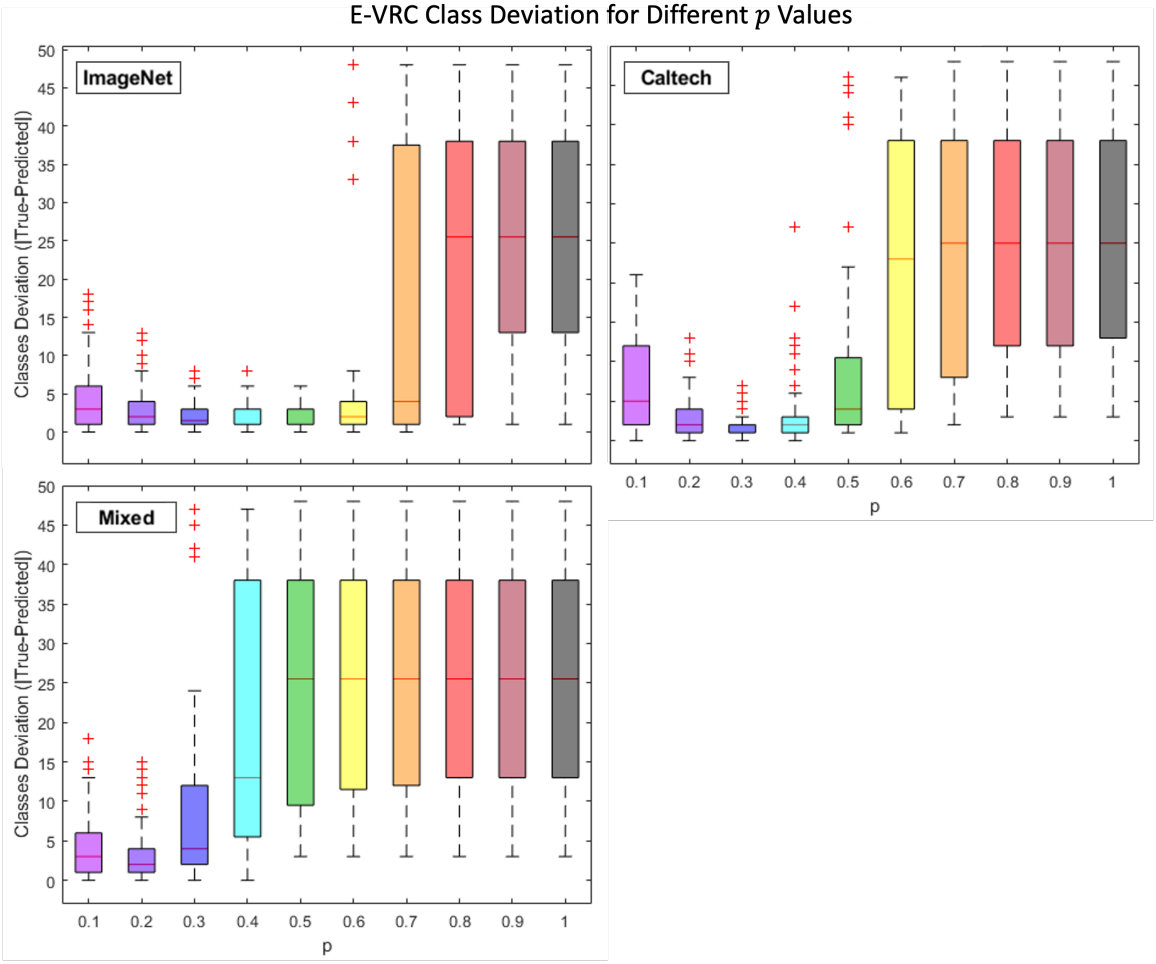

Figure 4.7: Box plots for the predicted classes deviation ΔK.

are scattered around the correct prediction line is not the only indicator of deterioration due to the dataset clustering difficulty.

It is worth noting the number of p values (that correspond to different colors in Figure 4.6) that are scattered around the $y = x$ line and do not lie close to the $x = 0$ horizontal line. For example, on the Mixed dataset bubble plot there are only three p values ($p = [0.1, 0.2, 0.3]$) that really converge towards the red correct prediction line, with a fourth $p = 0.4$ value yielding somewhat reliable cluster predictions only where $K_{true} \leq 25$. On the other hand, the Caltech and ImageNet plots have five or even six p values respectively that yield quality

results. This observation can be made even more clearly when studying Figure 4.7.

Figure 4.7 shows the cluster prediction error for the different values of p. For all K_{true} cases (10 runs each with randomly selected images from each class), we calculate the predicted classes deviation $\Delta K = |K_{pred} - K_{true}|$ and present the variation using box plots. Therefore, each exponent p box depicts the ΔK values for 100 different runs: ten different cases for each $K_{true} = [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]$ where each case is run 10 different times using a different set of images for the same classes. Consequently, this means that if a box in Figure 4.7 is "long" (i.e. has a high standard deviation), this specific p value predicts K_{pred} values that are very far from the true cluster number K_{true}.

For the ImageNet and Caltech datasets, the exponent that yields consistently and for all cases the lowest ΔK variability is $p = 0.3$, therefore it is the one suggested for use based on the conducted study. Yet again, the importance of having high quality features is present, as the Caltech image features -produced by a CNN trained on a different dataset- present a slightly lesser range of exponents yielding good results, and higher ΔK variability. This is particularly evident when clustering Mixed dataset pictures, as we observe that there is only three p values that make the E-VRC method work (more correctly, the range of p values between 0.1 and 0.3).

In conclusion, when the E-VRC method is faced with datasets that are easy to classify, the range of p values is large. For example, in the case of ImageNet there are six different values, a range of $p \in [0.10.6]$- that make the algorithm work. For the harder to classify (as discussed before) Mixed dataset there are five values, a range of $p \in [0.10.5]$- that make the algorithm work while for the hardest dataset (Mixed) of all, there are only three values. Despite the dependency on the quality of the features, the E-VRC method has a range of p

values that can make it work for even the toughest to classify datasets, making it a method that can be used -without any initializations- on any image dataset.

For all the different cases studied above, although variability was introduced in the number of classes, the number of images within each class was constant (50 images per class). Such an experimental setting does not reflect reality. It is most certain that in the case the E-VRC method is used in an operational setting, the different image classes would not contain the same number of images.

In order to demonstrate that the E-VRC method could be used in an operational clustering setting, in this Chapter we introduce a certain amount of image imbalance between the different classes, showing that the method can cluster datasets that are prone to randomness.

The tests presented in Figure 4-8 and 4.9 are similar to the ones of Chapter 4.2 where for all datasets we explore the E-VRC capabilities for ten different number of classes (5 to 50 with increments of 5) where each K_{true} case is run 10 different times using different random images from each class. The difference here is that instead of each class containing 50 images (constant), the number of images per class varies from 10 to 50. Here, in every K_{true} case 20% of the classes include 10 images per class; 20% of the classes have 20 images per class; 20% of the classes have 30 images per class; 20% of classes have 40 images per class; and the rest 20% of classes 50 images per class as in the normal examples. As an example, the case where $K_{true} = 5$, the first and fifth class contain 5 images, the second and fourth 10 and the third 20. Similarly, for the case where $K_{true} = 10$, the first, fifth, sixth and tenth class contain 5 images, the second, fourth, seventh and ninth classes contain 10 images, the third and eighth classes contain 20 images and so goes on for $K_{true} \geq 15$.

As observed in Figure 4.8, when faced with an imbalanced dataset the E-

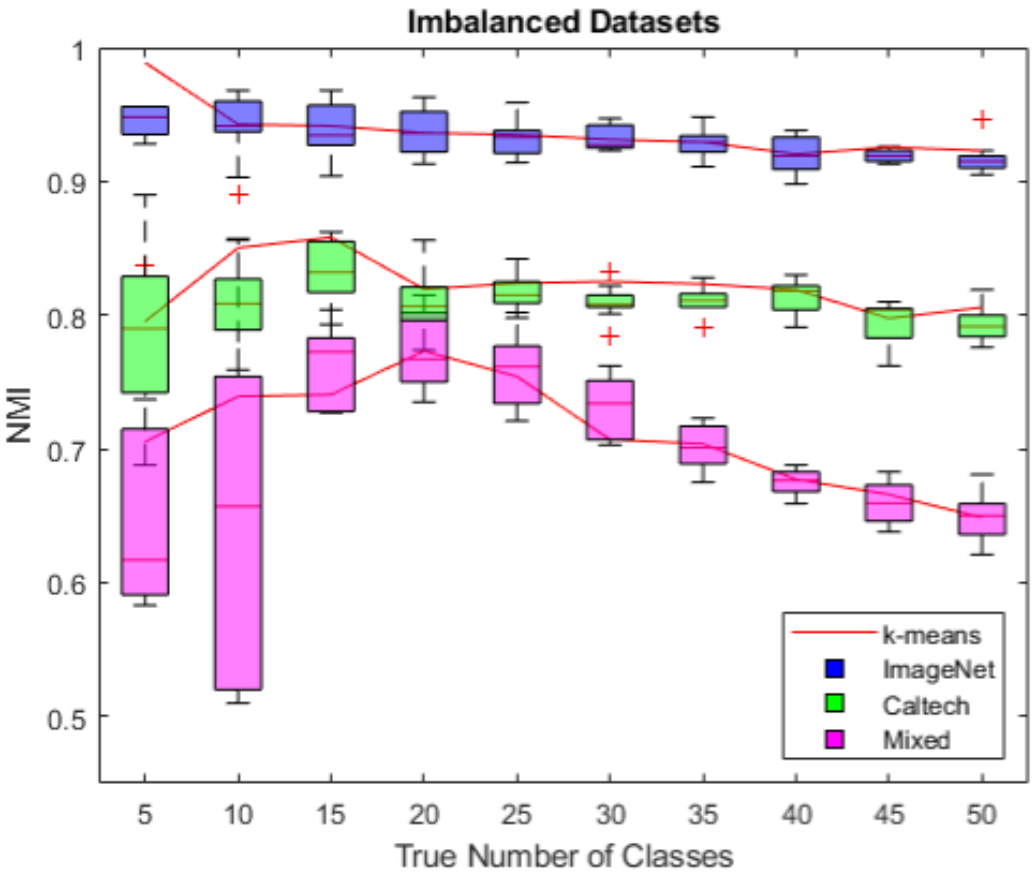

Figure 4.8: Box plots for the E-VRC NMI results when the three datasets have an imbalanced number of images per class.

VRC method produces results that are similar to the ones presented in Figure 4.4. Although a very small drop of the NMI index can be detected, this behavior is expected as the algorithm is presented with a more challenging clustering problem. A similar drop of the NMI index is observed when the K-means clustering method is used, where knowledge of the true number of clusters is available. Based on the class number prediction bubble plot seen in Figure 4.9, the NMI drop for the E-VRC method can be attributed to the fact that when faced with highly imbalanced datasets, the E-VRC method tends to slightly underestimate the correct number of classes.

Note that for the imbalanced case study too, both the true number of clusters

Table 4.4: Mean NMI values for imbalanced datasets (ImageNet)

K	**E-VRC**	**K-means**	**Difference**
5	0.936 ± 0.036	0.989 ± 0.017	-0.053
10	0.943 ± 0.022	0.943 ± 0.028	$+0.000$
15	0.939 ± 0.021	0.942 ± 0.019	-0.003
20	0.937 ± 0.018	0.936 ± 0.018	$+0.001$
25	0.932 ± 0.013	0.935 ± 0.014	-0.003
30	0.932 ± 0.010	0.931 ± 0.013	$+0.001$
35	0.928 ± 0.011	0.929 ± 0.007	-0.001
40	0.919 ± 0.013	0.920 ± 0.009	-0.001
45	0.919 ± 0.004	0.926 ± 0.010	-0.007
50	0.917 ± 0.012	0.923 ± 0.009	-0.006

K_{true} and the quality of the image features (dataset) have a significant effect on the NMI accuracy. As seen in Tables 4.4-4.6, the mean E-VRC NMI score across all tests for the ImageNet dataset is 0.930 ± 0.016 (with an upper limit of 0.943 ± 0.022 and a lower limit of 0.917 ± 0.012), for the Caltech dataset is 0.808 ± 0.024 (with an upper limit of 0.834 ± 0.024 and a lower limit of 0.786 ± 0.067) and for the Mixed dataset is 0.697 ± 0.046. If these results are compared to the ones presented in Tables 4.1 - 4.3 we see that the mean NMI score across all tests decreases only by 0.009 for ImageNet, 0.005 for Caltech and 0.037 for Mixed when these datasets have an imbalanced number of images per class. Yet again, if we examine the K-means results which is though to be the highest scoring method, we see that this drop in NMI accuracy can be attributed to the fact that the clustering problem is in general much more difficult.

For the well established K-means clustering method, we see in Tables 4.4-4.6 that the mean NMI values across all 100 different tests are 0.937 ± 0.014 (with an upper limit of 0.989 ± 0.017 and a lower limit of 0.920 ± 0.009) for ImageNet, 0.822 ± 0.023 (with an upper limit of 0.858 ± 0.023 and a lower limit of 0.795 ± 0.074) for Caltech and 0.697 ± 0.046 (with an upper limit of 0.773 ± 0.024 and a lower limit of 0.649 ± 0.016) for Mixed, when imbalance in the number

Table 4.5: Mean NMI values for imbalanced datasets (Caltech)

K	**E-VRC**	**K-means**	**Difference**
5	0.786 ± 0.067	0.795 ± 0.074	-0.009
10	0.813 ± 0.039	0.850 ± 0.028	-0.037
15	0.834 ± 0.024	0.858 ± 0.023	-0.024
20	0.809 ± 0.023	0.819 ± 0.020	-0.010
25	0.816 ± 0.013	0.824 ± 0.011	-0.008
30	0.810 ± 0.013	0.825 ± 0.015	-0.015
35	0.810 ± 0.013	0.823 ± 0.016	-0.013
40	0.815 ± 0.013	0.819 ± 0.014	-0.004
45	0.794 ± 0.018	0.798 ± 0.015	-0.004
50	0.792 ± 0.012	0.806 ± 0.011	-0.014

of images per class is introduced. If these results are compared to the ones presented in Tables 4.1 - 4.3 we notice that the mean NMI score across all tests decreases for the K-means clustering method too by 0.010, 0.006 and 0.049 for the ImageNet, Caltech and Mixed datasets too. What is interesting when looking into these numbers is that the K-means clustering method is affected more by the introduction of imbalance within classes than the E-VRC method, proving once again the robustness of the proposed method.

Tables 4.4-4.6 also note the differences between the proposed E-VRC method and the very well established K-means clustering method. For the ImageNet dataset, the mean difference between the two clustering methods ($NMI_{E-VRC} - NMI_{K-means}$), across all different K_{true} tests, is -0.007, with the biggest difference (-0.053) being that of the $K_{true} = 5$ case and the E-VRC method scoring higher than the K-means method twice for the $K_{true} = 20$ and $K_{true} = 30$ cases. For the Caltech dataset (Table 4.5), the mean difference between the two clustering methods, across all different K_{true} tests, is -0.014, double that of the ImageNet dataset runs (Table 4.4), with the biggest difference (-0.037) being that of the $K_{true} = 10$ case. Here, the E-VRC algorithm does not score equally well or better than the K-means method in any case. Lastly, for the tough-

Table 4.6: Mean NMI values for imbalanced datasets (Mixed)

K	**E-VRC**	**K-means**	**Difference**
5	0.639 ± 0.062	0.705 ± 0.050	-0.066
10	0.656 ± 0.129	0.739 ± 0.052	-0.083
15	0.723 ± 0.138	0.741 ± 0.032	-0.018
20	0.774 ± 0.027	0.773 ± 0.024	$+0.001$
25	0.760 ± 0.026	0.754 ± 0.029	$+0.006$
30	0.731 ± 0.022	0.707 ± 0.022	$+0.024$
35	0.702 ± 0.016	0.704 ± 0.019	-0.002
40	0.675 ± 0.010	0.677 ± 0.008	-0.002
45	0.660 ± 0.014	0.666 ± 0.014	-0.006
50	0.648 ± 0.019	0.649 ± 0.016	-0.001

est dataset to cluster (Mixed), the mean difference between the two clustering methods, across all different K_{true} tests, is -0.016, with the biggest difference (-0.083) being that of the $K_{true} = 10$ case and the E-VRC method scoring higher than the K-means method three times for the $K_{true} = [20, 25, 30]$ cases.

There are a couple of noteworthy details to comment on when studying the imbalanced datasets results. In most cases, the K-means method scores significantly better at the lower cluster cases ($K_{true} = [5, 10, 15]$) with differences ranging from -0.003 to a high of -0.083. These low K_{true} differences are dependent to the feature quality. We observe that when testing the ImageNet dataset the E-VRC method can even score equally well as the K-means (Table 4.4, $K_{true} = 10$ case) equivalent, whereas on the other hand, when testing the Mixed dataset, the K-means clustering outperforms the E-VRC method for the low K_{true} cases by a significant amount (-0.066, -0.083 and -0.018). This discrepancy can be probably attributed again to the fact that when the E-VRC fails to predict the exactly correct number of classes for low K_{true} cases, the effect is much larger than when it does so for larger K_{true} values. For high numbers of k_{true} we observe that the K-means clustering approach scores consistently better than the E-VRC, but by a very small amount.s

It is important to also pay attention to the cases where the E-VRC scores equally well or better than the K-means clustering (6 out of 30 cases). Out of these 6 cases, there is only one (Table 4.6, $K_{true} = 30$ case) where the difference benefiting the proposed method ($+0.024$) is significant. For the rest of the cases the difference ranges from 0.000 to $+0.006$ (the standard deviation of the difference is 0.021), with all the positive differences being observed where $K_{true} = [20, 25, 30]$ (middle range). For the cases where $K_{true} \geq 35$, the difference between the two methods discussed is small (close to the standard deviation) and benefits always the K-means clustering. These observations show the K_{true} range where the E-VRC method is mostly superior at. It is important to note again that although in the majority of the cases K-means beats (slightly) in terms of NMI accuracy the proposed method, the fact that the E-VRC algorithm can sometimes even cluster images better than a well-established supervised clustering method like the K-means, shows the quality and importance of the proposed method.

Finally, Figure 4.9 shows bubble plots that present the cluster prediction ability of the E-VRC method when faced with a highly imbalanced dataset. Similarly to all the aforementioned case studies, every K_{true} case is run ten times, using a different set of images from the same classes each time. Instead of single points we use bubbles to present the cases where different runs predicted the same K. As seen in the bottom right legend, based on its size, a bubble can be either a single run or either nine different runs where the E-VRC method predicted the same number of clusters. The same idea applies to all bubble plots in Figures 4.9, 4.6, 4.5 and 4.3.

As discussed above, the results when the three datasets we test have an imbalance introduced within the number of images per class are very similar to the experimental ones that do not (Chapter 4.2). Despite this fact, other than the

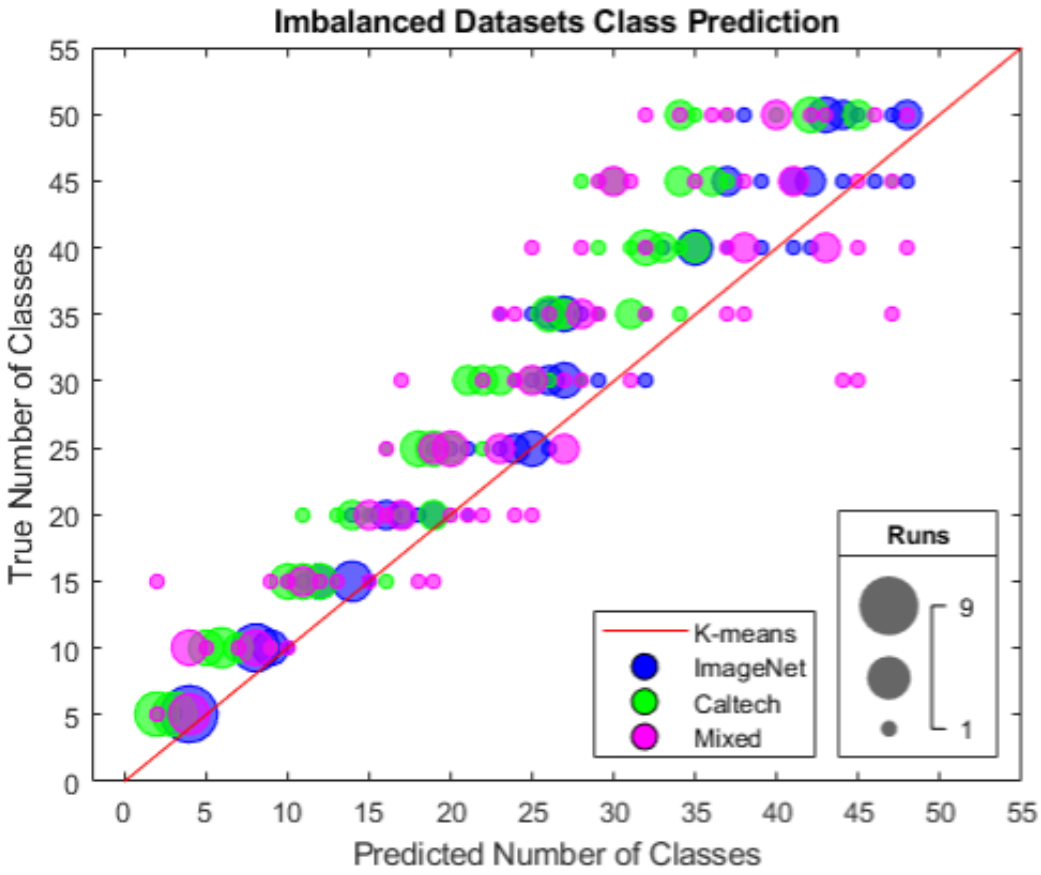

Figure 4.9: Bubble plots for the E-VRC cluster number prediction when the three datasets have an imbalanced number of images per class.

quality of the features, the small deterioration of the results can be attributed to the fact that in the imbalanced case the E-VRC method tends to underestimate the true number of clusters. As seen in Figure 4.9, not only the bubbles that represent the cluster prediction fall mostly over the red $y = x$ line, but also their spread is less tight around it compared to Figure 4.5, which suggests that the E-VRC method has a somewhat harder time estimating the exactly correct class.

Overall, the E-VRC algorithm not only seems to be able to produce quality clustering assignments for image data in pseudo-operational settings that have imbalance and randomness introduced to them, but can also produce competi-

tive results compared to other popular supervised (K-means) and unsupervised methods (PCA VRC, X-means, U-k-means). In the next and final Chapter, we will be discussing the benefits of all the aforementioned algorithms along with how the E-VRC method could be coupled with the OSLS classifier in order to process image data from unstructured environments.

CHAPTER V

Conclusion

5.1 Contributions

Before entering operational situations that threaten their uninterrupted operation, autonomous vehicles need an understanding of the environment which can be used in place of a human to provide context to situations where a user intervention is needed. More specifically, in reconnaissance applications, a capability is needed where objects of interest -such as adversary targets- are reliably distinguished from objects of no relevance.

During development and experimentation, these autonomous vehicles capture -through sensors mounted on ground vehicle systems- large amounts of raw, unlabeled image data. Although a modest amount of labeled examples for the targets might be available to use during the training of a convolutional neural network, labels for the irrelevant objects might be scarce or even not possible to obtain. An effort can be made to label a small portion of it, but not all. In such a case, the use of a supervised training approach for image recognition is impossible as the full data is unlabeled or the amount of annotated data is not adequate.

In this dissertation, we tackle this problem by integrating an image feature extraction framework based on Instance Discrimination (IDLS) with an Open-

Set Low-Shot classifier (OSLS), this dissertation presents a module that can identify targeted objects while at the same time recognize when candidate images do not belong to any one of the target classes, both in a very data-inexpensive way. Unseen images are also correctly placed in the category of irrelevant images.

The first step of this apparatus is training the OSLS Classifier using a modest number of labeled images from the relevant classes and unlabeled irrelevant images. A partially labeled target matrix is used for developing an analytically differentiated loss function for training the classifier. At each training epoch a random selection of irrelevant images is introduced. During the training an ROC approach is used for determining a threshold score value for each relevant class. The latter is used for providing a balanced performance between classifying relevant samples and identifying irrelevant images. During testing, this information is used for determining when a candidate image is either relevant, irrelevant or even unseen during training. The OSLS Classifier performs better compared to baseline classifying approaches, is able to handle the classification of many more classes compared to similar open-set approaches in the visual recognition literature and is able to demonstrate sufficient balance with high accuracy in classifying relevant images, identifying irrelevant images and correctly recognizing unseen images.

Instead of producing features for the OSLS Classifier using a pre-trained CNN, we use an unsupervised training method on the raw data and then train the low-shot classifier using the small portion of annotated data. The main advantage of this approach is that the feature extractor is trained on data related to the specific operation scenario, even though no labels are being used. By using SimCLR as the framework for unsupervised learning we also allow for our method to be dynamic: training can take place during operation and

therefore adapt to new battlefield environments and improve over time. The big advantage this second step presents is that in military applications, adversaries do not have access to the data that a transfer learning approach with a pre-trained feature extractor relies on. The IDLS is very good at recognising when irrelevant and unseen images are encountered while also having an accurate target classification quality compared to similar applications that use significantly more labeled data. The IDLS exhibits comparable performance with a CNN using less than 0.3% of the labeled data needed to train a CNN.

In the third and last step of the unstructured environments' image data processing apparatus an unsupervised soft-labeling method is developed based on an extended variance ratio criterion (E-VRC method). It can be used for sorting out images collected by operating military vehicles. It can also be used for further sorting the irrelevant images determined by the IDLS. This last, unsupervised clustering step, without the need of any initializations or prior knowledge, completes the task of leaving no candidate image unidentified. More specifically, the E-VRC method overcomes the disadvantages of the Variance Ratio Criterion by normalizing the feature data and introducing an exponent term on the VRC equation, making it applicable to image recognition datasets. An algorithm that can efficiently cluster groups of images without any prior knowledge about them can be proven useful in many applications such as autonomous vehicle navigation or the creation of deep learning image datasets for training CNNs.

Within this work, this three-part image processing apparatus is being tested on several different datasets, demonstrating that it is useful not only for autonomous exploration and reconnaissance operations but also for the efficient content management and retrieval tasks. Concluding, other than its direct use in classifying or clustering image datasets, extensions of this work could lead

to new, more efficient and capable unsupervised networks for image feature extraction. Therefore, the importance of this work lies on the fact that advances in clustering methods have an impact in myriads of machine learning applications, pushing the boundaries of how well machines can visually interpret the world.

5.2 Future Research

This dissertation reflects the research work that was conducted throughout the duration of a PhD program in the University of Michigan- Ann Arbor, College of Engineering for more than four years (nine academic semesters semesters). Although the visual recognition apparatuses devised by the author and outlined in this dissertation have been described theoretically and tested in practice using the cited algorithms (available online), they have not yet been utilized in a real world application.

Therefore, the first step forward is to combine the two main pieces of this thesis, the IDLS and the E-VRC algorithms into a common image processing and labeling machine. A second step forward would be to apply the proposed algorithms, either separately or all together as a complete and integrated apparatus, in a real-world operational setting. In the Introduction and Conclusion sections of this thesis, a couple possible applications were suggested, such as military or civilian off-road autonomous vehicles, image content sorting machines and other. The work would also benefit from testing its parts on more challenging data, with the goal being to understand the fundamental stability of the challenging data to perturbations in order to be separable via these approaches. Video data or diverse sets of images could be used along with new data augmentation approaches (especially for the IDLS capability), so that a full system setup is devised that performs in realistic situations (applications where

there's very few samples).

The knowledge produced and obtained throughout these nine academic semesters has been already utilized in a number of ways by the author through internships at the U.S. Army CCDC Ground Vehicle Systems Center (GVSC) and the NASA Goddard Space Flight Center. This effort of integrating these algorithms with real-life autonomous vehicles (even in an elementary testing phase) should be continued. The process of applying the algorithms introduced in this thesis to autonomous vehicle research has been already initiated, as the MATLAB and Python algorithms have been published in the US Army Data DirectoR (DDR) through the id2e.net servers (currently unavailable as they recently migrated to different servers) using the Robot Operating System (ROS) open-source middleware suite. For the continuation of this effort we rely on the sponsors of this PhD thesis which -as a last note- we would also like to acknowledge. Therefore, we conclude this thesis by thanking the Automotive Research Center (ARC) in the University of Michigan- Ann Arbor and the Ground Vehicle Systems Center (GVSC) in Warren, MI for their technical and financial support.

APPENDICES

APPENDIX A

Algorithms

A.1 OSLS Algorithm

Run Training Images Through ResNet
Normalize Features F Using Equation 2.1
Construct Target $\hat{y}$ Similar to Matrix 2.13
for every Class **do**
    for every Epoch **do**
        Calculate Class Gradient $\partial L/\partial x$ Using Equations 2.9-2.11
        Calculate Learning Rate η Using Algorithm A.2
        Take Gradient Step to Calculate Weight Vector W_{class}
    end for
    Add Class Weight Vector W_{class} to the Weight Matrix W
end for
Calculate Training Scores $s = WF_{norm}$
Calculate Class Threshold Values τ Using the ROC and s
Testing Using Validation Images, W and τ

A.2 Learning Rate Algorithm

Get Total Loss: $L_{total} = |L|$
Initialize Learning Rate: $\eta = 0.01$
while Flag = True **do**
    Calculate Temporary Class Weights W_{temp} Using Equation 2.12
    Calculate Error: $E = F \times W_{temp} - \hat{y}$
    Calculate Loss: $L = \exp(E^2)$
    if $|L| < L_{total}$ **then**
        Update Total Loss: $L_{total} = |L|$
        Update Learning Rate: $\eta_{new} = 2 \times \eta$
    else
        Flag = False
    end if
end while

A.3 IDLS Algorithm

Load Training Dataset $X = [x_1, x_2, ..., x_i]$
Load Testing Dataset
for Training Image x_i **do**
    Pick Data Augmentation Operators $t_1(\cdot)$ & $t_2(\cdot)$
    Produce two correlated views $\tilde{x}$
    Pass Views Through $f(\cdot)$ (CNN) to Get Feature Representations z
    Pass Representations Through Non-Linear Projection Head $h = g(\cdot)$
    Use Feature Vector Projections h To Calculate Contrastive Loss L
    Backpropagate Loss / Update $f(\cdot)$ Weights
end for
Extract Testing Dataset Features Using Trained $f(\cdot)$
Use Features As Input For OSLS Classifier

A.4 E-VRC Algorithm

Produce Image Features using ResNet34
Euclidean Normalization of Features
for $K = [K_{start}, K_{stop}]$ **do**
    Perform K-means using K and Features
    Calculate E-VRC using K-means clusters
    Record E-VRC Value (EV)
    if $EV_K \leq [EV_{K-1}, EV_{K-2}, ..., EV_{K-10}]$ **then**
        Break For Loop
    end if
end for
Find K where EV is maximum (K_{max})
Perform K-means using K_{max} and Features

APPENDIX B

Datasets

B.1 Dataset Tables

Table B.1: ATR Dataset.

ATR Dataset

	Class	Images
1	Pickup Truck	100
2	Sport Utility Vehicle	100
3	Personell Carrier	100
4	Tank 1	100
5	Tank 2	100
6	Tank 3	100
7	Tank 4	100
8	Human	100
	Total Images	800

Table B.2: List of irrelevant and relevant classes in the mixed dataset in alphabetical order.

	Irrelevant		Relevant		
	Class	**Images**	**Class**	**Images**	**ImageNet Key**
1	Abbey	15,100	Fireboat	1,300	n03344393
2	Alley	15,100	Fire Engine	1,300	n03345487
3	Amphitheater	7,129	Forklift	1,300	n03384352
4	Amusement Park	15,100	Freight Car	1,300	n03393912
5	Apartment Building	6,909	Garbage Car	1,300	n03417042
6	Aqueduct	8,871	Go-Kart	1,300	n03444034
7	Arch	15,100	Golf Cart	1,300	n03445924
8	Assembly Line	6,103	Half Track	1,300	n03478589
9	Bandlands	15,100	Horse Cart	1,300	n03538406
10	Bamboo Forest	5,746	Jeep	1,300	n03594945
11	Bayou	13,405	Limousine	1,300	n03670208
12	Boardwalk	15,100	Minibus	1,300	n03769881
13	Boat Deck	15,100	Minivan	1,300	n03770679
14	Botanical Garden	15,100	Missile	1,300	n03773504
15	Bridge	15,100	Model T	1,300	n03777568
16	Building Facade	15,100	Moped	1,300	n03785016
17	Butte	15,100	Motor Scooter	1,300	n03791053
18	Campsite	15,100	Mountain Bike	1,300	n03792782
19	Canyon	15,100	Moving Van	1,159	n03796401
20	Castle	15,100	Parachute	1,300	n03888257
21	Cathedral	7,350	Parking Meter	1,300	n03891332
22	Cemetery	15,100	Passenger Car	1,300	n03895866
23	Church	8,216	Police Van	1,300	n03977966
24	Coast	15,100	Projectile	1,300	n04008634
25	Corn Field	15,100	RV	1,300	n04065272
26	Cottage Garden	15,100	Revolver	1,300	n04086273
27	Courthouse	10,490	Rifle	1,300	n04090263
28	Creek	15,100	School Bus	1,300	n04146614
29	Crevasse	6,654	Schooner	1,300	n04147183
30	Crosswalk	13,742	Snowmobile	1,300	n04252077
31	Dam	15,100	Space Shuttle	1,300	n04266014
32	Desert	20,620	Speedboat	1,300	n04273569
33	Dock	15,100	Sports Car	1,300	n04285008
34	Excavation	9,858	Steam Locomotive	1,300	n04310018
35	Fairway	15,100	Street Car	1,300	n04335435
36	Field	23,285	Submarine	1,300	n04347754
37	Fire Station	15,100	Tank	1,300	n04389033
38	Forest Path	15,100	Torch	1,300	n04456115
39	Forest Road	15,100	Tow Truck	1,300	n04461696
40	Formal Garden	12,738	Tractor	1,300	n04465501
41	Garbage Dump	6,151	Trailer Truck	1,300	n04467665
42	Gas Station	15,100	Tricycle	1,300	n04482393
43	Golf Course	15,100	Trimaran	1,300	n04483307
44	Harbor	15,100	Trolleybus	1,300	n04487081
45	Herb Garden	15,100	Unicycle	1,300	n04509417
46	Hospital	15,100	Warplane	1,300	n04552348
47	Hot Spring	15,100	Street Sign	1,300	n06794110
48	Hotel	6,017	Traffic Light	1,300	n06874185
49	Iceberg	15,100	Ballplayer	1,300	n09835506
50	Islet	6,843	Scuba Diver	1,300	n10565667
	Total Irrelevant	663,324	**Total Relevant**	64,859	

Table B.3: For E-VRC.

	ImageNet1000			Caltech256		
	Class	**Images**	**Key**	**Class**	**Images**	**Key**
1	Tench	50	0000	Ak47	50	001
2	Brambling	50	0010	American flag	50	002
3	Water ouzel	50	0020	Backpack	50	003
4	Bullfrog	50	0030	Baseball bat	50	004
5	American chameleon	50	0040	Baseball glove	50	005
6	American alligator	50	0050	Basketball hoop	50	006
7	Night snake	50	0060	Bat	50	007
8	Harvestman	50	0070	Bathtub	50	008
9	Black grouse	50	0080	Bear	50	009
10	Lorikeet	50	0090	Beer mug	50	010
11	Black swan	50	0100	Billiards	50	011
12	Flatworm	50	0110	Binoculars	50	012
13	Fiddler crab	50	0120	Birdbath	50	013
14	Flamingo	50	0130	Blimp	50	014
15	Red-backed sandpiper	50	0140	Bonsai	50	015
16	Sea lion	50	0150	Boom box	50	016
17	Afghan hound	50	0160	Bowling ball	50	017
18	Irish wolfhound	50	0170	Bowling pin	50	018
19	Staffordshire terrier	50	0180	Boxing glove	50	019
20	Sealyham terrier	50	0190	Brain	50	020
21	Tibetan terrier	50	0200	Breadmaker	50	021
22	German pointer	50	0210	Buddha	50	022
23	Sussex spaniel	50	0220	Bulldozer	50	023
24	Shetland sheepdog	50	0230	Butterfly	50	024
25	Appenzeller	50	0240	Cactus	50	025
26	Siberian husky	50	0250	Cake	50	026
27	Chow chow	50	0260	Calculator	50	027
28	White wolf	50	0270	Camel	50	028
29	Grey fox	50	0280	Cannon	50	029
30	Jaguar	50	0290	Canoe	50	030
31	Tiger beetle	50	0300	Car tire	50	031
32	Ant	50	0310	Cartman	50	032
33	Damselfly	50	0320	CD	50	033
34	Wood rabbit	50	0330	Centipede	50	034
35	Zebra	50	0340	Cereal box	50	035
36	Ibex	50	0350	Chandelier	50	036
37	Otter	50	0360	Chess board	50	037
38	Guenon	50	0370	Chimp	50	038
39	Titi monkey	50	0380	Chopsticks	50	039
40	Eel	50	0390	Cockroach	50	040
41	Academic gown	50	0400	Coffee mug	50	041
42	Apiary	50	0410	Coffin	50	042
43	Banjo	50	0420	Coin	50	043
44	Basketball	50	0430	Comet	50	044
45	Beer bottle	50	0440	Computer keyboard	50	045
46	Bobsled	50	0450	Computer monitor	50	046
47	Breakwater	50	0460	Computer mouse	50	047
48	Candle	50	0470	Conch	50	048
49	Cash machine	50	0480	Cormorant	50	049
50	Chain mail	50	0490	Covered wagon	50	050
	Total ImageNet	2500		**Total Caltech**	2500	

BIBLIOGRAPHY

BIBLIOGRAPHY

Addagarla, S. K., and A. Amalanathan (2020), Probabilistic unsupervised machine learning approach for a similar image recommender system for e-commerce, *Symmetry*, *12*(11), 1783.

Agrawal, R., C. Faloutsos, and A. Swami (1993), Efficient similarity search in sequence databases, in *International conference on foundations of data organization and algorithms*, pp. 69–84, Springer.

Arbeláez, P., B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik (2012), Semantic segmentation using regions and parts, in *2012 IEEE conference on computer vision and pattern recognition*, pp. 3378–3385, IEEE.

Azizpour, H., A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson (2015), From generic to specific deep representations for visual recognition, in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 36–45.

Bachman, P., R. D. Hjelm, and W. Buchwalter (2019), Learning representations by maximizing mutual information across views, *arXiv preprint arXiv:1906.00910*.

Bansal, A. (2020), Improving task-specific representation via 1m unlabelled images without any extra knowledge, *arXiv preprint arXiv:2006.13919*.

Basha, S. S., S. K. Vinakota, V. Pulabaigari, S. Mukherjee, and S. R. Dubey (2021), Autotune: Automatically tuning convolutional neural networks for improved transfer learning, *Neural Networks*, *133*, 112–122.

Beck, A., and M. Teboulle (2009), A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM journal on imaging sciences*, *2*(1), 183–202.

Bendale, A., and T. E. Boult (2016), Towards open set deep networks, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1563–1572.

Berkhin, P. (2006), A survey of clustering data mining techniques, in *Grouping multidimensional data*, pp. 25–71, Springer.

Berry, M. W., and M. Browne (2005), *Understanding search engines: mathematical modeling and text retrieval*, SIAM.

Berry, M. W., S. T. Dumais, and G. W. O'Brien (1995), Using linear algebra for intelligent information retrieval, *SIAM review*, *37*(4), 573–595.

Boddeti, V. N., and B. Kumar (2014), Maximum margin vector correlation filter, *arXiv preprint arXiv:1404.6031*.

Bozdogan, H. (1987), Model selection and akaike's information criterion (aic): The general theory and its analytical extensions, *Psychometrika*, *52*(3), 345–370.

Caliński, T., and J. Harabasz (1974), A dendrite method for cluster analysis, *Communications in Statistics-theory and Methods*, *3*(1), 1–27.

Caron, M., P. Bojanowski, A. Joulin, and M. Douze (2018), Deep clustering for unsupervised learning of visual features, in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 132–149.

Caron, M., I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin (2020), Unsupervised learning of visual features by contrasting cluster assignments, *arXiv preprint arXiv:2006.09882*.

Chen, T., S. Kornblith, M. Norouzi, and G. Hinton (2020), A simple framework for contrastive learning of visual representations, in *International conference on machine learning*, pp. 1597–1607, PMLR.

Cheng, Z., H.-T. Zhang, M. Z. Chen, T. Zhou, and N. V. Valeyev (2011), Aggregation pattern transitions by slightly varying the attractive/repulsive function, *PLoS One*, *6*(7), e22,123.

Davies, D. L., and D. W. Bouldin (1979), A cluster separation measure, *IEEE transactions on pattern analysis and machine intelligence*, (2), 224–227.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009), Imagenet: A large-scale hierarchical image database, in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee.

Dhamija, A. R., M. Günther, and T. E. Boult (2018), Reducing network agnostophobia, *arXiv preprint arXiv:1811.04110*.

Dodge, S., and L. Karam (2016), Understanding how image quality affects deep neural networks, in *2016 eighth international conference on quality of multimedia experience (QoMEX)*, pp. 1–6, IEEE.

Donahue, J., Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell (), A deep convolutional activation feature for generic visual recognition, *UC Berkeley & ICSI, Berkeley, CA, USA*, *1*.

Dosovitskiy, A., J. T. Springenberg, M. Riedmiller, and T. Brox (2014), Discriminative unsupervised feature learning with convolutional neural networks, *Advances in neural information processing systems*, *27*, 766–774.

Dunn, J. C. (1973), A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.

Edwards, A. W., and L. L. Cavalli-Sforza (1965), A method for cluster analysis, *Biometrics*, pp. 362–375.

Estévez, P. A., M. Tesmer, C. A. Perez, and J. M. Zurada (2009), Normalized mutual information feature selection, *IEEE Transactions on neural networks*, *20*(2), 189–201.

Ge, W., and Y. Yu (2017), Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1086–1095.

Ge, Z., S. Demyanov, Z. Chen, and R. Garnavi (2017), Generative openmax for multi-class open set classification, *arXiv preprint arXiv:1707.07418*.

Girshick, R., J. Donahue, T. Darrell, and J. Malik (2014), Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Griffin, G., A. Holub, and P. Perona (2007), Caltech-256 object category dataset.

Hadsell, R., S. Chopra, and Y. LeCun (2006), Dimensionality reduction by learning an invariant mapping, in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 1735–1742, IEEE.

Hand, D. J. (2009), Measuring classifier performance: a coherent alternative to the area under the roc curve, *Machine learning*, *77*(1), 103–123.

Hand, D. J., and R. J. Till (2001), A simple generalisation of the area under the roc curve for multiple class classification problems, *Machine learning*, *45*(2), 171–186.

Harris, D., and S. L. Harris (2010), *Digital design and computer architecture*, Morgan Kaufmann.

Hartigan, J. A., and M. A. Wong (1979), Algorithm as 136: A k-means clustering algorithm, *Journal of the royal statistical society. series c (applied statistics)*, *28*(1), 100–108.

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

He, K., X. Zhang, S. Ren, and J. Sun (2016b), Identity mappings in deep residual networks, in *European conference on computer vision*, pp. 630–645, Springer.

He, K., G. Gkioxari, P. Dollár, and R. Girshick (2017), Mask r-cnn, in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969.

Hendrycks, D., and K. Gimpel (2016), A baseline for detecting misclassified and out-of-distribution examples in neural networks, *arXiv preprint arXiv:1610.02136*.

Huang, L., X. Liu, B. Lang, A. W. Yu, Y. Wang, and B. Li (2018), Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks, in *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ishioka, T., et al. (2020), An expansion of x-means for automatically determining the optimal number of clusters, Citeseer.

Jain, L. P., W. J. Scheirer, and T. E. Boult (2014), Multi-class open set recognition using probability of inclusion, in *European Conference on Computer Vision*, pp. 393–409, Springer.

Jia, K., D. Tao, S. Gao, and X. Xu (2017), Improving training of deep neural networks via singular value bounding, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4344–4352.

Joachims, T., et al. (1999), Transductive inference for text classification using support vector machines, in *Icml*, vol. 99, pp. 200–209.

Kalibhat, N. M., K. Narang, L. Tan, H. Firooz, M. Sanjabi, and S. Feizi (2022), Understanding failure modes of self-supervised learning, *arXiv preprint arXiv:2203.01881*.

Kasapis, S., G. Zhang, J. Smereka, and N. Vlahopoulos (2020), Using roc and unlabeled data for increasing low-shot transfer learning classification accuracy, *arXiv preprint arXiv:2010.00721*.

Kasapis, S., G. Zhang, J. M. Smereka, and N. Vlahopoulos (2022), Open-set low-shot classification leveraging the power of instance discrimination, *The Journal of Defense Modeling and Simulation*, p. 15485129221111172.

Kass, R. E., and A. E. Raftery (1995), Bayes factors, *Journal of the american statistical association*, *90*(430), 773–795.

Käster, T., V. Wendt, and G. Sagerer (2003), Comparing clustering methods for database categorization in image retrieval, in *Joint Pattern Recognition Symposium*, pp. 228–235, Springer.

Kenyon-Dean, K., A. Cianflone, L. Page-Caccia, G. Rabusseau, J. C. K. Cheung, and D. Precup (2018), Clustering-oriented representation learning with attractive-repulsive loss, *arXiv preprint arXiv:1812.07627*.

Keogh, E., K. Chakrabarti, M. Pazzani, and S. Mehrotra (2001), Locally adaptive dimensionality reduction for indexing large time series databases, in *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pp. 151–162.

Kermani, S., N. Samadzadehaghdam, and M. EtehadTavakol (2015), Automatic color segmentation of breast infrared images using a gaussian mixture model, *Optik*, *126*(21), 3288–3294.

Kim, H. Y., and N. Vlahopoulos (2012), A multi-level optimization algorithm and a ship design application, in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, p. 5555.

Kozerawski, J., and M. Turk (2021), One-class meta-learning: Towards generalizable few-shot open-set classification, *arXiv preprint arXiv:2109.06859*.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012), Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, *25*, 1097–1105.

Kvålseth, T. O. (2017), On normalized mutual information: measure derivations and properties, *Entropy*, *19*(11), 631.

Likas, A., N. Vlassis, and J. J. Verbeek (2003), The global k-means clustering algorithm, *Pattern recognition*, *36*(2), 451–461.

Liu, B., H. Kang, H. Li, G. Hua, and N. Vasconcelos (2020), Few-shot open-set recognition using meta-learning, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8798–8807.

Loshchilov, I., and F. Hutter (2016), Sgdr: Stochastic gradient descent with warm restarts, *arXiv preprint arXiv:1608.03983*.

Lu, J., T. Issaranon, and D. Forsyth (2017), Safetynet: Detecting and rejecting adversarial examples robustly, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 446–454.

Mancini, M., Z. Akata, E. Ricci, and B. Caputo (2020), Towards recognizing unseen categories in unseen domains, in *European Conference on Computer Vision*, pp. 466–483, Springer.

Mardia, K. (1988), Multi-dimensional multivariate gaussian markov random fields with application to image processing, *Journal of Multivariate Analysis*, *24*(2), 265–284.

McCallum, A., K. Nigam, and L. H. Ungar (2000), Efficient clustering of high-dimensional data sets with application to reference matching, in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 169–178.

McDaid, A. F., D. Greene, and N. Hurley (2011), Normalized mutual information to evaluate overlapping community finding algorithms, *arXiv preprint arXiv:1110.2515*.

McQueen, J. B. (1967), Some methods of classification and analysis of multivariate observations, in *Proc. of 5th Berkeley Symposium on Math. Stat. and Prob.*, pp. 281–297.

Pelleg, D., and A. Moore (1999), Accelerating exact k-means algorithms with geometric reasoning, in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 277–281.

Pelleg, D., A. W. Moore, et al. (2000), X-means: Extending k-means with efficient estimation of the number of clusters., in *Icml*, vol. 1, pp. 727–734.

Peres, D., and A. Cancelliere (2014), Derivation and evaluation of landslide-triggering thresholds by a monte carlo approach, *Hydrology and Earth System Sciences*, *18*(12), 4913–4931.

Pernici, F., F. Bartoli, M. Bruni, and A. Del Bimbo (2018), Memory based online learning of deep representations from video streams, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2324–2334.

Petangoda, J., N. A. Monk, and M. P. Deisenroth (2020), A foliated view of transfer learning, *arXiv preprint arXiv:2008.00546*.

Petralia, F., V. Rao, and D. Dunson (2012), Repulsive mixtures, *Advances in neural information processing systems*, *25*.

Pham, I., and M. Polasek (2014), Algorithm for military object detection using image data, in *2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC)*, pp. 3D3–1, IEEE.

Reite, A., S. Kangas, Z. Steck, S. Goley, J. Von Stroh, and S. Forsyth (2019), Unsupervised feature learning in remote sensing, in *Applications of Machine Learning*, vol. 11139, p. 111390H, International Society for Optics and Photonics.

Rousseeuw, P. J. (1987), Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of computational and applied mathematics*, *20*, 53–65.

Russakovsky, O., et al. (2015), Imagenet large scale visual recognition challenge, *International journal of computer vision*, *115*(3), 211–252.

Sanakoyeu, A., M. A. Bautista, and B. Ommer (2018), Deep unsupervised learning of visual similarities, *Pattern Recognition*, *78*, 331–343.

Scheirer, W. J., A. de Rezende Rocha, A. Sapkota, and T. E. Boult (2012), Toward open set recognition, *IEEE transactions on pattern analysis and machine intelligence*, *35*(7), 1757–1772.

Scheirer, W. J., L. P. Jain, and T. E. Boult (2014), Probability models for open set recognition, *IEEE transactions on pattern analysis and machine intelligence*, *36*(11), 2317–2324.

Shapiro, L. G., G. C. Stockman, et al. (2001), *Computer vision*, vol. 3, Prentice Hall New Jersey.

Simonyan, K., and A. Zisserman (2014), Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*.

Sinaga, K. P., and M.-S. Yang (2020), Unsupervised k-means clustering algorithm, *IEEE Access*, *8*, 80,716–80,727.

Sindhwani, V., and S. S. Keerthi (2006), Large scale semi-supervised linear svms, in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 477–484.

Tibshirani, R., G. Walther, and T. Hastie (2001), Estimating the number of clusters in a data set via the gap statistic, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(2), 411–423.

Vidal, R., Y. Ma, and S. S. Sastry (2016), Principal component analysis, in *Generalized principal component analysis*, pp. 25–62, Springer.

Wang, L., G. Hua, R. Sukthankar, J. Xue, and N. Zheng (2014), Video object discovery and co-segmentation with extremely weak supervision, in *European Conference on Computer Vision*, pp. 640–655, Springer.

Wu, Z., Y. Xiong, S. Yu, and D. Lin (2018), Unsupervised feature learning via non-parametric instance-level discrimination, *arXiv preprint arXiv:1805.01978*.

Xie, F., and Y. Xu (2020), Bayesian repulsive gaussian mixture model, *Journal of the American Statistical Association*, *115*(529), 187–203.

Xie, T., and Y. Li (2019), A gradient-based algorithm to deceive deep neural networks, in *International Conference on Neural Information Processing*, pp. 57–65, Springer.

Yu, L., G. Fan, J. Gong, and J. P. Havlicek (2015), Joint infrared target recognition and segmentation using a shape manifold-aware level set, *Sensors*, *15*(5), 10,118–10,145.

Zhang, J., W. Li, P. Ogunbona, and D. Xu (2019), Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective, *ACM Computing Surveys (CSUR)*, *52*(1), 1–38.

Zhang, L., Y. Tong, and Q. Ji (2008), Active image labeling and its application to facial action labeling, in *European Conference on Computer Vision*, pp. 706–719, Springer.

Zhou, B., A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba (2017), Places: A 10 million image database for scene recognition, *IEEE transactions on pattern analysis and machine intelligence*, *40*(6), 1452–1464.

Zuo, W., X. Wu, L. Lin, L. Zhang, and M.-H. Yang (2018), Learning support correlation filters for visual tracking, *IEEE transactions on pattern analysis and machine intelligence*, *41*(5), 1158–1172.