PUCRS

ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

MURILO SANTOS REGIO

AN EFFICIENT MODEL FOR IDENTIFYING FIREARM THREATS IN VIDEOS

Porto Alegre
2023

PÓS-GRADUAÇÃO - *STRICTO SENSU*

Pontifícia Universidade Católica
do Rio Grande do Sul

PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
COMPUTER SCIENCE GRADUATE PROGRAM

AN EFFICIENT MODEL FOR IDENTIFYING FIREARM THREATS IN VIDEOS

MURILO SANTOS REGIO

Master Thesis submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Prof. Isabel Harb Manssour

Porto Alegre
2023

Ficha Catalográfica

R336a Regio, Murilo Santos

An efficient model for identifying firearm threats in videos / Murilo Santos Regio. – 2022.

62 p.

Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientadora: Profa. Dra. Isabel Harb Manssour.

1. Surveillance. 2. Security Camera. 3. Computer Vision. 4. Weapon Threat Detection. 5. Model Compression. I. Manssour, Isabel Harb. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS com os dados fornecidos pelo(a) autor(a).

Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

MURILO SANTOS REGIO

AN EFFICIENT MODEL FOR IDENTIFYING FIREARM THREATS IN VIDEOS

This Master Thesis has been submitted in partial fulfillment of the requirements for the degree of Master in Computer Science, of the Computer Science Graduate Program, School of Technology of the Pontifical Catholic University of Rio Grande do Sul

Sanctioned on March 07, 2022.

COMMITTEE MEMBERS:

Prof. Cláudio Rosito Jung (PPGC/UFRGS)

Prof. Rodrigo Coelho Barros (PPGCC/PUCRS)

Prof. Isabel Harb Manssour (PPGCC/PUCRS - Advisor)

UM MODELO EFICIENTE PARA IDENTIFICAÇÃO DE EVENTOS DE AMEAÇA À MÃO ARMADA EM VÍDEOS

RESUMO

Para que uma sociedade prospere, seus membros devem se sentir seguros em suas vidas cotidianas; caso contrário, o medo começaria a tomar conta da população, causando estresse e pânico e, consequentemente, reduzindo a qualidade de vida. Diversas políticas e medidas costumam a ser adotadas para preservar a segurança das pessoas, mas a medida que a população cresce e armas de fogo se tornam mais acessíveis, a segurança da sociedade fica mais ameaçada. Preocupados com isso, diversos trabalhos buscaram explorar o uso de câmeras de segurança, uma das medidas de segurança mais utilizadas, e identificar um evento de ameaça. No entanto, esses trabalhos não possuem práticas comuns de comparação, conjuntos de dados padrão ou restrições para os conjuntos de dados usados. O principal objetivo deste trabalho é explorar métodos e estratégias para enfrentar o desafio da detecção de ameaça à mão armada, assumindo um cenário de sistema de vigilância com hardware limitado. Para atingir esse objetivo, buscamos redes neurais eficientes do estado da arte e técnicas de compressão de modelos para termos uma base sólida e estratégias bem desenvolvidas que pudessem melhorar ainda mais seu desempenho. Também propomos um novo conjunto de dados desafiador para identificar ameaças à mão armada que segue critérios rigorosos para garantir a qualidade dos dados utilizados. Até onde sabemos, o nosso é o maior conjunto de dados disponível na área com anotações para detecção de objetos e que usa apenas dados de mundo real. Nosso conjunto de dados está disponível online, juntamente com as ferramentas usadas para criá-lo, facilitando sua capacidade de expansão. Além disso, avaliamos o desempenho de alguns métodos do estado da arte nele, e os resultados obtidos corroboram sua dificuldade. Fornecemos um conjunto extenso de experimentos para demonstrar os pontos fortes e fracos de cada abordagem e seu impacto nas detecções. Também realizamos experimentos em diferentes

ambientes para avaliar como essas abordagens se comportavam em diferentes condições de hardware. Também evidenciamos quais são mais vantajosas ou mais versáteis e que melhor funcionam em nossos cenários.

Palavras-Chave: vigilância; câmeras de segurança; visão computacional; detecção de ameaça à mão armada; compressão de modelos.

AN EFFICIENT MODEL FOR IDENTIFYING FIREARM THREATS IN VIDEOS

ABSTRACT

For a society to prosper, its members must feel safe in their everyday lives; otherwise, fear would start to take over the population, causing stress and panic and, consequently, reducing the quality of life. Several policies and measures are usually adopted to preserve people's security, but as the population grows and firearms become more accessible, society's security becomes more threatened. Concerned with this, several works sought to explore the use of security cameras, one of the most commonly used security measures, and identify when a threatening event occurs. However, these works do not have common comparison practices, standard datasets, or constraints for the datasets used. The main goal of this work is to explore methods and strategies to address the challenge of firearm threat detection while assuming a scenario of a surveillance system with limited hardware. To achieve this goal, we sought well-known efficient neural networks from the state-of-the-art and model-compression techniques to have a solid basis to start from and well-developed strategies that could further improve their performance. We also propose a new challenging dataset for identifying firearm threats that follows rigorous controls to ensure the quality of the data used. To the best of our best knowledge, ours is the largest dataset available in the area based on frame-level annotations and that uses only real-world data. Our dataset is available online, alongside the tools used to create it, making it easier to expand it further. Moreover, we evaluated the performance of some state-of-the-art methods on it, and the obtained results corroborate with its difficulty. We provide an extensive set of experiments to present clearly each approach's strengths and weaknesses and their impact on the detection performance. We also conducted experiments on different environments to evaluate how these approaches performed on different hardware conditions. We also clarified which ones are most advantageous or are more versatile and work well on our scenarios.

Keywords: surveillance; security camera; computer vision; weapon threat detection; model compression; .

CONTENTS

1. INTRODUCTION

Security has always been a major concern for human beings within society, and as the war industry develops and more people have access to firearms, the more fragile the security of society becomes [54, 80, 33]. This relationship exists because firearms enable a single person, without the need for advanced training, to be able to cause a worrisome amount of fatalities, especially in public environments, leading to situations such as school shootings [25] and mass shootings [45]. Several measures have been implemented to combat this type of event, among which the most common is monitoring these environments using security cameras.

Although the usage of security cameras presents advantages [67], such as recording an event to be analyzed later, a significant disadvantage that this method has is the constant need for supervision. The cameras alone only record the event, requiring someone to monitor the recordings to notify the authorities if an event occurs. In addition, in most cases, just one security camera is not enough to cover, for example, an entire public space or the various environments of a building. Therefore, it becomes necessary to use and monitor several cameras recording 24 hours a day, seven days a week, making the monitoring process more difficult and creating many chances for distractions and other human errors [12].

Some qualities such as maintaining concentration and remaining alert for long periods are required for an effective CCTV-based vigilance [15]. However, CCTV operators may not realize their attention levels have dropped, especially when dividing their attention across multiple tasks. When analyzing the attention level of the operators, the work proposed by [87] shows that the operators can maintain their focus for 20 minutes on average. After that, they start missing details in the footage. The work developed by Ainsworth [2] goes further, showing that the operators miss around 45% of the details in the scene after 12 minutes. This value goes up to 95% of details overlooked after 22 minutes.

The initial solution to relying less on the operator's undivided attention was the proposal of semi-automated monitoring systems. These approaches studied the most common human errors during CCTV vigilance and proposed solutions trying to work around those issues [11]. However, they were part of a complex process since they required studying the problem to be solved, the operator's performance, and how to minimize human faults. So, as Computer Vision methods advanced and results became more reliable, fully automated monitoring systems became more and more prevalent [14].

Many works, have addressed this issue recently, some focusing less on the data used and more on the models and results achieved in the Firearm Detection task, as can be seen in the work proposed by [23, 88, 13]. In other cases, the data used had a particular goal, as, for example, the works that address Concealed Weapon Detection, as proposed

by [70, 40, 36]. Finally, some researchers do not focus on the task by itself, but they address multiple events at once, such as detecting abandoned luggage [49], fire [55], and violence in general [64].

Several authors have expressed concern with the available datasets' quality, most of those being discussed as future work [60, 47], while only a few propose new datasets or improvements. Moreover, those who did could not satisfy the requirements stipulated by the research area. We have also identified that many works express concern with performance in their future work, given its importance in real-life situations. However, all these discussions considered the availability of plentiful computational resources and the usage of GPU accelerators when estimating the model's performance. However, small-scale security systems, such as middle-class residences or small stores, will not have these resources available most of the time. For cases like these, solutions must be projected with these restricted conditions to achieve the desired performance.

With these restrictions in mind, we decided to focus our research on CPU-based systems with no hardware accelerators. Addressing these systems makes our model more accessible and usable by people that can not invest a large portion of their earnings in large surveillance systems. With these restrictions, in ideal cases, we would like to achieve a performance of at least $10 fps$, since we identified that surveillance systems usually record between $15 fps$ and $5 fps$. However, in more extreme cases of limited hardware, we would be satisfied with achieving a performance close to $5 fps$. Thus, achieving either of those performance levels would enable our model to process most of the recorded frames as they were made available, satisfying our near-real time goals.

Considering this, the main goal of this work is to perform near-real time firearm threat detection on systems with few computational resources available. Moreover, while most works identified in the literature address the issue of Firearm Detection, by locating the weapon itself, we decided to explore a slightly different challenge: firearm threat detection, which consists of identifying the human wielding the weapon. We decided to make this distinction because, in real-world data, the weapon might not be entirely on display (due to camera angles or occlusion, for example). However, by analyzing body language, we may be able to recognize the threatening individual. Motivated by this, we developed a novel and flexible benchmark dataset for firearm threat detection to address and fill this gap. Our dataset was created with real-world scenes by following a defined procedure for image selection and annotation. We also developed a set of tools to facilitate its extension.

Thus, our model should be capable of assisting in monitoring different spaces, public or private, by processing videos taken by security cameras and identifying events that compromise the security of those environments. We expect that by identifying that a dangerous event is starting, it is possible to notify those responsible for monitoring the environment, who can then alert the authorities to take immediate action, preserving the innocent's safety before the event escalates to a more dangerous level.

This work's contributions can be summarised as follows:

- Introduction of a novel challenging dataset with 6942 images of real-world situations, called FiDaSS (Firearm Dataset for Smart Surveillance). The dataset images are annotated for object detection encompassing both the assailant and the held weapon. Our dataset is different from the ones presented in the literature by containing real-world scenes from various scenarios and cultures, more detailed annotations, and novel data representing COVID-19 preventive measures.
- Assessment of the quality and difficulty of FiDaSS using state-of-the-art neural network architectures.
- Proposal of a challenging scenario as a case study to validate our model by simulating a small and low-investment surveillance system without specialized hardware. We also provide an evaluation of various model compression techniques in our case study, comparing both their performance and detection results.

The remainder of this work is organized as follows. Chapter 2 presents some important concepts, challenges, and techniques in the area. Following that, in Chapter 3, we describe the literature review procedure we followed, our insights gained from it, and the gaps identified in the area. Next, Chapter 4 presents in-depth each step followed to develop this research. In Chapter 5, we discuss in detail each step followed to create our dataset, the tools used, and statistical information about our dataset and comparisons to the ones existing in the literature. Then, in Chapter 6, we address how we planned our proposed model, the architectures we explored, and the techniques we used for our experiments. We discuss the results achieved in Chapter 7, including whether the chosen architectures can learn and generalize our dataset well, how the performance of these architectures is affected by the compression techniques chosen, and the performances achieved on our case study. Finally, Chapter 8 addresses our contributions, limitations, and plans for future work, followed by Chapter 9, which contains our conclusions and final remarks.

2. BACKGROUND

This chapter describes some important concepts and techniques that are relevant to the issue this work addresses. Computer vision, neural networks, and model compression are examples of content covered in the following sections.

2.1 Techniques using Handcrafted Features

This section addresses some strategies and Computer Vision techniques that are commonly employed on solutions for automated Firearm Detection. The first one simplifies the problem by dividing it into smaller sub-problems. This technique is known as Sliding Window [62] and consists of methodically extracting patches from the image, which are then evaluated separately, and the results of the patches can be joined and mapped to the original image accordingly. Sliding Window approaches are well suited for tasks that rely heavily on multi-scale local features but should be avoided when global features are more important. Although optimization strategies can be employed to minimize the number of patches extracted and analyzed, the technique is very computational heavy and requires many resources

The second technique discussed is called Bag Of Features [10], which consists of a vocabulary of features formed from a set of training images. Each feature on the bag represents descriptors of a local area of a training image, using information extracted by methods such as Zernike Moments [89] and Gabor Filters [22], which leads to an extremely large amount of features. Thus, to reduce the number of features and generalize them so that they can be applied to new images, it is required to apply a clustering step so that the features can be summarized in a “visual vocabulary”. When evaluating a new image, it is only necessary to match the new features extracted to the nearest matching cluster centers in the vocabulary. This technique is best fitted to image classification and retrieval tasks based on similar images, and it is not recommended for tasks that require a semantic understanding of elements in the image, such as object detection and keyword-based image retrieval.

Another important technique to be described in the context of this work is the Scale Invariant Feature Transform [51] (SIFT) technique, which expands on the existing Bag Of Features techniques. The technique consists of four steps: Keypoint Localization, Keypoint Filtering, Orientation Assignment, and Keypoint Descriptor. In the first step, to find candidates Keypoints in the image, a Difference of Gaussian is applied, as a less-expensive alternative to Laplacian of Gaussian, using small and large kernels to identify potential keypoints independent of their scale. In the next step, keypoints mainly consisting of edges and

low-contrast areas are removed, leaving only strong interest points. Then the orientation of each keypoint is extracted, and copies of each keypoint are created, but with different orientations, making the descriptors less dependent on the orientation of the features. And finally, in the last step, the feature descriptors of each keypoint are extracted and clustered. Similar to the Bag of Features technique, features are matched to the nearest matching centers, but in this case, some additional checks are suggested to reduce the number of false matches. Being an improvement over the Bag Of Feature technique, the SIFT technique is recommended or not for the same tasks. This technique is preferred over the previous since it is robust to different illumination levels, scales, and local affine distortions in the recommended cases.

The last technique presented is the Speeded-Up Robust Features [4] (SURF) technique, which was proposed as a faster alternative to SIFT, being almost three times faster than it. The first change proposed by SURF is to substitute the Difference of Gaussian for the Box Filter convolution, which can take advantage of parallelism when computing different kernel sizes. The next change proposed is the orientation extraction by computing the wavelet responses of each keypoint, while also providing an alternative method, called Upright-SURF, that skips the Orientation Assignment step, speeding up applications that do not require considering multiple orientations. The SURF approach also provides the option of using a more compact descriptor, reducing the dimensionality of each feature. Another step proposed during the matching process is to only compare features with similar contrast, reducing the number of comparisons required. This technique is similar to SIFT and is recommended for the same tasks, but it improves the algorithm's performance. But which one should be chosen between SIFT and SURF depends on the objectives to be achieved: if performance is a major concern, then SURF is the preferred technique; otherwise, if accuracy is more important, then SIFT is the preferred one.

2.2 Neural Networks

It is widely known that Neural Networks have an enormous influence on modern-day Computer Vision [27], having achieved state-of-the-art results in many different challenges. Neural Networks fulfill primarily three main tasks in Computer Vision [19]: Image Classification, Object Detection, and Image Segmentation. Each of those tasks has its own scientific implication, but with ascending complexity.

The image classification task focuses on discerning if the image contains a characteristic of interest, such as a person or a person performing a specific action. This task is responsible for many important advancements in the area, resulting in architectures such as the Visual Geometry Group [81] (VGG) and ResNet [31]. The task's state of the art is, most commonly, evaluated based on the Imagenet database [77], where some of the current

best-performing approaches are the Meta Pseudo Labels [66] and Big Transfer [41] models. Although this task is simple, it is the basis of the more advanced ones, where the proposed solutions are used as the backbone of more complex tasks.

The object detection task focuses on identifying the position in the image that contains instances of interest. This task, extending the previous examples, finds a bounding box that indicates precisely where a person is located on the image or where there are people performing actions of interest. Using AlexNet [42] as its backbone, the R-CNN architecture [24] was one of the first works to address this task with neural networks, proposing a two-stage process to perform the detections. The architecture underwent a series of modifications, later on turning to the Faster R-CNN architecture [74], which is still one of the architectures in the literature with the best results. But being dependent on two stages makes the R-CNN family computationally heavy and slow. With this in mind, architectures such as You Only Look Once [72] (YOLO) and Single-Shot Detector [48] (SSD) were created. These architectures unified the backbone and the bounding box computation into the same process, highly improving performance but providing slightly worse results. As the next class of difficulty following the classification task, this task is significantly more challenging but provides more exciting results and applications.

Finally, the segmentation task tries to identify instances of interest on a pixel level of precision, i.e., instead of giving the position, the task seeks to identify each pixel that constitutes each instance. To address this task, one of the initial proposals introduced the strategy of Fully Convolutional Networks [50], where the authors started with architectures proposed for image classification (such as VGG) and changed all the fully-connected layers to convolutional layers. Although the model did not achieve the ideal results, it provided important insights necessary to develop new approaches. Segmentation is far more challenging than the previous tasks, which is evident from the fact that there is a lot yet to be improved even with all advances being proposed.

From the presented tasks, the object detection task is the best suited for us, in the context of this work, as it encompasses well all our goals when processing the frames. Alongside the task, we presented briefly important architectures in its history and some interesting one-stage architectures for object detection. In the remainder of this section, we will discuss those one-stage architectures more in-depth and introduce some other relevant architectures for single-shot object detection.

The first model we will discuss is the YOLO architecture [72], whose first proposed architecture is shown in Figure 2.1. This architecture went through many changes and optimizations in the last years [71, 73, 6], but we will focus on describing the initial strategies employed to create the model. The main contribution proposed by the YOLO model is dividing the image in an arbitrary grid, where each cell in this grid makes a set number of bounding box predictions alongside predicting the probability of each class being present in that cell. Then, by combining the information from the bounding boxes with the probability

predicted by the cells, the model can provide a confidence score that can be translated into the probability of each box containing the predicted class. By using a single processing stage, this model can save a lot of time when compared to the two-stage detectors previously discussed. To illustrate this difference, the original YOLO model was six times faster than the Faster R-CNN model [74], although having less accurate results.

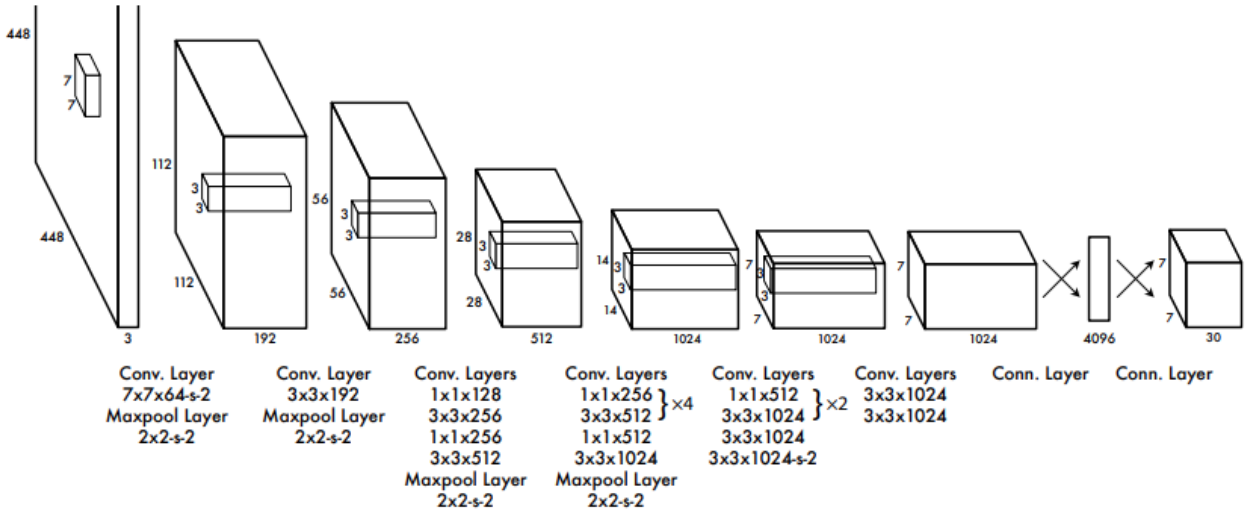

Figure 2.1 – Diagram representing the scheme of the original YOLO architecture.

The next model we will address is the YOLOv3 Nano architecture [92], illustrated in Figure 2.2. This architecture proposes a highly compact deep convolutional neural network based on the design principles of YOLO architectures. The authors also followed a process of Machine-Driven Design Exploration to determine the optimal micro-architecture that meets the original design requirements and a set of constraints. The three design constraints used by the authors were: (i) the architecture should achieve $\geq$ 65% mAP on VOC 2007; (ii) the computational cost should be $\leq 5B$ operations; (iii) the weights should abide by 8-bit precision. Compared to the YOLOv3 architecture, YOLOv3 Nano is approximately 8.3$\times$ smaller and requires 17% fewer operations while increasing the mAP by more than 10%.

Another model we will address is the SSD architecture, illustrated in Figure 2.3. This architecture proposes using a sequence of feature maps with decreasing size, allowing for multi-scale detections. Each of those extra feature maps can produce a set of detection predictions using convolutional filters. Finally, this architecture also proposes the usage of "default boxes" on the initial feature maps, with a similar purpose to the *anchor boxes* proposed by the Faster R-CNN architecture, but the default boxes are applied in different shapes and multiple feature maps. Apart from the aforementioned advantage of using a single processing stage, strategies such as the multi-scale convolutional feature maps make the model 3 times faster than the Faster R-CNN model while also achieving better results.

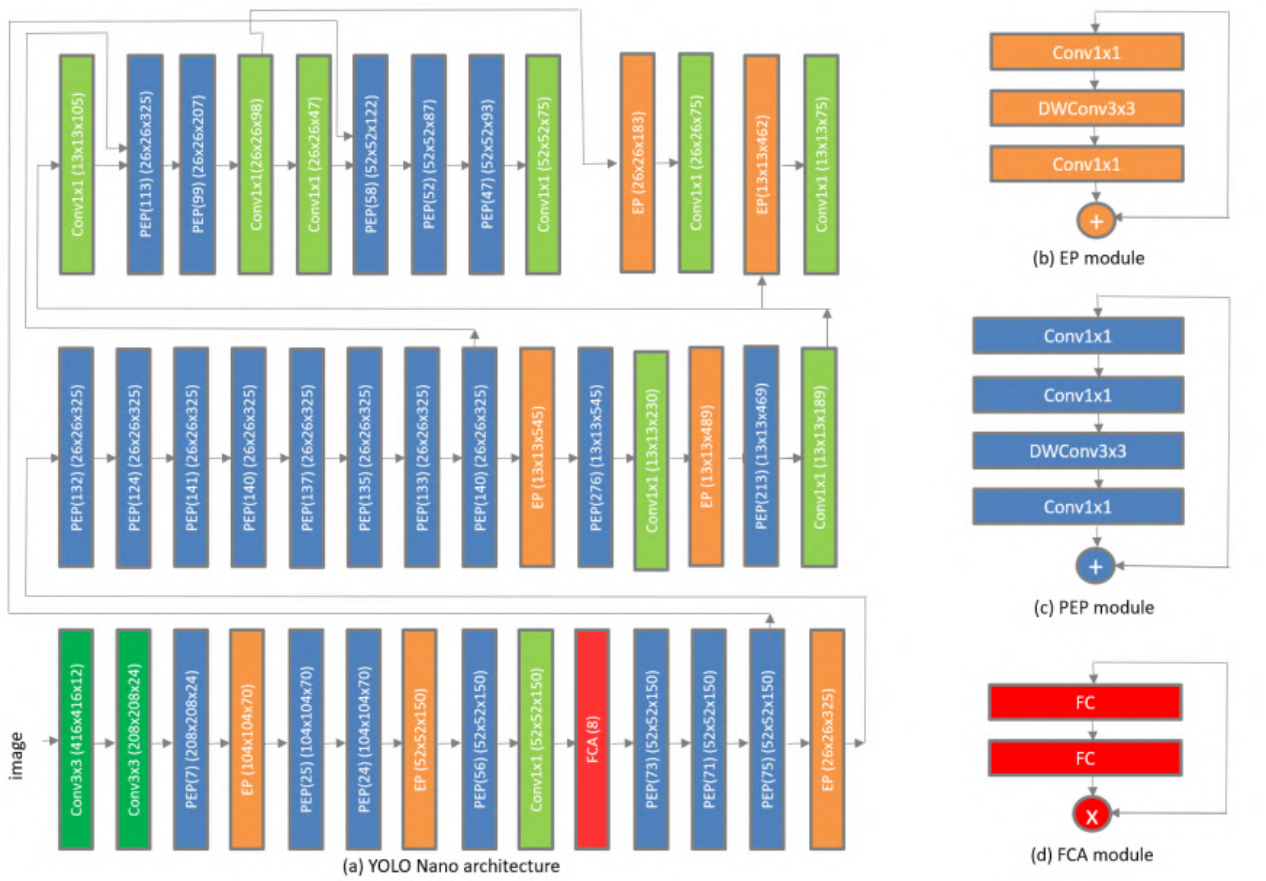

Figure 2.2 – Diagram representing the scheme of the YOLO Nano architecture.

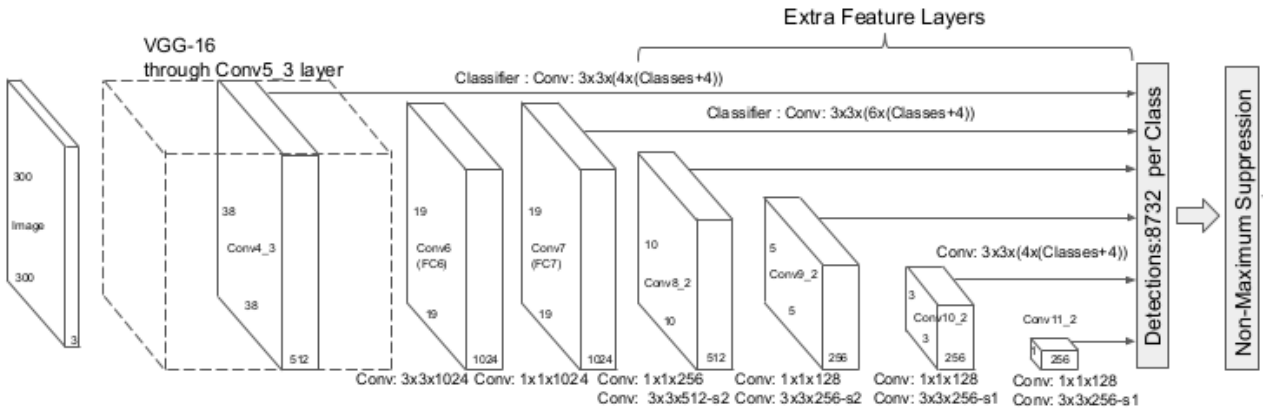

Figure 2.3 – Diagram representing the SSD architecture scheme.

The final model we will present is the M2Det architecture [95], presented in Figure 2.4. The architecture proposes the usage of a Multi-Level Feature Pyramid Network (MLFPN) for feature extraction, which is composed of three modules: Feature Fusion Modules (FFM), Thinned U-Shape Modules (TUM), and a Scale-wise Feature Aggregation Module (SFAM). The first module is responsible for fusing feature maps of the backbone to enrich their semantic information. The second module allows for the extraction of multi-level

multi-scale features. The final module aggregates the features by using scale-wise concatenation and attention modules. Then finally, similar to the SSD architecture, they estimate the dense bounding boxes and categories based on the learned features. In their model, the authors use a modified version of the backbone network where the fully connected layers are removed, which improves significantly their performance when combined with the MLFPN approach proposed. The model's results compete with the state of the art of two-stages detectors and achieved the best results out of the one-stage detectors when it was proposed, while still being 4.5 faster than the Faster R-CNN model.

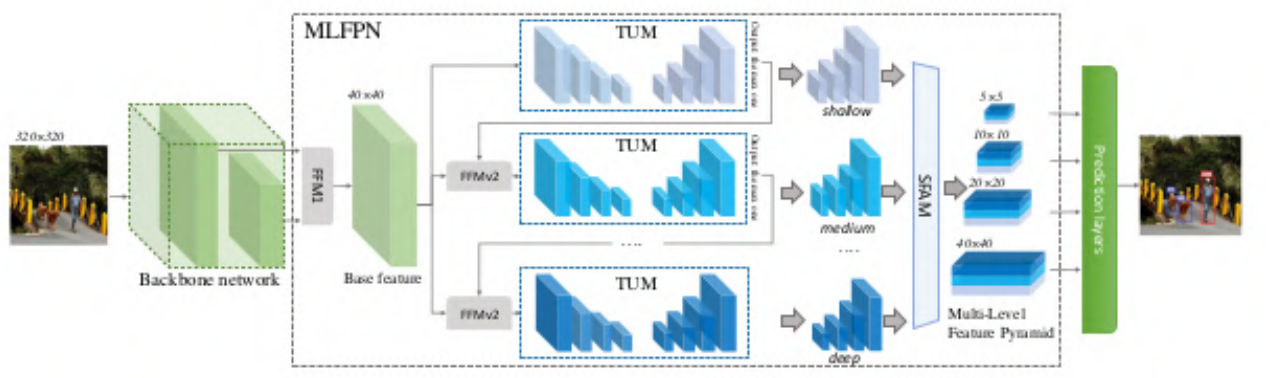

Figure 2.4 – Diagram representing the M2Det architecture scheme.

2.3 Model Compression

While neural networks have been achieving high results in various computer vision tasks, they also come with an inherently high cost of computational resources. This section discusses valuable techniques that optimize resource usage, reducing the amount of Multiply-Accumulate (MAC) operations while affecting detection results with as minimal impact as possible.

The first technique we are going to highlight is Pruning [44], which aims to remove unnecessary elements from models. This technique can be applied on multiple levels for differing levels of impact on the resources saved and detection results. The lowest level pruning is called Weight Pruning, where zeros replace redundant and non-relevant weights, but it is the pruning strategy with the least impact. The next level of pruning strategies is called Neuron Pruning (or Filter Pruning for convolutional layers), where the most negligible neurons or filters are cut from the model, which is a more effective strategy than singling out individual values. Finally, for deeper networks, it might be interesting to perform the highest level pruning strategy: Layer Pruning, where layers that do not provide new relevant information are cut from the network, thus having a high impact on both the resources used and the results achieved by the model.

The following technique we are interested in discussing is Quantization [21], which changes the internal representation of the model's weights to a smaller number of bits. This technique allows us to reduce both the complexity of MAC operations and the final size of the trained model. In addition to reducing the precision of the weights to smaller precision, clustering can also be explored for quantization. In this strategy, each cluster is assigned the value of a full precision weight, and each weight can be quantized to an index mapping its corresponding cluster, where it can then retrieve its full precision value.

The last technique described will be Knowledge Distillation [7], which aims to train a smaller student model to generalize as well as a bigger teacher model – which can be even an ensemble of individual networks. In this technique, first, the teacher model is trained to generalize the dataset well, then the student model is trained to replicate the generalization capabilities of the teacher model. This technique allows us to achieve comparable results by expending less computational resources and MAC operations for the same inputs.

3. RELATED WORK

We conducted a literature review consisting of two stages to study state-of-the-art works addressing Firearm Detection to identify what has already been done in the past years and what needs improvement. In this chapter, we detail the procedure for each of the two stages followed, discuss some of the selected works, present the datasets available in the literature, and describe the conclusions of our findings.

In the following sections, we present the analysis and insights we found for each group of papers. Section 3.1 describes the methodology followed to study the literature. We start by presenting the datasets identified in the literature. Then, in Section 3.2 we present the works that addressed gun detection with different goals in Section 3.3. Finally, in Section 3.4, we discuss our findings from this study.

3.1 Methodology

Our literature review was divided into two stages: a Snowball procedure and a query-based search. The first stage is responsible for analyzing the state of the art, which allowed us to identify the research gaps that could be addressed. In the second stage, we constructed a query tailored around the gap found, thus allowing us to select works that addressed that specific issue – even if it was not their primary concern. By following these stages, we aimed to encompass a representative portion of the state of the art and focused on works that could significantly impact our work. The process followed for our literature review is shown step-by-step in Figure 3.1, highlighting where each stage begins and ends.

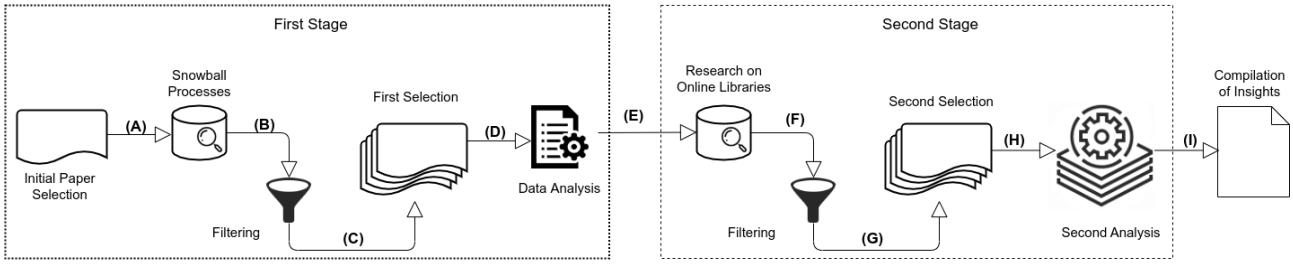

Figure 3.1 – The process followed step-by-step during the literature study realized.

The first stage consisted of a two-step Snowball procedure parting from the work of Lim et al. [47] (Figure 3.1-A), where we first applied a backward step, then a forward step. By applying these steps, we arrived at a total of 50 papers (Figure 3.1-B), which were then reduced to 37 after a selection process (Figure 3.1-C) by following the criteria presented in Table 3.1. By analyzing these selected papers (Figure 3.1-D), we were able to identify that

the efficiency of the models was a concern of most works, although in most cases it was left as future work and few papers addressed it directly.

Based on these findings, we then constructed a query to search for works that address the issue of efficiency, more specifically, focusing on works that address near-real time efficiency (Figure 3.1-E). Using this query, we were able to retrieve a total of 139 papers that met the criteria (Figure 3.1-F), which were quickly reduced to 118 by removing the duplicates from our initial collection of 50 papers. Then, due to the aim of this selection being more specific and well-defined, we applied a selection process with more rigorous criteria, as shown in Table 3.1. So, we reduced this selection to only five papers deemed as relevant for our work (Figure 3.1-G).

Once concluded these two stages, we arrived at a total of 42 relevant papers, 37 of those being more broad-scoped and encompassing the state of the art in a general manner, and five of those that discuss more in-depth near-real time solutions. To understand how the state of the art has evolved and to complement our original literature review that considered papers published until early 2020, we conducted a small-scale query-based search following the same procedure as the original. During this final step, we first selected 14 initial papers that met our criteria, but we only chose six of these as pertinent to our objectives. With this final search, we gathered a total of 48 works for our research.

We present the selected works in Table 3.2, ordered by when they were added to our selection. Each row is colored by a shade of blue indicating the step in which each work was added, starting with a lighter color for the initial paper for the snowball and getting progressively darker for each snowball iteration, and then reaching the darkest shade on the papers added after the query search.

In the following sections, we will present the analysis and insights we found for each group of papers. The first group includes the works identified in the first stage and that address firearm detection with varying goals. The second one contains the works selected due to having addressed performance as a major concern.

Criteria	Was used in
Addressed Gun Detection as one of its main focuses	Both stages
Proposed between 2015 and early 2020	Both stages
Was available online as Gray Literature or have access provided by the University	Both stages
Was written either in English or Portuguese	Both stages
Addressed high performance as one of its main focuses	Query-based retrieval
Presented state-of-the-art results	Query-based retrieval

Table 3.1 – List of criteria we required for each work to fulfill, depending on which stage they were identified, so that they entered our final selection.

Work	Published in
Gun Detection in Surveillance Videos using Deep Neural Networks [47]	APSIPA ASC
A Computer Vision based Framework for Visual Gun Detection using Harris Interest Point Detector [85]	IMCIP
Developing a Real-Time Gun Detection Classifier [43]	Tech report
Automatic Handgun Detection Alarm in Videos Using Deep Learning [60]	Neurocomputing
A computer vision based framework for visual gun detection using SURF [84]	EESCO
Automated Detection of Firearms and Knives in a CCTV Image [29]	Sensors
Object Detection Binary Classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance [65]	Knowledge-Based System
A binocular image fusion approach for minimizing false positives in handgun detection with deep learning [61]	Information Fusion
Firearm Detection from Surveillance Cameras Using Image Processing and Machine Learning Techniques [23]	ICSICCS
A Review on State-of-the-Art Violence Detection Techniques [69]	IEEE Access (Vol. 7)
The Need for marker-less computer vision techniques for human gait analysis on video surveillance to detect concealed firearms [57]	Tech report
Detection Of Concealed Weapons Using Image Processing Techniques: A Review [52]	ICSCCC
Firearm Detection and Segmentation Using an Ensemble of Semantic Neural Networks [18]	EISIC
Convolutional Models for the Detection of Firearms in Surveillance Videos [75]	Applied Sciences
Use of Deep Learning for Firearms Detection in Images [8]	XV WVC
A Novel Approach to Detect Crimes and Assist Law Enforcement Agency using Deep Learning with CCTVs and Drones [64]	IJRASET
ADoCW: An Automated method for Detection of Concealed Weapon [70]	ICIIP
Automatic Handgun and Knife Detection Algorithms: A Review [90]	IMCOM
Localizing Firearm Carriers by Identifying Human-Object Pairs [3]	ICIP
Gun Detection System Using Yolov3 [91]	ICSIMA
Detection and Recognition of Handguns in the Surveillance Videos using Neural Network [30]	IJRASET
Gun and Knife Detection Based on Faster R-CNN for Video Surveillance [20]	IbPRIA
Weapon Classification using Deep Convolutional Neural Network [17]	ICoICT
Crime Scene Prediction by Detecting Threatening Objects Using Convolutional Neural Network [58]	IC4ME2
Graph clustering for weapon discharge event detection and tracking in infrared imagery using deep features [5]	Pattern Recognition and Tracking XXVIII
Firearm Detection using Convolutional Neural Networks [13]	ICAART
AI Based Automatic Robbery/Theft Detection using Smart Surveillance in Banks [39]	ICECA
Accelerated pistols recognition by using a GPU device [53]	INTERCON
Crime Intention Detection System Using Deep Learning [59]	ICCSDET
Hybrid weapon detection algorithm, using material test and fuzzy logic system [36]	Computers & Electrical Engineering
A handheld gun detection using faster r-cnn deep learning [88]	ICCCT
An alternative method to discover concealed weapon detection using critical fusion image of color image and infrared image [34]	ICCCI
Concealed weapon detection from images using SIFT and SURF [40]	IC-GET
Cascaded Neural Networks for Identification and Posture-Based Threat Assessment of Armed People [1]	HST
Suspicious Activity Detection in Surveillance Footage [49]	ICECTA
Fire and Gun Violence based Anomaly Detection System Using Deep Neural Networks [55]	ICESC
Gun source and muzzle head detection [97]	Electronic Imaging
A Systematic Review of Intelligence Video Surveillance: Trends, Techniques, Frameworks, and Datasets [79]	IEEE Access (Vol. 7)
Intelligent Surveillance System to Handle Sudden Arms Attack in Less Secured Areas [63]	JAT
Development of an AI-based System for Automatic Detection and Recognition of Weapons in Surveillance Videos [93]	ISCAIE
Deep autoencoder for false positive reduction in handgun detection [86]	Neural Computing and Applications
Real-time gun detection in CCTV: An open problem [26]	Neural Networks
Deep multi-level feature pyramids: Application for non-canonical firearm detection in video surveillance [46]	EAAI
A Dataset and System for Real-Time Gun Detection in Surveillance Video Using Deep Learning [68]	IEEE SMC
Automatic Handgun Detection with Deep Learning in Video Surveillance Images [78]	Applied Sciences
TYolov5: A Temporal Yolov5 Detector Based on Quasi-Recurrent Neural Networks for Real-Time Handgun Detection in Video [16]	CoRR
Detection of weapon possession and fire in Public Safety surveillance camera [56]	ENIAC 2021
Handgun Detection Using Combined Human Pose and Weapon Appearance [76]	IEEE access (Vol. 9)

Table 3.2 – List of works that we selected, highlighting in which stage they were added to our selection.

3.2 Literature Datasets

Several datasets for detecting weapons [35], crimes [82], and other objects [55] and violent events [9] have been found in the literature. By studying the selected works, we

identified no standard dataset in the area, and the existing datasets can be roughly categorized according to the data they use. This insight is crucial for us to quickly filter undesirable datasets and focus on those adequate to our objectives. The categories identified are as follows:

- **Movie Data:** Some works focus on data taken from movies to base their models on since there is a large amount of data available to allow for large datasets to be created, such as the IMFDB [35]. But, because of a few characteristics of movies, such as video quality and camera positioning, models based on these datasets don't perform well on real-world data.
- **Enacted Data:** Other works try to emulate real-life events to make their models more appropriate for real-life applications than the movie-based counterpart. Although such datasets are a better representation of real-life scenarios, they are generally much smaller because they require much effort to create, which can be seen in datasets such as the videos created by [28].
- **Real Data:** The final class of relevant datasets is composed of surveillance videos made public. Even though these are the most representative datasets, they are pretty rare, and few in numbers, and are not very large (sometimes being even a subclass of a larger dataset, as in [82].

ID	Dataset	Type of Data	Year	Uses
[D01]	Weapons-Detection [60]	Movie Data	2018	15
[D02]	IMFDB [35]	Movie Data	2015	7
[D03]	UCF Crime [82]	Real-world Data	2018	4
[D04]	Gun Movies Database [28]	Acted Data	2013	3
[D05]	Monash Guns Dataset [46]	Acted Data	2021	3

Table 3.3 – This table presents the most frequently used datasets in the works we selected.

In our analysis, we identified 33 datasets used and sometimes proposed by the literature reviewed. Among these datasets, only five of them, which are presented in Table 3.3, were commonly used among the selected works. It is important to note that we assigned an ID to those five datasets, which are used in future discussions to reference back to this table. In contrast to those, another eleven proposed datasets were used in just one work. And the 17 remaining datasets were created but not made available. Thus we will not be considering them for our analysis.

Although most works discussed the usage of firearm detection in CCTV cameras scenarios, only three of the publicly made datasets focused on real-world data. Even considering enacted data, which had the most datasets at size unique proposals, the number of works focused on security camera data was only the minority. For the movie category, five

datasets were proposed and looking at Table 3.3, we can see that this category was vastly more used than the others since the two most used datasets were part of it. This leads us to a situation where most works explore the possibility of using firearm detection models in real scenarios, but most of them use inadequate data for such scenarios. One possible justification for this decision might be the difference in availability and size of these datasets since it is a lot easier to acquire a large dataset of movies than in real-life cases.

3.3 Selected Works

We could identify two main research lines by analyzing the selected works: proposals for firearm detection and the detection of concealed weapons. However, since our search was primarily directed towards firearm detection in general, only a minority of the selected works address concealed weapons as their primary objective. Moreover, to further filter the works selected, we separated them into categories based on the research line they followed: Works that used feature engineering [23, 29, 40], works that used deep features with a two-stage detector [43, 88, 70], and works that used deep features with a one-stage detector [47, 13, 93]. In the following sections, we present some proposals for each of the previously mentioned categories, and give a brief description of the strategies employed.

3.3.1 Gun Detection with Computer Vision Techniques

Among the works that use engineered features, we will highlight three works proposed for Gun Detection [29, 23, 36] and one work proposed for Concealed Gun Detection [40].

The works of Grega et al. [29] and Gelana and Yadav [23] propose a very similar pipeline, where their model extracts patches from the image using foreground filters and a sliding window method, which are then classified by a neural network. However, the work by Grega et al. applies an intermediate step, using the PCA method to reduce dimensionality before classifying the patch, while in their work, Gelana and Yadav address this problem by using a more advanced network. With their pipeline, Grega et al. achieved a sensitivity of 36% when testing on [D04], and Gelana and Yadav achieved 93.8% sensitivity on the same dataset.

Ineneji and Kusaf [36] propose a hybrid pipeline, using data from sensors to aid the main pipeline, composed of 3 stages: feature extraction using a Bag Of Features method, feature clustering using K-means, and an SVM for classification. And lastly, the work proposed by Kaur and Kaur [40] uses a pipeline based on the combination of the SIFT and SURF methods to select candidate regions, then classify those regions based on a set of

rules. Both works experimented on their own dataset, which were not available. However, they achieved high metrics, with Ineneji and Kusaf achieving an accuracy of 80% and Kaur and Kaur achieving a sensitivity of 90%.

3.3.2 Gun Detection using Two-Stage Detectors

For works that use deep features, we will highlight three two-stage proposals, where two of them address Gun Detection [43, 88] and one addresses Concealed Gun Detection [70].

The model proposed by Lai and Maples [43] uses the Overfeat architecture to perform the detection, achieving an accuracy of 89% using images from [D02]. For their work, Verma and Dhillon [88] use the Faster R-CNN architecture, using the VGG-16 backbone, achieving 93% accuracy on [D02] images. Similar to the previous work, the work by Raturi et al. [70] also employed the Faster R-CNN architecture, although not specifying the backbone used. They built their own dataset, taking images from datasets such as [D01], and achieved 95.8% accuracy, but did not make the dataset public.

3.3.3 Gun Detection using One-Stage Detectors

For the second group that uses deep features, we will highlight three one-stage proposals [47, 13, 93], all of them addressing Gun Detection.

The work proposed by Lim et al. [47] employs the M2Det [95] architecture, using a VGG-16 [81] backbone, seeking to improve performance when compared to the state-of-the-art. However, they lose significantly in their detection results, which is evidenced by the 22.3% accuracy achieved on the UCF Crime dataset (see Table 3.3). In their work, De Azevedo et al. [13] went a step further, using the YoloV2 [71] architecture to achieve a performance gain but still achieving high detection results with 96.3% accuracy on the custom test set made. And lastly, the work by Xu et al. [93] proposes the usage of the SSD [48] architecture with a MobileNet [32] backbone, seeking to have a slightly better performance than the heavier models while not losing too much on the results reached since they achieved 85.2% precision on the dataset they made.

3.4 Discussion

Through the analysis of the selected papers, we identified that the issue of performance was not explored very often in this area, although being one of the most frequent suggestions for future work, as shown in Figure 3.4, but it has gained substantially more attention in recent years [23, 79, 16]. Monitoring the videos from security cameras is essential to identify the moment a particular event begins so that the safety of the innocent involved can be preserved as much as possible. This shows the importance of having a high performance to process each instant of the video quickly, and hence, the importance of studying efficient architectures such as those previously presented.

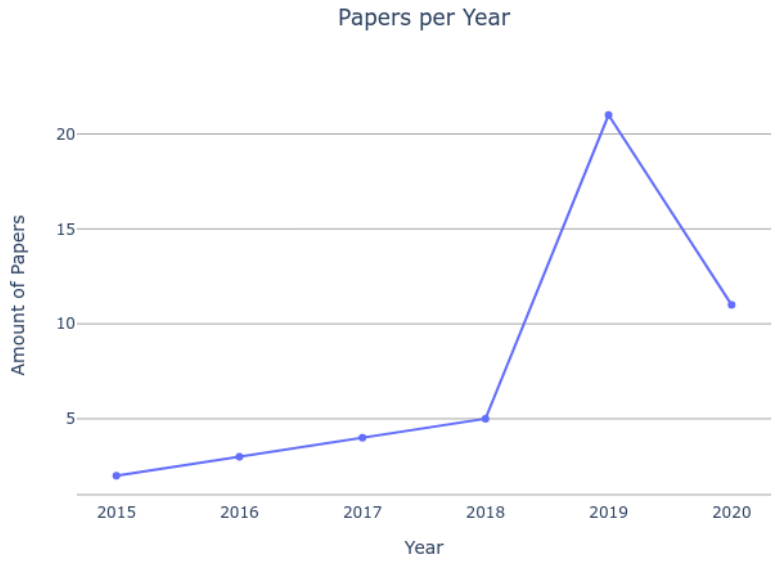

Figure 3.2 – Amount of papers selected, grouped by year of publication.

Although research in this area has received more attention in recent years, as shown in Figure 3.2, few works addressed the performance issue in real-life scenarios. We identified that few papers proposed contributions to solve this problem and, analyzing the charts in Figure 3.3, it is also observed that, recently, this problem is rarely addressed as a proposal for future work, even though it is still open [26] and is clearly very important for innovative solutions to be applied in real scenarios.

From the overview presented throughout Section 3.3, we can observe that it is hard to fairly compare the methods in the area, as many works use diverging datasets and focus on different types of data. Even when analyzing the most frequently used datasets discussed in Section 3.2, we can see no standard between them. In Figure 3.4, we can see the recurring future work proposals identified in our study, where achieving better datasets is by far the most frequently mentioned proposal. Thus, the biggest concern in the area is

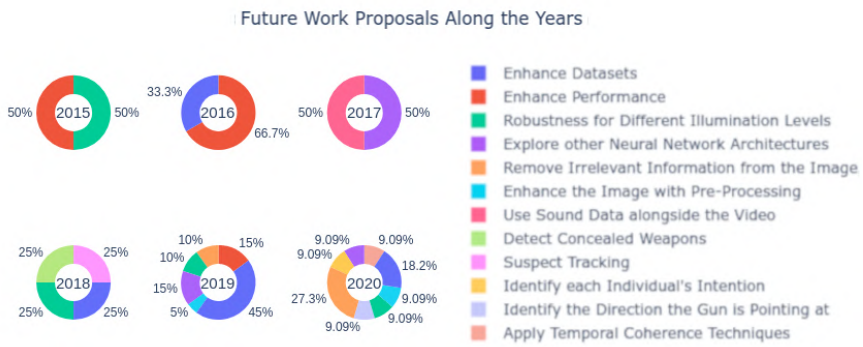

Figure 3.3 – Distributions of future work proposals over the years, as observed in our selection. Works that did not propose any future work were not considered for this analysis.

the lack of representative datasets. While most of the datasets found in the literature are composed of acted scenes (see Section 3.2), some works, such as Sultani et al. [82] and Lim et al. [47], stand out for presenting data from actual events captured by security cameras and made available to the public. Nonetheless, while those datasets contain exciting data, they lack amount, diversity, and structure since most of them constitute a set of videos or contiguous frames marked as having or not the object of interest instead of more a precise annotation.

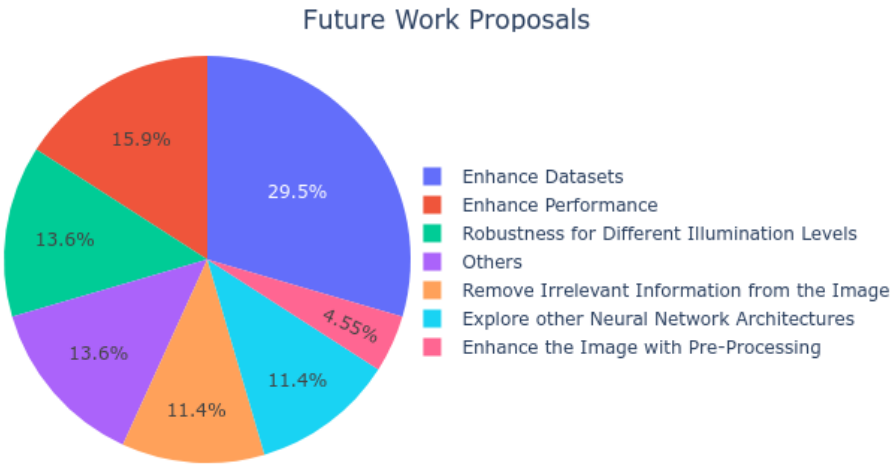

Figure 3.4 – Distribution of future work proposals by the selected papers.

Through the analysis of the works, it was also possible to identify that the most used technique was the Faster-RCNN [74] architecture, being also the technique adopted by the papers that presented the best results in representative datasets, such as Pérez-Hernández et al. [65] and Raturi et al. [70]. However, although this technique presents excellent results, it does not have an ideal performance and is a “heavy” architecture. Thus, it requires many computational resources, making it unsuitable for limited-hardware scenarios.

After this study, which showed the need to have an enhanced dataset, we decided to create a new dataset to try to fulfill all the lacking characteristics identified on the ones present in the literature:

1. We chose to include only real-world data on our dataset since we want to encourage research applied in real-world scenarios.
2. We gathered data from multiple sources, thus introducing high variability and quantity of data.
3. We also chose frame-level annotations for object detection that can be easily adaptable for other tasks, such as scene or clip classification.

4. RESEARCH METHODOLOGY

To develop this project, we planned the ten activities shown in Figure 4.1. The first four activities encompassed studying and understanding the literature. In activities five and six, we defined the main goal, the contributions, and the requirements for this work. Activities seven through nine are responsible for the development of our solution. The last activity involves compiling our findings, results, and limitations for writing the dissertation and submitting a paper to a conference or journal. The following sections detail these activities, which were grouped into three phases.

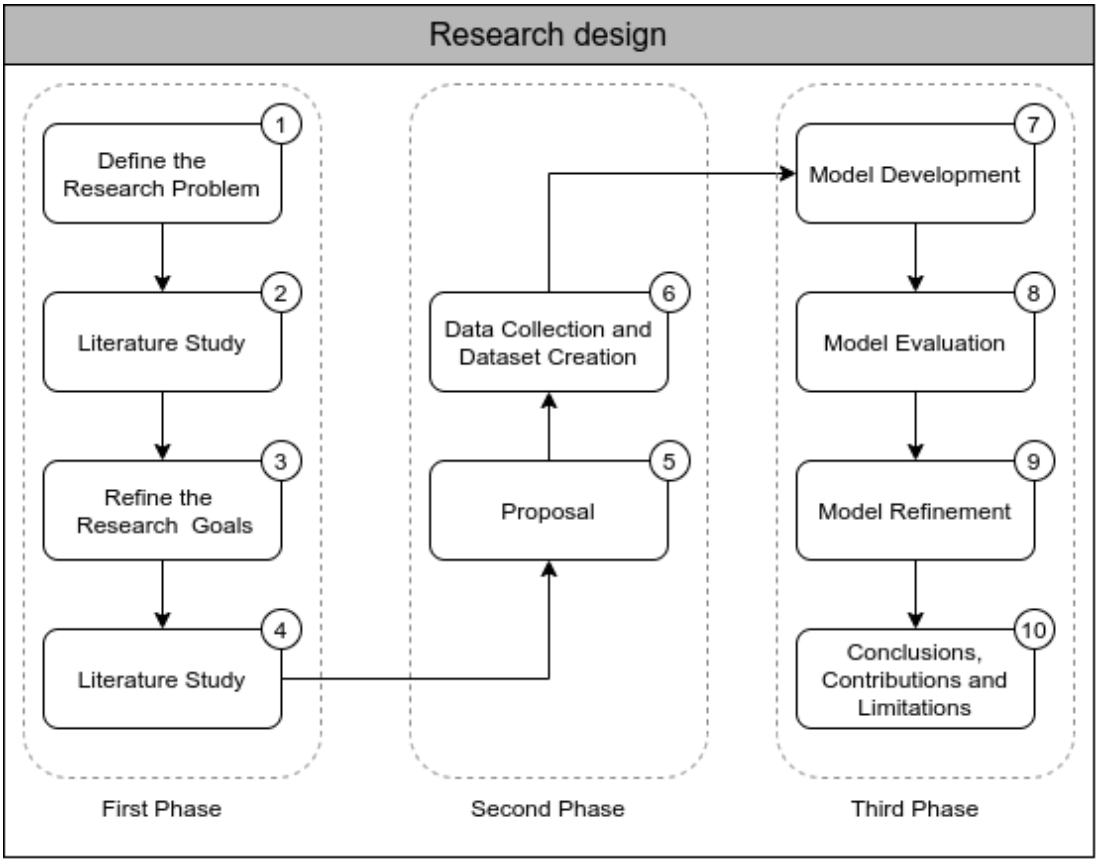

Figure 4.1 – List of the main activities constituting this project.

4.1 First Phase

When choosing the research problem to be addressed in this work, we sought to explore areas that had real-world implications in our current society. Our initial step (Figure 4.1-1) was to investigate the surveys in the literature about surveillance video processing. Then, we identified that Firearm Detection was growing as a research area with still many challenges to be addressed. After deciding on firearm threat Detection as a research prob-

lem, we conducted the literature study discussed back in Chapter 3, through which we could then finally identify a challenge significant for the chosen research problem (Figure 4.1-1– 4) that was not fully addressed in the literature.

4.2 Second Phase

Once we decided precisely the points we wanted to address, we needed to formally stipulate what we would need to do to achieve our goals (Figure 4.1-5). Our first step was to decide the dataset we would use as a basis for the evaluation of our approach. For this, we searched the literature for datasets available online that contained interesting data for our challenge. Meanwhile, we also started studying the area of Model Compression and chose a set of techniques that we were interested in exploring further. These techniques were chosen based on effectiveness and complexity to understand. We decided on this because we were just learning these techniques, and we were afraid that misunderstanding and misusing them would have an undesirable impact on our model.

Then, we gathered some datasets to join them into a final group of data (Figure 4.1-6), which is described in detail in Chapter 5. This stage was needed because, as mentioned before in Section 3.2, real-life data is very scarce, and we could not find a satisfactorily large dataset. Thus, our approach to this issue was manually joining the best datasets, among the previously identified, and including novel data we gathered to form a challenging and diverse final dataset. This process introduced a few challenges since the datasets we found varied by how the annotations were done and different data formats, e.g., images and videos.

Next, to assess whether the dataset is adequate or needs improvements, we used it to train some state-of-the-art models and extracted some metrics. These metrics were used to assert that the data is ample and diverse enough that the models could properly learn from it, instead of being forced into pitfalls such as overfitting, and demonstrate how challenging our final dataset is.

4.3 Final Phase

In the next activity (Figure 4.1-7), we studied more in-depth and experimented with the chosen Model Compression techniques. We explored many combinations and hyper-parameters to see which would yield satisfying results and which were not fit for our work.

Once the most cost-efficient strategies are selected, we advanced to the next activity (Figure 4.1-8), where a set of experiments were conducted both on ideal and limited conditions. This way, we can compare the strategies amongst themselves and contrast their

performance against how the state-of-the-art performs on both conditions. Through this analysis, we could acquire unbiased insights about how the models perform since some strategies may be more appropriate for one scenario rather than the other. Making this disparity will highlight the advantages of each strategy for the scenario they excelled.

Then, by analyzing the results and metrics extracted, our next activity (Figure 4.1-9) focused on studying the weak points of the strategies selected and making adjustments to address those weaknesses directly. Once the refinements were done, we could assign a score to each strategy to objectively rank the strategies amongst themselves, encompassing their performance and robustness.

On our last activity, after finishing these refinements, we selected our best strategy and compiled the information gathered from the set of experiments performed. By doing so, we can fully state our main contributions, what metrics support our claims, and our limitations. Furthermore, having these formally stated, we concluded the dissertation to focus on producing a paper.

5. PROPOSED DATASET

We initially did a literature research to identify which datasets were being used, their characteristics, and what could be improved, as discussed in Section 3.2. After, we decided that the paramount quality we want on our dataset is a high variability amongst the images, i.e., we are not interested in a contiguous sequence of frames but frames that present new information not shown by previous frames. Then we started gathering images, preprocessing them, and lastly annotating and anonymizing faces on them. The following sections describe in detail each of the steps presented in Figure 5.1 and followed to develop FiDaSS. Our dataset and the tools designed to create it are available online[1].

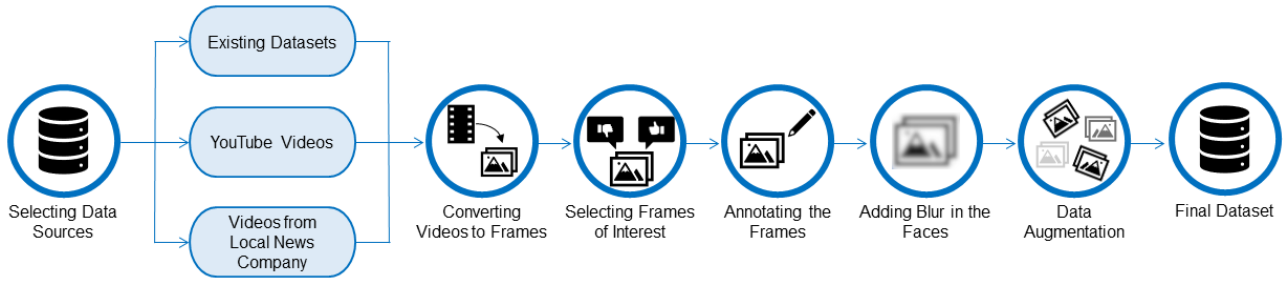

Figure 5.1 – Presentation of the steps followed to create our dataset.

5.1 Data Collection

The creation of FiDaSS was divided into two stages. First, we analyzed the datasets described in the literature, starting with the most-used ones shown in Table 3.3. Then, we searched the lesser-known ones to understand what we could use and to build a strong basis of what we wanted in our dataset. In the second stage, we selected a collection of videos from YouTube and a local news company to further enrich FiDaSS with various scenarios and situations. Even though our work focuses on everyday events, most recordings are not shared online due to belonging to security companies or other privacy matters, making collecting an ample amount of varied data difficult.

After analyzing existing datasets, we selected those that focused primarily on real-world data from security cameras: *UCF Crime* and *Weapons in Images* [38]. When choosing which videos to analyze further, we chose those that contained at least one moment that clearly displayed the weapon (where it could not be confused with another object). We also demanded that a criminal appear (thus avoiding situations including only cops, for example). Since there was an intersection between the datasets selected, we conducted a manual

[1] https://github.com/MuriloRegio/FiDaSS

verification to remove all overlapping data from the selection – which we performed once more at the end of the next stage.

Then, to complement the data that we selected from the datasets in the literature, we sought new and still unexplored data from YouTube. In the first step, we conducted a query-based search to create an initial selection of videos using the keywords *[surveillance video armed robbery, CCTV assaults, guns in CCTV, assault caught on camera, assaltos à mão armada]*. So, we filtered the videos found based on our previously defined criteria (clearly displaying a weapon and with a criminal appearing), arriving at a total of 39 videos. For the next step, we started collecting new videos based on YouTube's recommendations alone, leading us to new videos from different cultures that were not represented by our query, which were then filtered by the previously presented criteria, thus accumulating 162 more videos for our dataset. The playlist with the selected and filtered videos is available online[2]. Thus, in the end, we selected 201 videos from YouTube depicting crime scenes from different countries and cultures.

Finally, to further expand our dataset, we contacted a local news company, requesting access to some videos provided to them depicting recent crime scenes from the region. Upon receiving their approval, they provided us with 13 novel videos.

5.2 Dataset Annotation

Before annotating FiDaSS, since we are interested in image annotations instead of video annotations, we converted the videos we selected into a set of frames. Then, to satisfy our constraint of data variability, we manually analyzed all of the frames extracted and chosen for our final dataset only the frames that provided new variations from their predecessors (e.g., different angles, illumination levels, positioning). Following this procedure, we transformed seven hours of video to form our final dataset of a total of 6942 images. These images were then annotated so that the bounding boxes would include both the gun and the person handling them, thus making our dataset not focused on the weapons themselves but the action of people using them. We present a selection of a subset of images present in FiDaSS and their corresponding annotations in Figure 5.2.

We adopted some measures to provide a thorough and fair annotation process since some scenes were not clear and, thus, not intuitive whether they should be annotated and how. We conducted two parallel annotation processes, where each person analyzed all the selected images and decided which threats to annotate and which were too ambiguous. Then, after both were done, we conducted a discussion session to decide what to do with annotations that differed from each person. Therefore, our final dataset is the consensus reached between researchers and carefully made annotation decisions.

[2]https://www.youtube.com/playlist?list=PLnq5fLsdu5RqPUGq3r4rgyY5m_pM9h3HB

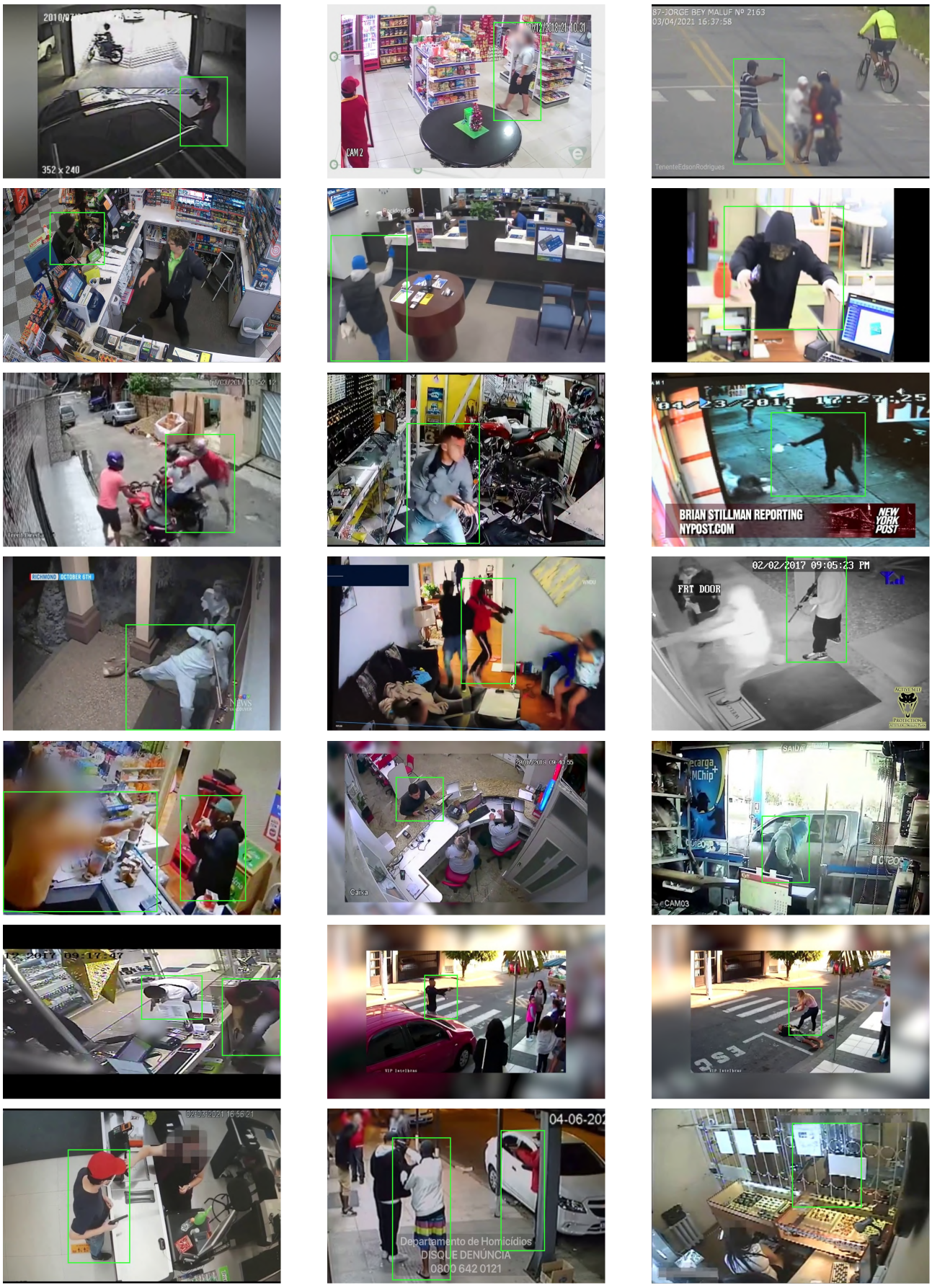

Figure 5.2 – Example of images from our dataset, showcasing the diversity of situations present in the images we have gathered with their corresponding annotation.

While annotating the images, we only considered information from the current frame to avoid an “unfair” ground truth on some of them. This was needed because some frames had ambiguous or hard to discern objects since security cameras usually record poor-quality videos. In these cases, we would only know for sure what these objects were if we had information from past and future frames. We present some cases in Figure 5.3. Thus, since we are interested in frame-level detections, we avoided using knowledge from other frames and factors such as body language when deciding if a firearm threat was within our scope.

Following this methodology, we annotated a total of 4307 firearm threats over 3942 images of our dataset. Furthermore, we also selected 3000 additional images for our dataset that did not include our event of interest or similar situations, which were included to help enhance the recall of the predictions. As explained before, since some images rely on subtle information such as body language, we analyzed that it was beneficial to include counter-examples so the learning algorithms could better differentiate the situations.

Furthermore, in cases where the image quality was clear enough to recognize someone’s face, we applied a Gaussian Blur on people’s faces to preserve their anonymity. We used the face detector proposed by Zhang et al. [94] to apply the blur automatically. However, we had to review all the images and manually apply the blur where the algorithm failed to detect a face. Then, using the tools developed (which are available together with the dataset), we annotated the images to include the whole body of the person posing as an armed threat – as shown in the cropped annotations in Figure 5.3.

a. **b.** **c.**

Figure 5.3 – Example of an image from our dataset (**a**) and the annotation in the subsequent frames. Images in (**b**) are adequate for our dataset, and in (**c**) are inadequate due to being too hard to identify the gun without temporal context. The images shown in (**b**) and (**c**) are cropped exactly in the area of corresponding annotations. The red circles however were only included as visual aid.

5.3 Dataset Augmentation

To promote more diversity in the data of FiDaSS, we applied data augmentation techniques to create variations of the images we had previously selected. Given both the number of images we had and the number of random changes we applied to each one, we decided that creating three new variations of each image would be enough to enhance the performance of learning algorithms without introducing issues such as overfitting.

We applied a pipeline of seven transformations to each of our images, with random parameters, to create still recognizable but highly variable outputs. The transformations applied in our pipeline were, in order:

- Gaussian noise;
- HSV-space variation;
- Horizontal flips;
- Scale reduction;
- Plane translation;
- Plane rotation;
- Shear mapping.

Figure 5.4 presents some examples of input images and the variations introduced by our data augmentation pipeline. We developed a tool to apply this pipeline and created, with pseudo-random parameters, three new augmented variations of all training images. However, we discarded cases where a transformation omitted an annotated region of the image, thus providing 14372 additional examples to complement our dataset.

5.4 Datset Statistics

This section encompasses a discussion on some properties of FiDaSS and compares it to those presented in Section 3.2. We also address the necessity of annotating data from the datasets highlighted in Section 5.1 and why the original annotations were insufficient.

One important characteristic of FiDaSS is that we made the annotations directed towards the task of object detection in real-life scenarios. To the best of our knowledge, considering our literature analysis, there is no dataset presenting those characteristics, and

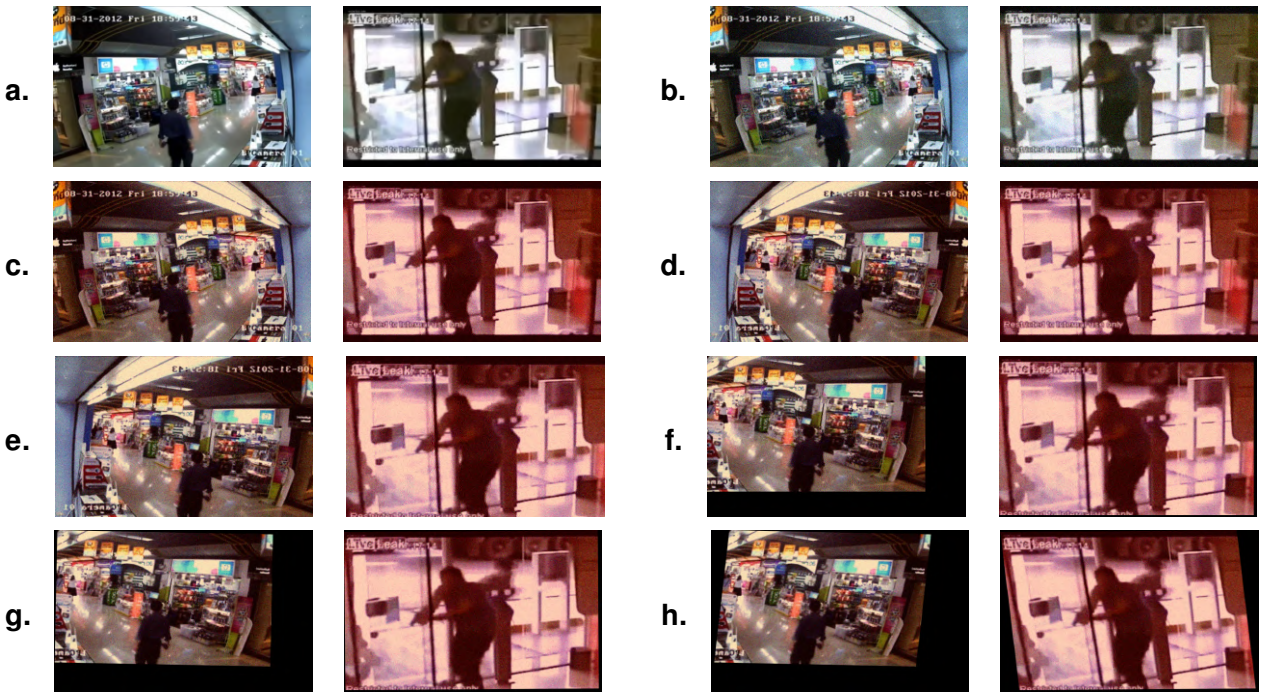

Figure 5.4 – Examples of our data augmentation on sample images, shown step by step: (**a**) the original images; (**b**) Gaussian noise; (**c**) HSV-space variation; (**d**) random horizontal flips; (**e**) scale reduction; (**f**) plane translation; (**g**) plane rotation; (**h**) shear mapping.

that also contains a substantial amount of images. Among the datasets highlighted in Table 3.3, although they have a large amount of data, only one of them presented exclusively real-world data, and none had annotations for the object detection task, only for image or video classification.

Number of annotations	Quantity	Percentage
0	2955	42.84%
1	3611	52.36%
2	303	4.39%
3	23	0.33%
4	4	0.06%
5	1	0.01%
Total	6897	100%

Table 5.1 – Distribution of objects per image of our dataset.

We present in Table 5.1 the composition of the dataset we built and, in sequence, the composition of our augmented dataset in Table 5.2. These tables show how many annotations each image has and how many images contain that amount of annotations. It is important to mention that we used 70% of the dataset for training and only created augmented versions for this portion of the dataset. Also, when separating the frames into subsets (training, test, and validation), we ensured that frames from the same video were in the same subset to not have similar frames from the same video in the training and test sets, for example.

Number of annotations	Quantity	Percentage
0	6165	42.90%
1	7541	52.47%
2	599	4.17%
3	55	0.38%
4	9	0.06%
5	3	0.02%
Total	14372	100%

Table 5.2 – Distribution of objects per image our augmented dataset.

Dataset of Origin	Amount of selected Videos	Total Selected Frames
Youtube Playlist	201	1531
UCF Crime	392	5034
News Company's Videos	13	258
Weapons in Images	11	74
Total	617	6897

Table 5.3 – Total data acquired from each data source.

Table 5.3 presents the datasets we used as the basis to create ours. It also shows the number of videos taken from each dataset and the total number of frames selected from all those videos. Although most of FiDaSS already existed in the literature, constituting 74.06% of it (54.62% of images with objects of interest), we have carefully and rigorously selected the most relevant frames, which were previously only available as raw videos. Moreover, the remaining 25.94% (45.38% of annotated images) of FiDaSS contains novel data, including people wearing masks due to COVID-19's security norms.

In Figure 5.5 we present the distribution of size occupied by the annotated regions in images of our dataset. Since our data relies solely on images from security cameras, we can observe that most objects of interest are further away (i.e., they occupy a smaller portion of the image). However, we can also see that some images are zoomed in or close to the incident, as some annotated regions occupy up to 86% of them. Just in one extreme case, an annotation occupied 100% of the image.

After describing FiDaSS' properties, it is important to compare it with the most used datasets identified, shown in Table 3.3. While the datasets Weapons-Detection and IMFDB contain the most data among those analyzed and ours, they have images from movies or without context primarily and thus are not ideal for real-world applications. The Gun Movies Database provides more accurate data by recording scenes with a security camera but only provides seven videos shot in a laboratory. A more interesting dataset, given our goals, was the recently released Monash Guns Dataset which provides a plethora of data and bounding box annotations for object detection. This dataset is very exciting and has a lot of potentials, shown by the number of works that have already been using it in the relatively short time it has been available. However, this dataset is has a considerable portion of it made of acted

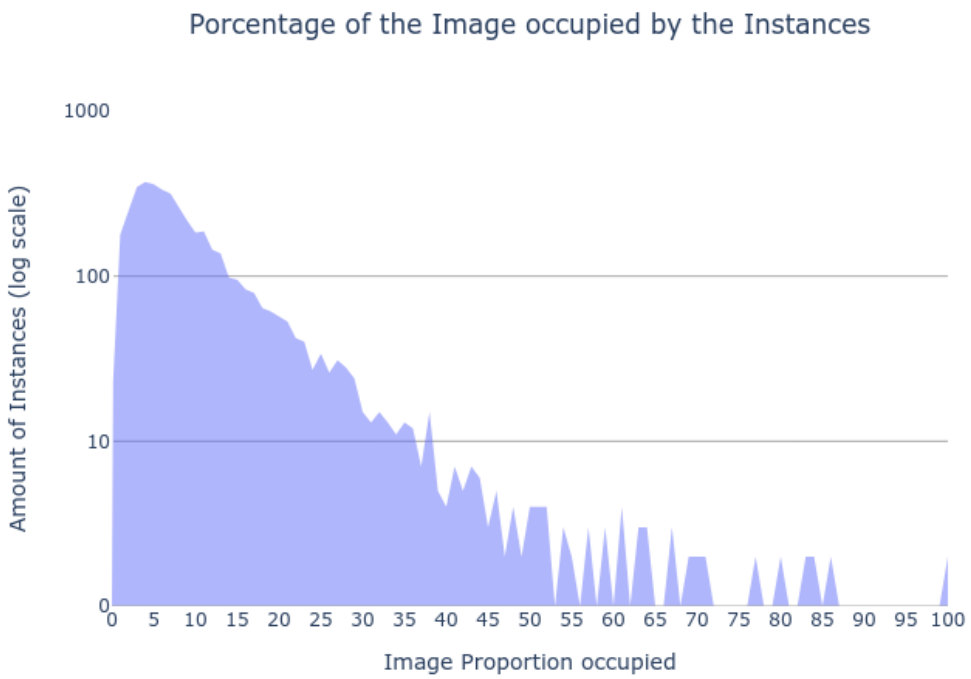

Figure 5.5 – Size distribution of each object of interest in FiDaSS compared to how much of the image it covers.

data, but we wanted a dataset exclusively of real data. Finally, the UCF Crime dataset provides real-world recordings of several events, the ones relevant to this work being *Robbery* (with 150 videos available) and *Shooting* (with 50 videos available). With the desired type and a vast amount of data, this dataset was close to ideal. Its most significant issue was how the data was provided with clip-level labels instead of more detailed frame-level annotations. Thus, with the development of our dataset, we tried to cover these issues, providing a large number of images obtained from real scenarios and with frame-level annotations.

6. PROPOSED MODEL

This chapter presents in detail how we approached the issue of firearm threat detection, how our model is subdivided, and what challenges were addressed. In the following sections, we present the model overview, the architectures and Model Compression techniques chosen, and how they were used to address the challenge.

6.1 Model Description

Our proposed solution consists of three main states, which are illustrated in Figure 6.1: on state **A** our model reads a frame from the input channel; once a frame is read, our model advances to state **B**, where it will use the firearm threat detection architecture developed to look for any use of firearms in the scene; then, our model reaches state **C**, where the detection results are collected and, before looping back to state **A**, our model sends an alert or notifies the personnel responsible for the CCTV monitoring, in case the results indicate that there is a dangerous event happening, so that they may take the necessary actions for the given situation.

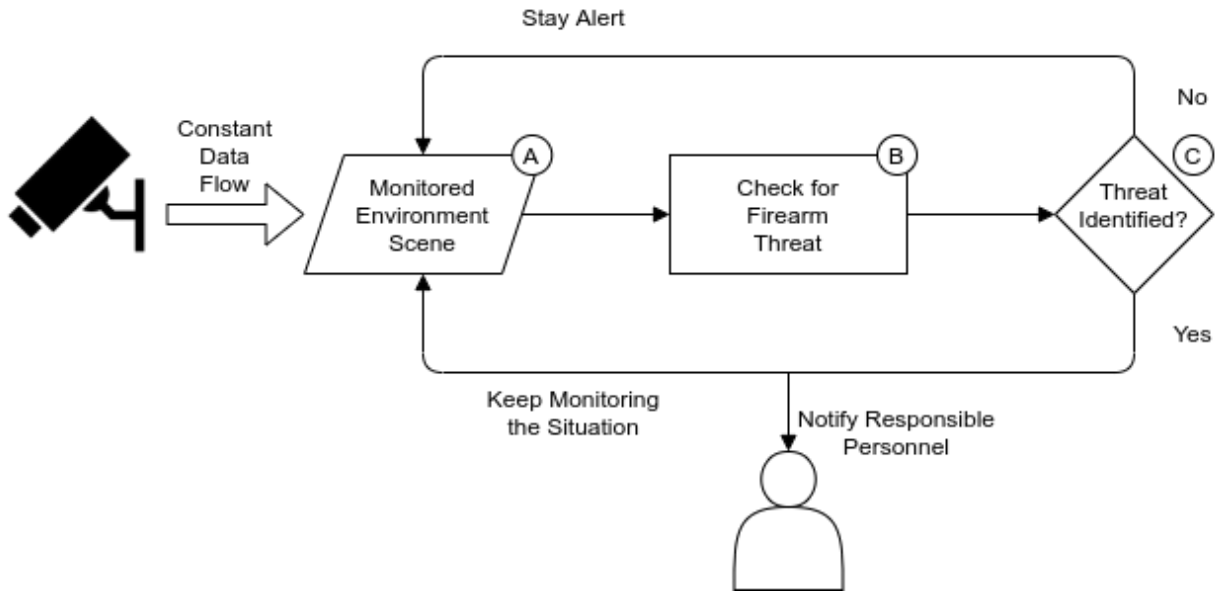

Figure 6.1 – Overview of model states.

Since our model aims to preserve the safety of people involved in dangerous events, the earlier we identify such an event starting, the safest the people involved will be. Because we intend for our model to be accessible to the general public without requiring high investment in hardware or complex systems, our model needs to run efficiently on a vast majority of systems, especially considering budget options. Due to the heavy hardware

restrictions imposed, we foregone processing data in clips and worked only on frame-by-frame detections. Although clip-level detection could provide better results, it requires more complex and specialized approaches. Furthermore, the hardware we explored could not satisfy our performance restrictions with these approaches. However, our model still requires a human operator to verify the gravity of the identified situations, especially since we had to sacrifice the detection quality for more efficiency.

Because we intend for our model to be usable by a broad spectrum of systems and hardware constraints, we planned our model around two key concepts: flexibility and versatility. That means that the states presented for our model are constituted of independent modules that can be changed, adapted, or optimized freely to suit each system better. State **A** encompasses modules responsible for capturing data from the environment, receiving it from the camera, and preprocessing it. For example, a small surveillance system might use an IP camera to monitor the environment transmitting the frames to the processing unit through the local network. Upon receiving them, the model would adjust some of the frames' qualities (e.g., their dimensions). Then, state **B** contains the modules responsible for processing, identifying, and locating firearm threats in the received images. Continuing the previous example, the processing unit would evaluate the frames using one of the trained networks available. Finally, on state **C**, we process the detection results and make decisions based on them, to decide whether personnel should be notified or not. Concluding the example, in this state, the model could be arranged to send a message to the person responsible for the environment's security with a copy of the accusing frame when a dangerous situation is identified.

For our case study, we wanted to evaluate an implementation of our model on a system that had few computational resources available, and that would be easy to acquire and operate. By satisfying these conditions, we would guarantee that a large number of people would be able to use our model, and would leave the option for more knowledgeable people to adapt our model to fit optimally to their needs. The system we decided was the best fit for these conditions was a smartphone, as people could use their old smartphone after buying a new one and most people are familiar with how to operate a smartphone, and finally, older smartphones fit perfectly into our restrictions of having few resources available and no specialized hardware. Thus, the way we implemented our model in our case study was to use the smartphone's camera to capture data on state **A**. The images were then transmitted to the next module through a socket in the localhost, then resized and transformed to fit the network's necessities. Once the images are ready, we feed them to a YOLO network in small batches during state **B**. Finally, the detection results are processed on state **C** by applying a small set of postprocessing computations, using these to decide which cases should be notified and which should not. Since our model implementation allowed for easy substitution of modules, we could easily make use of another smartphone's camera a as method of capturing data for state **A** or change the network architecture used in state **B**.

This implementation of our model is available alongside FiDaSS[1]. In the following sections we discuss the network architectures we explored for our model, followed by the model compression techniques we employed to achieve a satisfying performance.

6.2 Network Architecture

When deciding the network architecture we would focus on and modify, considering all the restrictions of our case study, we concluded that the YOLO architecture would be a solid basis to build upon to achieve our goals. When addressing the issue of performance in neural networks, few architectures stand out as much as the YOLO architecture, famous for excellent performance while also having great results for object detection.

The YOLO architecture is helpful since it was made with efficiency being one of its main concerns. However, another significant advantage of using this architecture is the variety of options available. The YOLO-Tiny model is an example of an available option and the most important for our study. This model, in particular, is specially crafted for higher performance and less computational cost than the base YOLO model. Another class of models has been explored in recent years, the YOLO-Nano model. These studies started with the YOLOv3 architecture [92] and have received increasing attention, especially with the recently released YOLOv5-Nano [16]. Such models have achieved exceptional performance results while still maintaining great detection results.

Inspired by this and employing the newly acquired knowledge of Model Compression, we proceeded to further expand on Tiny and Nano's key features and lower its hardware restriction even further. The following section addresses the Model Compression techniques we explored and how they were employed.

6.3 Compression Methods

As mentioned in the previous section, because of how restrictive our case study scenario is, we have chosen state-of-the-art efficient neural networks as the basis for the implementation of our model. However, after some initial experiments, we identified that they still were not efficient enough to achieve the desired performance on our limited-hardware scenario. Thus, to address the challenge of adapting a neural network into our hardware-restricted scenario, we decided to adopt some model compression techniques to reduce the number of computational resources required by the network during inference. To achieve this, we selected the techniques pruning and quantization, which directly affect the number

[1] https://github.com/FiDaSS/FiDaSS_dataset

of resources the network requires and experimented with combining different strategies to approach each of these techniques. It is important to note that whenever we experimented with a combination of these techniques, we also included a fine-tuning step to guarantee that the compressed model would achieve the best results it could.

The first technique we explored was pruning, which allowed us to minimize the number of redundant and irrelevant information stored on the trained model. We focused on two main approaches to pruning: Weight Pruning and Filter Pruning. The process of Weight Pruning consisted of identifying the least important weights, and a portion of those that least contribute to the network's output, based of a percentage informed as a hyper-parameter, are substituted by zero by applying a binary mask to the weights of each filter, When applying Filter Pruning we removed irrelevant weight values by applying a zero-mask to each filter, changing those specific weights to zero, and then proceeding to cut a percentage of the least relevant filters from the model.

Then, after we had chosen which prune approach we would use, if any, we proceeded to choose a quantization approach. We experimented with two quantization approaches, in both cases mapping the float weights to $8 - bit$ values. The first quantization we used is made by Google and proposed by Jacob et al. [37], which allows for integer-only operations during inference. The second one is called DoReFa-Net and was proposed by Zhou et al. [96], which specializes in low bitwidth representations.

7. EXPERIMENTAL RESULTS

In this chapter, we provide an assessment of FiDaSS on Section 7.1, demonstrating how challenging the dataset is by presenting an evaluation of state-of-the-art neural networks trained on it. Additionally, we also provide comparisons between the model compression techniques explored. On Section 7.2.1 we present comparisons between the results each technique achieved, while on Section 7.2.2 we discuss how the techniques compare when applied to our case study.

7.1 Dataset Evaluation

FiDaSS aims to promote more research in the area by introducing a novel, challenging object detection dataset. We used FiDaSS to train five state-of-the-art architectures: YOLOv4-Full [6]; YOLOv4-Tiny [6]; YOLOv3-Nano [92]; SSD300 [48] with an EfficientNetB3 [83] backbone; SSD512 [48] with a VGG-16 [81] backbone. After which, we used them to evaluate our case study and assess our dataset. For the training of our models, and later for the inference, we used a machine with an NVidia GeForce RTX 2080 GPU with an 8GB memory, 64GB RAM, and an i5-9400F 2.90GHz CPU with six cores. The metrics we adopted to compare the results achieved were: Precision, Recall, Average Precision (AP_{50}), and F-measure (F1).

Configuration ID	Annotated Images		Non-Annotated Images		Amount of Images		
	Original	Augmented	Original	Augmented	Training	Validation	Test
Configuration #1	**X**				2716	623	603
Configuration #2	**X**	**X**			10776	623	603
Configuration #3	**X**		**X**		4770	1068	1056
Configuration #4	**X**	**X**	**X**		12830	1068	1056
Configuration #5	**X**	**X**	**X**	**X**	18994	1068	1056

Table 7.1 – Description of the different data configurations we experimented with.

Before starting our experiments, we organized our dataset in five different configurations, shown in Table 3.3, each using a unique combination of data. Each configuration corresponds to the presence or absence of the following data groups: images containing firearm threats, images not containing firearm threats, and their respective augmented versions. Instead of only training one dataset, we were interested in analyzing how adding and removing data would impact the results achieved by the networks. However, when comparing multiple architectures, we only used configuration #1 as a basis and used all five configurations only on YOLOv4-Tiny, which we focused on in this research.

We present the results achieved by the five architectures trained on Table 7.2. We can observe that the results obtained are not ideal and would need to be improved a lot

Architecture	Precision	Recall	AP_{50}	F1
YOLOv4-Full	65.27%	67.28%	55.46%	59.97%
YOLOv4-Tiny	73.56%	33.49%	29.19%	46.02%
YOLOv3-Nano	39.97%	24.53%	18.57%	30.40%
SSD300	69.77%	61.62%	53.02%	65.24%
SSD512	71.74%	61.11%	53.20%	66.00%

Table 7.2 – Results achieved by training each architecture on FiDaSS.

before they could be used in a real-world scenario, given the severe implications incorrect predictions may implicate. While the YOLOv4-Full and SSD architectures achieve overall better results, the YOLOv4-Tiny architecture achieves the best precision, although it also has a very low recall. While we were interested in exploring the YOLOv3-Nano architecture, since it is optimized for high performance, it could not generalize our dataset well and its results were a lot lower than we expected.

Training	Precision	Recall	AP_{50}	F1
#1	73.56%	33.49%	29.19%	46.02%
#2	69.30%	36.57%	30.10%	47.88%
#3	70.93%	43.67%	36.49%	54.06%
#4	73.49%	37.65%	33.06%	49.80%
#5	73.90%	31.02%	26.88%	43.70%

Table 7.3 – Results of each of our training configurations.

Table 7.3 presents an assessment of YOLOv4-Tiny on the five configurations of our dataset. We can see that changing which groups we used had little effect on precision, but including images without firearm threat caused a significant increase in the model's recall. Something surprising and unexpected is that including augmented images without firearm threat lowered the recall a lot, being the lowest recall of all five experiments, but also had a small positive impact on precision, achieving the highest of the experiments.

7.2 Model Compression Evaluation

To identify the most valuable combination of the model compression techniques studied, we conducted two sets of experiments: an evaluation of the results achieved by each combination, discussed in Section 7.2.1, and an analysis of the performance they achieved on our case study, presented in Section 7.2.2.

7.2.1 Detection Performance

In this section, we address the experiments about the results achieved, discussing how each technique performed on average, what were the best combinations, and the insights gained from these experiments.

Configuration ID	Prunning Strategy	Quantization Strategy
CFG_A	No Pruning	No Quantization
CFG_B		Google's Quantization
CFG_C		DoReFa-Net Quantization
CFG_D	Weight Pruning	No Quantization
CFG_E		Google's Quantization
CFG_F		DoReFa-Net Quantization
CFG_G	Filter Pruning	No Quantization
CFG_H		Google's Quantization
CFG_I		DoReFa-Net Quantization

Table 7.4 – Description of the different model compression combinations we experimented with.

We performed this set of experiments with a very similar methodology to those presented in Section 7.1. We started by organizing the combinations we would be applying to the network, presented in Table 7.4, and deciding on hyper-parameters such as the percentage of the model that would be pruned. Although we initially conceptualized using four different pruning percentages, those being [0.1%, 5%, 10%, 25%], our initial experiments showed no interesting new results from using most of them as generally they would either underfit or overfit to our dataset. Thus we decided to focus only on a pruning percentage of 0.1%, as it was the one that showed the most exciting results. Then we proceeded to set up our training environment, using the same system as the experiments in Section 7.1, as well as preparing a small set of tools to help us manage the training sessions and the evaluation results.

To evaluate each combination, we decided that it would be interesting to see how they performed after being trained by different group configurations of our dataset, from those presented in Table 7.1, deciding ultimately on configurations #1, #3, and #5. However, since we are comparing the results achieved by the different techniques and combinations, we decided it would be better to use the same validation and test sets for all trained networks, as it provides a more fair evaluation. By looking at Table 7.1, we can see that configurations #3 and #5 already shared the same test and validation sets, while configuration #1 used different ones. Thus, for the context of this evaluation only, we made so configuration #1 would share the same data sets as the other two configurations, making the training set the only distinguishable feature between the three configurations.

Network Configuration	Pruning Percentage	Train Set #1			Train Set #3			Train Set #5		
		P	R	AP_{50}	P	R	AP_{50}	P	R	AP_{50}
CFG_A	NA	73.60%	33.50%	29.20%	70.90%	43.70%	36.50%	73.90%	31.00%	26.90%
CFG_B	NA	53.70%	68.20%	53.90%	59.80%	54.60%	52.10%	55.90%	42.80%	41.20%
CFG_C	NA	54.70%	67.70%	52.30%	63.60%	47.20%	51.10%	52.60%	47.40%	46.10%
CFG_D	0.1%	74.00%	32.00%	40.40%	57.70%	45.00%	45.00%	66.10%	37.30%	44.50%
CFG_E	0.1%	72.80%	34.30%	44.30%	53.40%	48.10%	48.50%	62.00%	36.40%	42.20%
CFG_F	0.1%	68.00%	38.50%	46.10%	50.90%	50.50%	49.10%	57.30%	46.30%	47.20%
CFG_G	0.1%	72.20%	30.00%	35.50%	57.60%	51.00%	48.30%	45.40%	45.20%	42.20%
CFG_H	0.1%	76.20%	33.80%	46.20%	56.90%	46.10%	48.00%	62.00%	36.40%	42.20%
CFG_I	0.1%	69.80%	38.30%	45.80%	50.90%	50.50%	49.10%	52.80%	50.30%	48.80%

Table 7.5 – Results achieved by each explored combination of model compression techniques. It is important to note that the hyper-parameter "pruning percentage" is not applicable (NA) to the first three configurations, due to them not employing any pruning technique.

The results achieved by the combinations explored are presented in Table 7.5, where each training realized corresponds to a triple of (*Precision*, *Recall*, *Average Precision*). We start by presenting the results achieved by the base network, which were already discussed in Section 7.1, but served as a relevant basis of comparison. However, it is important to note that these base results were acquired using a different resolution from the one we used in these new experiments. For our previous experiments, we used a rectangular input of dimensions $(768, 480)$, which is approximately the average of the many different original resolutions from the data we gathered for FiDaSS. However, for these new experiments, we used a square input of dimensions $(512, 512)$ to reduce the number of pixels on the image, thus improving the performance of our model by both reducing the load needed during I/O and the amount of data the neural network needed to process.

Analyzing our results, the first thing we noticed was also something we did not expect: applying model compression to our trained networks caused a significant increase in their average precision. While we did expect it to change, since we were removing extra and irrelevant information, the amount changed was beyond our expectations and demonstrates that model compression can be beneficial not only to performance but also to improve the model's overall quality. This is further reinforced by the results of configurations CFG_B and CFG_C, where adding quantization and reducing the input's dimensions caused an improvement on each training's recall, especially those trained on configuration #1. Although a major part of the massive improvement in this configuration, in particular, is because while the original training used only data with firearm threat, it was fine-tuned using images without these events, thus allowing it to differentiate better between these two situations. It is also interesting to note that, while CFG_B with Google's quantization method achieved the best average precision for configurations #1 and #3, the combination of Filter Pruning and DoReFa-Net's quantization method were responsible for the best average precision of the training configuration #5.

7.2.2 Time Performance

In this section, we present the final set of experiments realized, which corresponds to the evaluation of the model compression techniques on our case study. These experiments were conducted in a Samsung Galaxy S7, with eight cores ($4\times$ $2.60GHz$ *Exynos M1* and a $4\times$ $1.59GHz$ *Cortex-A53*) with 4GB RAM. It is important to note that the performance shown is an average of five executions, which ran in sequence after rebooting the smartphone. We did this to avoid having any cached data influencing the performance of the executions, thus ensuring that we can compare our results fairly.

Before checking how the compressed networks performed on our case study, we first analyzed the performance of the original architectures on it. These initial results were not

ideal as the YOLOv4 architectures performed very poorly on the limited-hardware scenario we proposed. Ideally, we would want to use the YOLOv4-Full for our model since it achieved the best results on our dataset (as discussed in Section 7.1). However, it was the worst-performing architecture of these initial experiments by performing about 10$\times$ slower than the YOLOv4-Tiny, the second-best performing model of this initial batch of experiments. As expected, the YOLOv3-Nano architecture was the one that achieved the best performance out of the architectures we explored, but we had already identified that it could not generalize our dataset well and thus was not a good fit for our model. Therefore, based on these initial results and the ones presented in Table 7.2, we decided to focus the model compression evaluation primarily on the YOLOv4-Tiny architecture, even if it did not achieve a performance of even 1*fps*. This architecture is also a lot more malleable and has more room for changes and improvements than the YOLOv3-Nano, which is already a product of a model compression approach.

Network Configuration	Batch Size			
	1 Frame	2 Frames	4 Frames	8 Frames
CFG_B	1.9	3.9	3.9	3.9
CFG_C	1.9	3.8	4.0	4.1
CFG_D	2.2	4.1	4.2	4.3
CFG_E	1.8	3.5	3.9	4.0
CFG_F	1.9	3.8	3.9	4.0
CFG_G	2.2	3.9	4.1	4.1
CFG_H	1.9	3.8	3.9	4.0
CFG_I	1.9	3.4	3.8	4.1

Table 7.6 – Performance in frames per second achieved on our case study by each of our experiments of combining model compression techniques on the YOLOv4-Tiny architecture.

Table 7.6 presents the performance achieved by the compressed networks we discussed in the previous section, showing how their performance changes based on the number of images being processed at once in a batch. By analyzing all the performances achieved, we can see that while we managed to improve upon YOLOv4-Tiny's performance, we did not manage to achieve a high rate of processed frames per second. This further corroborates how difficult the challenge we addressed is, especially in extreme cases such as our case study. By looking specifically at how the performances changed based on the batch size, we can see that increasing the batch size caused a minimal gain of performance that diminishes rapidly, due to the low amount of memory available. Considering that all the performances achieved processed at least 4*fps*, we found that the optimal usage of the memory was feeding the network two images at a time. Finally, we can see that the performance achieved by the pruning strategies did not differ much from each other, which we believe is due to the low prune percentage value we chose.

8. DISCUSSION

This chapter summarizes our contributions to the research area, the limitations of the proposed approach, and our plans for continuing the research, contemplating both the dataset created and the proposed model.

8.1 Contributions

We propose a dataset for firearm threat detection that improves pre-existing ones by adding novel images and new information to the annotations. FiDaSS is one of the few datasets made with real data, being the largest one in this context, as far as we know, with annotations for object detection. Along with our dataset, the tools we developed to create it are also available online. Having these available makes FiDaSS easily expandable and adaptable for varying needs that may arise from approaching the issue from different angles.

We also propose a model for managing a surveillance environment and identifying dangerous situations involving firearms. Our model helps the surveillance process by constantly processing images depicting the environment from a CCTV camera feed and notifying responsible personnel when it recognizes that a dangerous situation is starting. Additionally, we also provide an extensive set of comparisons over the performance of state-of-the-art network architectures on the proposed dataset, demonstrating their detection capabilities and the performance they can achieve when combined with specific model compression strategies. By analyzing these results, we can determine which combination of neural network architecture and compression technique is the best fit for the challenge and should be chosen for our proposed model.

8.2 Limitations

Our dataset has comprehensive data from South and North America (gathered primarily from our YouTube playlist and existing datasets, respectively). However, we still lack volume for other continents, especially those in the orient. A limitation of FiDaSS, although we sought diversity in our dataset, is the minimal number of examples of some cultures while having ample data about others. This characteristic implies that approaches based on our dataset are more inclined to identify everyday situations from the Americas but may not be as prepared to recognize events from oriental countries. This lack of volume also makes our data unbalanced towards specific demographics, which may introduce a particular bias on models learning from our dataset.

As discussed throughout Sections 7.1 and 7.2.1, the models trained have achieved less than desirable detection results, especially regarding the recall. Although these models can run efficiently even in our limited hardware scenario, their detections are not reliable enough for our approach to be realistically used in a surveillance environment yet. Even if our approach can consistently identify a dangerous situation, it needs further improvements before it reaches the point where people can trust it with their security. Furthermore, the alarming amount of false alarms raised would also be an issue since our approach still raises too many false positives to help monitor an environment.

While we did achieve a desirable level of performance for our approach, we did not get to explore the more sophisticated compression techniques, and processing four frames each second might miss some key moments that a more efficient model would be able properly identify. Although the techniques we used were the most versatile and valuable in most cases, they are also the ones with the least impact compared to the ones we did not explore as in-depth. Therefore, our approach can still be optimized further to enhance performance and detection results.

8.3 Future Work

We want to enhance FiDaSS by adding more data and classes to it for future work. We expect that adding a class for unarmed people might provide new exciting information while also helping the neural networks achieve a better recall, which is an strategy adopted by Lim et al.[46] on their dataset. Further effort will also be dedicated to expanding FiDaSS with new and unexplored data to introduce more diversity of situations and images on our dataset, seeking to provide a more substantial and representative amount of data for each culture and demographic. We also plan to develop new tools to facilitate expanding the dataset. For example, we plan to organize a strategy to select frames more systematically when gathering data from new videos, thus minimizing manually analyzing frames looking for the best fit for our dataset. Another tool we are interested in improving is the one responsible for anonymizing the frames selected by the previously mentioned stage. With the new classes added to the dataset, we expect to reliably automate the blurring process and minimize the number of incorrectly blurred images we need to check and correct manually.

As for our model, the first options we want to explore are different model compression techniques and other network architectures. We will start with deeper research of advanced techniques and then experiment and combine them with our best-performing trained models, trying to achieve faster performance and be less penalized on the detection results achieved. Furthermore, another interesting experiment is trying the various compression techniques on different architectures in the literature. While we did focus on the YOLOv4

architectures in this work, many other great one-stage detectors can yield interesting results – including the recent YOLOv5 [16].

Finally, we are also interested in taking the research in a new direction, focusing on different challenges around firearm threat detection. Our experiments' consistently below-average detection results motivated us to readjust our goals. We intend to study and seek further information we can extract from the frames to help our decision process decide if there is an armed threat on the scene or just a false positive from the network. However, it is essential to note that we are still considering performance even if we are adding more steps into our pipeline. Nevertheless, with these new goals, we will be easing the restrictions of the environment responsible for running the model, thus allowing us to make a model more reliable at the cost of a more considerable hardware investment.

9. CONCLUSIONS

This work presented a novel dataset for firearm threat detection containing images from real-world situations, focusing on object detection. It has 6942 images with high variability, i.e., without a contiguous sequence of frames that do not present new information. The available scripts used to assemble the dataset allow us to extend it easily. Moreover, our experiments assert how challenging it is, showing that state-of-the-art methods have difficulty with it, primarily when dealing with false negatives, making it an exciting alternative for future research. We attribute this to the high similarity of a person holding a gun and holding an arbitrary object that was not caught well by the camera.

We also propose a model to aid in managing surveillance systems environment by performing firearm threat detection, identifying, and notifying responsible personnel as fast as possible before the situation escalates. As a case study, we implemented our model focusing on small systems and requiring low investment to achieve the desired performance. The model processes images captured from a smartphone camera using a YOLO architecture optimized with model compression techniques. Thus, we make it accessible to a broader spectrum of people regardless of the scale of the environment being monitored.

By employing the pruning and quantization techniques, we managed modify the YOLOv4-Tiny architecture to best fit the needs of our case study, managing to achieve a performance that would satisfy our needs without deteriorating the architecture's detection results. However, our analysis both on detection results and performance achieved shows that that there is still room for improvements, and we hope our research helps instigate more research in this area. Finally, in addition to promoting research in the area, we hope to contribute to security in our everyday lives.

REFERENCES

[1] Abruzzo, B.; Carey, K.; Lowrance, C.; Sturzinger, E.; Arnold, R.; Korpela, C. “Cascaded neural networks for identification and posture-based threat assessment of armed people”. In: Proceedings of the IEEE International Symposium on Technologies for Homeland Security, 2019, pp. 1–7.

[2] Ainsworth, T. “Buyer beware”, *Security Oz*, vol. 19, May, 2002, pp. 18–26.

[3] Basit, A.; Munir, M. A.; Ali, M.; Werghi, N.; Mahmood, A. “Localizing firearm carriers by identifying human-object pairs”. In: Proceedings of the IEEE International Conference on Image Processing, 2020, pp. 2031–2035.

[4] Bay, H.; Tuytelaars, T.; Van Gool, L. “Surf: Speeded up robust features”. In: Proceedings of the European conference on computer vision, 2006, pp. 404–417.

[5] Bhattacharjee, S. D.; Talukder, A. “Graph clustering for weapon discharge event detection and tracking in infrared imagery using deep features”. In: Proceedings of the Pattern Recognition and Tracking XXVIII, 2017, pp. 154–163.

[6] Bochkovskiy, A.; Wang, C.; Liao, H. M. “Yolov4: Optimal speed and accuracy of object detection”, *Computing Research Repository*, vol. abs/2004.10934, Apr, 2020.

[7] Buciluǎ, C.; Caruana, R.; Niculescu-Mizil, A. “Model compression”. In: Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, pp. 535–541.

[8] Cardoso, G. V. S.; Ciarelli, P. M.; Vassallo, R. F. “Use of deep learning for firearms detection in images”. In: Anais do Workshop de Visão Computacional, 2019, pp. 109–114.

[9] Cheng, M.; Cai, K.; Li, M. “Rwf-2000: An open large scale video database for violence detection”. In: Proceedings of the International Conference on Pattern Recognition, 2021, pp. 4183–4190.

[10] Csurka, G.; Dance, C.; Fan, L.; Willamowski, J.; Bray, C. “Visual categorization with bags of keypoints”. In: Proceedings of the Workshop on statistical learning in computer vision, ECCV, 2004, pp. 1–2.

[11] Dadashi, N.; Stedmon, A. W.; Pridmore, T. P. “Semi-automated cctv surveillance: The effects of system confidence, system accuracy and task complexity on operator vigilance, reliance and workload”, *Applied ergonomics*, vol. 44–5, Sep, 2013, pp. 730–738.

[12] Darker, I.; Gale, A.; Ward, L.; Blechko, A. “Can cctv reliably detect gun crime?” In: Proceedings of the 41st Annual IEEE International Carnahan Conference on Security Technology, 2007, pp. 264–271.

[13] de Azevedo Kanehisa, R. F.; de Almeida Neto, A. “Firearm detection using convolutional neural networks”. In: Proceedings of the International Conference on Agents and Artificial Intelligence, 2019, pp. 707–714.

[14] Dee, H. M.; Velastin, S. A. “How close are we to solving the problem of automated visual surveillance?”, *Machine Vision and Applications*, vol. 19–5, Jan, 2008, pp. 329–343.

[15] Donald, F. M.; Donald, C. H. “Task disengagement and implications for vigilance performance in cctv surveillance”, *Cognition, Technology & Work*, vol. 17–1, Feb, 2015, pp. 121–130.

[16] Duran-Vega, M. A.; Gonzalez-Mendoza, M.; Chang-Fernandez, L.; Suarez-Ramirez, C. D. “Tyolov5: A temporal yolov5 detector based on quasi-recurrent neural networks for real-time handgun detection in video”, *Computing Research Repository*, vol. abs/2111.08867, Nov, 2021.

[17] Dwivedi, N.; Singh, D. K.; Kushwaha, D. S. “Weapon classification using deep convolutional neural network”. In: Proceedings of the IEEE Conference on Information and Communication Technology, 2019, pp. 1–5.

[18] Egiazarov, A.; Mavroeidis, V.; Zennaro, F. M.; Vishi, K. “Firearm detection and segmentation using an ensemble of semantic neural networks”. In: Proceedings of the European Intelligence and Security Informatics Conference, 2019, pp. 70–77.

[19] Feng, X.; Jiang, Y.; Yang, X.; Du, M.; Li, X. “Computer vision algorithms and hardware implementations: A survey”, *Integration*, vol. 69, Nov, 2019, pp. 309–320.

[20] Fernandez-Carrobles, M. M.; Deniz, O.; Maroto, F. “Gun and knife detection based on faster r-cnn for video surveillance”. In: Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, 2019, pp. 441–452.

[21] Fiesler, E.; Choudry, A.; Caulfield, H. J. “Weight discretization paradigm for optical neural networks”. In: Proceedings of the Optical interconnections and networks, 1990, pp. 164–173.

[22] Gabor, D. “Theory of communication. part 1: The analysis of information”, *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, vol. 93–26, Nov, 1946, pp. 429–441.

[23] Gelana, F.; Yadav, A. “Firearm detection from surveillance cameras using image processing and machine learning techniques”. In: Proceedings of the Smart innovations in communication and computational sciences, Jan, 2019, pp. 25–34.

[24] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. “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.

[25] Gius, M. “The effects of state and federal gun control laws on school shootings”, *Applied economics letters*, vol. 25–5, Mar, 2018, pp. 317–320.

[26] González, J. L. S.; Zaccaro, C.; Álvarez-García, J. A.; Morillo, L. M. S.; Caparrini, F. S. “Real-time gun detection in cctv: An open problem”, *Neural networks*, vol. 132, Dec, 2020, pp. 297–308.

[27] Goodfellow, I.; Bengio, Y.; Courville, A. “Deep Learning”. MIT Press, 2016, http://www.deeplearningbook.org.

[28] Grega, M.; Lach, S.; Sieradzki, R. “Automated recognition of firearms in surveillance video”. In: Proceedings of the IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support, 2013, pp. 45–50.

[29] Grega, M.; Matiolański, A.; Guzik, P.; Leszczuk, M. “Automated detection of firearms and knives in a cctv image”, *Sensors*, vol. 16–1, Jan, 2016, pp. 47.

[30] Gupta, M. “Detection and recognition of handguns in the surveillance videos using neural network”, *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, Jun, 2020, pp. 1566–1570.

[31] He, K.; Zhang, X.; Ren, S.; Sun, J. “Deep residual learning for image recognition”. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[32] Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”, *Computing Research Repository*, vol. abs/1704.04861, Apr, 2017.

[33] Hurka, S.; Knill, C. “Does regulation matter? a cross-national analysis of the impact of gun policies on homicide and suicide rates”, *Regulation & Governance*, vol. 14–4, Oct, 2020, pp. 787–803.

[34] Hussein, N. J.; Hu, F. “An alternative method to discover concealed weapon detection using critical fusion image of color image and infrared image”. In: Proceedings of the First IEEE International Conference on Computer Communication and the Internet, 2016, pp. 378–383.

[35] imfdb, L. “Internet movie firearms database”. http://www.imfdb.org/index.php?title=Main_Page&oldid=911151. Last accessed in 18/01/2021.

[36] Ineneji, C.; Kusaf, M. “Hybrid weapon detection algorithm, using material test and fuzzy logic system”, *Computers & Electrical Engineering*, vol. 78, Sep, 2019, pp. 437–448.

[37] Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. “Quantization and training of neural networks for efficient integer-arithmetic-only inference”. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2704–2713.

[38] Jubaer, A. N. M. “Weapons in images”. https://www.kaggle.com/jubaerad/weapons-in-images-segmented-videos/. Last accessed in 15/02/2021.

[39] Kakadiya, R.; Lemos, R.; Mangalan, S.; Pillai, M.; Nikam, S. “Ai based automatic robbery/theft detection using smart surveillance in banks”. In: Proceedings of the 3rd International conference on Electronics, Communication and Aerospace Technology, 2019, pp. 201–204.

[40] Kaur, A.; Kaur, L. “Concealed weapon detection from images using sift and surf”. In: Proceedings of the Online International Conference on Green Engineering and Technologies, 2016, pp. 1–8.

[41] Kolesnikov, A.; Beyer, L.; Zhai, X.; Puigcerver, J.; Yung, J.; Gelly, S.; Houlsby, N. “Big transfer (bit): General visual representation learning”. In: Proceedings of the European conference on computer vision, Oct, 2020, pp. 491–507.

[42] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. “Imagenet classification with deep convolutional neural networks”, *Advances in neural information processing systems*, vol. 25, Dec, 2012, pp. 1097–1105.

[43] Lai, J.; Maples, S. “Developing a real-time gun detection classifier”, Research report, Stanford University, 2017.

[44] LeCun, Y.; Denker, J. S.; Solla, S. A. “Optimal brain damage”. In: Proceedings of the Advances in neural information processing systems, 1990, pp. 598–605.

[45] Lemieux, F. “Effect of gun culture and firearm laws on gun violence and mass shootings in the united states: A multi-level quantitative analysis”, *International Journal of Criminal Justice Sciences*, vol. 9–1, Jan, 2014, pp. 74.

[46] Lim, J.; Al Jobayer, M. I.; Baskaran, V. M.; Lim, J. M.; See, J.; Wong, K. “Deep multi-level feature pyramids: Application for non-canonical firearm detection in video surveillance”, *Engineering applications of artificial intelligence*, vol. 97, Jan, 2021, pp. 104094.

[47] Lim, J.; Al Jobayer, M. I.; Baskaran, V. M.; Lim, J. M.; Wong, K.; See, J. “Gun detection in surveillance videos using deep neural networks”. In: Proceedings of the Asia-Pacific

Signal and Information Processing Association Annual Summit and Conference, 2019, pp. 1998–2002.

[48] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A. C. “Ssd: Single shot multibox detector”. In: Proceedings of the European conference on computer vision, 2016, pp. 21–37.

[49] Loganathan, S.; Kariyawasam, G.; Sumathipala, P. “Suspicious activity detection in surveillance footage”. In: Proceedings of the International Conference on Electrical and Computing Technologies and Applications, 2019, pp. 1–4.

[50] Long, J.; Shelhamer, E.; Darrell, T. “Fully convolutional networks for semantic segmentation”. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

[51] Lowe, D. G. “Object recognition from local scale-invariant features”. In: Proceedings of the seventh IEEE international conference on computer vision, 1999, pp. 1150–1157.

[52] Mahajan, R.; Padha, D. “Detection of concealed weapons using image processing techniques: A review”. In: Proceedings of the First International Conference on Secure Cyber Computing and Communication, 2018, pp. 375–378.

[53] Martínez-Díaz, S.; Palacios-Alvarado, C. A.; Chavelas, S. M. “Accelerated pistols recognition by using a gpu device”. In: Proceedings of the IEEE XXIV International Conference on Electronics, Electrical Engineering and Computing, 2017, pp. 1–4.

[54] McDowall, D. “Firearm availability and homicide rates in detroit, 1951–1986”, *Social Forces*, vol. 69–4, Jun, 1991, pp. 1085–1101.

[55] Mehta, P.; Kumar, A.; Bhattacharjee, S. “Fire and gun violence based anomaly detection system using deep neural networks”. In: Proceedings of the International Conference on Electronics and Sustainable Communication Systems, 2020, pp. 199–204.

[56] Moura, N. S.; Gondim, J. M.; Claro, D. B.; Souza, M.; de Cerqueira Figueiredo, R. “Detection of weapon possession and fire in public safety surveillance cameras”. In: Anais do Encontro Nacional de Inteligência Artificial e Computacional, 2021, pp. 290–301.

[57] Muchiri, H.; Ateya, I.; Wanyembi, G. “The need for marker-less computer vision techniques for human gait analysis on video surveillance to detect concealed firearms”, *International Journal of Computer*, vol. 29, Apr, 2018, pp. 107–118.

[58] Nakib, M.; Khan, R. T.; Hasan, M. S.; Uddin, J. “Crime scene prediction by detecting threatening objects using convolutional neural network”. In: Proceedings of

the International Conference on Computer, Communication, Chemical, Material and Electronic Engineering, 2018, pp. 1–4.

[59] Navalgund, U. V.; Priyadharshini, K. “Crime intention detection system using deep learning”. In: Proceedings of the International Conference on Circuits and Systems in Digital Enterprise Technology, 2018, pp. 1–6.

[60] Olmos, R.; Tabik, S.; Herrera, F. “Automatic handgun detection alarm in videos using deep learning”, *Neurocomputing*, vol. 275, Jan, 2018, pp. 66–72.

[61] Olmos, R.; Tabik, S.; Lamas, A.; Perez-Hernandez, F.; Herrera, F. “A binocular image fusion approach for minimizing false positives in handgun detection with deep learning”, *Information Fusion*, vol. 49, Sep, 2019, pp. 271–280.

[62] Pappas, T. N.; Jayant, N. S. “An adaptive clustering algorithm for image segmentation”. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 1989, pp. 1667–1670.

[63] Pavan, V.; Yeshwanth, D.; Bharath, S.; Narasimha, P. L. “Intelligent surveillance system to handle sudden arms attack in less secured areas”, *Journal of Xi’an University of Architecture & Technology*, vol. 12–4, Apr, 2020, pp. 3407–3411.

[64] Pawar, M. “A novel approach to detect crimes and assist law enforcement agency using deep learning with cctvs and drones”, *International Journal for Research in Applied Science and Engineering Technology*, vol. 7, Dec, 2019, pp. 653–662.

[65] Pérez-Hernández, F.; Tabik, S.; Lamas, A.; Olmos, R.; Fujita, H.; Herrera, F. “Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance”, *Knowledge-Based Systems*, vol. 194, Apr, 2020, pp. 105590.

[66] Pham, H.; Xie, Q.; Dai, Z.; Le, Q. V. “Meta pseudo labels”, *Computing Research Repository*, vol. abs/2003.10580, Mar, 2020.

[67] Piza, E. L.; Welsh, B. C.; Farrington, D. P.; Thomas, A. L. “Cctv surveillance for crime prevention: A 40-year systematic review with meta-analysis”, *Criminology & Public Policy*, vol. 18–1, Feb, 2019, pp. 135–159.

[68] Qi, D.; Tan, W.; Liu, Z.; Yao, Q.; Liu, J. “A dataset and system for real-time gun detection in surveillance video using deep learning”. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2021, pp. 667–672.

[69] Ramzan, M.; Abid, A.; Khan, H. U.; Awan, S. M.; Ismail, A.; Ahmed, M.; Ilyas, M.; Mahmood, A. “A review on state-of-the-art violence detection techniques”, *IEEE Access*, vol. 7, Jul, 2019, pp. 107560–107575.

[70] Raturi, G.; Rani, P.; Madan, S.; Dosanjh, S. “Adocw: An automated method for detection of concealed weapon”. In: Proceedings of the Fifth International Conference on Image Information Processing, 2019, pp. 181–186.

[71] Redmeon, J.; Farhadi, A. “Yolo9000: better, faster, stronger”. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.

[72] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. “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.

[73] Redmon, J.; Farhadi, A. “Yolov3: An incremental improvement”, *Computing Research Repository*, vol. abs/1804.02767, Apr, 2018.

[74] Ren, S.; He, K.; Girshick, R.; Sun, J. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: Proceedings of the Advances in neural information processing systems, 2015, pp. 91–99.

[75] Romero, D.; Salamea, C. “Convolutional models for the detection of firearms in surveillance videos”, *Applied Sciences*, vol. 9–15, Aug, 2019, pp. 2965.

[76] Ruiz-Santaquiteria, J.; Velasco-Mata, A.; Vallez, N.; Bueno, G.; Álvarez-García, J. A.; Deniz, O. “Handgun detection using combined human pose and weapon appearance”, *IEEE Access*, vol. 9, Sep, 2021, pp. 123815–123826.

[77] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al.. “Imagenet large scale visual recognition challenge”, *International journal of computer vision*, vol. 115–3, Dec, 2015, pp. 211–252.

[78] Salido, J.; Lomas, V.; Ruiz-Santaquiteria, J.; Deniz, O. “Automatic handgun detection with deep learning in video surveillance images”, *Applied Sciences*, vol. 11–13, Jun, 2021, pp. 6085.

[79] Shidik, G. F.; Noersasongko, E.; Nugraha, A.; Andono, P. N.; Jumanto, J.; Kusuma, E. J. “A systematic review of intelligence video surveillance: Trends, techniques, frameworks, and datasets”, *IEEE Access*, vol. 7, Nov, 2019, pp. 170457–170473.

[80] Siegel, M.; Ross, C. S.; King III, C. “The relationship between gun ownership and firearm homicide rates in the united states, 1981–2010”, *American journal of public health*, vol. 103–11, Nov, 2013, pp. 2098–2105.

[81] Simonyan, K.; Zisserman, A. “Very deep convolutional networks for large-scale image recognition”, *Computing Research Repository*, vol. abs/1409.1556, Sep, 2014.

[82] Sultani, W.; Chen, C.; Shah, M. “Real-world anomaly detection in surveillance videos”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6479–6488.

[83] Tan, M.; Le, Q. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: Proceedings of the International conference on machine learning, 2019, pp. 6105–6114.

[84] Tiwari, R. K.; Verma, G. K. “A computer vision based framework for visual gun detection using surf”. In: Proceedings of the International Conference on Electrical, Electronics, Signals, Communication and Optimization, 2015, pp. 1–5.

[85] Tiwari, R. K.; Verma, G. K. “A computer vision based framework for visual gun detection using harris interest point detector”, *Procedia Computer Science*, vol. 54, Jan, 2015, pp. 703–712.

[86] Vallez, N.; Velasco-Mata, A.; Deniz, O. “Deep autoencoder for false positive reduction in handgun detection”, *Neural Computing and Applications*, vol. 33–11, Jun, 2021, pp. 1–11.

[87] Velastin, S. A.; Boghossian, B. A.; Vicencio-Silva, M. A. “A motion-based image processing system for detecting potentially dangerous situations in underground railway stations”, *Transportation Research Part C: Emerging Technologies*, vol. 14–2, Apr, 2006, pp. 96–113.

[88] Verma, G. K.; Dhillon, A. “A handheld gun detection using faster r-cnn deep learning”. In: Proceedings of the International Conference on Computer and Communication Technology, 2017, pp. 84–88.

[89] von F, Z. “Beugungstheorie des schneidenver-fahrens und seiner verbesserten form, der phasenkontrastmethode”, *physica*, vol. 1–7-12, Feb, 1934, pp. 689–704.

[90] Warsi, A.; Abdullah, M.; Husen, M. N.; Yahya, M. “Automatic handgun and knife detection algorithms: A review”. In: Proceedings of the International Conference on Ubiquitous Information Management and Communication, 2020, pp. 1–9.

[91] Warsi, A.; Abdullah, M.; Husen, M. N.; Yahya, M.; Khan, S.; Jawaid, N. “Gun detection system using yolov3”. In: Proceedings of the IEEE International Conference on Smart Instrumentation, Measurement and Application, 2019, pp. 1–4.

[92] Wong, A.; Famuori, M.; Shafiee, M. J.; Li, F.; Chwyl, B.; Chung, J. “Yolo nano: a highly compact you only look once convolutional neural network for object detection”, *Computing Research Repository*, vol. abs/1910.01271, Oct, 2019.

[93] Xu, S.; Hung, K. “Development of an ai-based system for automatic detection and recognition of weapons in surveillance videos”. In: Proceedings of the IEEE Symposium on Computer Applications & Industrial Electronics, 2020, pp. 48–52.

[94] Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. “Joint face detection and alignment using multitask cascaded convolutional networks”, *IEEE Signal Processing Letters*, vol. 23–10, Aug, 2016, pp. 1499–1503.

[95] Zhao, Q.; Sheng, T.; Wang, Y.; Tang, Z.; Chen, Y.; Cai, L.; Ling, H. “M2det: A single-shot object detector based on multi-level feature pyramid network”. In: Proceedings of the AAAI conference on artificial intelligence, 2019, pp. 9259–9266.

[96] Zhou, S.; Ni, Z.; Zhou, X.; Wen, H.; Wu, Y.; Zou, Y. “Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients”, *Computing Research Repository*, vol. abs/1606.06160, Aug, 2016.

[97] Zhou, Z.; Etinger, I. C.; Metze, F.; Hauptmann, A.; Waibel, A. “Gun source and muzzle head detection”, *Electronic Imaging*, vol. 2020–8, Jan, 2020, pp. 187–1.

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br