A Novel Approach on the Fine-Grained Task of Classifying Firearm Brands and Subtypes

A Thesis Presented for
the Master Artificial Intelligence

Author: Raphaël Furtunato
Student Number: 4170512
First Supervisor: Dr. ir. Ronald W. Poppe
External Supervisor: Ben Kessler
Second Supervisor: Dr. Heysem Kaya

In Cooperation With:

POLITIE

Beta Science
University of Utrecht
The Netherlands
April 22, 2021

Abstract

The registration of legal and illegal firearms takes up a substantial amount of man-hours, in addition to requiring expert knowledge of firearm brands and subtypes. Therefore, Dutch police force are looking into an alternative way to classify of firearms which would facilitate the registration process. No dataset containing brand and subtype annotation was yet available. Consequently, the current study constructed three firearm datasets containing brand and subtype annotation, through the means of web scraping and photo shoots at one of the Dutch police forces depots. Due to data shortage, visual similarities of the firearms and the wide range of applicable situations required by the Dutch police force, three approaches were formulated in order to implement an effective prototype. These approaches were: a baseline approach that experimented with the effect of data augmentation and image size, a fine-grained approach that tested the novel combination of two fine-grained loss functions to increase the attention for fine-grained details, and finally, an object detection approach that added an object detection pipeline to the classification algorithm. Subsequent implementation of each approach resulted in increased performances, achieving a final balanced accuracy score of 80.22% on the main benchmark dataset. Taken together, this study revealed that automating the process of firearm brand and subtype classification is feasible.

Table of Contents

List of Abreviations

ANN . . . . . Artificial Neural Network

CNN . . . . . Convolutional Neural Network

ConvNet . . . . . Convolutional Network

FGC . . . . . Fine-Grained Categorization

Grad-CAM . . . . . Gradient-weighted Class Activation Mapping

IoU . . . . . Intersect of Union

MA-CNN . . . . . Multi-Attention CNN

mAP . . . . . mean Average Precision

PPHI . . . . . Plain Police Handgun Images

R-CNN . . . . . Region-based Convolutional Network

RA-CNN . . . . . Recurrent Attention CNN

ResNet . . . . . Residual Network

t-SNE . . . . . t-distributed stochastic neighborhood embedding

TASN . . . . . Trilinear Attention Sampling Networks

1 Introduction

Dutch police forces come in contact with firearms every day, be it legal through regulatory monitoring, or illegal by means of confiscation. Both legal and illegal firearms need to be registered in databases for security reasons. During this process of registration, information like brand, subtype and serial number need to be entered into database. Given that 197,357 legal firearms were in circulation in the Netherlands in 2019 [43] and almost 6,000 firearms are confiscated each year [33], a lot of time goes into the registration of firearms. This is not only a time-consuming and costly process, but also a cumbersome and difficult task. Because there exist an abundance of brands with hundreds of different subtypes, that visually only differ in small details like barrel length or ammunition size, firearm registration depends on expert knowledge. Take for example the Glock, which is a semi-automatic pistol existing of more than 31 subtypes differing only in small visual details (see Figure 1). Therefore, the police forces are looking for a way to automate the classification of firearms by their brand as well as their subtype.

Figure 1: Example of the visual similarities between different types of the Glock. From left to right the Glock 17, 19, 21, 30, 34 and 45.

One field that could potentially contribute to the automation of firearm brand and subtype classification is computer vision. Given that one of the aims of computer vision is to automate visual classification tasks. During the last decade computer vision has made great leaps in image recognition. Through the development of improved hardware and the increase in data available on the internet, computer vision algorithms such as the Convolutional Neural Network (CNN) have gone from a negligible performance to approaching human level skills in the field of object classification [10, 36]. Hence, the Dutch police force are looking into the possibility of utilizing computer vision to automate the process of classifying firearm brands

and subtypes.

1.1 Scope

1.1.1 Requirements

The current study aims to enable automatic classification of firearm brands and subtypes for the Dutch police force. First and foremost, this study serves as a test to determine whether a prototype could be implemented that is capable of classifying firearms. Given that a prototype could be implemented successfully, the applications within the police force are plentiful. One of the main applications arose during the start of the corona pandemic. Because of the restrictions enforced by the pandemic, it became impossible to meet legal gun owners in person. Therefore, the regulatory checkups were held online via video calls. This triggered the idea of further digitizing the process and thereby automating more of the administration such as the classification. In this case the gun owner would hold up the firearm in front of the webcam of the computer and the classification software would recognize the brand and subtype, facilitating the administrative process. Another application would be the automatic classification of firearms in confiscated footage on cell phones or other capturing devices. Finally, a third possible application would support officers on the street, whom encounter a firearm while lacking sufficient knowledge of firearm types to be able to classify them. Given that the applications of the software could vary widely, from detecting firearms through a video call to classifying a firearm on photos from confiscated cellphones, the requirements on the input data should be as lenient as possible. Meaning that, if possible, the software should be able to classify firearms from a range of different angles, positions and backgrounds.

1.1.2 Challenges

Several challenges need to be overcome in order to implement a well-performing prototype. To start off, the prototype needs to be able to distinguish firearms from different angles, backgrounds and light exposures. This by itself can be a very challenging task, given that firearms are in general visually similar. This means that peripheral matters such as the angle the picture is taken in, have much more influence on the image features than the type of firearm that is in the image. In effect, there is very little visual information that distinguishes one type of firearm from another (i.e. both have almost identical characteristics, such as a barrel, trigger and a pistol grip). To this end, the current study needs to find a way to deal with the challenge of classifying data that has low inter-class variation (images from different categories differ relatively little) and high intra-class variation (images from the same category differ much relatively).

In addition to the low inter-class variation and high intra-class variation, the wide variety of poses

the firearm can take on also posits challenges. Moreover, the high variation in locations of the firearm in a picture adds another challenge to this assignment. Therefore, the current study also needs to assess ways of localizing the firearm in the picture, before classification is even possible.

Finally, the current study needs to overcome the lack of an adequate dataset. An open source firearms dataset, containing firearm brand and type annotation, does not exist on the internet. Furthermore, the strict regulations of the Dutch police forces when it comes to sharing sensitive information (i.e. such as images of confiscated weapons), causes them to collect very little image data. Therefore, the current study will have to gather data available on the internet. Moreover, the data obtained on the internet might be heavily skewed, meaning that some categories have more images than others. This could weaken the performance of the implemented algorithms. For this reason, it will be useful to study methods to augment data for categories that are relatively small.

1.1.3 Relevance

The current study is relevant for society in several ways. First, automating the classification of a firearm through a well-performing prototype would alleviate some of the administrative workload Dutch police officers face during working hours. In addition, less expert knowledge in terms of recognizing firearms would be needed by police officers, which would mean that more time can be spent on other parts of the job.

Second, the officers that come in contact with firearms often have little knowledge of firearms and are not well-suited to the task of recognizing firearms. This causes an increased risk of fraudulent cases through illegal modifications or false registration by the legal gun owners. Moreover, it also results in increased difficulties to trace the origin of illegal firearms. Hence, automating firearm classification could contribute towards a safer society.

Finally, there is no software currently available that is capable of automatically classifying firearms. Therefore, a working firearm classifier would offer the scientific community more insight into the computer vision field and would show that the state-of-that-art vision algorithms are capable of classifying challenging objects such as firearms. Even in the case that suboptimal results would be obtained, this study can offer valuable insights as well as a challenging dataset, which can help to improve future algorithms.

2 Literature Review

This section will provide an overview of the relevant literature used for the current study. It will go over the three challenges presented in the introduction, which are the classification of hard to classify objects, the shortage of data and finally, the localization of the firearms.

2.1 Firearm Classification

2.1.1 Hand-Crafted Feature Extraction

To the best of our knowledge, no studies have been executed on the subject of firearm brands and subtype classification. However, one study showed similarities [22] by studying the effectiveness of gun type (e.g. handgun, shotgun, submachine gun) classification. This study used multiple hand-crafted feature extractors to perform the classification. A combination of the Canny Edge Detector [5], the Susan Corner Detection [40] and Template Matching were used to extract the important features and match these with prototype firearms of each category. Although this method performed well with a relative small dataset, the firearms in the images were limited in their pose and no additional background noise was allowed. Furthermore, despite the fact that hand-crafted feature descriptors could potentially preform well, they are based on predefined rules that are limited in their capabilities to extract complex features in noisy data, such as classifying a firearm in a variety of backgrounds, firearm poses and light exposures. Therefore, the current study will focus on algorithms capable of learning complex features and patterns from the data. One such class of algorithms is the Artificial Neural Network (ANN). The ANN has been proven to be able to find patterns in a variety of complex tasks [4, 30, 36, 56]. Consequently, the current study will look further into the application of ANNs.

2.1.2 Artificial Neural Networks

ANNs are computational models that try to mimic the complex interaction between the neurons in the human brain. The building block of an ANN is the artificial neuron, which is connected with other neurons through weights. A neurons output consists of a weighted sum with an additional bias term. Because this mechanism by itself can only solve linear problems, activation functions are introduced to add nonlinearity. In addition, those activation functions also help to transform the output to be used for regression or classification tasks. In case of classification, the softmax function is often used on the output of the ANN, which is a function that creates a probability distribution from a vector of outputs.

The weights and biases (parameters) of an ANN are often initiated at random. Therefore, an ANN has to be fitted to the data to make meaningful predictions. During this training process the dataset is passed through the network in small batches and the loss is calculated. The loss or cost function is a function that

calculates the divergence between the predicted outcome and the ground truth. A prominent loss function for classification is the Cross Entropy loss, which measures the entropy between two probability distributions. The loss is attempted to be minimized through backpropagation, an algorithm that computes the gradient for all the parameters with respect to the loss. As such, the direction is calculated in which the parameters need to move in order to minimize the loss. However, adding the gradients directly to the parameters would cause unstable learning. Therefore, a learning rate is implemented to smoothen the training process.

$$\theta_t = \theta_{t-1} - \alpha_t \nabla_{\theta_{t-1}} \mathcal{L}_t \tag{1}$$

Equation 1 shows the updating process, where θ refers the networks parameters, with t being the time step. α is the learning rate that is multiplied with the gradient of the loss with respect to the parameters, $\nabla_{\theta_{t-1}} \mathcal{L}_t$. This process of optimizing the ANN parameters is called stochastic gradient descent.

2.1.3 Convolutional Networks

A Convolutional Network (ConvNet) is a subclass of ANNs that has been successfully applied to a multitude of computer vision tasks [24, 16, 30]. A ConvNet architecture consists of convolutional and pooling layers (see Figure 2). These layers are repeated and perform operations at different scales and sizes on the input. The initial layers extract generic features from the image such as edges, while layers deeper in the network combine these generic features into more complex patterns and objects. Just like in other types of the ANN, the features are learned through backpropagation, which is a major advantage. In addition, ConvNets are great at extracting spatial information and contain comparatively fewer parameters, which makes them easier to optimize.

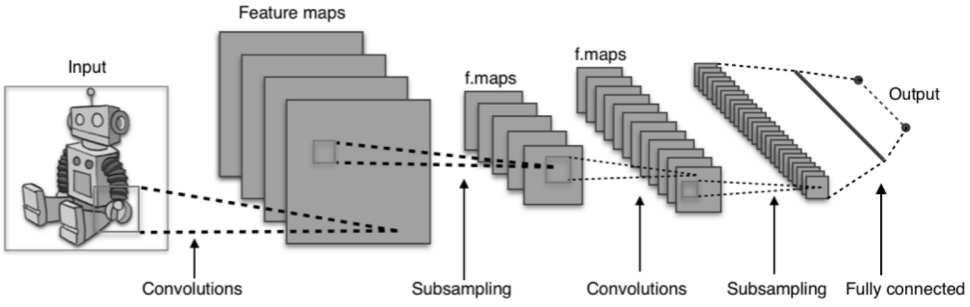

Figure 2: Visualization of a CNN

A ConvNet extracts features through its convolutional layers. In a convolutional layer, weights are stacked into filters (also called kernels) that slide over the input. The filters perform matrix multiplication with the part of the input they currently sit on, creating a matrix of features with an equal or smaller size

than the original input. Every convolutional layer consists of multiple filters, which create what is called a feature map (activation map). In order to make the ConvNet more robust to locational variances and to reduce the resolution of the feature maps, pooling is used. The pooling operation splits the feature map in blocks and performs an operation (e.g. max, mean) on those blocks to reduce the overall size of the feature maps.

For classification tasks ConvNets are often combined with a classifier. In this case, the ConvNet functions as a feature extractor and the classifier outputs predictions base on the extracted features. The predominant method of combination is to combine a ConvNet with a fully connected network and is referred to as a CNN. The fully connected network is another subcategory of the ANN, which owes its name to the connections between the neurons. Each neuron is connected with all the neurons of the previous layer as well as all the neurons of the next layer.

In theory, deeper networks should be capable of extracting more complex features and thereby increase the performance [41, 39]. This, however, is not always the case because of the problem of vanishing and exploding gradients. Vanishing and exploding gradients are caused by the computation of the gradient of deeper layers with respect to the loss. To calculate the gradient deeper in the network, the gradient goes through multiple multiplication terms that can either cause the gradient to explode or vanish. These vanishing or exploding gradients cause either extremely large or small updates to occur deep in the network. Although the problem of vanishing and exploding gradients can partly be solved by using batch normalization (see section 2.2.3), the problem of training deeper neural network still persists. This indicates that there might also be an optimization problem that causes the lack in performance of deep networks. Therefore, the Residual Network (ResNet) [18] is proposed as a way to cope with this optimization problem. The ResNet adds shortcuts between multiple layers, which are called Residual Blocks. (see Figure 3). The main idea behind the ResNet is, that a larger network is a superset of a shallower network. Therefore, a larger network should always be capable of performing at least as well as the shallower network, by just copying the weights of the shallower network with the identity mappings (i.e. $f(x) = x$). However, the identity mappings are hard to learn. Therefore, the residual block should facilitate in pushing the layers output to zero if necessary. This aids the optimization of networks and has been found to significantly increase performance in deeper ConvNets [18, 28].

The effectiveness of the ConvNet lies in its feature extraction operations which are hierarchically constructed, thereby matching the hierarchical nature of vision itself. The first few convolutional layers extract more generic features, such as edges. Along the way, those generic features are combined into more complex patterns and shapes. In spite of the effectiveness of the feature extraction, standard CNNs do not perform well when differences between categories are small and differences within categories are relatively

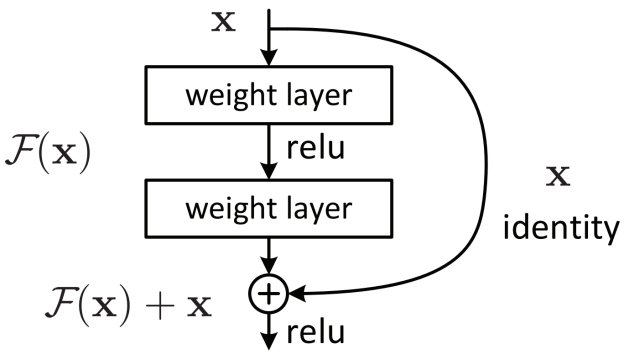

Figure 3: Residual Block

big, which is the case for firearms. Therefore, the focus will turn towards solving the problem of hard to distinguish objects.

2.1.4 Fine-Grained Categorization

The Fine-Grained Categorization (FGC) task refers to recognizing objects that are visually hard to distinguish such as different types of birds, cars and airplanes [44, 50, 29]. In large-scale visual datasets different object classes are often recognizable by their shape, size, or color, but in FGC datasets there is very little inter-class variation and differences are often local and subtle. Moreover, intra-class variation is high, since angles, background, and light exposure make up most of the variation in the image. In addition to the low inter-class high intra-class variance, data is often scarce and non-uniformly distributed over the classes. Given that FGC data is often hard to come by and labeling requires expert knowledge.

The combination of mentioned difficulties makes FGC a challenge for standard CNN architectures. Loss functions such as Cross Entropy loss enforce networks to become discriminative with high confidence. In Large-scale visual datasets, where inter-class variation is high and intra-class variation is low, this has been found to work well. However, in FGC, where inter-class variation is relatively high, loss functions such as Cross Entropy loss will cause the network to discriminate sample specific differences and thereby to generalize poorly to unseen samples [12]. Consequently, to improve the performance on FGC tasks it is important to pay attention to subtle differences in fine-grained details of an object (e.g. different types of firearm barrels). Therefore, research in the field of FGC focuses on enforcing attention on specific parts of the object, since those parts contain vital information about the objects' category.

Early research added extra annotation to datasets in the form of bounding boxes and object part annotation. This was done in order to handle both the tasks of classification and object/part localization as a supervised learning task [53, 52, 2, 49] (see Figure 4). In this approach, an object detection algorithm

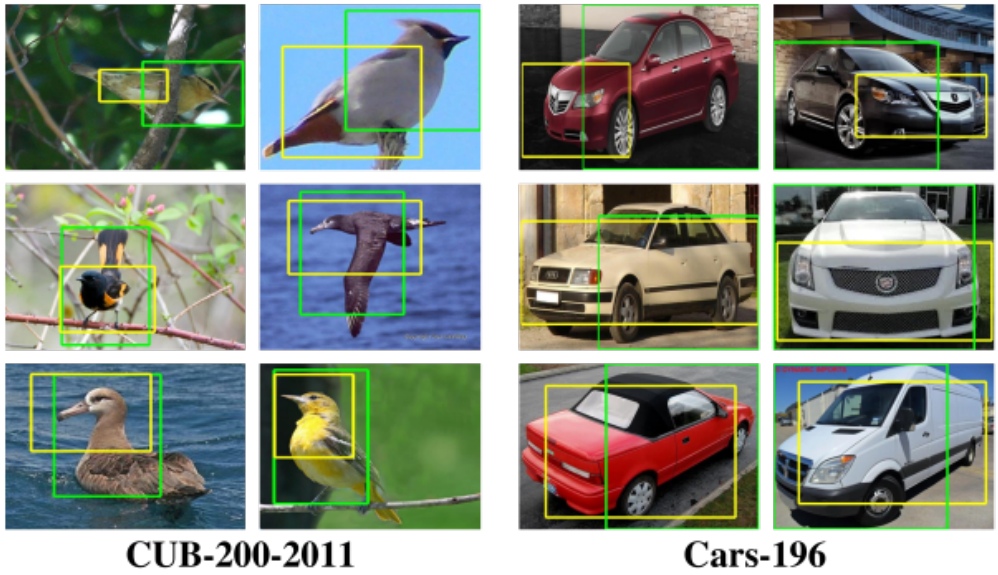

Figure 4: Part Annotation for FGC

would learn to localize the important features and the classification algorithm would learn to classify these features. At test time, the algorithms should have learned which features to focus on and do not need the extra annotation anymore. However, since all the extra annotation needs to be done manually, this approach does not scale well. Therefore, novel research focuses on combining an attention mechanism with object recognition. The object recognition algorithm would then be responsible for classification, while the attention mechanism would be responsible for proposing interesting regions and be trained in a weakly supervised way, without the need for extra annotation.

2.1.5 Attention Mechanisms

Different architectures are proposed that combine attention and recognition in a single algorithm. Lin et al. [27] suggested a Bilinear CNN that consists of two ConvNets forming a two stream network that is merged by means of an outer product pooling layer and a softmax layer. Although not further investigated, it is noted that this might work like the human brain, where the ventral stream is responsible for 'what' you see and the dorsal stream is responsible for 'where' you see it. Despite the fact that this architecture does not need any extra annotation it performed comparable to algorithms that need extra annotation, showing that manually defined part annotations are not necessary to perform well on FGC tasks.

Fu et al. [14] builds upon the idea of a multi-convnet architecture by proposing a Recurrent Attention CNN (RA-CNN). The RA-CNN consists of a CNN that splits into two subnetworks, an attention proposal subnetwork and a classification subnetwork. The classification subnetwork is responsible for producing a

classification, whereas the attention proposal subnetwork proposes an attention region, containing important features of the object. This region of attention is cropped from the original image and again passed into the network repeating the previous steps until reaching the number of passes defined beforehand. After all the passes through the network are completed, the outputs of the classification branch, which made a prediction after every pass, are combined into the definitive prediction. In this manner, the algorithm is optimized to localize important features in the image and at the same time to learn to classify the object on different scales. Optimization is done through a multi-task loss function, that optimizes the classification as well as the attention on fine-grained details.

The Multi-Attention CNN (MA-CNN) [54] takes the attention mechanism one step further by incorporating the attention mechanism and the classification in a single pass. A multi-attention layer is implemented that, sorts the feature maps into spatially correlated attention groups. In order to learn both the attention grouping and the classification, a multi-task loss is designed. The first part of the loss function captures the classification loss, while the second part captures the attention grouping loss, where the focus lies on grouping regions that are close to each other while still keeping diversity in the features. The main advantage of the MA-CNN over the RA-CNN is that everything is done in one pass, speeding up the training and inference procedure.

Although these algorithms perform well on highly complex FGC data, they still have a flaw. That is, the number of attention regions is predefined, by either the number of passes through the network or the feature map grouping. The Trilinear Attention Sampling Networks (TASN) [55] tries to deal with this downside by proposing a separate attention mechanism that is not bound by a predefined number of attention regions. The TASN consists of three parts: the trilinear attention module, the attention sampler and the feature distiller. First, the attention maps are extracted from the feature maps by the trilinear attention module. This is done by integrating the feature maps according to their spatial relationship. Second, the attention sampler produces a structure-preserved and a detail-preserved image. The structure-preserved image contains all important fine-grained features and the detail-preserved image focuses on single fine-grained details. Third, a part and master CNN are introduced in a master-student like style. The part CNN receives the detail-preserved image and the master CNN the structure-preserved image. The aim of this pair of networks is for the part CNN to distill knowledge of the fine-grained details to the master CNN, such that the master CNN learns which features to pay attention to.

2.1.6 Loss Functions

Recently, more consideration has gone into improving the loss function for FGC tasks [6, 12, 47, 45]. Although the attention mechanisms described above perform well, they add a lot of complexity, requiring modifications

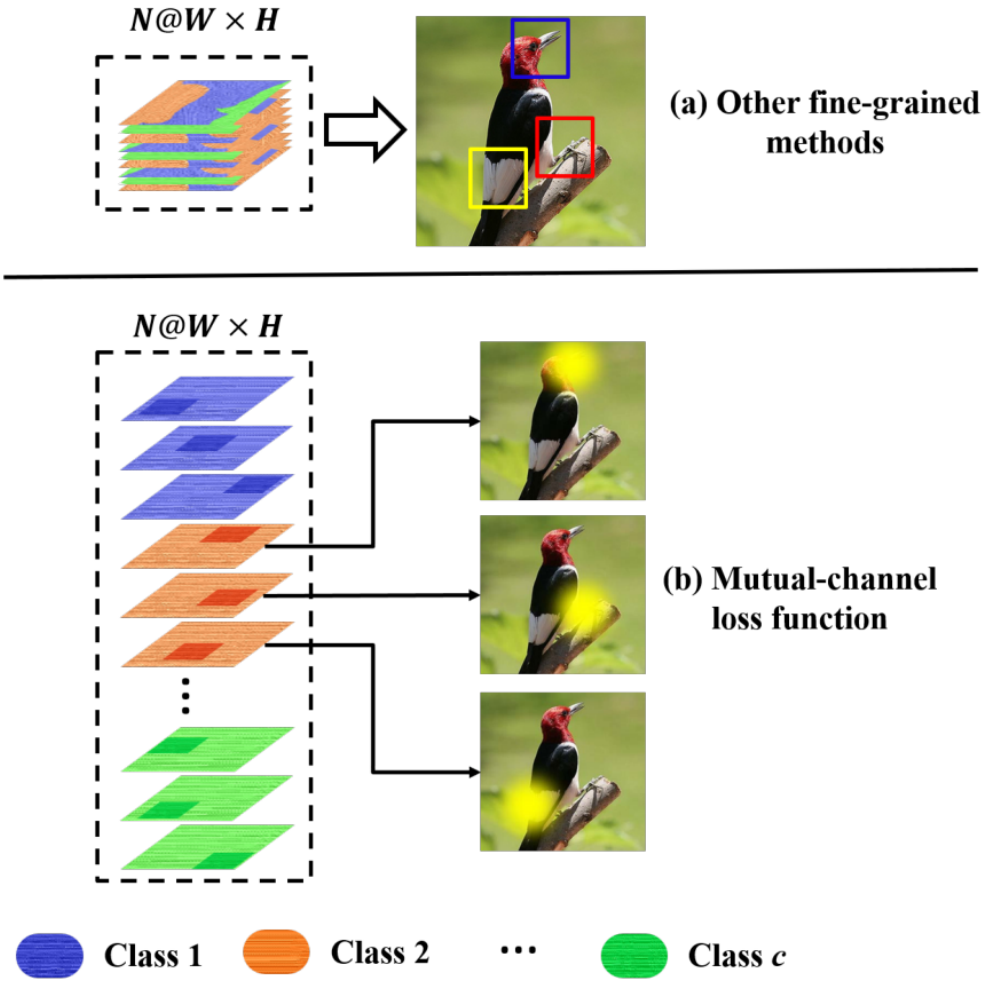

Figure 5: Example of the class-aligned feature maps enforced by the Mutual Channel Loss

to be made to the algorithm. In addition, training such attention mechanisms often comes with extra computational costs. The Mutual Channel loss [6] has been proposed as a loss function that does not add addition cost in terms of computation. Furthermore, the loss function solves the fine-grained problem by directly enforcing the feature maps to be diverse and discriminative. The discriminative component forces feature maps to be class-aligned and each feature map belonging to a particular class to be discriminative. This is done via a novel Channel-Wise Attention operation that randomly samples a few feature maps from every category group during training, such that every feature map contains sufficient discriminative information to make a valid prediction. The diversity component is a distance function for the feature maps, that enforces spatial diversity between feature maps belonging to the same category. The end result, therefore, should be a feature map that is class-aligned, where each individual feature map is discriminative on mutually

distinct local parts (see Figure 5).

Whereas the Mutual Channel loss aims to deal with the high inter-class variances by paying attention to fine-grained details, the Pairwise Confusion loss [12] attempts to increase low intra-class variance. The Pairwise Confusion loss is a loss function designed to add label confusion, increasing the low inter-class variance, thereby facilitating the algorithm to learn the subtle fine-grained details. To apply the Pairwise Confusion loss the training batch is split in two. Next, for each pair of images a euclidean distance penalty is added over the softmax output of that given image pair. However, this penalty only applies for images that are not of the same label, such that the penalty term works to increase the inter-class variance.

2.2 Dealing with a Data Shortage

Although ConvNets are a very good solution for extracting features from image data, they do have some downsides. First, they are computationally expensive, requiring high-end hardware in order to train state-of-the-art ConvNets. Especially because deeper ConvNets are capable of extracting more complex features from the data, but do introduce more parameters, which in turn increases the computational burden [18]. Second, optimizing machine learning algorithms like ConvNets is a data-driven process, requiring an immense amount of data. In the case of supervised learning, knowing the true category of the data is necessary, in order to calculate the error between the predicted outcome and the ground truth. In case labeled data is at a shortage, the chances of overfitting increase. Overfitting happens when a machine learning algorithm adjusts too closely to the training data, thereby failing to fit to unseen data. Overfitting is a substantial problem in the field of machine learning, especially when working with a small dataset and an algorithm that contains a lot of parameters. This section will delve into approaches that can help to overcome the lack of data as well as ways to deal with small datasets. These approaches include techniques such as web scraping, data augmentation and regularization techniques.

2.2.1 Web Data

The rise of the internet has provided us with a vast amount of images shared online. The current study will try to collect this data and use it for the purpose of training the algorithms. One approach of web scrapping uses search engines to find the images online [23]. Search engines already contain an up to date index of a large portion of the images available on the web. Therefore, they facilitate the process of finding the images on the web. Subsequently, only the link has to be followed to the website that houses the image.

Although a great amount of data can be gathered using web searches, there is also a downside to using this method; web searches often return noisy data. The two forms of noise that are most relevant are

cross domain noise and cross category noise [23]. Cross domain noise refers to the portion of images that are not of any category (i.e. images that do not contain one of the relevant firearm categories). Cross category noise describes the portion of images with wrong labels (i.e. an image of a Glock 17 which has the label of a Glock 25). A lot of research has gone into the validity of using noisy web data, showing promising results [23, 7, 8]. However, most of the studies have used web data as an extension to a human annotated dataset (active learning). Therefore, the current study will look further into the validity of using solely web images as means of training.

2.2.2 Data Augmentation

Data augmentation is another way to deal with a shortage of data [38]. It encompasses multiple image processing techniques, with the goal of improving generalizability. The key idea behind data augmentation is to artificially increase the size of the train dataset, through manipulation of the data in such a way that it seems like novel data, while still preserving the label. The most popular data augmentation techniques are:

- **Flipping:** flips an image left to right
- **Rotating:** rotates the images randomly by a certain degree
- **Cropping:** zooms in on the center of the image by some random margin
- **Color Jitter:** changes the brightness, contrast and saturation

2.2.3 Regularization techniques

Transfer Learning Transfer learning is taking knowledge gained in a specific domain and utilizing it on a different domain. In CNNs, transfer learning can be an effective way to deal with the computational expensiveness and data hungriness of optimizing a CNN. Research has shown that the features learned by the convolutional layers are very generalizable [51], meaning that regardless of whether convolutional layers are trained on a dataset of cars or birds, they will still look for the same features in the image. By replacing the fully connected layers with a new network that fits the category size of the new domain, a CNN can get a head start using transfer learning. Given that only the fully connected layers have to be trained from scratch. Therefore, CNNs are often trained on a large dataset such as ImageNet [24] to get the benefits of the generic nature of feature extracting.

Batch Normalization Batch normalization is a regularization technique that normalizes the inputs of individual layers inside a CNN [20]. By using batch normalization parameters are less affected by updates

to other parameters in their neighborhood, which increases training time and improves overall accuracy. Moreover, batch normalization allows for the use of larger learning rates and makes networks less affected by parameter initialization.

2.3 Firearm Detection and Segmentation

Considering that the goal is to classify firearms in a variety of poses and locations on the image, a classification only approach might not be sufficient to comply with the applications of the Dutch police force. Most state-of-the-art algorithms for image classification are competent at classifying objects when they are clearly presented in the image, but struggle when the image contains other objects or the desired object is not in the center of the image. Therefore, object detection might be a solution as it has found to work well in combination with classification [53, 52].

Object detection consists of two parts. First the objects in the image have to be localized and, second, the objects in the image have to be classified as a particular class. There exist two main approaches in object detection which are the regression approach and the classification approach. The former approach frames the objective as a regression problem and tries to regress all the objects in a image in one single pass [34, 32]. The latter approach uses a classifier which is used to predict objects in specific regions [16]. These regions are selected either by a region proposal algorithm or a sliding window approach. The regression approach offers an advantage in speed and simplicity of training since everything is computed in a single forward pass. However, the classification approach offers better performances and extensibility in the form of segmentation and key point prediction. Therefore, the current study will focus on the classification approach.

There exist two ways in which the classification approaches localizes objects in the images, which are the sliding window approach and the region proposal approach. The sliding window approach slides across regions of the image and tries to classify whether an image is present in that region [11, 1]. One major disadvantage is the huge computational cost, because an n by n image has an order of n^4 number of rectangles that all need to be visited. Therefore, region proposal algorithms are introduced as a less computational costly way to localize possible objects. Region proposal algorithms propose image patches that might contain an object as an interesting region (region of interest) [17]. These algorithms are mostly non-learning computer vision algorithms that work by detecting edges, color and texture. The advantage of region proposal algorithms over sliding windows approach, is that they scale well to larger images and significantly reduce the set of possible object locations in an image. Region proposal algorithms are effectively implemented and improved by the Region-based Convolutional Network (R-CNN) series.

The R-CNN [16] is the first algorithm in the R-CNN series. The R-CNN architecture consists

of multiple parts (see Figure 6). First, the region proposal algorithm explained above, proposes around 2000 regions of interest, that are wrapped into a standard size. Subsequently, these wrapped proposals are passed through a ConvNet, that extracts the important features from the regions of interest. Next, the architecture branches into a classification stream and a bounding box regression stream. The classification stream consist of a support vector machine that classifies the objects according to the predefined classes, including a background class for regions that do not contain an object. The bounding box regression branch consists of a fully connected ANN and regresses offsets to the size of the region of interest, that become the bounding boxes. These bounding box positions are an offset on the original region proposal boxes, since those regions of interest do not always surround the whole object. Although the R-CNN algorithm offers much improved performance in the task of object detection, its main downside is its speed. Because the region proposal algorithm proposes 2000 regions of interests, the network needs to make that many forward passes as well, which causes extremely slow performance at during training as well as during inference.

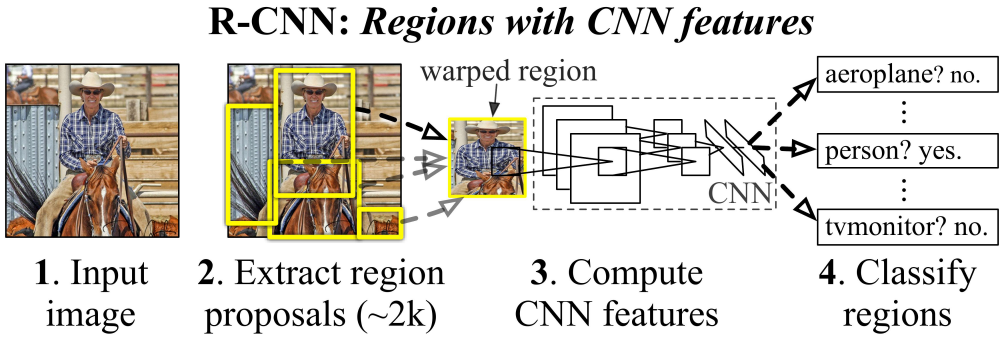

Figure 6: R-CNN Framework

Fast R-CNN [15] is introduced in order to improve the speed of the feature extraction. Instead of generating regions of interest on the original image, the image is first passed through the ConvNet, which generates high resolution feature maps. Subsequently, the region proposal algorithm is applied on the feature maps instead of on the image. Next, a Region of Interest Pooling layer is used to crop the regions of interest (which are now crops of the feature maps instead of the image) to the desired size. Finally, the network branches out again into a classification stream, that now consists of a fully connected ANN, and the unchanged bounding box stream. The improvement of speed of the Fast R-CNN over the original R-CNN is significant. However, the bottleneck of the algorithm now lies with the region proposal algorithm.

Faster R-CNN [35] is proposed to improve the speed of the region proposal algorithm. The Faster R-CNN does away with the previously used region proposal algorithm based on standard computer vision methods and implements a region proposal network with learnable parameters. This has two main advantages,

the speed of the region proposals improves dramatically and regions of interest can now be learned, which has positive effects when dealing with unconventional objects.

Mask R-CNN [19] builds on the previous iterations of the R-CNN and extends the algorithm with the ability to do instance segmentation as well. The Mask R-CNN algorithm consists of an extra branch next to the classification and bounding box regression branches. This branch consists of a ConvNet that generates a binary mask based on the region of interests, meaning that the success of the segmentation masks depends on the quality of the bounding box regression.

3 Research Question

The main goal of this study will be to develop a prototype that is able to classify firearm brands and subtypes. The introduction has proposed three hurdles to overcome in this process, namely the lack of a large dataset, the low inter-class, high intra-class variation and, lastly, the localization of the firearm in the picture. Accordingly, three research questions are formulated to study the effectiveness in dealing with these challenges.

3.1 Research Question 1

Does data augmentation improve the classification performance?

Although training machine learning algorithms such as the CNN requires a lot of data, techniques such as transfer learning and data augmentation are proposed to increase the performance on small datasets. Consequently, these techniques will be applied to develop a prototype that generalizes well to unseen data and does not overfit on the training data. Despite the fact that multiple methods of both data augmentation and regularization will be used, the main focus will be on data augmentation. Considering that, in addition to the augmentation of the dataset in terms of the size, data augmentation is also capable of increasing the variance of the firearm poses. This increase in poses might be especially useful given the earlier formulated requirements of a prototype that is capable of generalizing to multiple situations.

3.2 Research Question 2

Does the combination of Mutual Channel loss and Pairwise Confusion loss increase the classification performance?

Based on literature, it is hypothesized that a CNNs solely trained using Cross Entropy loss will not provide the expected results on the fine-grained task of classifying firearms. Accordingly, a novel approach is taken, combining two fine-grained loss functions to improve attention on fine-grained details. The first, Mutual Channel loss, is found to deal well with the high inter-class variance by enforcing attention on fine-grained details. The second, Pairwise Confusion loss, is proposed as a way to increase the low intra-class variance by adding intra-class noise to the loss. The current study expects these two loss functions might work well in tandem, since they both deal with a separate part of the high inter-class low intra-class variance problem of FGC.

3.3 Research Question 3

Does an object detection approach outperform a classification only approach?

A classification only approach performs well when the object is clearly presented in the image and the background noise is limited. However, the variety of applications that the Dutch police force suggested (see section 1.1.1) might present a problem for a classification only approach. Therefore, the current study will experiment with the use of an object detection pipeline that localizes the objects, followed by a classification algorithm that classifies the localized object. It is hypothesised that the classification will benefit from the object detection, given that the object detection algorithm is able to cut out much of the surrounding background noise by cropping the part of the image that contains the firearm. In addition, the cropped image patch could maintain a higher resolution than without the object detection, since the crop will be taken from the original full size image, which might further benefit the classification.

4 Methodology

This section will outline the procedure taken by the current study and will consist of four parts. First, the data used for this study will be described. Second, the algorithms trained on the data will be explained. Third, the metric used to evaluate these algorithms will be outlined and, finally, the training procedure followed to train these algorithms will be described.

4.1 Data

The current section will consist of two parts. First, the process of data collection will be outlined. Considering there was no dataset available online that included firearm brand and subtype annotation, part of this study involved around constructing datasets to train and test the classification algorithms. Second, in the dataset analysis section the data used in this study will be analyzed.

4.1.1 Data Collection

For the firearm classification task, three different datasets were constructed, each consisting of 15 different categories plus a negative category. The negative category is meant for unknown firearms that do not belong in any of the other 15 categories (see appendix 8 for the list of firearms in the negative category). Collecting these datasets was done in two ways. First, a dataset was constructed of images found through the search engines: Google, Bing and DuckDuckGo. This was done by entering the fully written official name of the firearms as a search query (see appendix 9 for the full names). This dataset will from now on be referred to as the Web Handgun dataset. Because search engines return a lot of noisy data (i.e. images that contain guns of a different category or no gun at all etc), the invalid images had to be removed manually from the Web Handgun dataset. This was done according to the following criteria:

- The image should contain the correct type of firearm.
- The firearm should be fully visible without obstructions.
- Only one firearm should be visible in the picture.
- The image should be bigger than 224 by 224.
- There should not be any distracting background noise in the image.

Second, two datasets were assembled from images taken in one of the Dutch police depots. Two types of images were taken in order to be able to evaluate the algorithms and check the generalizability of the algorithms in different applications (see Figure 7). The first of these two datasets consists of images that

contain a gun on a plain background, as shown by the two leftmost images. This dataset is called the Plain Police Handgun Images (PPHI) dataset. The second dataset consist of images of firearms being held by a person as seen on the rightmost images and will be referred to as the Handheld dataset. This combination of different kinds of datasets will offer a good evaluation of the performance of the algorithms in terms of attention to fine-grained details in addition to the capability of firearm localization in more ambiguous poses and locations. Furthermore, the PPHI and Handheld dataset contain the exact same firearms, whereas the Web Handgun dataset contains the same type of firearms but different variants.

Figure 7: Example of the PPHI (left) and Handheld (right) dataset.

4.1.2 Data Analysis

This section will explain the analysis of the datasets used for the current study, beginning with the classification datasets and finishing with the object detection datasets.

Classification For the classification algorithms, the three datasets mentioned in section 4.1.1 are used. After the process of deleting unqualified images, the Web Handgun dataset contains a total of $16,676$ images. The average height of the Web Handgun dataset images is 914.71 ($SD = 650.57$) and the average width is

1,202.89 ($SD = 870.02$). Next, the two datasets from the police depot are the PPHI dataset with a size of 2,459 and the Handheld dataset with a size of 1,668 images. The average height of the PPHI dataset images is 3,985.30 ($SD = 525.63$) and the average width is 5,120.92 ($SD = 733.95$). For the Handheld dataset the average image height is 3,672.0 ($SD = 0.0$) and the width is 4,896.0 ($SD = 0.0$). See Table 1 for an overview of the class distribution for all classification datasets.

Table 1: Class distribution of the three classification datasets

Firearm	Web	PPHI	Handheld
Colt M1911	572	131	37
Glock 17	428	183	97
Glock 19	310	274	198
Glock 21	291	68	98
Glock 26	335	81	99
Glock 30	307	41	99
Glock 34	316	118	99
Glock 45	217	51	97
HecklerKochUSP	111	116	146
Jericho941F	242	145	100
Negative	11,908	486	283
SigSauerP220	392	145	126
TanfoglioCG	134	127	37
WaltherP22	362	123	44
WaltherP99	443	188	42
WaltherPPQ	319	183	66
Total	16,676	2,459	1,668

A t-distributed stochastic neighborhood embedding (t-SNE) [42] test is conducted to visualize the Web Handgun and PPHI dataset in a 2D plain (see Figure 8). Moreover, the input used for the t-SNE is the output of the final convolutional layer of a ResNet50 trained on ImageNet. Comparing both datasets shows a significant difference between the type of images that the datasets contain. One thing that stands out is, that the PPHI dataset seems to contain more variances in term of camera angle as well as background. Most of the images of the Web Handgun dataset contain a white background and show the firearm in canonical view. The difference within the datasets are substantial as well, with the most salient factor being the background. It is noteworthy that the ResNet50 distinguishes the images mostly based on the type of background to a greater degree than for example rotation or firearm type. This factor might be a good indication that a non-fine-grained approach, which does not focus on enforcing attention to small details, might not work well on this data.

Object Detection To train the object detection algorithm, four datasets are used (see Figure 9). Since datasets labeled for segmentation task are scarce, only one dataset is found, being the segmentation part of the handgun category of the Open Image Dataset [25] containing 523 images. However, considering the

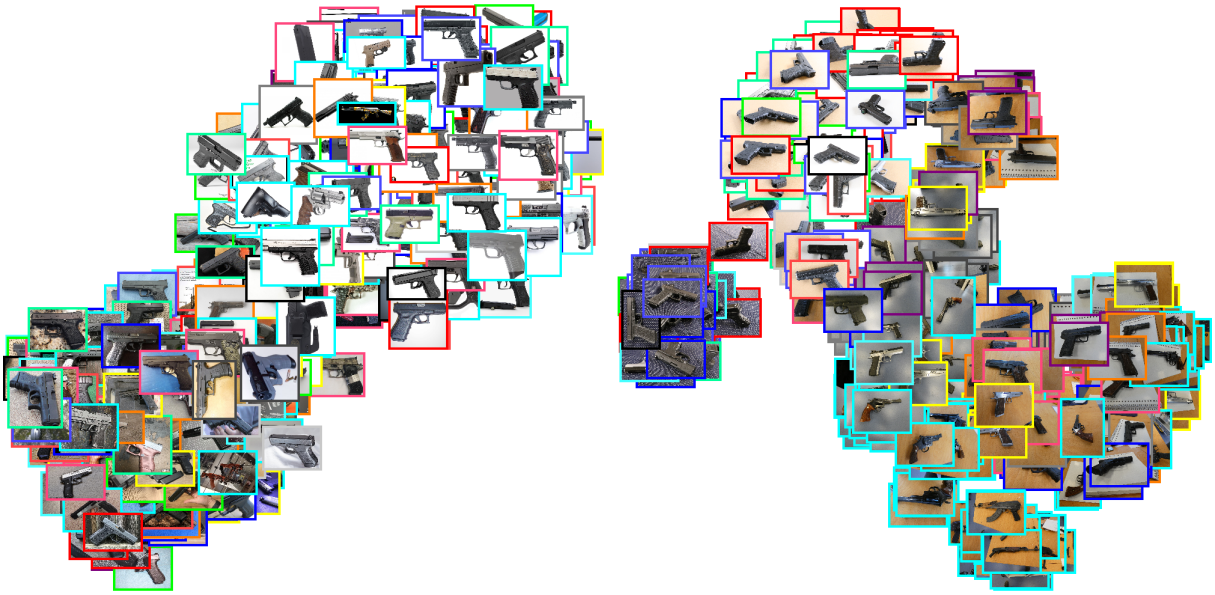

Figure 8: t-SNE plot of the Web Handgun (Left) and Police dataset (Right). (Zoom in to see better)

segmentation branch in the Mask R-CNN depends on the classification and bounding boxes branches, it can be beneficial to the algorithm as a whole to train on datasets without segmentation labeling as well. Therefore, three different datasets are utilized, only containing bounding box annotation for firearms. These datasets are: the bounding box part of the handgun category of the Open Image Dataset [25] with 1055 images, the firearms class of the ImageNet dataset [36] existing of 380 images and the pistol class of the Weapon Detection Dataset [31] of 3000. For an overview of the datasets see Table 2.

Table 2: Overview of the Object Detection Datasets with Annotation

	Size	Bounding Box	Segmentation
Open Image Dataset (Segmentation)	523	✔	✔
Weapon Detection Dataset	3000	✔	
ImageNet	380	✔	
Open Image Dataset (Detection)	1055	✔	

4.2 Algorithms

This section will outline the three different algorithms used in this study. The three algorithms used in the current study are the baseline, fine-grained and the pipeline algorithm.

Figure 9: Example images of the object detection datasets. From left to right: ImageNet, Weapon Detection, Open Images (Bounding Box), Open Images (Segmentation).

4.2.1 Baseline Algorithm

The baseline algorithm consists of a ResNet50, a state-of-the-art CNN, that has been proven to perform well on large-scale datasets as well as FGC datasets [36, 6]. Given its wide range of applications in the field of computer vision [16, 48, 28], the current study will use this algorithm as a baseline.

4.2.2 Fine-Grained Algorithm

Research shows that ConvNets trained solely with a classification loss function tend to perform worse on FGC datasets, since they do not look for fine-grained details and are more likely to overfit on sample specific features [12]. A lot of studies try to enforce attention through complex attention mechanisms, that add extra computational overhead [14, 54, 55]. Loss functions, on the contrary, try to enforce attention within the network without adding any overhead. In addition, loss functions seem to perform on par with the more complex attention mechanisms. Therefore, the current study will implement a combination of the Mutual Channel loss [6] and the Pairwise Confusion loss [12] to increase the algorithms' attention to fine-grained details.

The Mutual Channel loss is a loss function which is applied on the feature maps of the convolutional layers, directly enforcing them to be class-aligned and discriminative on mutual distinct local parts. The mutual channel loss is added to the classification loss. Consequently, to balance these two loss functions, a weighting parameters has to be selected. In addition, a value for the hyperparameter ξ will have to be considered. This hyperparameter determines how much filter channels will be dedicated to each category. Therefore, it should fit both the number of categories and the number of filter channels. The datasets contain 16 different categories of firearms, dividing the number of feature maps in the final layers by the number of categories should offer the best value for ξ. In the case of the ResNet50 the final layer consists of 2048 feature maps, meaning that ξ will be 128, and therefore 128 feature maps will be designated to each category.

The second loss function will be the Pairwise Confusion loss. This loss function is hypothesised to work well in combinations with the Mutual Channel loss. Whereas the Mutual Channel loss focuses on

capturing the low intra-class difference through focussing on fine-grained features [6], the Pairwise Confusion loss aims to deal with high inter-class variances by adding label confusion, such that the network is less likely to overfit on sample-specific features. The Pairwise Confusion loss adds an extra regularization term on top of the classification and Mutual Confusion loss. The regularization term is only added in case an image pair is not of the same label. Moreover, the amount of confusion depends on the distance of the two probability distributions that the network outputs.

4.2.3 Object Detection Algorithm

The final algorithm extends the classification with an extra object detection algorithm, to facilitate the localization of the firearm as well as the attention to fine-grained details. The first step of the pipeline is to localize the firearm in the original full scale image, at which point the firearm is cut out of the original image to remove the unnecessary background noise. Subsequently, the cropped image is used as input for the classifier algorithm which returns a probability distribution over the possible classes. An additional advantage of using object detection is that the crop is taken from the original full scale image. Therefore, the crop will have a high resolution, thereby preserving as much important details as possible.

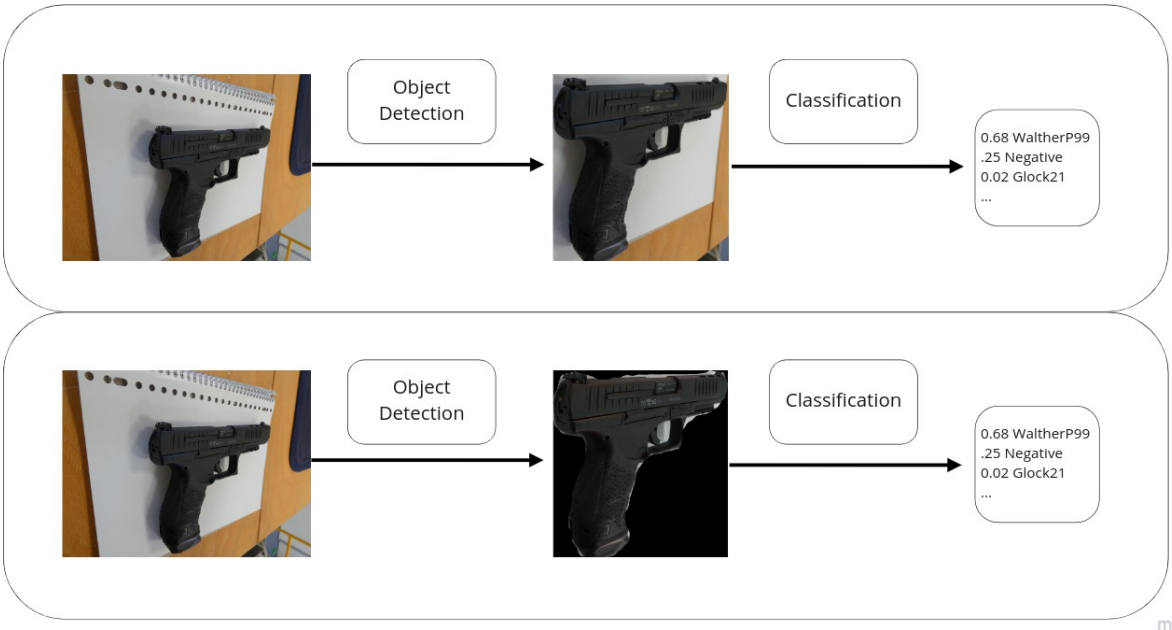

Figure 10: An overview of the bounding box method (top) versus the mask method (bottom).

To obtain the location of the firearm, a Mask R-CNN [19] is implemented and trained to detect firearms in images. The Mask R-CNN algorithm is capable of predicting bounding box locations as well

as generating a binary mask of the object on pixel level. This gives way to two possible methods for the cropping of the image. A bounding box crop or a segmentation crop (see Figure 10). Finally, the Mask R-CNN is pre-trained on the Coco dataset [26] which contains more than 2.5 million labeled instances in 328 thousand images. To fine-tune the algorithm, the Mask R-CNN is trained on the four different object detection datasets mentioned in section 4.1.2.

4.3 Metrics

4.3.1 Classification Metric

Classification of the firearms is the main objective of this study, and therefore it is important to use a metric that adequately evaluates the performance of each algorithm. Especially given that the class distribution of the classification datasets contains an imbalance. Accuracy may be a misleading metric when used on an imbalanced dataset and may incorrectly suggest an above-chance performance. Take for example the distribution of the Web Handgun dataset. When the accuracy score would be used, a classification algorithm that solely predicts the most occurring class would already achieve an accuracy of around 70%. Subsequently, the balanced accuracy metric [3] is used. The balanced accuracy score is calculated by taking average of the recall of each class, such that every class has the same amount of weight on the performance no matter the class distribution. Returning to the example, the previously mentioned classifiers' would achieve no more than a score of $\frac{1}{numclasses}$ using the balanced accuracy. This is clearly a better reflection of the classifiers performance over the total class distribution.

4.3.2 Object Detection Metric

Evaluating object detection algorithms is not as evident as evaluating classification algorithms. Object detection involves multiple steps, in terms of localization and classification of an object. Therefore, each step has to be evaluated separately. Given that this becomes inconvenient when comparing different object detection algorithms, some encompassing evaluation metrics are designed to make the comparison more suitable. The most prominent metric being the mean Average Precision (mAP) metric.

In order to correctly detect an object, three conditions have to be satisfied according to the mAP metric. First, the confidence score has to be higher than a certain threshold. The confidence score is a probability distribution containing all the possible classes plus a background class (for regions that do not contain an object). Second, the distribution of the confidence score must point to the ground truth and third, the bounding box and or segmentation mask must have an Intersect of Union (IoU) higher than a certain

threshold. The IoU is calculated by dividing the area of intersect by the area of union as seen in Equation 2.

$$IoU = \frac{\text{area of intersect}}{\text{area of union}} \tag{2}$$

When these three conditions are met, a prediction is categorized as a true positive, meaning that the prediction corresponds to the ground truth. However, if one of the latter two is violated, the prediction is evaluated as a false positive, meaning that the prediction states that a object is present while this is not the case. On the contrary, if an object is present, but the confidence score is lower than the threshold, the prediction is evaluated as a false negative, implying that the object was present, but not detected. These classification mechanisms can be used to define two important concepts for object detection: precision and recall.

- Precision measures the likelihood of a positive case actually being a positive case. It is calculated by dividing the true positive cases by all predicted positive cases (see Equation 3).

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{3}$$

- Recall measures the likelihood of a positive case being predicted as such and is calculated by dividing the true positives by the total of positive cases (see Equation 4).

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{4}$$

Improving either one of these two metrics decreases the other metric, since improving precision means setting a higher confidence threshold for positive prediction, thereby increasing the change of a false negative, while increasing the recall causes the effect the other way around. The Precision-Recall-curve models this trade off between recall and precision for different confidence thresholds. By calculating the area under the Precision-Recall-curve, the average precision is found. To smoothen the curve the max precision value is taken for any of the recall values. Finally, the mAP is calculated by taking the mean of the average precision of all the classes. Moreover, changing the IoU threshold changes the mAP and can be done to evaluate the mAP for different localization thresholds.

4.4 Training Procedure

The current section will describe the procedures followed during training for the classification and object detection algorithms.

4.4.1 Classification

The classification algorithms are trained using the Web Handgun dataset. A 90-10% split is created for training and validation purposes. Moreover, this dataset has a significant class imbalance, as can be seen in Table 1. Machine learning classifiers might perform worse when classes are imbalanced, since they tend to learn the class distribution instead of class specific features [21]. To cope with this imbalance, weighted sampling is used. Weighted sampling ensures equal sampling of all classes during training, meaning that the algorithms might see some of the images of small classes more frequent than classes of large size. In addition, the balanced accuracy score (see section 4.3.1) is used instead of accuracy as our main metric for classification. Before the images are fed into the algorithm, the images are resized and normalized according to the ImageNet pixel value distribution. Following these steps, data augmentation can be applied, depending on the experiment (see section 5). All classifiers use a ResNet50 pre-trained on the ImageNet dataset. Furthermore, all classifiers are fine-tuned on the Web Handgun dataset for 30 epochs using stochastic gradient descent. An initial learning rate of 0.001 and a learning rate scheduler which decreases the learning rate with a factor of 10 after 2 epochs without validation loss improvement. Moreover, a l2 regularization penalty of 0.0005 is used for all classification algorithms. Finally, during training a checkpoint is made each time the algorithm improves on the balanced accuracy score of the validation set. In the end checkpoint with the highest performance is used as the final algorithm.

4.4.2 Object Detection

The training of the Mask R-CNN is split up in two different phases, because two different types of data are used to train the model. For both of the training phases the original image size is used. Furthermore, data augmentation is used to artificially increase the dataset size. Data augmentation is done in such a way as to not remove any of the firearm objects from the image. The methods used are:

- Random horizontal flipping with probability .5
- Random vertical flipping with probability .5
- Random cropping with a maximal size of the original image size and minimal size of 448 by 448
- Random rotation between -45° and 45°.

In the initial phase, the three bounding box datasets are used to train the Mask R-CNN with a frozen segmentation branch. The ImageNet handgun class is split into two, where one half of the dataset is used as a validation dataset and one to train on. The Weapon Detection and Open Image (Detection) datasets are utilized as training data entirely. During phase an initial learning rate of 0.005 is used, with a learning

rate scheduler that decreases the learning rate every 10 epochs by a factor of 10. Lastly, l2 regularization of 0.0005 is used as a penalty. During the second training phase the segmentation branch is trained on the segmentation part of the handgun class of the Open Image Dataset. A 90-10% test-train split is used to train and evaluate the segmentation branch. During this stage all hyperparameters are kept the same, other than for the learning rate. The learning rate of the entire network is reduced to 0.0001, except for the segmentation branch, which starts off with a learning rate of 0.005.

5 Experiments

This study will evaluate three different approaches. First, the baseline approach, using a ResNet50. Second, the fine-grained approach, which implements specialized loss functions to increase the algorithms attention on fine-grained details. And third, the detection approach, which combines the classification algorithm with an object detection algorithm. All experiments, except for the initial baseline, are constructed in such a way that the hyperparameters of the best performing previous experiment are implemented in the next experiment. For the baseline, hyperparameters from the Pytorch transfer learning tutorial are used [9], because assessing all possible hyperparameters would be too time-consuming. Moreover, all experiments are run on an Nvidia Tesla V100 16GB.

5.1 Baseline Approach

The Baseline approach uses a ResNet50 to classify the firearms in a classification only approach. Initially, a ResNet50 will be trained without the use of data augmentation and an input size of 224 by 224. Next, the baseline approach will test two different cases. For the first case, which is data augmentation, the same initial ResNet50 is trained this time with data augmentation. The data augmentation methods used during the experiments are:

- Horizontal flipping
- Rotation between 90 and -90 degrees
- Cropping between 1 and 0.8 of the standard image size
- Random jitter of the brightness, contrast and saturation

For the second case, namely input size, a bigger image sizes will be used as input to test the effect of input size on the algorithms' performance. The input size that is tested is 448 by 448. Hypothesized is that both the data augmentation and the larger input image will have a positive effect on the performance of the algorithm.

5.2 Fine-Grained Approach

The Fine-Grained approach will test two different loss functions designed to cope with the fine-grained details of firearms. The first loss function that is evaluated is the Mutual Channel loss. This loss function is calculated over the output of the feature maps and added to the Cross Entropy loss with a certain weight. It is expected that the addition of Mutual Channel loss will improve the attention of the algorithm on important fine-grained details. Next, the Pairwise Confusion loss is evaluated. This loss function adds an euclidean

distance penalty to the loss in case two images are not of the same category. This is achieved by separating the output of a training batch in two and adding the penalty whenever two labels of the output do not correspond. Hypothesized is that the Pairwise Confusion loss will decrease the tendency to overfit on image specific features, which will work well in combination with the Mutual Channel loss. Furthermore, a grid search will be conducted to find the best combination of weights for the Mutual Channel loss (alpha) and the Pairwise Confusion loss (lambda).

5.3 Object Detection Approach

The last approach that will be tested is the Object Detection approach. This approach tests the combination of object detection and classification. The approach will evaluate two different cases. First, the approach will evaluate the performance of cropping the mask versus cropping the bounding box of a firearm. The crops cut out a lot of the unnecessary background noise in particular present in the Handheld dataset. In addition, the crops are taken of the full resolution image. Therefore, the image patch containing the firearm will have a higher resolution than when using the original image as input, which should aid the classification algorithm. Furthermore, crops taken from the segmentation mask are more precise, and therefore capable of removing more background noise, such as the hands that hold the firearm, compared to the bounding box crops. However, the mask crops can also have a downside, given that masks are more error prone, because the chances of removing important features is greater. Second, this approach will evaluate the benefits of training the pipeline in an end-to-end manner. This means that during training of the classification algorithm, the object detection algorithm will first detect the firearm in the image after which the classification algorithm will receive the cropped firearm image as input instead of the original image. The current study hypothesizes that training the pipeline in this end-to-end fashion might aid the algorithm to learn more fine-grained features.

Finally, the performance of the object detection algorithm on the PPHI and Handheld datasets will have to be measured. Since the test datasets do not contain any extra annotation apart from the category label, the performance measurement will have to be done by hand. This will be done as a binary classification. Given that fine-grained details of firearms matter a lot, the performance measurement will be done in a very strict way, meaning that the IoU (measured by hand) will have to be high in order to classify as a correct detection. In case an image crop clearly contains the firearm, a detection will be classified as true positive. On the contrary, crops that do not contain the complete firearm will be classified as a false positive. Crops that contain no firearm will be classified as a false negative and finally, when no crop is found and no firearm is present in the image the detection will be classified as a true negative. Given that almost every image

in the test datasets contains a firearm, true and false negatives should not occur often. Furthermore, the pipeline approach might aid the classification in most instances, but also makes the whole process more error prone, given that a faulty detection can scrutinize the classification. When the object detection algorithm performs well, this problem might be irrelevant. However, the current study will have to evaluate the effects of faulty detections on the overall performance of the pipeline, in order to find what problems impact the algorithm the most.

6 Results

This section will outline the results of the conducted experiments across all three approaches. As discussed in the Experiments (see section 5), all tests will be run in a stacked approach, meaning that every experiment will be run on the previous best experiment. This stacked approach is taken to reduce the computational burden of training a combinatorial number of algorithms, as well as to speed up the process. The best model is chosen based on the balanced accuracy score on the PPHI dataset, given that this dataset best demonstrates the capability of recognizing firearms. Furthermore, since localization on object level as well as on feature level, is a major part of the current study, the Gradient-weighted Class Activation Mapping (Grad-CAM) [37] will be used to visualize important learned regions for the prediction of the firearm categories. Grad-CAM is a technique to produce visual explanations of the otherwise black box algorithms. By tracing the gradients back into the network, Grad-CAM is capable of highlighting areas in the image that are important for the classification.

6.1 Baseline Approach

The baseline approach consists of three different conditions, namely a baseline, the use of data augmentation (DA) and the additional use of a larger image size (448 by 448). The results of the baseline experiments can be found in Table 3. The initial test with no data augmentation and an image size of 224 by 224 achieves a

Table 3: Balanced accuracy scores for all three test cases

	Web Handgun	PPHI	Handheld
Baseline	88.75%	38.91%	9.00%
Data Augmentation (DA)	90.41%	52.63%	8.41%
DA + Image Size (448 by 448)	**92.42%**	**65.46%**	**11.28%**

high balanced accuracy score on the Web Handgun dataset, but a poor score on both the PPHI and Handheld dataset. This is to be expected, as the Web Handgun has little variance in angles and background and is therefore easier to learn. The PPHI and Handgun dataset on the contrary, contain a lot of variance, thus fitting well to the Web Handgun dataset does not automatically mean good performance on the PPHI and Handheld datasets. Next, data augmentation is introduced to add more variance in rotation to the training set. Data augmentation significantly improves the performance on the Web Handgun dataset, but especially on the PPHI dataset. This increase in accuracy can be contributed to rotation and flipping transformations, which add variances in rotation that the training data previously lacked. Furthermore, the performance of the Handheld dataset decreases with a small amount, showing the inability to localize the firearms. Because the data augmentation condition has outperformed the initial baseline model, the next experiment, which is

image size, builds on the data augmentation experiment. The increased image size outperforms the other experiments on all three datasets, the biggest jump in balanced accuracy is again found on the PPHI dataset. Given that a larger image size means more information can be extracted from the image, the improvement in performance is not unexpected, especially since small details matter a lot in FGC. However, there is a downside to using a higher resolution being, higher resolution images increase the number of parameters and therefore also the computational cost.

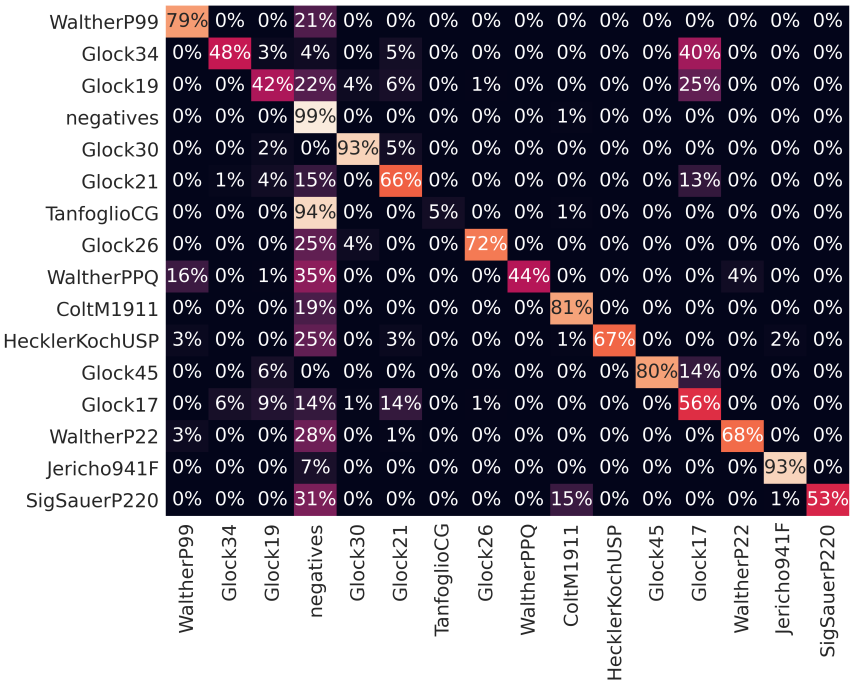

Figure 11: Confusion Matrix of the best performing baseline model on the PPHI dataset. Column presents the ground truth; Row presents the prediction of the algorithm.

Figure 11 shows the confusion matrix of the best performing baseline model (data augmentation + image size) on the PPHI dataset. There is still a lot of variety in the per-class performance of the algorithm. For example, the negative class achieves an accuracy score of 99%. On the contrary, a lot of categories perform poor and are often mistaken for the negative class, for instance: the WaltherPPQ, WaltherP22, SigSauerP220, TanfoglioCG and the Glocks. This might be due to the large number of different brands and subtypes that the negative class contains. Multiple types of the SigSauer, Tanfoglio, Heckler & Koch, Walther and the Glock are present in the negative category, which causes the negative class to have a lot of variance. The high variance of the negative category compared to the lower variance of the other categories might tilt the algorithm more towards predicting the negative class. Furthermore, two categories stand out, being the Glock34 and the TanfoglioCG. First, the Glock34 is often mistaken for the Glock17, but not the other way around. The visual similarities are plentiful; both guns have the almost same size and the same

caliber size and proportions. One visual difference that stands out the Glock34 from the Glock17 is the slightly increased size of the barrel that the Glock34 has. Given that the algorithm mistakes the Glock34 for the Glock17, but not the other way around, suggest that the algorithm learned to discriminate features that set apart the Glock17 and Glock34 from other categories, but did not entirely succeed in learning this one important discriminative feature of the Glock34 and is therefore, not capable of distinguishing the Glock34 from the Glock17. Second, the TanfoglioCG has an accuracy of 5%, which is way below the average of the best baseline approach. Apart from the TanfoglioCG, there are two other types of the Tanfoglio, that are present in the negative category. All types of the Tanfoglio contain a lot of visually similar features as well as a high inter-class variance, making the logo and the shape of the barrel (TanfoglioCG has a small extension on the end of the barrel) the most distinguishable features. In addition, the appearance of the TanfoglioCG in the trainset differs from the appearance of the TanfoglioCG in the testset. Section 6.4 will further investigate the poor performance of the TanfoglioCG.

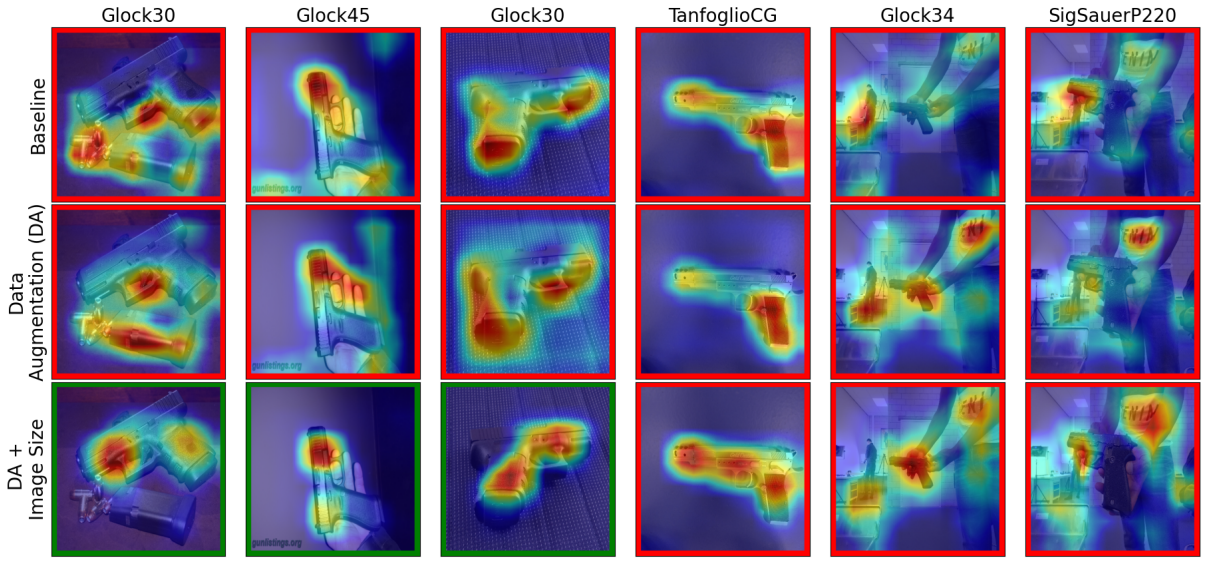

Figure 12: Grad-CAM of the experiments on the three datasets. Green boxes: correct prediction; red boxes: wrong prediction.

Figure 12 shows Grad-CAM heatmaps for the three conditions. The baseline demonstrates poor attention on details on all the samples but especially on the last two images from the Handheld dataset, where it fails to even localize the firearm. This is expected when looking at the poor performance of the baseline on the PPHI as well as the Handheld dataset. Surprisingly, the second algorithm, trained with data augmentation does not seem to be able to localize important features much better. It was hypothesized that the added variance in rotation by the data augmentation would increase the performance of the algorithm on

the PPHI and the Handheld dataset, which contain a lot of rotational variance. The unexpected result in terms of localization might be due to a lack of resolution, which inhibits the capability of good localization. This is hypothesized given the fact that the final approach, the combination of data augmentation with a larger image size, seems to improve the localization of fine-grained details significantly. The final experiment of the baseline approach shows good localization for all images except for the final two images of the Handheld dataset. When looking at the first four images of the Web Handgun and PPHI dataset, the improved attention on important regions such as the barrel and the pistol grip is visible, especially when looking at the two images of the Glock30. Therefore, it can be concluded that the combination of data augmentation with an increased image size can improve the attention on important fine-grained details in firearms.

6.2 Fine-Grained Approach

For the Fine-Grained approach, two experiments are conducted with different types of loss functions which should alleviate the high inter-class and low intra-class variance problem. First, the Mutual Channel loss (MC) is added to the previous best performing algorithm, the algorithm using data augmentation and an image size of 448 by 448. Table 4 shows the increased balanced accuracy score on the Web Handgun as well as on the PPHI dataset. Contrarily, the performance on the Handheld dataset decreases, which might

Table 4: Balanced accuracy scores for the fine-grained test cases

	Web Handgun	PPHI	Handheld
Previous Best	92.13%	65.46%	11.28%
Mutual Channel loss (MC)	93.39%	67.43%	10.52%
MC + Pairwise Confusion loss	**95.17%**	**69.69%**	**11.07%**

be attributed to difficulties in localizing firearms in the Handheld dataset. Considering the increase in performance of the Mutual Channel loss on the PPHI dataset compared to the previous best algorithm, the second loss function, the Pairwise Confusion loss, is combined with the Mutual Channel loss. The combination of Pairwise Confusion and Mutual Channel loss further increases the performance on the Web Handgun and PPHI dataset. Furthermore, the performance on the Handheld dataset has stagnated. These results confirm the expected improvement in terms of generalizability of the fine-grained loss functions. In addition, the conducted experiments demonstrate that the two utilized fine-grained loss functions perform well when combined.

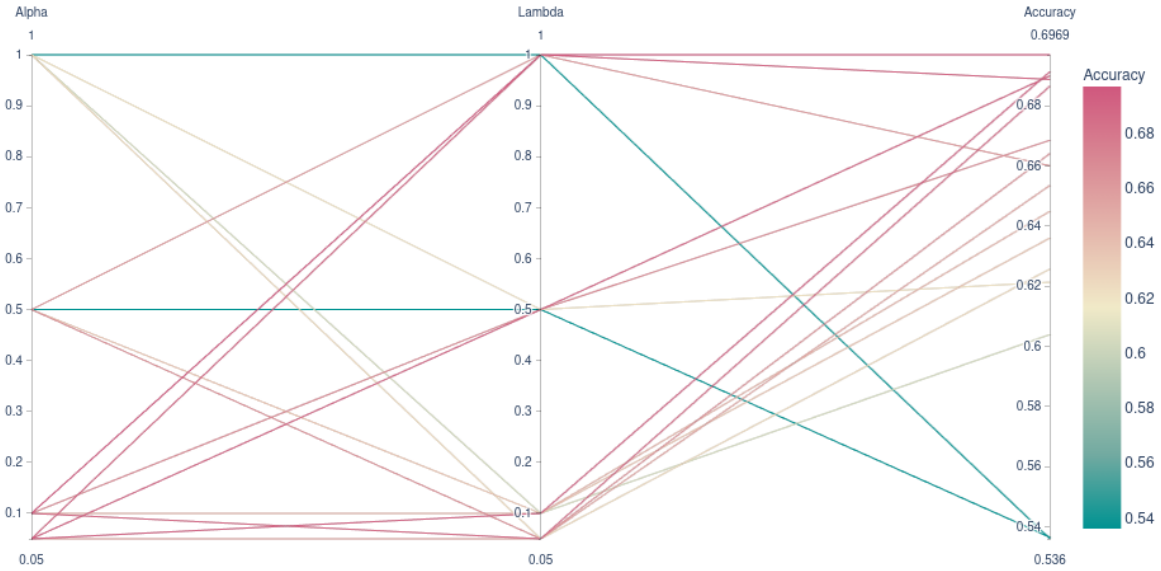

Figure 13: Balanced Accuracy for different values of Alpha and Lambda.

A grid search is performed to find the best weights for the Mutual Channel loss and the Pairwise Confusion loss. Figure 13 shows the optimal value Mutual Channel weight (alpha) to be 0.1, while the optimal value of the Pairwise Confusion loss is found to be at 1. Overall, the trend shows that a lower alpha between 0.1 and 0.05 is best for the Mutual Channel loss. On the contrary, the lambda used for the Pairwise Confusion loss does not seem to effect the performance as much, which is in line with findings the of Dubey et al. [12].

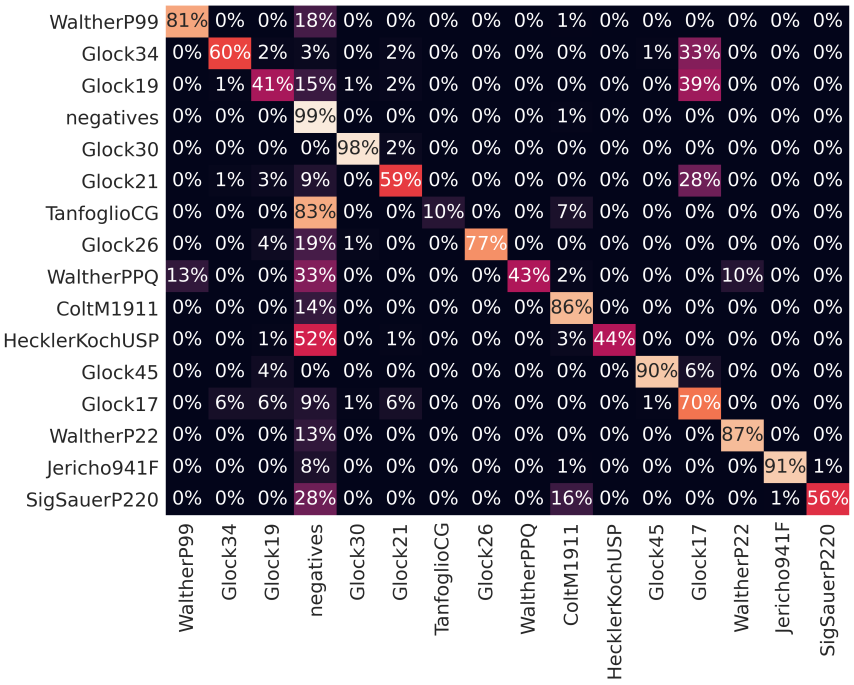

Figure 14: Confusion Matrix of the best performing baseline model on the PPHI dataset. Column presents the ground truth; Row presents the prediction of the algorithm.

The confusion matrix of the best performing algorithm is shown in Figure 14. On average, the per-class accuracy increased from the best baseline, as expected. However, there are classes that perform significantly worse than in the baseline approach, being the Glock21, the WaltherPPQ and the HecklerKochUSP. The HecklerKochUSP seems to be mistaken for subtypes of the Heckler & Koch inside the negative category. Besides, there seems to be a difference between the HecklerKochUSP in the PPHI and Handheld dataset compared to those in the Web Handgun dataset (see section 6.4). Interestingly, the WaltherPPQ is less often mistaken for a WaltherP22 or WaltherP99 than for a firearm from the negative category. Further investigation demonstrates that the WaltherPPS located in the negative class has a higher resemblance with the WaltherPPQ, than the other types of the Walther. Another interesting finding is that the fine-grained algorithm confuses other types of the Glock more often for the Glock17 than the baseline algorithm. This seems to be due to a high overlap in visual similarities of the different types of the Glock.

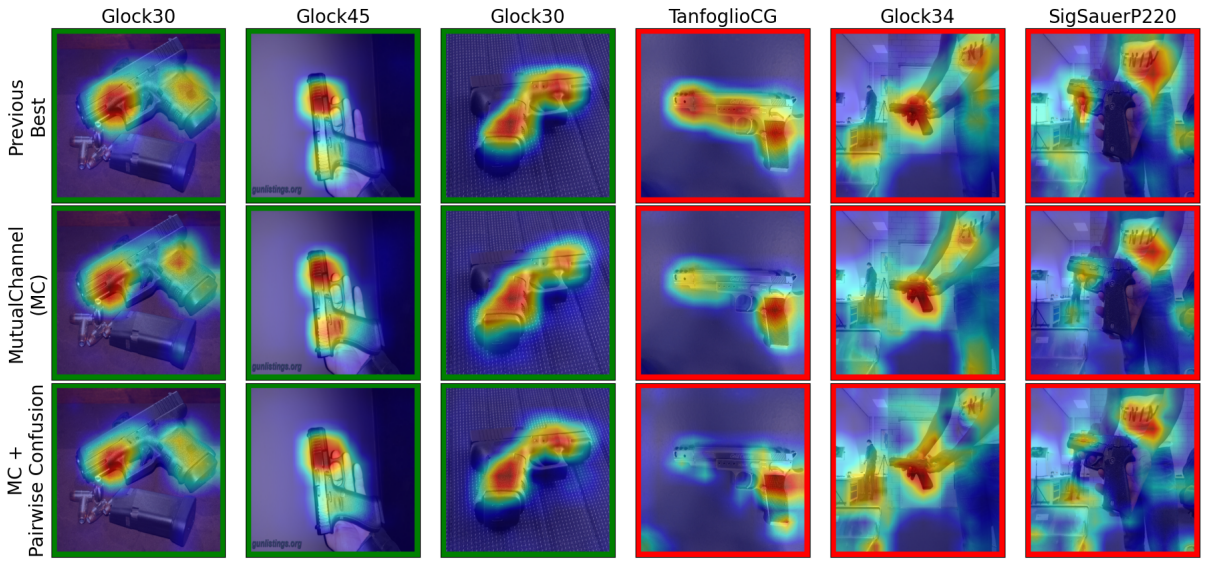

Figure 15: Grad-CAM of the experiments on the three datasets. Green boxes: correct prediction; red boxes: wrong prediction.

The Grad-CAM heatmaps in Figure 15 shows little improvements in terms of localization for the first experiment. Given that the experiment using the Mutual Channel loss only added a few percentages of balanced accuracy compared to the previous best experiment, it is understandable that no clear visual improvements in terms of attention to fine-grained details are seen. However, small improvements in terms of feature localization can be seen for the final experiment. This shows itself in more attention on the barrel, seen in the first three images of the two types of the Glock. Moreover, the barrel seems to be an important visual feature for firearm classification. Although the fine-grained loss functions seem to perform well in terms of attention to fine-grained details, the localization needed to perform well on the Handheld dataset still lacks. Moreover, it seems like the localization of firearms in the Handheld dataset is too difficult for a classification only approach and object detection is needed to solve this problem.

6.3 Object Detection Approach

The final approach to be tested is the detection approach, consisting of an object detection model in addition to the classification model. First, the results of the object detection model on the object detection datasets will be outlined, followed by the results of the entire detection pipeline on the three classification datasets.

6.3.1 Object Detection

The results on the handgun segmentation part of the Open Image Dataset can be seen in Table 5. Especially the bounding box results ($mAP_{50} = 94.1$, $mAP_{75} = 63.4$) are significantly higher compared to previous research ($mAP_{50} = 85.46\%$) [13]. Furthermore, the performance drops off when using a higher IoU threshold for the bounding boxes as well as the segmentation masks. This drop off might be due to the images in which the firearm only takes up a very small amount of space, which in turn makes in harder to locate the firearm more precisely. Consequently, this drop in mAP for high IoU thresholds should not pose a problem, since the firearms in the classification datasets all take up a significant part of the images.

Table 5: mAP on the validation part of Open Image Dataset

	Bounding Box	Segmentation
mAP_{50}	94.1%	94.3%
mAP_{75}	63.4%	65.0%

Table 6 shows the results of the object detection on the PPHI and Handheld dataset measured as a binary classification. All the measurements are done by hand and Figure 16 shows examples of good and bad detection. Only detections with a high IoU are classified as a correct detection, given that important fine-grained details could be left out by missing small but significant parts of the firearm. Therefore, the first two examples of the bottom images are classified as an incorrect detection even though the detection algorithm detected the majority of the firearm. Despite these strict criteria set by the authors, the results of the object detection on the PPHI and Handheld dataset are very promising and in line with the performance on the validation set. Furthermore, this shows that the object detection should not be a bottleneck in the pipeline.

Table 6: Confusion Matrix for PPHI (left) and Handheld (right) dataset

	Positive	Negative
Positive	2411	42
Negative	6	1

	Positive	Negative
Positive	1632	31
Negative	4	1

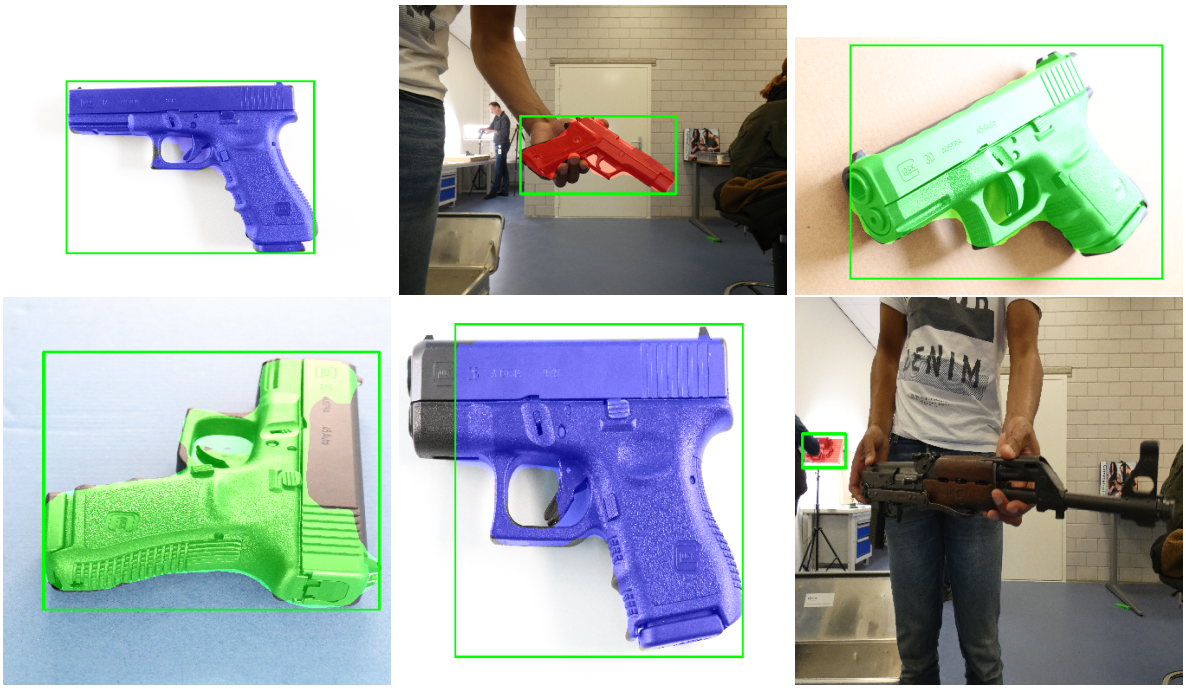

Figure 16: Examples of correct (top) and incorrect (bottom) object detection

6.3.2 Classification

The current section evaluates two cases. First, the bounding box and mask detection methods are compared to the best classification only algorithm and second, the two methods will tested in an end-to-end trained pipeline. The results displayed in Table 7 show a significant increase in balanced accuracy score for the

Table 7: Balanced accuracy scores for all three test cases of the pipeline approach

	Web Handgun	PPHI	Handheld
Previous Best	95.17%	68.28%	11.35%
Bounding Box	**98.03%**	**80.22%**	56.88%
Mask	94.82%	71.24	66.16%
Bounding Box e2e	92.57	78.02	72.58
Mask e2e	92.79	79.33	**73.74**

bounding box detection method on the PPHI dataset as well as a significant increase on the Handheld dataset. Beforehand, it was hypothesized that the algorithm would benefit from the added localization on the Handheld dataset, but the results on the PPHI dataset show that the algorithm benefits as well from the increased resolution caused by the cropping transformation in the pipeline. Furthermore, the mask method improves the balanced accuracy score for the Handheld dataset even more, although it does not outperform the bounding box method on the PPHI dataset. The bounding box method might perform better on this dataset, because the classification algorithm is not trained to receive a mask crop type input with variable

edges. On the contrary, the performance of the mask method on the Handheld dataset demonstrates that the mask input does have its benefits as it is capable of more precision. This precision of the mask might be capable of removing fingers and arms close to the firearm, which remain in the image when utilizing the bounding box method. The next experiment compares the two types methods trained in an end-to-end manner. The bounding box trained end-to-end (Bounding Box e2e) performs worse on the Web Handgun dataset compared to all previous methods, but does outperform most previous methods on the PPHI dataset, except for the previous bounding box approach. Furthermore, the Bounding Box e2e outperforms the normally trained detection methods on the Handheld dataset. Finally, the trained in a end-to-end manner (mask e2e) does improve on the bounding box e2e on all three the datasets. Although the balanced accuracy score on the Web Handgun, and to a lesser extent the PPHI dataset, still lags behind the standard bounding box method, the Mask e2e does significantly increase the performance on the Handheld dataset. Moreover, given that the e2e methods performed well on both the PPHI and Handheld dataset, it seems there is a considerable benefit to training the algorithm in an end-to-end manner.

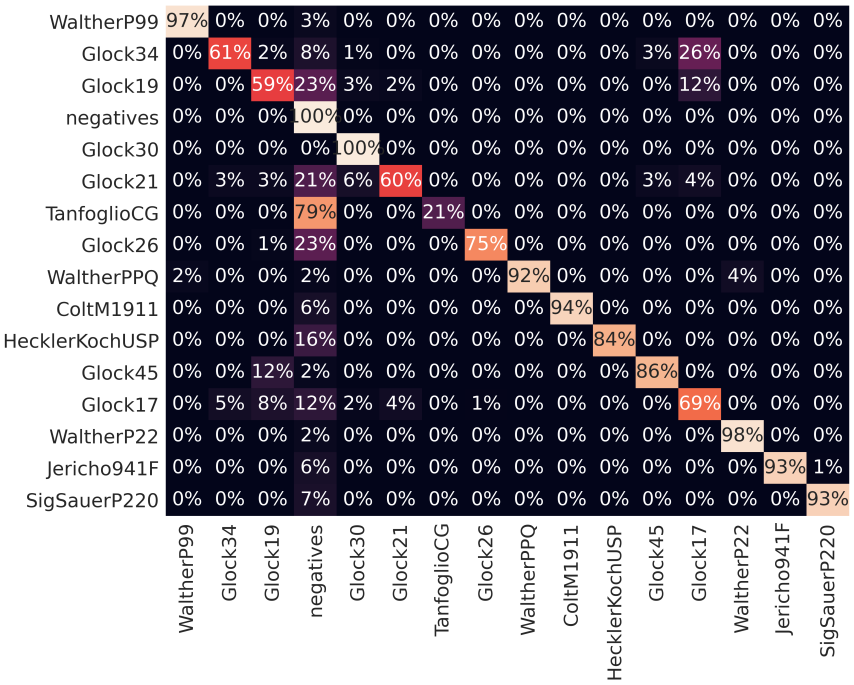

Figure 17: Confusion Matrix of the best performing baseline model on the PPHI dataset. Column presents the ground truth; Row presents the prediction of the algorithm.

Figure 17 shows the improvement in per-class accuracy for the best pipeline approach (standard bounding box method). The added detection algorithm seems to have given the classification an extra boost in terms of attention on fine-grained details, which shows itself in increased performance on almost all classes of the dataset. Furthermore, the performance for most of the different types of the Glock has improved.

Likewise, the amount of times any of the different types of the Glock are mistaken for a Glock17 has decreased significantly. Finally, the performance of the TanfoglioCG is still lagging behind. Although a ten percent increase is made by the detection approach, the majority of the time the TanfoglioCG is still classified as a negative firearm.

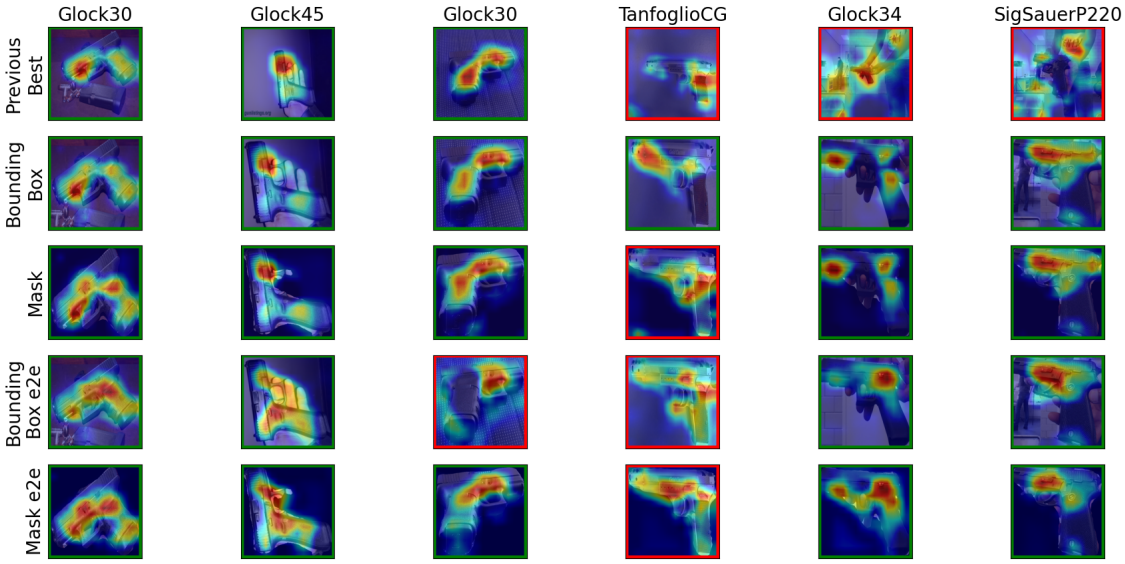

Figure 18: Grad-CAM of the experiments on the three datasets. Green boxes: correct prediction; red boxes: wrong prediction.

The Grad-CAM heatmaps are shown in Figure 18 for both test cases. For the standard detection methods, an improvement can be seen in the localization of objects, (see Glock34) as well as in the attention on important features (see WaltherPPQ, Glock34 and SigSuaerP220). This is in line with the improved results of the pipeline approach in comparison to the previous approaches. Consequently, it can be concluded that the classification algorithm benefits not only from the improved object localization, but also from increased attention on important features caused by the higher resolution of the images and less background noise. For the end-to-end trained methods there is a decrease in attention on details visible when looking at the first three Glock images. This is unexpected given the results achieved by the end-to-end methods. One possible explanation is that the object detection algorithm has a tendency to not fully detect the barrel of a firearm. Given that the barrel seems to be one of the most discriminative points, missing part of the barrel could complicate the process of finding small discriminative regions for the classification algorithm. This in turn might prevent the classification algorithm from learning fine-grained details especially present in the barrel.

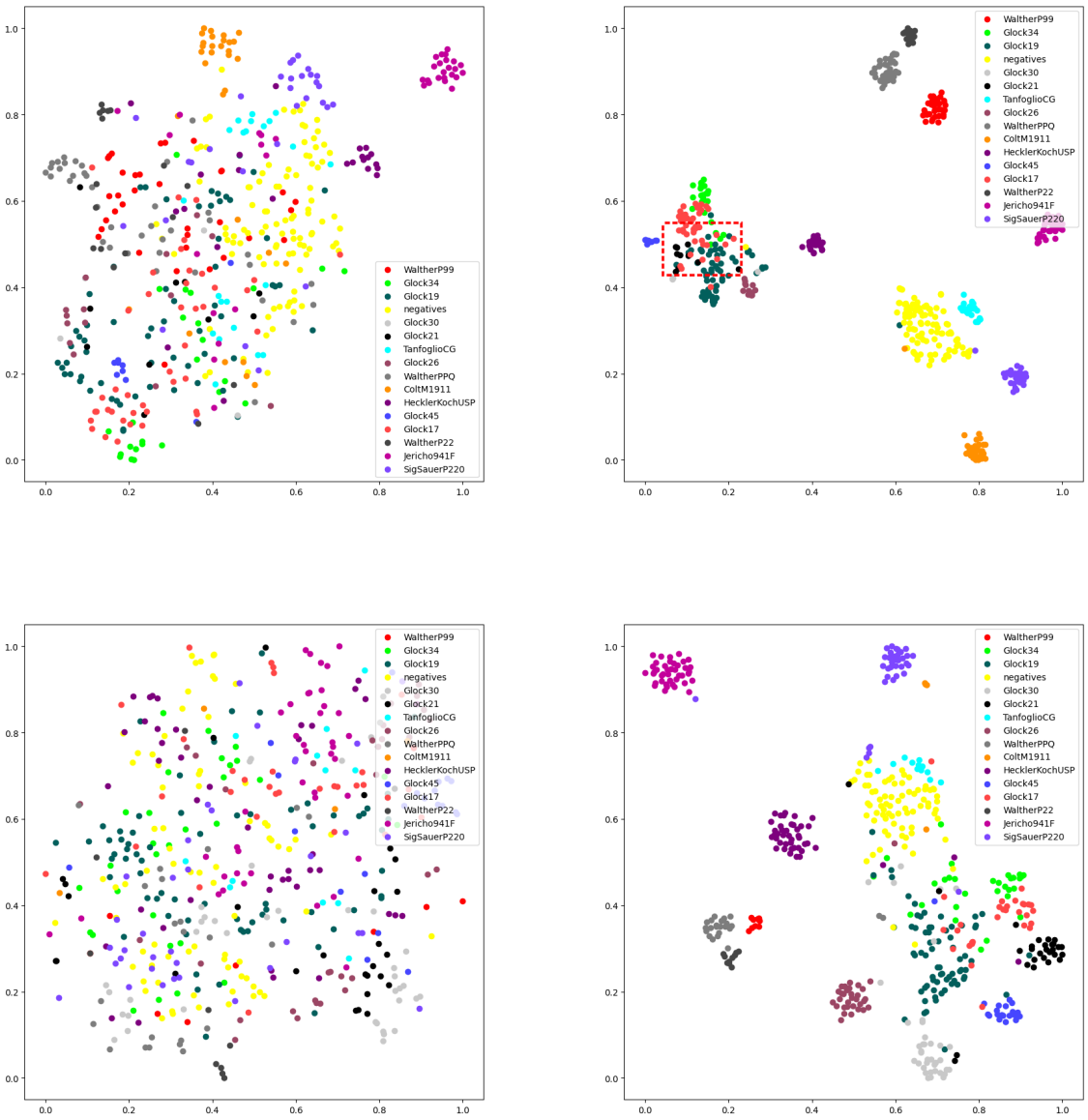

Figure 19: t-SNE plot using the final fully connected layer of the baseline (left) and best model (right) on the PPHI (top) and Handheld dataset (bottom)

Figure 19 shows the output of the t-SNE for the final fully connected layer of the baseline model and the best performing pipeline model (standard bounding box method) for both the PPHI and Handheld dataset. The first thing that stands out is the high overlap of the categories in the t-SNE plots of the baseline versus the relatively ordered best pipeline model. The plot of the baseline algorithm on the PPHI (top right) only shows two categories that are separated from the other categories, which are the Jericho941F and the ColtM1911. These two categories also are among the better performing categories, as expected. Moreover,

the plot of the baseline algorithm on the Handheld dataset (bottom right) demonstrates no order at all, as reflected by the baselines' performance on the Handheld dataset. The best pipeline algorithm shows a much more ordered picture compared to the baseline, where most categories except a few are nicely separated from each other. It should not come as a surprise that the Glocks are the only categories with a large amount of overlap. Especially the Glock17, 34, 21 and 19 demonstrate a lot of overlap. Moreover, the Glock17 seems to spread out over the Glock34 and the Glock19, which reflects the high amount of Glocks that are classified as a Glock17. Lastly, the TanfoglioCG appears really close to the negative category for both the PPHI and the Handheld dataset, which is reflected by the difficulty the algorithm has in distinguishing the TanfoglioCG from the different kind of Tanfoglios in the negative category.

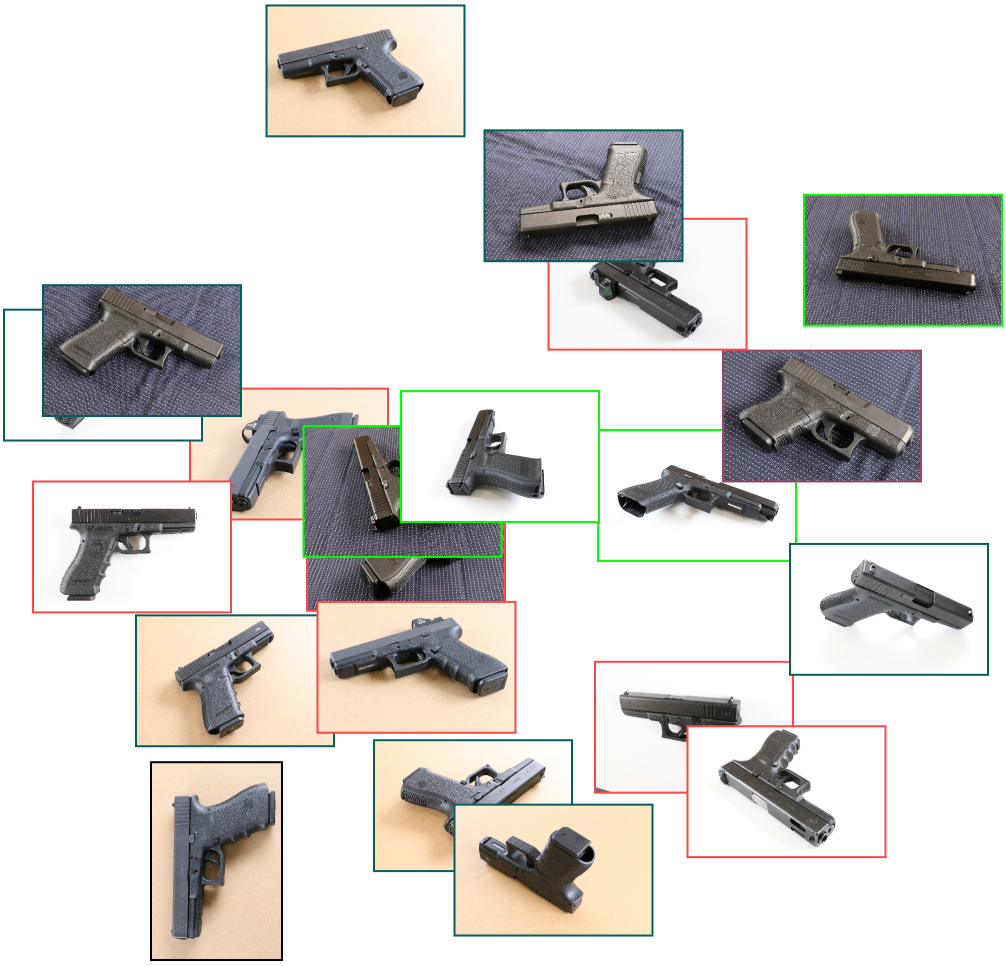

Figure 20: Zoomed in view of subregion of Figure 19

Figure 20 shows a zoomed in version of the best algorithms t-SNE output on the PPHI dataset, as seen in Figure 19. A few things stand out. First, the high amount of visual similarities is clearly the main factor is the high overlap between the Glocks, given that it is hard to find visual differences between the firearms. Secondly, the background plays less of a factor than for the ResNet50 trained on ImageNet (See Figure8). That is testimony to the addition of the fine-grained losses and the object detection, which improve the attention on fine-grained details. Finally, there seems to be a high inter-class variance in the Glock19, which causes the algorithms to mistake the Glock19 for a Glock17 and is also visible in the high overlap between the Glock17 and Glock19 in the t-SNE plots. See section 6.4 for more details on this matter.

6.4 Qualitative Analysis

In all three experiments there are a few categories that performed worse compared to the other firearm categories, such as the TanfoglioGC and some types of the Glock. One of the main culprits for the poor performance is the low inter-class variance and high intra-class variance that is especially evident for some bad performing categories. Figure 21 shows the different types of the Tanfoglio in the dataset with their visual differences and similarities. The TanfoglioGC found in the PPHI and Handheld dataset (top left) shows clearly more visual similarities with the two types of the Tanfoglio in the Negative category (bottom) than with the TanfoglioGC (top right) of the Web Handgun dataset. Given that all the algorithms are trained on the Web Handgun dataset it is unsurprising that the TanfoglioGC is so often mistaken for the negative category.

Figure 21: TanfoglioGC from the PPHI and Handheld dataset (top left) compared to the TanfoglioGC from the Web Handgun dataset (top right) and two different types of Tanfoglio's in the Negative category of the Web Handgun dataset (bottom)

To a lesser extent the same can be said for the HecklerKochUSP (see Figure 22). The HecklerKochUSP in the PPHI and Handheld (top left) dataset has a metal colored barrel making it less visual similar to the HecklerKochUSP in the Web Handgun dataset (top right). The Heckler & Koch categories in the Negative dataset (bottom) have on average more firearms with metal colored barrel. This mismatch, in addition to the already high similarity of the Heckler & Koch firearms makes it significantly more difficult to classify the HecklerKochUSP.

Figure 22: HecklerKochUSP from the PPHI and Handheld dataset (top left) compared to the HecklerKochUSP from the Web Handgun dataset (top right) and two different types of the Heckler & Koch in the Negative category of the Web Handgun dataset (bottom)

Figure 23: Example of two different types of Glock19 in PPHI and Handheld dataset

Figure 23 shows the two different types of Glock19 that exist in the PPHI and Handheld dataset. The two firearms show two visual differences, which are the groove at the bottom of the barrel and the notch in the pistol grip. As shown by the Grad-CAM images, the barrel and the pistol grip are both important regions for the classification of the Glocks. Furthermore, the version of the Glock19 without the barrel groove and notch in the pistol grip seems to be relatively sparse in the Web Handgun dataset, making it harder for the algorithms to learn the variances within the Glock19 category. In the Discussion (see section 7), the low overlap between training and test data will be further discussed.

7 Discussion & Conclusion

This final section will outline the conclusions that can be drawn from the formulated research questions, as well as limitations and possible future research encountered during this study and finally, recommendations for the Dutch police force.

The first research question states: Does data augmentation improve the classification score? The baseline approach has experimented with the use of data augmentation during training. It is found that using data augmentation does increase the performance of the baseline algorithm on both the Web Handgun and PPHI dataset. However, performance did not increase on the Handheld dataset. These findings show that although data augmentation does increase generalizability of important features, it does not necessarily improve the localization of such features in images that do not clearly present the firearm. This is substantiated by the Grad-CAM images in Figure 12, which do not demonstrate visible improvements in terms of localization.

The second research question states: Does the combination of Mutual Channel loss and Pairwise Confusion loss increase the performance? The current study hypothesized that the combination of Mutual Channel loss and Pairwise Confusion loss would perform well, given that they both focus on a specific part of the high inter-class low intra-class problem. This hypothesis is confirmed; as seen by the increased results of the combination of the two fine-grained loss functions compared to the Mutual Channel loss alone. Furthermore, a grid search was performed to find the best weight for each of the loss functions. One remark however, is the fact that the grid search is performed using transfer learning. Therefore, future research should analyze whether these weights are also optimal when training an algorithm from scratch. Moreover, future research should further look into whether this increase in performance holds for other datasets and algorithms as well.

The third research question states: Does an object detection approach outperform a classification only approach? It was hypothesized that the addition of an object detection algorithm would facilitate the classification of the firearm by reducing background noise and increasing the resolution of the firearm. This hypothesis is found to be true as both the bounding box method and the mask method outperformed the best previous algorithm (Mutual Channel loss + Pairwise Confusion loss) on all test datasets. The only remark is the performance stagnation of the mask method on the Web Handgun dataset. What stands out most is the improved attention to fine-grained details that both methods demonstrate. Removing all the background noise forces the network to pay attention to the fine-grained details of the firearm. Moreover, the major improvements on the Handheld dataset are a further testament to the contribution of object detection on the task of firearm classification. In addition, both methods are trained and tested in an end-to-end manner,

resulting in a significant increase on the handheld dataset. Although the results on the PPHI dataset slightly decreased, these end-to-end methods further closed the gap in performance between the datasets. Making the classification of a firearm held in the hand almost as effective as the classification of a firearm on a plain background.

7.1 Limitations & Future Work

All three approaches perform exceptionally well on the validation part of the Web Handgun dataset, compared to the PPHI and the Handheld dataset. One reason for the high performance on the Web Handgun dataset might be the high overlap in types of images that the Web Handgun dataset contains. A large portion of the images of firearms found online are thumbnails from online firearm shops. These images often present the firearms in a standard pose with a plain white background. This means that the Web Handgun dataset contains little variance and images in the validation part of the dataset are visually similar to images in the training dataset. Furthermore, although all exact duplicates are removed during preprocessing, quasi duplicates might still exist in the form of duplicate images containing different watermarks, aspect ratio or resolutions. One way to deal with this problem of quasi duplicates is to use a more aggressive duplicate removing process such as used by Wang et al. [46].

Whereas the Web Handgun dataset contains low variety in terms of backgrounds and poses, the PPHI and Handheld dataset contain low variety in terms of the number of different firearms used to create the datasets. The total number of different firearms in the PPHI and Handheld datasets is 21. This has the implication that categories in the test datasets consisting of a single firearm that does or does not comply visually with firearms of the same category in the training dataset can heavily influence the performance of the algorithms.

Future research should look into improving the algorithm by adding a rotational transformation to canonical pose to the pipeline. This transformation might further decrease the gap between the different amount of rotational variance in the Web Handgun and the PPHI and Handheld dataset. Given that the segmentation mask of the firearm is known, the pose could be deducted from this mask and the firearm could be rotated resulting in the realignment of the barrel with the x-axis. This might help bridge the gap between the performance on the Web Handgun and PPHI and Handheld dataset.

Both the Pairwise Confusion loss and the Mutual Channel loss performed best when applied to a Bilinear CNN [6, 12]. Future research should assess the novel combination of these two loss functions on the Bilinear CNN, to see whether the performance might increase.

7.2 Recommendations

This final section will offer recommendations for the Dutch police force to further improve the prototype. One such recommendation is to create a human-in-the-loop system. In such a system the user could directly provide feedback on false predictions the prototype makes. This type of feedback is very valuable in machine learning, given that the algorithms can directly learn from their mistakes on unseen data and prevent these mistakes from happening again. Another method that might drastically increase the performance of the algorithms is to limit the possible poses the firearms are allowed to take on in images. A lot of mistakes in classification could be caused by firearm poses that make classification of certain firearms impossible (i.e. by important features being out of sight). At the start of this study the goal was to classify firearms from a wide variety of poses, locations and backgrounds. However, during this study, the police started to lean towards limited the variety in order to obtain a higher performance. For an initial prototype, limiting the poses of the firearms appears to be the logical choice, given that the variety in poses contributes highly to false predictions. Lastly, it might be an option to combine hard to distinguish brands such as subtypes of the Tanfoglio and the Glock into a single category. This will result in sacrificing the classification of some subtypes in exchange for a more reliable prototype. Furthermore, sub algorithms could be created with the task to classify specific subcategories such as Glock for example. This sub algorithms could be trained to focus on very specific differences such as logos and text labels and could thereby improve the classification of hard to distinguish firearm brands.

7.3 Conclusion

Taken all together, this study has introduced an automatic firearm classification prototype for the Dutch police force. With the use of a new dataset containing different firearm brands and subtypes, this thesis has shown the viability of fine-grained methods for this novel task. Furthermore, this is the first study to combine the fine-grained loss functions, Mutual Channel and Pairwise Confusion loss, which has been proven to be effective. To enable the applicability of the prototype, object detection has been added, resulting in accurate predictions in a wide variety of settings. This software will contribute to decreasing the workload of Dutch police officers, by increasing firearm classification efficiency and speed.

References

[1] Anna Bosch, Andrew Zisserman, and Xavier Munoz. "Representing shape with a spatial pyramid kernel". In: *Proceedings of the 6th ACM international conference on Image and video retrieval*. 2007, pp. 401–408.

[2] Steve Branson et al. "Bird species categorization using pose normalized deep convolutional nets". In: *arXiv preprint arXiv:1406.2952* (2014).

[3] Kay Henning Brodersen et al. "The balanced accuracy and its posterior distribution". In: *2010 20th international conference on pattern recognition*. IEEE. 2010, pp. 3121–3124.

[4] Tom B Brown et al. "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165* (2020).

[5] John Canny. "A computational approach to edge detection". In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), pp. 679–698.

[6] Dongliang Chang et al. "The devil is in the channels: Mutual-channel loss for fine-grained image classification". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 4683–4695.

[7] Xinlei Chen and Abhinav Gupta. "Webly supervised learning of convolutional networks". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1431–1439.

[8] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. "Neil: Extracting visual knowledge from web data". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 1409–1416.

[9] Sasank Chilamkurthy. *Transfer Learning for Computer Vision Tutorial*. 2017. URL: https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html.

[10] Dan Ciregan, Ueli Meier, and Jurgen Schmidhuber. "Multi-column deep neural networks for image classification". In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 3642–3649.

[11] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee. 2005, pp. 886–893.

[12] Abhimanyu Dubey et al. "Pairwise confusion for fine-grained visual classification". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 70–86.

[13] M Milagro Fernandez-Carrobles, Oscar Deniz, and Fernando Maroto. "Gun and knife detection based on faster R-CNN for video surveillance". In: *Iberian Conference on Pattern Recognition and Image Analysis*. Springer. 2019, pp. 441–452.

[14] Jianlong Fu, Heliang Zheng, and Tao Mei. "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4438–4446.

[15] Ross Girshick. "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[16] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[17] Chunhui Gu et al. "Recognition using regions". In: *2009 IEEE Conference on computer vision and pattern recognition*. IEEE. 2009, pp. 1030–1037.

[18] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[19] Kaiming He et al. "Mask R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

[20] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[21] Nathalie Japkowicz. "The class imbalance problem: Significance and strategies". In: *Proc. of the Int'l Conf. on Artificial Intelligence*. Vol. 56. Citeseer. 2000.

[22] ML Kulthon Kasemsan. "The classification of gun's type using image recognition theory". In: *International Journal of Information and Electronics Engineering* 4.1 (2014), p. 54.

[23] Jonathan Krause et al. "The unreasonable effectiveness of noisy data for fine-grained recognition". In: *European Conference on Computer Vision*. Springer. 2016, pp. 301–320.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[25] Alina Kuznetsova et al. "The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale". In: *IJCV* (2020).

[26] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV].

[27] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. "Bilinear cnn models for fine-grained visual recognition". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1449–1457.

[28] Dhruv Mahajan et al. "Exploring the limits of weakly supervised pretraining". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 181–196.

[29] Subhransu Maji et al. "Fine-grained visual classification of aircraft". In: *arXiv preprint arXiv:1306.5151* (2013).

[30] Volodymyr Mnih et al. "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (2013).

[31] Roberto Olmos, Siham Tabik, and Francisco Herrera. *Automatic Handgun Detection Alarm in Videos Using Deep Learning*. 2017. arXiv: 1702.05147 [cs.CV].

[32] Patrick Poirson et al. "Fast single shot detection and pose estimation". In: *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE. 2016, pp. 676–684.

[33] Politie. "Zorgen bij politie en OM om aanhoudende stroom illegale vuurwapens". In: *Politie.nl* (Apr. 2020). URL: https://www.politie.nl/nieuws/2020/april/20/zorgen-bij-politie-en-om-om-aanhoudende-stroom-illegale-vuurwapens.html.

[34] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[35] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[36] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[37] Ramprasaath R Selvaraju et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[38] Connor Shorten and Taghi M Khoshgoftaar. "A survey on image data augmentation for deep learning". In: *Journal of Big Data* 6.1 (2019), p. 60.

[39] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[40] Stephen M Smith and J Michael Brady. "SUSAN — a new approach to low level image processing". In: *International journal of computer vision* 23.1 (1997), pp. 45–78.

[41] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[42] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[43] Jerry Vermanen and Tanne van Bree. "Legaal wapenbezit op laagste punt in 12 jaar". In: *pointer.kro-ncrv.nl* (June 2019). URL: https://pointer.kro-ncrv.nl/legaal-wapenbezit-op-laagste-punt-in-12-jaar.

[44] Catherine Wah et al. "The caltech-ucsd birds-200-2011 dataset". In: (2011).

[45] Hao Wang et al. "Cosface: Large margin cosine loss for deep face recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 5265–5274.

[46] Jiang Wang et al. "Learning fine-grained image similarity with deep ranking". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1386–1393.

[47] Yandong Wen et al. "A discriminative feature learning approach for deep face recognition". In: *European conference on computer vision*. Springer. 2016, pp. 499–515.

[48] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. "Wider or deeper: Revisiting the resnet model for visual recognition". In: *Pattern Recognition* 90 (2019), pp. 119–133.

[49] Lingxi Xie et al. "Hierarchical part matching for fine-grained visual categorization". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 1641–1648.

[50] Linjie Yang et al. "A large-scale car dataset for fine-grained categorization and verification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3973–3981.

[51] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems*. 2014, pp. 3320–3328.

[52] Han Zhang et al. "Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1143–1152.

[53] Ning Zhang et al. "Part-based R-CNNs for fine-grained category detection". In: *European conference on computer vision*. Springer. 2014, pp. 834–849.

[54] Heliang Zheng et al. "Learning multi-attention convolutional neural network for fine-grained image recognition". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 5209–5217.

[55] Heliang Zheng et al. “Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5012–5021.

[56] Jun-Yan Zhu et al. “Unpaired image-to-image translation using cycle-consistent adversarial networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.

8 Appendix A

The complete list of firearms in the negative category:

- Ruger GP100
- Glock 43*
- Heckler & Koch HK4
- Heckler & Koch USP Expert
- Heckler & Koch USP Match
- AK47*
- Ruger LCP
- SIG P320 M17
- Sig Sauer P320
- Sig Sauer P320 M17
- Sig Sauer P365
- Smith & Wesson model 442*
- Springfield Armory Hellcat
- Springfield XDs
- Tanfoglio Limited
- Tanfoglio Match
- Taurus G2c
- Walther P5
- Walther PPK
- Walther PPS

* These firearms are also present in the negative category of the PPHI and Handheld dataset.

9 Appendix B

The list of firearm category abbreviations in the datasets:

- **Glock17** - Glock 17
- **Glock19** - Glock 19
- **Glock21** - Glock 21
- **Glock26** - Glock 26
- **Glock30** - Glock 30
- **Glock34** - Glock 34
- **Glock45** - Glock 45
- **WaltherP99** - Walther P99
- **WaltherPPQ** - Walther PPQ
- **WaltherP22** - Walther P22
- **HecklerKochUSP** - Heckler & Koch USP
- **TangfolioGC** - Tangfolio Gold Custom
- **Jericho941F** - Jericho 941F
- **ColtM1911** - Colt M1911
- **SigSauerP220** - Sig Sauer P220