A REVIEW THREAT OBJECT DETECTION IN X -RAY IMAGES USING SSD, R-FCN AND FASTER R -CNN

A THESIS SUBMITTED TO

THE FACULTY OF ARCHITECTURE AND ENGINEERING

OF

EPOKA UNIVERSITY

BY

JOLA KOÇI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR

THE DEGREE OF MASTER OF SCIENCE

IN

COMPUTER ENGINEERING

July, 2020

Approval sheet of the Thesis

This is to certify that we have read this thesis entitled **"Threat object detection in X-ray images using SSD, R-FCN and Faster R-CNN"** and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Dr. Ali Osman TOPAL
Head of Department
Date: July 23, 2020

Examining Committee Members:

Assoc. Prof. Dr. Dimitrios Karras	(Computer Engineering)
Assist. Prof. Dr. Arban Uka	(Computer Engineering)
Dr. Ali Osman Topal	(Computer Engineering)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Jola Koci

Signature:_____

ABSTRACT

THREAT OBJECT DETECTION IN X-RAY IMAGES USING SSD, R-FCN AND FASTER R-CNN

Koci, Jola

M.Sc., Department of Computer Engineering

Supervisor: Prof. Dr. Ali Osman Topal

Baggage inspection for threat objects using X-ray images is a priority task that is in charge of making the risk of crime and terrorist attacks more reducible. Nowadays, the checking of baggage is based on a semi-automated system that consists of both human and also image detection. The main purpose of this thesis is to make the automatisation system more reliable. This task is mainly dependent on object detection models and algorithms. However, another part of this problem is mainly prone to the lack of the data. Furthermore, obtaining an X-ray image dataset with many types of threat objects is quite difficult. This is why this thesis is composed of two main parts: data stimulation and object detecting approaches. On the first part, due to the lack of data, an older dataset containing only four classes of threat objects are used as a base for the new objects to be stimulated into. In total, the newly simulated dataset contains seven types of threat objects consisting of: handguns, razor-blades, knife, shuriken, battery, wires and mortar. After generating the new data, they are processed and augmented by applying random types of rotations, flippings, zoomings etc. Once the images are processed, they are passed into transfer learning. Transfer learning consists of using predefined models for training. The models that are taken into consideration are: Single Shot Detector, Regions Fully Convolutional Network (R-FCN) and Faster R-CNN. These models are used by applying different techniques of feature extraction, such as: Inception-v2, MobileNet-v2 and ResNet101. Combining the object detection models and object detection architectures in total the images are trained and tested in five different approaches. In conclusion, the best detection was

achieved by the combination of Faster-RCNN detection model and ResNet101 feature extractor by 87.58% ± 0.75 accuracy.

Keywords: *Object Detection, Threat objects, X-ray Images, Data Stimulation, Data augmentation*

ABSTRAKT

DETEKTIMI I OBJEKTEVE TE RREZIKUT NE IMAZHET ME RREZE ‘X’ DUKE PERDORUR SSD, R-FCN DHE FASTER R-CNN

Koci, Jola

Master Shkencor, Departamenti i Inxhinierisë Kompjuterike

Udhëheqësi: Prof. Dr. Ali Osman Topal

Inspektimi i bagazheve për objektet e kërcënimit që përdorin imazhe me rreze X është një detyrë prioritare që është përgjegjëse për të zvogëluar rrezikun e krimit dhe sulmeve terroriste. Në ditët e sotme, kontrollimi i bagazheve bazohet në një sistem gjysëm të automatizuar i cili përbëhet si nga kontrolli njerëzor ashtu edhe nga pajisjet kompjuterike. Qëllimi kryesor i kësaj teze është ta bëjë sistemine automatizimit më të besueshëm. Kjo detyrë varet kryesisht nga modele dhe algoritme të detektimit të objekteve. Sidoqoftë, një pjesë tjetër e këtij problemi është kryesisht e prirur nga mungesa e të dhënave. Për shkak se marrja e një të dhëne me imazhe me rreze X qe të përmbajë shumë lloje të objekteve kërcënuese është mjaft e vështirë. Kjo është arsyeja pse kjo tezë është e përbërë nga dy pjesë kryesore: stimulimi i të dhënave dhe qasjet për detektimin e objekteve. Në pjesën e parë, për shkak të mungesës së të dhënave, një bazë e të dhënave më e vjetër që përmban vetëm katër klasa të objekteve kërcënuese përdoren si bazë për objektet e reja që do të stimulohen. Në total, të dhënat e reja të simuluara përmbajnë shtatë lloje të objekteve kërcënuese, të cilat përbëhen nga: pistoleta, brisku, thikë, shuriken, bateri, tela dhe mortar. Pas gjenerimit të të dhënave të reja, ato përpunohen dhe shtohen duke aplikuar lloje të rastësishme të rotacioneve, zmadhimeve etj. Pasi imazhet përpunohen, ato kalohen në transfer learning. Transfer learning konsiston në përdorimin e modeleve të paracaktuara për trajnim.

Modelet që merren në konsideratë janë: Single Shot Detector, Regional Fully Convolutional Network (R-FCN) dhe Faster R-CNN. Këto modele përdoren duke aplikuar teknika të ndryshme të ekstraktimit të veçorive, siç janë: Inception-v2, MobileNet-v2 dhe ResNet101. Nga kombinimi i modeleve dhe arkitekturave të detektimit të objekteve në total, imazhet trajnohen dhe testohen në pesë qasje të ndryshme. Si përfundim, detektimi më i mirë u arrit nga kombinimi i modelit të zbulimit Faster-RCNN dhe ekstraktuesit të veçorive ResNet101 me saktësi 87.58%.

Fjalët kyçe: *detektimi i objekteve, objekte kërcënuese, imazhe te rrezeve X, stimulimi i të dhënave*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IED	Improvised Explosive Devices
CNN	Convolution Neural Network
R-CNN	Region Convolutional Neural Network
MS COCO	MicroSoft Common Objects in COntext
SSD	Single Shot Detector
R-FCN	Region Fully Convolutional Network
ROI	Region of Interest
RPN	Region Proposed Network
YOLO	You Only Look Once
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

CHAPTER 1

INTRODUCTION

In this chapter, it will be given the breakdown of the concepts in the thesis such as: threat objects, object detection models and object detection architectures. Also, will be introduced the main research questions and hypotheses that will be addressed during the thesis.

1.1 Threat object detection

In every country, one of the highest priorities is security. The pertinence of flying security has expanded dramatically as of late and there has been generous advancement with respect to screening innovation, particularly in the field of programmed danger discovery systems. So as to accomplish better outcomes in computerized methods of being secure, object detection is a needed task. Over the years, there have been many very successful automated solutions since nowadays almost every airport has a semi-automated security policy. The automation is mainly based on scanning luggage into X-Ray images and then the next part is left out to the security guard to determine whether there is any threat revealed. Except for the long lines of people waiting to be checked, this causes many drawbacks in the security issue. One main drawback is that it leaves a window open for human error, which sometimes can be unaffordable. Therefore, automatic object-detection is a promising approach, since it does not suffer from these human limitations. Even though detection of threat objects based on material properties is already used in practice, appearance-based detection of objects in X-ray images is not yet common [1].

Baggage inspection for threat objects using X-ray images is a priority task that is in charge of making the risk of crime and terrorist attacks more reducible. The automatisation task is mainly dependent on object detection models and algorithms. Despite the algorithms another part of this problem is mainly prone to the lack of the data. Furthermore, obtaining an X-ray image dataset is quite difficult. This is why the

main part of this thesis is also focused on generating new data based on an old dataset. Having the right amount of data and the quality ones is really important in object detection. It also helps to make the automatisation a lot more accurate.

When considering the importance of this task it should be taken into consideration the latest statistics regarding this topic. Based on the latest Transportation Security Administration of USA national press release, in total 4,432 firearms were discovered in carry-on bags only for 2019 [1]. This would consist of approximately 12.1 firearms per day. On the other hand, if a closer look is given to these statistics over the years, it would show an increase of the number of firearms brought in airports (*Fig. 1*). As per the information recovered from the Global Terrorism Database (GTB) [1], there were 10,900 psychological oppressor assaults the world over in 2017 that slaughtered in excess of 26,400 people. In the USA, 692 psychological oppressor occurrences were recorded with 496 passings and 674 wounds. Equipped attack and besieging have the most noteworthy number of occurrences, which are 257 and 168, separately. With this information, the USA was recorded in the main 4 for the most number of fear monger occurrences in 2017. To forestall fear based oppressor episodes, numerous foundations, train stations, and air terminal terminals actualized tight safety efforts. Most train stations and air terminal terminals have an X-ray machine at the passageway to examine the packs of each traveler. The undertaking of the administrator is to search for danger objects like guns, blades, and explosives.

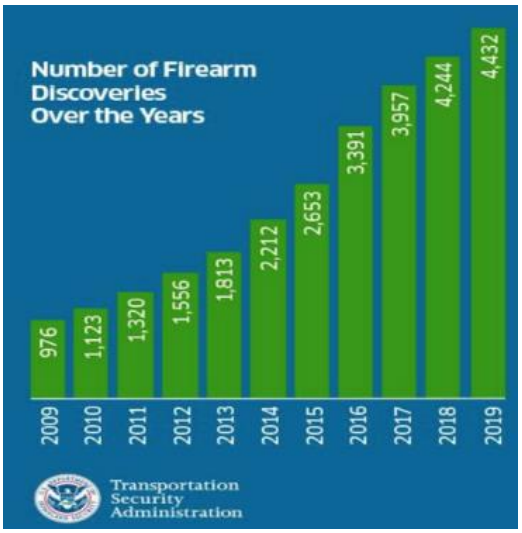


Figure 1 *Number of firearm discoveries over years.*

On the other hand, when taking into consideration threats that can occur in an airport there are many types of them. Initially when scanning the baggage there have been found numerous types of threats. Starting from the least dangerous to the most this is the list of some of the most found threat objects in x-ray images of baggage: shuriken, razor-blades, wires, batteries, laptops, knives and even types of handguns. Separately taking into consideration the wires, batteries and laptops it should not have caught any attention. Despite that, each of this things being combined with each other and with a mortar[1] would easily form improvised explosive devices (IED). IED devices are not like every other type of explosive. They are mostly, as the name suggests, "homemade" bombs with different types of matter. So, not having a predisposed dataset to make the training part of the dataset makes this task a lot harder. However, when talking about IEDs there are some main components that can be detected separately and that are consistent in any type of IEDs: battery, wires and mortar. As it can be acknowledged that IEDs are a real threat since they can be made out of components that are easily found and constructed together. The way these threats are now being processed is that each baggage that comes through an x-ray scanner machine is then processed and distinguished by the human employees. This can be a reason for most missed cases and also including the human error factor is quite unpredictable when or where the same errors as in the past will occur again. The only thing left to do is to make sure that there is a lot more weight on detecting objects put on the scanners and object detection algorithms rather than on humans. This task is dependent on more training on different types of threat objects and more high accuracy from the algorithms. As a solution, this thesis will use different types of threat object to train and test the algorithm, including: handgun, knife, razor-blade, shuriken, battery, wires and mortar. These classes will be trained and tested in different object detection models which are all based on CNN (Convolutional Neural Network). In addition to CNN, the concept of transfer learning will be applied. Taking into account that transfer learning is primarily based on using pre-trained models seams as the best way to get a higher accuracy rate.

[1] Mortar - a short smoothbore gun for firing shells (technically called bombs) at high angles[9]

1.2 Statement of purpose

The main idea of this thesis is to build a reliable approach for the detection of the threat objects. The aim of this thesis is to take a look at the current threat detection literature to determine the seriousness of this problem, to provide a brief summary of the evolution of research in this field, as well as current trends in object detection and its legal remedies for it. This thesis will provide an overview of the issues and challenges that still prevail in this area of research. In this paper, a summary of the threat detection problem, the history of attacks and best approaches to avoid these attacks from happening in the same ratio again. The taxonomy of different types of threat objects will also be ensured. To add more security to airports, the purpose of this thesis is also to experiment with real and latest data. After determining which of the research techniques is thought to yield the highest results, three of these techniques will be applied to real data to see how they can be improved. This experimental work aims to find the best method and/or algorithm to detect threat objects.

To rule the examination procedure and to accomplish a fantastic outcome at long last, the exploration process is part in four phases:

1. Development of a dataset.

First stage in the examination procedure is to develop an adequately enormous and delegate dataset that covers a wide scope of situations and potential articles that can be experienced in the operational organization. This is a significant and basic step, which must guarantee that the detection algorithm and learning approach are really usable, all things considered, and not just in a specific situation. In the second research stage this dataset is required for enhancement of the object detection algorithm and in the third stage for the extraction of target and foundation models.

2. Processing of images.

The image processing section will introduce some of the best approaches of data augmentation. After having simulated new threat objects in the existing dataset the newly images will undergo an image data augmentation process which will

include processes like: *horizontal/vertical shifting*, *horizontal/vertical flipping*, *random rotation* and *random zoom augmentation*.

3. Transfer learning object detection models

The third stage is devoted to the application of the selected object detection models. As a beginning stage a lot of highlights that demonstrated promising outcomes in existing writing will be utilized to prepare different classifiers. Each of the selected object detection models will require as a first step the feature extraction from transfer learning. So this is a two-phase detection plot where different feature extractors can be applied to each algorithm. One of the purposes of this step is to determine which of the feature extractors is better to be applied to every selected object detection model. Finally, the composition between feature extractors and object detection algorithms will be passed to execution and show their performances.

4. Performance analysis.

The accentuation of this work will lie in the last stage, in which the presentation of the order step will be broken down by researching the impact of different metrics as for both the extraction of the classifier just as the article recognition framework in general. Essential objective of this stage is to accumulate information about the presentation of the new two-phase recognition process and to recognize basic factors of the characterization framework. Every change will be contrasted with deference with the standard acquired in sync and the presentation of the underlying two-phase recognition plot acquired in stage three.

Also, based on the research work, the research questions will be answered at the conclusion of the thesis:

Research questions:

1. Which of the selected techniques will perform better in terms of accuracy?
2. Which of the selected techniques will perform better in terms of lowering the False/Positive results?

3. Which of the selected techniques will perform better in terms of lowering the False/Negative results?

1.3 Thesis structure

In Chapter 2, will be introduced the existing techniques for detecting the threat objects phenomenon. Based on this literature on techniques it will be decided which of them will be continued to experiment with. Also, a good representation of each literature's limitations will be presented.

In Chapter 3, a presentation of the selected dataset will be made. The choice of dataset helps prevent some limitations in calculating the accuracy of each of the algorithms. The chapter will be divided in two main parts that consist of: *image processing approaches* and o*bject detection models*. The image processing section will introduce some of the best approaches of data augmentation and data acquisition. On the other hand, the second part will take a good look into the details of mainly transfer learning which consists of selecting the object detection models described in details and also the feature extraction pre-defined models. In summary, will be provided the newly formed dataset with all mentioned classes and transfer learning concept description.

In Chapter 4, an experiment will be performed on each of the selected algorithms on the new dataset. Initially the presentation will be done with the ways of measuring accuracy. The methods chosen to measure the accuracy are: Accuracy average, F1-score and Confusion Matrix. Then the application of the methodology and the obtaining of the results for each of the algorithms will be done step by step. In conclusion, there will be a detailed discussion of which of the algorithms has achieved the highest accuracy and also the reason for which that algorithm outperformed others.

In Chapter 5, the conclusions of the work in this thesis will be given. The research question will also be answered. In conclusion, suggestions will be given for the continuation of this work in the future.

CHAPTER 2

LITERATURE REVIEW

2.1 Object detection

In the course of recent years, automated inspection frameworks have been created to direct X-ray images adequately and proficiently and to perform troublesome, repetitive and now and again hazardous assignments. Object detection utilizing X-ray screening is a need task that lessens the danger of wrongdoing and fear based oppressor assaults. Since 9/11 [1], aeronautics security screening with X-ray scanners has become a significant procedure at air terminals. However, inspection is a complex task because threat items are very difficult to detect when placed in closely packed bags, occluded by other objects, or rotated, thus presenting an unrecognizable view.

In addition, another negative aspect is that X-Ray images are represented as shadow images that correspond to projections of objects [1]. There is a lot of research that uses many different methods to get better views from the X-Ray images. N. Megherbi, T. P. Breckon, and G. T. Flitton [2] in their paper suggested using image segmentation before going into object detection. Another approach was proposed by B. Abidi, Y. Zheng, A. Gribok, and M. Abidi [3] in their work introduced the pseudo-color algorithms into weapon detection. On the other hand, in [1] T. Franzel, U. Schmidt, and S. Roth use automated detection of threat objects with single or multiple views on a single or dual-energy X-ray. In their paper, by experimenting, they achieved to reduce distortions in X-ray images. They also introduced an efficient non-maximum suppression scheme in the context of in-plane rotations. To conclude, most importantly, their work introduced a novel multi-view integration approach

to deal with out-of-plane object rotations. Their integration will be key to some other best known approaches. One of them introduced in [4] by M. Bas ̧tan, W. Byeon, and T. Breuel, use Support vector machine classifiers (SVM) with dual-energy multiple views with X-ray images which have initially been introduced by [1]. They achieved to raise the accuracy by 5-7% from the initial method.

Research on threat detection in luggage security can be grouped based on three imaging modalities: single-view X-ray scans [5], multi-view x-ray scans [2] [3], and computed tomography (CT) [4]. Classification performance usually shows improvements with the number of utilized views, with detection performance ranging from 89% true positive rate (TPR) with 18% false positive rate (FPR) for single view imaging [5] to 97.2% TPR and 1.5% FPR in full CT imagery [4]. The general consensus in the baggage research community is that the classification of x-ray images is more challenging than the visible spectrum data, and that direct application of methods used frequently on natural images (such as SIFT, RIFT, HoG, etc.) does not always perform well when applied to x-ray scans [1]. However, identification performance can be improved by exploiting the characteristics of x-ray images by: augmenting multiple views; using a colored material image or employing simple (gradient) density histogram descriptors [1] [5] [2]. Also, the authors of [1] discuss some of the potential difficulties when learning features using deep learning techniques, on varying size images with out-of-plane rotations.

On the other hand, after researching the visualization of X-Ray images, the next step is to see some effective classifying threat objects approaches. Classification of X-ray images could be a difficult task in computer vision. The explanation is that pictures from X-ray are normally occluded by different objects and can't be recognized simply once revolved [6]. To resolve this drawback, researchers planned to change the detection of threat objects. In [7], D. Turcsany, A. meat and T. P. Breckon, bestowed a picture classification technique for object detection in X-ray pictures exploitation set visual words in AN SVM classifier framework. The planned technique was evaluated in gun recognition and mobile detection. At constant time, in [8] evaluated trendy computer vision techniques for luggage review, bags of words (BoW's), thin representations, codebooks and deep networks. This is often the primary experiment in baggage review that uses deep learning. Their work shows that the best recognition

was achieved by ways supported visual vocabulary and deep networks with 95% accuracy.

This methodology was proved to be very effective, since over the time it was the main ideology used in many other researches. The main one, was the research concluded by Reagan L. Galvez, Elmer P. Dadios, Argel A. Bandala, and Ryan Rhay P. Vicerra in [9] where they experimented of four different threat classes such as blade, gun, knife and shuriken. After using the ideology of BoW's and codebooks, introduced in [8], they applied the new concept of transfer learning into their data. They would define transfer learning or information transfer as a way of re-using a model trained from an outsized dataset to unravel another drawback or task. Their experiment results showed that by victimization the idea of transfer learning with information augmentation and fine-tuning, threat objects may be classified at 99.5% accuracy. This can be considered the highest accuracy over the most research.

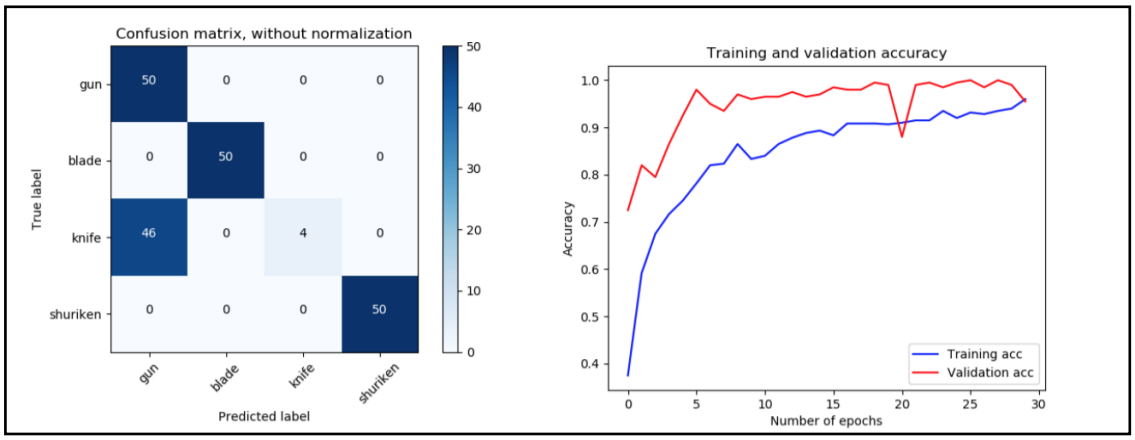


Figure 2. *The metrics of four classes [9].*

The dataset, shown below, used in these two researches was from the same source [10], a new dataset consisting of 19,407 X-ray images. These images were shown so that they could be used as a benchmark in order to test and compare the performance of different approaches on the same data, or also, the database can be used in the training programs of human inspectors.

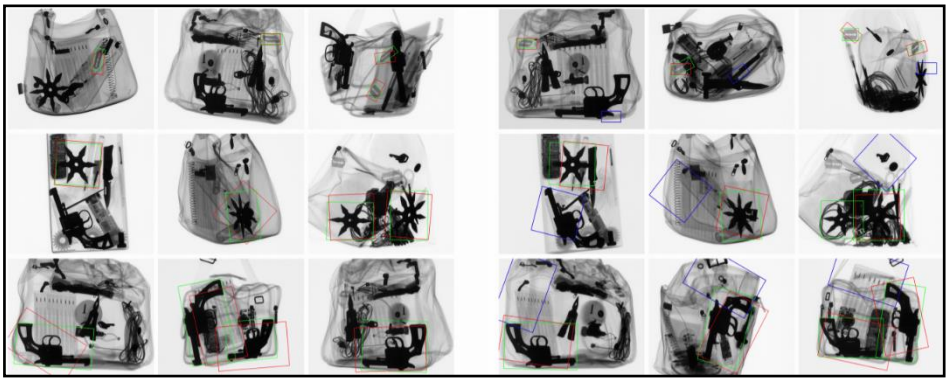


Figure 3. *GDXray luggage dataset [10]*

Transfer learning concepts became a very useful methodology. This technique is useful once the accessible information is tiny and saves time labeling massive amounts of data. In [11] which was one of the earliest research based in transfer learning, S. J. Pan and Q. Yang, said that transfer learning allows the domains, tasks, and distributions used in training and testing to be different. The idea that they came through was that low level features like edges, colors, textures, shapes and lighting can be shared among different tasks. Training of data from scratch can be avoided by adopting a pre-trained network instead of random initialization of parameters and use it as a starting point of the training. In the work of [11] in medical image analysis, showed that sufficient fine-tuning of pre-trained deep convolutional neural networks eliminates the need to train from scratch. In addition, transfer learning finds a big application even in other areas including medical, manufacturing and baggage screening.

Regardless of the drawback of transfer learning which consists of manually labeling the objects, this step is the main reason for the outstanding performance of machine learning (ML) algorithms. Once it is included in the transfer learning concept most of the algorithms perform better. The impact of transfer learning in Convolutional Neural Network (CNNs). In [12], Morris, T. Chien and E. Goodman have applied Convolution Neural Networks (CNNs) to the task of automatic threat detection, specifically conventional explosives, in security X-ray scans of passenger baggage. They obtained an AUC (Area Under The Curve) of the ROC (Receiver Operating Characteristics) of 0.95 for the best model, which means that their model achieved to divide between classes 95% of the time. The models they selected to

evaluate were three different prebuilt CNNs: VGG19 [13], Xception [14], and InceptionV3 [15]. They use the implementation of these models provided by Keras.

They discuss that each model has a 'top': the last stages of the network that prepares the data coming from the earlier convolutional layers for the final classification. They didn't include the top defined by Keras, but instead defined their own. For Xception and InceptionV3, they define the top as a 2D global average pooling layer, followed by a fully-connected layer with 1024 nodes, a fully-connected layer with 2 nodes (threat and non- threat), and then the soft-max function. They also introduced a dataset, the Passenger Baggage Object Database (PBOD). The closest to their work, related work, is Akc ̧ay et al. [16], which explores CNNs for detection of weapons such as guns.

In conclusion, including preprocessing and transfer learning into R-CNN could lead to better results. This is why the main objective of the proposed thesis is based on experimenting on R-CNN object-detection architecture.

2.2 Dataset Description

Based on the literature review, it is concluded that most of the recent approaches are mainly tested in GDB X-ray dataset of images [10]. On that dataset as shown in literature [9], the highest possible accuracy was achieved by 99.8%. Based on the future work of that paper [9], it is suggested for another dataset to try their recommended approach. Taking into consideration their suggestion and due to the lack of other data, in the dataset for this thesis will be included also a simulated data including other classes of threat objects. GDB X-Ray dataset, as mentioned, has overall distribution as shown in Table 1. The stimulated data will belong to the device explosives components, more specifically IEDs. The classes of stimulated images will be formed by: battery, wires and mortar.

Table 1. *GDXray luggage dataset object distribution [10]*

Threat object	***Count of objects in all the images***
Handgun	250
Knife	250
Razor-Blade	250
Shuriken	250

CHAPTER 3

METHODOLOGY

In this chapter will be introduced the methodological approach chosen to be applied in the thesis. The proposed thesis will represent a feasibility study based on the research question. It will be determined if it is possible to achieve higher results. In this sense, the methodology that will be used will be an experimental one, since there will be different approaches applied to achieve a higher accuracy. The thread objects that will be taken into consideration are: handguns, knives and blades. The methodology that will be taken into consideration will be divided into two main steps: *processing of images* and *object detection*.

In conclusion, this methodology will be analyzed both qualitative and quantitative. In the qualitative part, there will be a wide discussion about the missed cases of the algorithm. In the quantitative part, there will be many types of metrics taken into consideration and discussion in order to measure the results more efficiently

3.1 X-ray images processing

The main purpose of the processing phase is to be able to deliver into training an image which would assure the detection algorithm the best accuracy possible. So, since the main focus is being able to detect multiple objects inside an image and then classifying these objects in their classes, it can be more manageable if the images are well processed. When talking about processing of the images there are many things that can be done including rotations, translations, shifting, flipping the images and many more. But before going to the process that includes all the above mentioned terms, which is also known as data augmentation, due to the lack of dataset that include all the threat object classes that are taken into consideration there is a need to initially simulate some of the images.

So, in terms of steps the image below shows the necessary steps that will be included in the X-ray image processing phase.

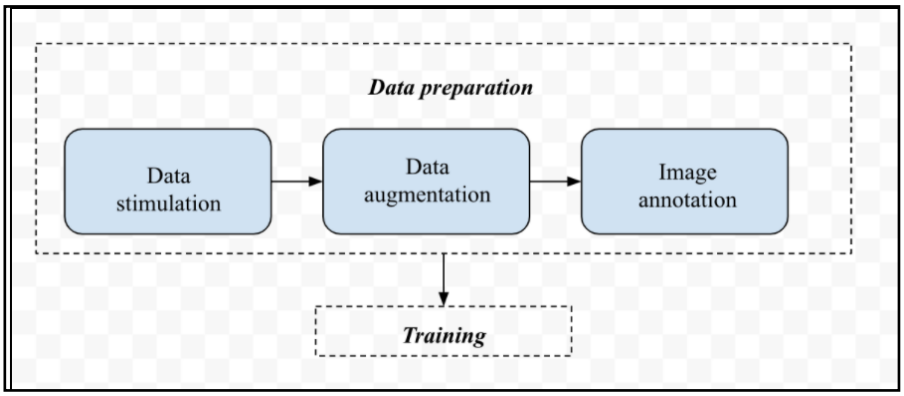


Figure 4. *Image processing flow to be applied in the thesis*

3.1.1 X-ray images simulation

Data simulation as a concept consists of generating random data compatible with the problem that is being solved. In this thesis, data simulation consists of adding additional threat objects to the images. The selected classes to be added in simulation are: battery, wire and mortar as shown in figure below.

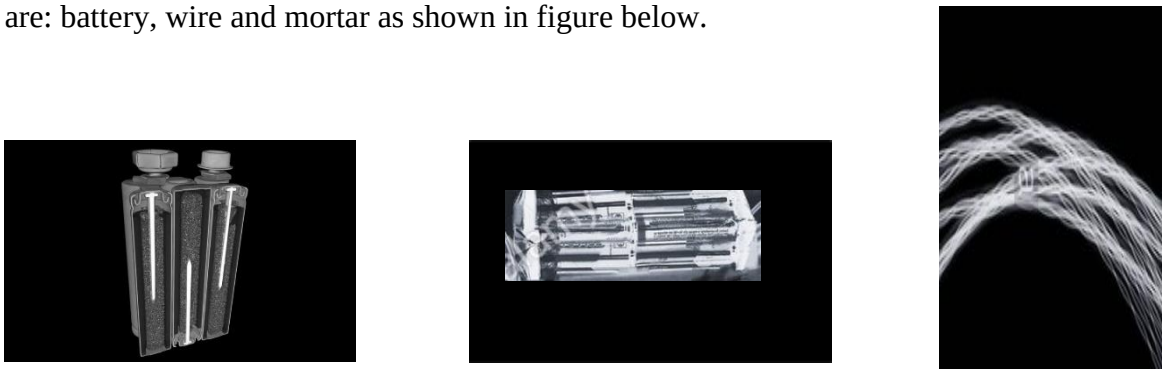


Figure 5. *Threat objects to be inserted in data simulation [17]*

In order for the data simulation on the images not to have interference with the object detection model accuracy, this thesis will be basing the simulation on [18]. In this paper, object stimulation inside the image can be achieved quite easily and without affecting the performance of any of the applied algorithms afterwards. In this logarithmic approach it is initiated the work based on the absorption law which

characterizes the intensity distribution of the image [18]. This law is presented also in equation 1.

$$\varphi(d)=\varphi_0 e^{-[?][?]d} \tag{1}$$

$$\varphi=\varphi_0 \exp\left(-\sum_{i=1}^{n}\mu_i d_i\right) \tag{2}$$

Where:

- μ absorption coefficient,
- d thickness of the irradiated matter,
- φ_0 incident energy flux density, and
- φ energy flux density after passage through matter with the thickness of d.

In an X-ray digital image, the grey value of a pixel can be linearly modeled as [**Error! Bookmark not defined.**]:

$$I = A \cdot \varphi + B. \tag{3}$$

Where:

- A and B are constant parameters of the model.

Following the models (2) for the energy flux density and (3) for the digital image, it is possible to model the X-ray image of the foreground (I_f), e.g. a battery, and the background (I_b), e.g. a cluttered bag, as illustrated in Figures 5. Thus,

$$I_f = A^*\varphi_f + B \qquad I_b = A^*\varphi_b + B \tag{4}$$

Where

$$\varphi_f = \varphi_0 e^{-\mu_f d_f} \qquad \varphi_b = \varphi_0 e^{-\mu_b d_b} \tag{5}$$

In this case μf and μb are the absorption coefficients of the foreground and background respectively. It is worth mentioning, that $\mu_b d_b$ represents $\Sigma_j \ \mu_j d_j$ considering all cluttered objects j that lie on the X-ray. The total X-ray image, called It, can be modeled as:

$$\varphi_f = \varphi_0 e^{-\mu_f df} e^{-\mu_b db} \tag{6}$$

$$I_t = A * \varphi_t + B = C e^{-\mu_f df} e^{-\mu_b db} + B \tag{7}$$

where C = A·φ0. It is clear, that (7) can be used to simulate new X-ray images from If and Ib: replacing (5) in (4), we obtain

$$e^{-\mu_f df} = \frac{I_f - B}{C} \qquad e^{-\mu_b db} = \frac{I_b - B}{C} \tag{8}$$

From (8) and (7), it yields

$$\frac{I_t - B}{C} = \frac{I_f - B}{C} * \frac{I_b - B}{C} \tag{9}$$

It can be normalized the X-ray images by subtracting B and dividing by C, e.g., J_t = (I_t − B)/C. Thus, using the normalized images for total, foreground and background images, it can be obtained:

$$J_t = J_f * J_b. \tag{10}$$

So, the total image can be computed by:

$$I_t = \text{C} \cdot J_f \cdot J_t + \text{B}. \tag{11}$$

Indeed, image It in Fig. 6c was simulated from *If* and *Ib* in Fig. 6a and 6b respectively using (11).

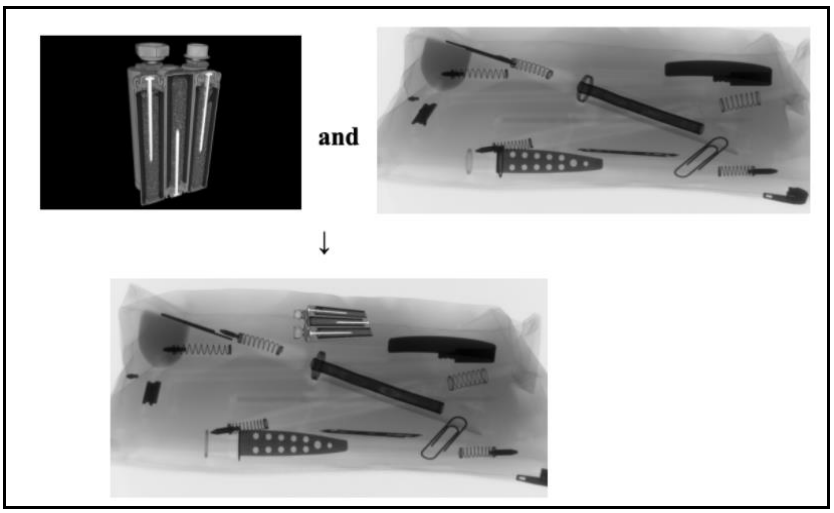


Figure 6. *a) The threat object, b) Background, c) Bag with the object*

So, after applying the data stimulation now the new dataset consists of a total of 1450 threat objects in the images. Previously each of the four classes had respectively 250 threat objects in the images, which makes a total of 1000 threat objects. Now, the new dataset with 150 stimulated objects for each additional class that will be added, makes a total of 450 additional objects.

The distribution of the newly added objects is also shown in table 2 below:

Table 2*. The distribution of the threat objects in the simulated dataset*

Threat object	***Count of objects in all the images***
Handgun	250
Knife	250
Razor-Blade	250
Shuriken	250
Battery	150
Wires	150
Mortar	150

3.1.2 X-ray images data augmentation

The main purpose of the processing phase is to be able to deliver into training an image in which a target object may be discovered from a ‘good pose’ that assures its detection. Given the securing of an essential picture, the accompanying three circumstances can happen:

- Detection from a ‘*good pose*’: On the off chance that the underlying posture compares to a decent perspective and the target item is recognized, it won't be important to move (pivot) the item into another position i.e., in this case, the image is put through only in appearance change stages and nor rotation neither translation is needed to be applied.
- Detection from a ‘*bad pose*’: On the off chance that the underlying target object is detected in a ‘bad pose’, then data augmentation will be applied by placing translated images.
- *No object detection*: On the off chance that no underlying target object is identified during the detection stage, then the testing object is arbitrarily rotated to a new position, that is different from the first, and repeating the detection stage.

Despite these circumstances, data augmentation will be applied in every image. The difference is what type of augmentation will be applied in each case. However, when talking about data augmentation, it can be said that it is a method to falsely make new training information from existing data. This is accomplished by applying domain-specific strategies to models from the training information that make new and diverse training models. Image data augmentation is maybe the most notable approach used for information increase and includes making changed adaptations of pictures in the training dataset that have a place with a similar class as the first picture. These adaptations include a range of operations of image manipulation, such as shifts, flips, zooms, brightness adjustment, rotations, translations and more. The purpose is to expand the training dataset with new, possible examples and also to fix the existing images to make the threat object inside much more noticeable. This means, variations of the training set images that are likely to be seen by the model. For example, if the IED device is rotated and translated in many forms it would still be an IED device that needs to be detected.

All things considered, obviously the decision of the particular data augmentation methods utilized for a training dataset must be picked cautiously and inside the setting of the preparation dataset and information on the difficult space. Furthermore, it may very well be helpful to try different things with data expansion techniques in isolation and in order to check whether they bring about a quantifiable improvement to demonstrate detection, maybe with a little prototype dataset, model, and training run. Current deep learning algorithms, for example, the convolutional neural network, or CNN, can learn features that are invariant to their area in the picture. All things considered, augmentation can additionally help in this change invariant way to deal with learning and can help the model in learning features that are likewise invariant to changes, for example, left-to-right to top-to-bottom ordering, light levels in photos, and that's only the tip of the iceberg. Differently from the data preparation that consists of resizing and pixel scaling and is also applied to all the dataset, data augmentation can only be applied to the training dataset and not the validation/testing one.

The data augmentation used when creating the new dataset with the newly added classes consists of: horizontal/vertical shifting, horizontal/vertical flipping, random rotation and random zoom augmentation. The figure (6) shows this data augmentation being applied to the battery image chosen to be implemented as an additional class in the training dataset.

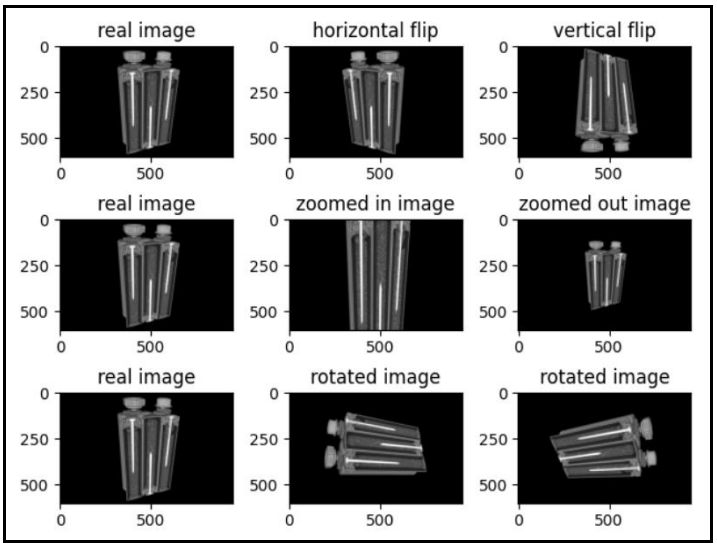


Figure 7. *Data augmentation being applied to the battery x-ray image.*

3.1.3 X-ray image annotation

The last section of the image processing phase is based on the labelling of the images. The labelling is considered as the returning point from the image dataset into a features dataset where from every image is extracted the following information in a .XML format containing object tags just like in the figure below. Inside the <object> tag there are two main features extracted: name and bndbox. The name consists of the label that it was given to the specific object. The `bndbox` consists of four main position pixels from where the object was labeled. These pixels represent the starting point of the drawing on the top-left and the ending point on the right-bottom. Having these points can easily be drawn in a rectangle shape that covers the whole image.

```
<object>
<name>razor blade</name>
<bndbox>
<xmin>439</xmin>
<ymin>283</ymin>
<xmax>524</xmax>
<ymax>366</ymax>
</bndbox>
</object>
```

Figure 8. *Data augmentation being applied to the battery x-ray image.*

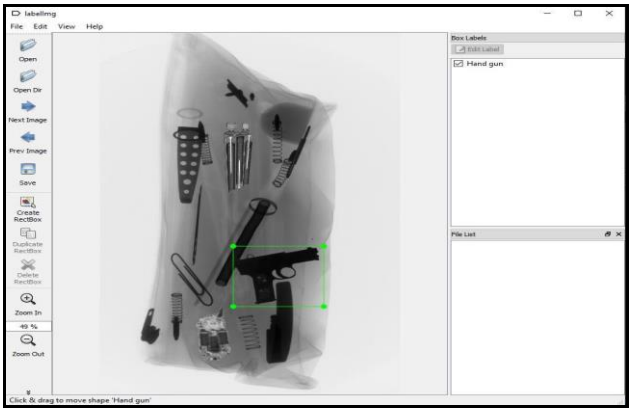


Figure 9. *Data labeling using the Label Img software.*

3.2 Transfer Learning

Transfer learning is a machine learning technique where a trained model on one task is reused on a second training model. "*Transfer learning and domain adaptation refer to the situation where what has been learned in one setting … is exploited to improve generalization in another setting*"— stated in page 526, Deep Learning, 2016. Transfer learning only works in deep learning if the model features learned from the first task are general. In this way training the data from scratch can be avoided by using these predefined training models that are also trained. In other words, instead of using random initialization of features it can be used as an initialized model whose parameters are already approximated by previously trained data. In the figure below it is shown the simple transfer learning pipeline. On the left of the image is a predefined model taken from the previous model of MS COCO (MicroSoft Common Objects in Context) [19]. In the left pipeline it is shown that MS COCOs convolutional features are extracted and are 'transferred' in the target domain. On the other hand, on the right is shown that the targets domain features are frozen due to their possibility to affect the weights of the convolutional features that are being transferred from the source domain.

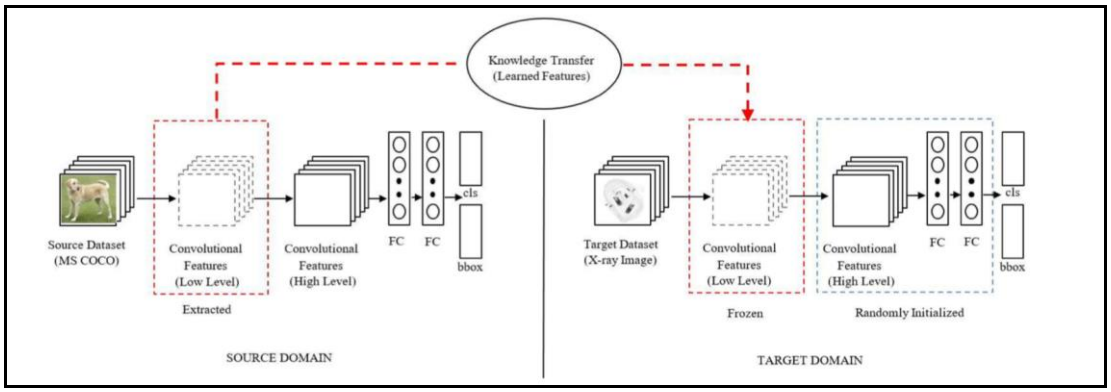


Figure 10. *The transfer learning concept that will be applied [12].*

The most important decision when using transfer learning is mostly based on the selection of *object detection models* and *feature extraction strategies*. An object detection model is trained to detect the presence and placement of multiple categories of objects. The models that will be taken into consideration in this thesis are: *Faster-RCNN*, *Single Shot Detector* (SSD) and *Region-based Fully Convolutional Networks*

(R-FCN). Faster R-CNN, Single Shot Detectors, and Regional Fully Convolutional Network can be regarded as the three meta-architectures of CNN-based detectors.

3.2.1 Faster R-CNN

Neural networks are considered as the building blocks of deep learning algorithms. To understand better how Faster R-CNN object detection algorithms work, it is needed to review the history of the R-CNN algorithm and how it is differently from Faster R-CNN. This way will be understated why it is even possible to to be faster than R-CNN.

R-CNN was presented in 2014 and increased a ton of enthusiasm for the computer vision network. The possibility of R-CNN was to utilize a Selective Search (SS)approach to propose around 2000 Regions-Of-Interest (ROI), which were then taken into a Convolutional Neural Network (CNN) to extract features. These features were utilized to group the pictures and their item limits utilizing SVM (Support Vector Machines) and regression methods. This was immediately trailed by Fast R-CNN, a quicker and better methodology of object detection, in the middle of 2015[1]. Fast R-CNN utilized a ROI pooling approach, which shares the features over the entire image and uses an altered type of spatial pyramid pooling technique to separate features in a computationally effective manner. The issue with Fast R-CNN is that it is still moderate since it needs to perform SS which is computationally moderate. Although Fast R-CNN accepts 0.32 seconds rather than 47 seconds at test time to do a detection, it takes 2 seconds for producing 2000 ROI's. This indicates 2.3 seconds for each picture. This latency drove specialists to create Faster R-CNN where the test time per picture is just 0.2 seconds with district proposition. This is because of the way that this most recent methodology is a completely differentiable model utilizing start to finish preparing.

Based on their work, Girshick et al. [20], created an extra part to the R-CNN design, a Regional Proposal Network. The Region Proposal Network (RPN), as its name suggests, wont to generate object proposals and also the second is employed to predict the particular category of the article. So the primary differentiator for Faster R-CNN is the RPN which is inserted after the last convolutional layer. This is trained to produce region proposals directly without the need for any external mechanism like

Selective Search. After this Faster R-CNN uses ROI pooling and an upstream classifier and bounding box regressor similar to Fast R-CNN.

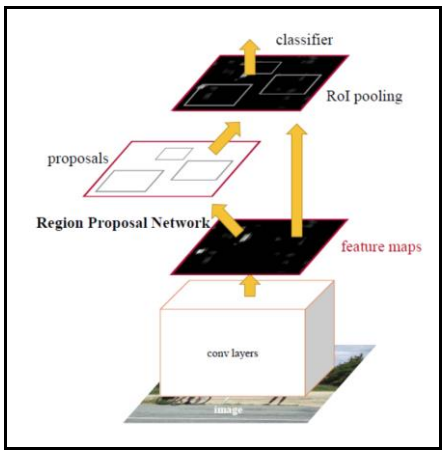


Figure 11. *The Faster R-CNN architecture. [20]*

In the figure (11) it can be seen the general steps in the Faster-RCNN architecture. After the image is inserted to training, the first component after that is the case of CNN network. This CNN network in this thesis is defined as the pre-trained model which is used for transfer learning and is used as a feature extractor. The reason why Faster R-CNN uses transfer learning is because features that are learned by a particular layer are generally transferable to other tasks outside the pre-defined network. This makes it more reliable. After features are extracted they are passed to the next phase of RPN. In Region Proposal Network (RPN) a set of anchors are accepted and after that it outputs the suggested proposals for where the image would be. RPN does not specify where the object really is, but just proposes main areas where it could be. The next step consists of RoIP(Region of Interest Pooling), that achieves to extract feature maps from each proposed region that has come as an input. Finally, a R-CNN is used to make the final label predictions and further define the locations for better accuracy.

3.2.2 Single Shot Detector

Single Shot Multibox Detector or SSD, was released in 2015 and this model is a solitary stage object location model that goes directly from picture pixels to bouncing box arrangements and class probabilities. The model engineering depends on rearranged leftover structure where the info and yield of the remaining square are

slight bottleneck layers rather than conventional lingering models. Besides, nonlinearities are expelled from the middle of the road layers and lightweight depthwise convolution is utilized. The SSD is a single deep neural network object detector that uses multiple-scale feature maps and default boxes for detection. It eliminates bounding box proposals and feature resampling stage, as a result, increases the speed of detection compared with Faster R-CNN and YOLO [21]. The SSD approach is predicated on a feed-forward convolutional network that produces a fixed-size assortment of bounding boxes and scores for the presence of object category instances in those boxes, followed by a non-maximum suppression step to provide the ultimate detections.

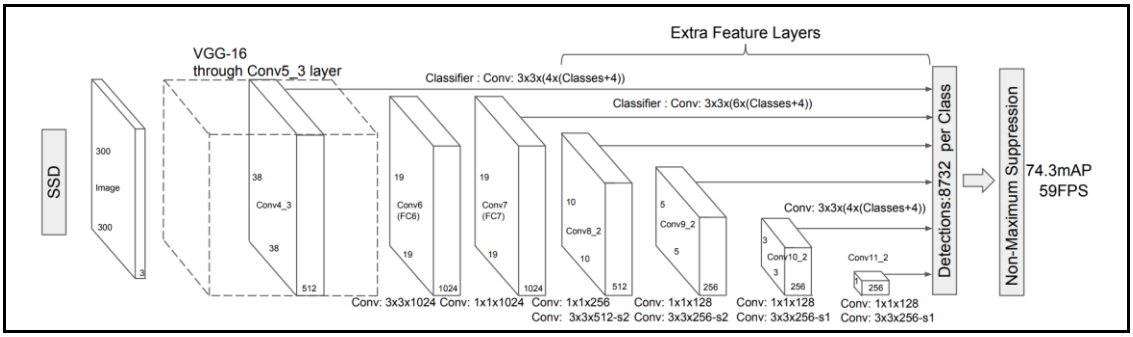


Figure 12. *The SSD model [22]*

SSD model adds a few component layers to the furthest limit of a base system, which foresee the counterbalances to default boxes of various scales and viewpoint proportions and their related confidences. The primary methodology of SSD is anticipating classification scores and box counterbalances for a fixed arrangement of default jumping boxes utilizing little convolutional channels applied to highlight maps. To accomplish high discovery exactness SSD produces expectations of various scales from include maps of various scales, and expressly isolates forecasts by viewpoint proportion.

A typical CNN network gradually shrinks the feature map size in each step and increases the depth as it goes to the deeper layers. When talking about CNN it can also be taken into consideration the fact that it requires a pre-trained image classification network as a feature extractor. Instead of using a sliding window, SSD divides the image using a grid and have each grid cell be responsible for detecting objects in that region of the image. Detection objects merely suggest that predicting the category and

placement of associate degree objects among that region Differently from Faster R-CNN, SSD requires no proposal network. Instead, it provides default boxes which are a set of fixed object positions, scales or aspect ratios. Then after applying the CNN network of the predefined model, the SSD network generates scores for each default box.

3.2.3 Region-based Fully Convolutional Networks

Region based Fully Convolutional Networks or *R-FCN* is a R-CNN based network approach. Like the other R-CNN based detectors, even R-FCN processes the object detection in two different stages. The first stage consists of generating region proposals (RoI) areas and the second one that makes classification and localization predictions from ROIs. In chapter 3.2.1 when explaining about the Faster R-CNN it was mentioned that the initial composition came based on lack of speed of Fast R-CNN and its problem with computing about 2000 ROIs per image. Following the same logic, R-FCN improves speed but differently from Faster R-CNN here speed is improved by reducing the amount of work needed for each ROI. The region-based feature maps are independent of ROIs and can be computed outside each ROI. The remaining work is much simpler and therefore R-FCN is faster than Fast R-CNN or Faster R-CNN. In difference from the other R-CNN based detectors, that apply a costly per-region subnetwork hundreds of times, the region-based detector is fully convolutional with almost all computation shared on the entire image. When talking about fully convolutional let keep in mind the difference between a fully-connected layer. A fully-convolutional layer is much more specialized, and efficient, than a fully connected layer. In a fully connected layer each node is connected to every node in the previous layer, and each connection has its own weight. This is a totally general purpose connection pattern and makes no assumptions about the features in the data. It's also very expensive in terms of memory which can be counted as weights and computation which can be counted from the connections. In contrast, in a convolutional layer each neuron is only connected to a few nearby (aka local) nodes in the previous layer, and the same set of weights is used for every node. This connection pattern only makes sense for cases where the data can be interpreted as spatial with the features to be extracted being spatially local and equally likely to occur at any input

position. The typical use case for convolutional layers is for image data where, as required, the features are local.

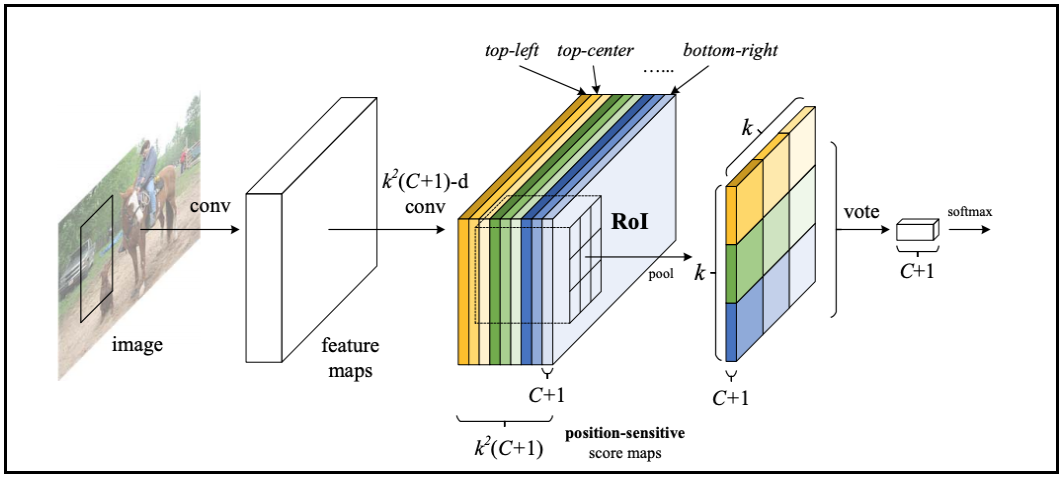


Figure 13. *The R-FCN model [23]*

The key idea of R-FCN for object detection can be shown in the figure above. In the illustration, there are k × k = 3 × 3 position-sensitive score maps generated by a fully convolutional network. For each of the k × k bins in an RoI, pooling is only performed on one of the k^2 maps (marked by different colors in figure) [23].

3.2.4 Detection Architectures

Faster R-CNN, Single Shooting Detectors and the R-FCN can be considered the three meta-architectures of CNN-based detectors. By changing their parameters, their performance can significantly be changed. In late 2016, a group of Google researchers published the paper with extensive comparison of these meta-architectures, and the impact of feature extractors on accuracy and speed. Meta-parameters include basic neural networks selected as feature extractors, the number of region proposals, the input resolution for the image, and the feature steps. There are several basic architectures like MobileNet, Resnet and Inception. In this thesis will be taken into consideration the following feature extractors:

1. *Inception-v2* [24]

   Inception v2 and Inception v3 were given within the [24] paper. During this paper, the authors projected a variety of upgrades that inflated the accuracy and

reduced the procedure quality. origination v2 reduces the realistic bottleneck. The intuition was that neural networks perform higher once convolutions didn't alter the size of the input drastically. Reducing the size an excessive amount might cause loss of knowledge, called a "representational bottleneck". victimisation sensible resolving ways, convolutions may be created additional economical in terms of procedure quality. Inception-v2 factorizes a 5x5 convolution to 2 3x3 convolution operations to boost procedure speed. though this might appear unreasonable, a 5x5 convolution is a pair of.78 times costlier than a 3x3 convolution. therefore stacking 2 3x3 convolutions after all results in a lift in performance. this can be illustrated within the below image. The reason why this version of the Inception-based architecture is chosen, is mainly due to the fact of batch normalization and reduction of overfitting. The term batch normalization refers to the fact of normalized inputs to every layer before sending the data to the activation function. Here, batch normalization is achieved factoring the 5x5 convolution into a 3x3. Batch normalization helps a lot in which is equal to lower training speed by allowing higher learning rates. On the other hand, it also has an effect in lowering the overfitting levels. This brings the main reason why version two is selected. The other versions of Inception architecture become more and more reductive in each layer by making the convolution even 1xn then nx1. The 1 factor there would result in a lot of overfitting of the data. This is why the 3x3 architecture seems the best choice between the other versions.

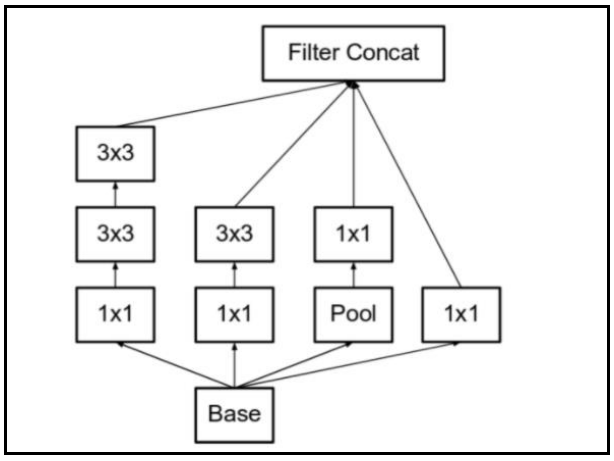


Figure 14. *The Inception-v2 architecture*

2. *ResNet-101* [25]

ResNet is an architectural design that is mainly used due to its deeper training layers and the good performance. This architecture performs well in image recognition and achieves to train deeper. Most other architectures can not go in deeper layers due to the loss of accuracy and increase of the computational cost. RestNet achieves to go even to the deepest layers without having any decrease in performance. This is mainly due to the fact that it works based on adding skip connections and then forming a residual block. So, the problem can be stated as follows: when deeper networks start converging, a degradation problem starts exposing which means that the network depth starts increasing, accuracy starts soaking and then decreases rapidly. In the worst case, deeper model's early layers can be replaced with a shallow network and the remaining layers can just act as an identity function. In the best case, the additional layers better approximate the mapping than it's shallower counterpart and reduce the error by a significant margin. In the worst case, both the network and deeper variants of it should give the same accuracy. In the best scenario case, the deeper model should give better accuracy than any other counterpart.

3. *MobileNet-v2* [26]

MobileNets are little, low-latency, low-power models parameterised to satisfy the resource constraints of a range of use cases. In line with the analysis paper, MobileNetV2 improves the progressive performance of mobile models on multiple tasks and benchmarks furthermore as across a spectrum of various model sizes. It's an awfully effective feature extractor for object detection and segmentation. maybe, for detection, once paired with Single Shot Detector, MobileNetV2 is regarding thirty five p.c quicker with an equivalent accuracy than MobileNetV1.

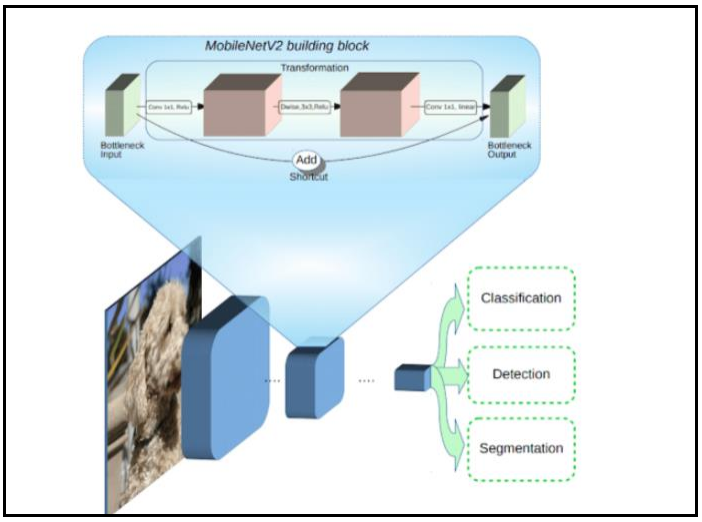


Figure 15. *The MobileNet-v2 architecture*

Based on the previous works and on how each object detection model and architectures work, in the next chapter will be experimented with the combination as shown in table below:

Table 3*. The selected Object detection models and respective architectures*

Nr	***Object Detection Model***	***Object Detection Architecture***
1.	SSD	Inception-V2
2.	SSD	MobileNet-V2
3.	R-FCN	ResNet-101
4.	Faster R-CNN	ResNet-101
5.	Faster R-CNN	Inception-V2

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter, an experiment will be performed on each of the selected algorithms on the new dataset. Initially the presentation will be done with the ways of measuring accuracy. The methods chosen to measure the accuracy are: Accuracy average, F1-score and Confusion Matrix. Then the application of the methodology and the obtaining of the results for each of the algorithms will be done step by step. In conclusion, there will be a detailed discussion of which of the algorithms has achieved the highest accuracy and also the reason for which that algorithm outperformed others.

4.1 Accuracy metrics

Choosing the right metric is essential when evaluating object detection models. Different metrics have been proposed to evaluate ML models in different applications, and it may be helpful to provide a summary of popular metrics, to better understand each metric and the applications for which they can be used. In some applications you see a single metric may not give the whole look of the problem being solved, and it may be desirable to use a submission of the metrics discussed in this thesis, to have a concrete assessment of the models

4.1.1 Cross validation

Cross-validation. Cross-validation is a statistical method used to assess the ability of machine learning models. It is commonly used in ML applied to compare and select a model for a particular modeling prediction problem because it is easy to understand, easy to implement, and results in assessments of skills that generally have a lower bias than other methods. Cross-validation is a resumption procedure used to evaluate ML models in a limited data sample.

The procedure has a single parameter called k which refers to the number of groups in which a given data sample should be divided. As such, the procedure is often called k-fold cross-assessment. When a specific value for k is chosen, it can be used instead of k in reference to the model, such as k = 10 by making a 10-fold cross-assessment. Cross-validation is mainly used in ML applied to assess the ability of a machine learning model in unprecedented data. This means using a limited sample in order to assess how the model is expected to perform in general when used to make predictions on data that are not used during model training. It is a popular method because it is easy to understand and because it generally results in a less one-sided or less optimistic assessment of model skills than other methods, such as a simple training / testing split.

The general procedure is as follows:

1. Shuffle the data at random.
2. Data are divided into groups k
3. For each unique group:
    a. The group is taken as a set of data without proof or proven
    b. The remaining groups are taken as a set of training data
    c. A model is placed in the training group and it is evaluated in the evidence group
    d. The evaluation result is kept and the model is thrown
4. Summarize the ability of the model using the example of model evaluation points

Most importantly, every observation within the knowledge sample is appointed to a private cluster and stays in this group for the length of the procedure. This suggests that every sample is given the chance to be utilized in the expected group 1 time and to be used to train the k-1 model.

It is also important that any data preparation prior to model assembly takes place in the CV-specific training data within the loop rather than in the broader data set. This also applies to any hyperparameter tuning. A failure to perform these operations within the loop may result in data leakage and an optimistic assessment of model capabilities. The results of a valid value of k-folds are often summarized with

the average of the model's skill points. It is also good practice to include a measure of skill point variance, such as standard deviation or standard error.

On the other hand, a very important issue in this case is the selection of the coefficient k. The k value must be carefully selected for your data example. A poorly chosen value for k can result in a misrepresentative idea of the model's capabilities, such as a result with a high variance (which can vary greatly based on the data used to fit the model). , or a high bias, (such as an overestimation of model capabilities) .Three common tactics for choosing a value for k are as follows:

- Representative: The value for k is chosen so that any training / test of data samples is large enough to be statistically representative of the wider data.
- k = 10: The value for k is fixed at 10, a value that has been found through experimentation to generally result in a model skill assessment with a low bias and a modest variance.
- k = n: The value for k is fixed at n, where n is the data size to give each sample a chance to be used in the database. This approach is called left-one-out cross-assessment.

"*The choice of k is usually 5 or 10, but there is no official rule. As k becomes larger, the difference in size between the training group and the remodeling subsections becomes smaller. As this change diminishes, the bias of the technique becomes smaller* "- Page 70, Applied Predictive Modeling Book, 2013.

The performance measure reported by the k-fold cross-assessment is then the mean of the values calculated in the loop. This approach can be expensive in a computational way, but it does not consume much data (as is the case when arranging an arbitrary validity group), which is a major advantage in problems such as reverse inference, where the number of samples is very small.

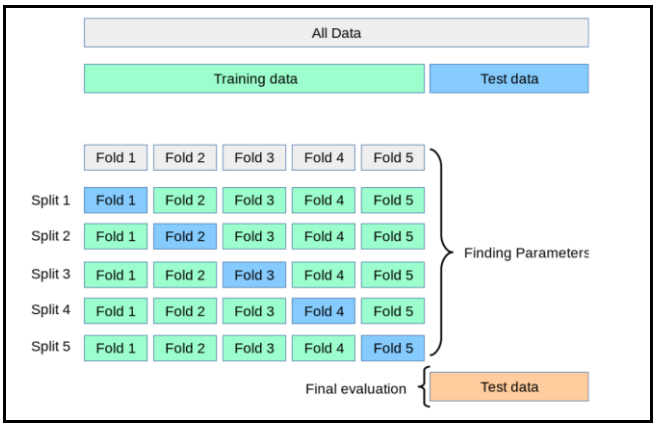


Figure 16. *Example how Cross Validation works for k=5*

4.1.2 Confusion Matrix

A confusion matrix could be a table typically wont to describe the performance of a classification model (or "classifier") in a very set of take a look at knowledge that true values are notable. The matrix of confusion itself is comparatively easy to know, however the connected nomenclature may be confusing.

Let's start with a confusion matrix for a binary classifier, as is the phishing problem. In phishing it is a binary problem as there are a total of 2 classification classes: legitimate or phishing. Let’s take some possible results to make a more detailed description of the confusion matrix, as in Figure 17.

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

Figure 17. *Example exercise for Confusion Matrix*

From the above matrix it can be learned that:

- There are two possible classes provided: "yes" and "no". If we were to predict the presence of an object, for example, "yes" would mean that object is found in the image, and "no" would mean that the object is not detected.

- The classifier made a total of 165 predictions (e.g., 165 images were being tested if they contain the object or not).
- Of those 165 cases, the classifier predicted "yes" 110 times, and "no" 55 times.
- In reality, 105 images in the sample had the specified object, and 60 images had not.

Let's now define the most basic terms, which are integers (not norms):

- *TruePositive*: These are cases in which the image contained that object and it was predicted that the image had the object.
- *True Negative*: Predicted not, and the images did not contain the object.
- *False Positive*: The images were predicted to have the objects in them, but in reality the object was not present. (Also known as an "I-type error")
- *False Negative:* The images were predicted to not have the objects in them, but in reality the object was present.. (Also known as a "type II error.")

These terms have been added to the matrix below:

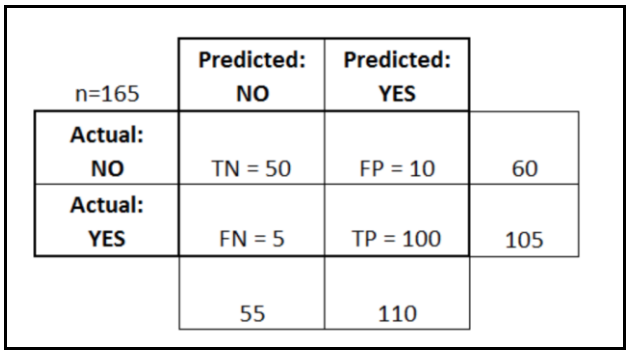


Figure 18. *Terms TP, FP, FN and TP in the Confusion Matrix*

This is a list of norms that are often calculated by a confusion matrix for a binary classifier:

- Accuracy: In general, how often is the classifier correct?

  Accuracy = (TP + TN) / total *(12)*

  ex. Accuracy = (100 + 50) / 165 = 0.91

- Poor-classification rate: In general, how often is it wrong?

Rate = (FP + FN) / total *(13)*

ex. Rate = (10 + 5) / 165 = 0.09

- True Positive Scale: When it's actually yes, how often does it predict yes?

  TPScale = TP / current yes *(14)*

  TP Scale = 100/105 = 0.95

- False Positive Scale: When it's actually not, how often do you predict yes?

  FP Scale = FP / current number *(15)*

  FPScale = 10/60 = 0.17

- True Negative Scale: When it's actually not, how often does it predict?

  TN Scale = Current TN / number *(16)*

  TN Scale= 50/60 = 0.83

- Precision: When predicting yes, how often is it correct?

  Precision = TP / predicted yes *(17)*

  Precision = 100/110 = 0.91

- Prevalence: How often does the condition occur in our example?

  Prevalence = current / total *(18)*

  Prevalence = 105/165 = 0.64

4.1.3 F1-Score

In applied mathematics analysis of binary classification, the f1-score (also the F-score or F-score) could be a live of the accuracy of a take a look at. Considers each the accuracy of p and therefore the recall of the take a look at to calculate the result: p is that the number of correct positive results divided by the quantity of all positive results came back by the classifier, and r is that the number of correct positive results divided by the quantity of all relevant samples (all samples that ought to are known as positive). First let's get acquainted with the terms precision and recall. In model recognition, information learning and machine learning, precision (also called positive predictive value) is the part of the corresponding instances between the instances obtained, while recall (also known as sensitivity) is the part of the total sum of instances relevant that were actually taken. Both precision and recall are based on an understanding and measure of importance. Respectively, precision and recall are calculated in the form shown in equations 19 and 20.

$$precision = \frac{TRUE\,POSITIVE}{TRUE\,POSITIVE + FALSE\,POSITIVE} \tag{19}$$

$$recall = \frac{TRUE\,POSITIVE}{TRUE\,POSITIVE + FALSE\,NEGATIVE} \tag{20}$$

Now can be considered the f1-score which is a precision and recall function:

$$f1 = 2[?][?][?]\frac{precision[?][?][?]recall}{precision + recall} \tag{21}$$

The F1 score is required after you need to appear for a balance between exactitude and Recall. All right ... what's the distinction between F1 Score and Accuracy then? it's been seen before that accuracy may be greatly contributed by an outsized variety of real negatives who in most business circumstances, we have a tendency to don't target abundant whereas False Positive and False Positive sometimes has business prices (tangible and intangible), therefore the F1 Score could also be the most effective live to use if we've got to appear for a balance between exactitude associated Recall and has an unequal category distribution (a sizable amount of current negatives)

4.2 Results

In order to fully complete the transfer learning approach, in this thesis is used the Tensorflow API. This already prepared library and API make it more simple to make testing with different types of architectures. The API has the following structure:

1. It requires manually loading and separating the labelled images into train and test files. As described in the third chapter the labelling of the images requires the transformation in a .xml file with the corresponding data.

2. It converts the .xml files into .csv files so that the image and object data are easily accessible by the Tensorflow library.
3. It downloads the predefined model files from the MS-COCO dataset. In order for the API to automatically download the base model it requires a JSON as an input that specifies the model as a key and also requires the batch number as a value as shown in the figure below:

```
'ssd_mobilenet_v2': {
'model_name': 'ssd_mobilenet_v2_coco_2018_03_29',
'pipeline_file': 'ssd_mobilenet_v2_coco.config',
'batch_size': 12
}
```

4. It configures the training pipeline based on the config file that is sent in the JSON object.
5. It sends the trained model and pipeline into training based on the number of steps and number of evaluation steps.
6. Once the training job is complete, you need to extract the newly trained inference graph, which will be later used to perform the object detection.

4.2.1 Simulation results

In order to make sure that the approach chosen to make the new stimulated images would not affect the accuracy of the detection models, it is followed the following procedure. Since the missing objects from the dataset were: battery, wire and mortar, the testing can not be done by using those objects. Instead, in the dataset will be stimulated the handgun as an object where it is missing and then will be trained. So, the same algorithm will be chosen to initially detect the old dataset and the new one together with the simulations and will come into a conclusion of how this stimulation affects the accuracy of the algorithm.
The stimulation is done as shown in the figure below:

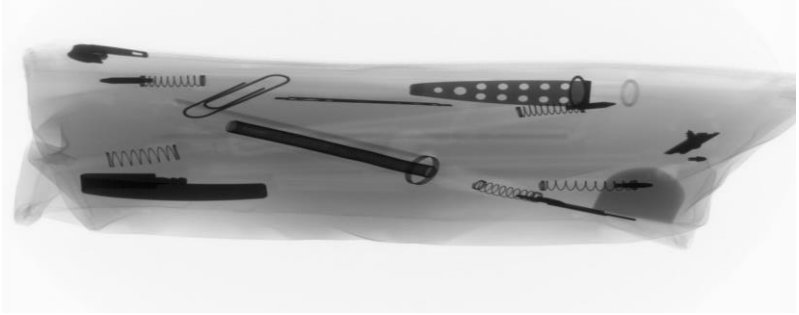


Figure 19. *Original Image*

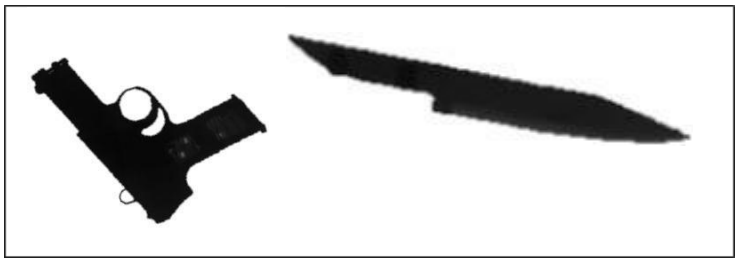


Figure 20. *Objects to be inserted into the image*

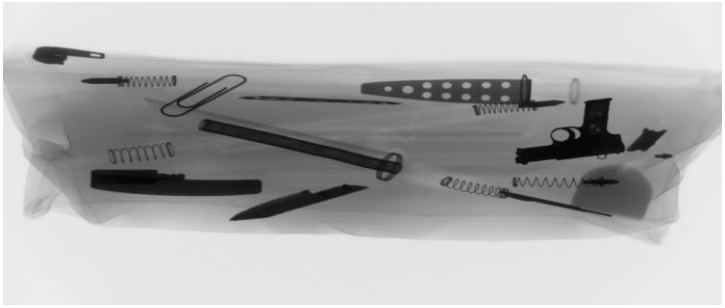


Figure 21. *Simulated image with added handgun and knife*

As mentioned in chapter 2, the dataset contains 250 images for each of the classes including: handgun, knife, razor-blade and shuriken. After making the insertions into the images of the threat objects of handgun and knife their count in total went upto: 320 handguns and 300 knives in total. The results after training in Faster R-CNN previously and after are shown in the figure below.

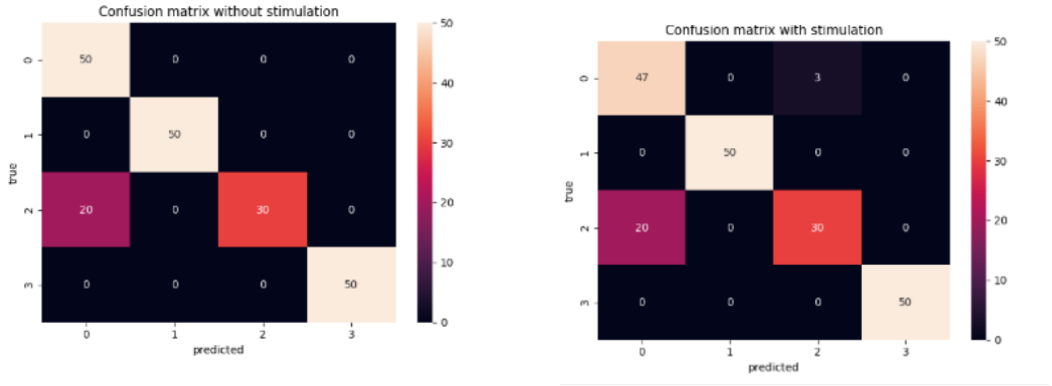


Figure 22. Confusion Matrix of Faster R-CNN with Inception-v2 applied to the old dataset(on the left) and to the simulated dataset(on the right), both containing 50 objects in testing

Based on the results from the results of the confusion matrix it can be seen that the Faster R-CNN algorithm applied to the data without stimulation consists of only 30 FN values where knives were predicted as handguns. (Keeping in mind that in the confusion matrix the classes are distributed as follows: 0: handgun, 1: razor-blades, 2: knife, 3: shuriken). Translating this result in terms of accuracy it would mean that the results were as following:

- Accuracy of data without stimulation: ***90%***
- Accuracy of data with stimulation: ***88.5%***
- This leads to an approximate absolute error of ***1.6%.***

4.2.3 Object detection results

The results of this experiment will be calculated based on different metrics, but the main one is considered to be the Accuracy. The experiments are done using the defined approaches in Table 2. Their results are measured in many different steps. The large amount of steps determine the possibilities to monitor the training until a certain point. The goal then is that between the considered amount of steps there is a maksimum found. In the graphs below it is calculated by using mean average precision (mAP). Average precision is the interpolation between precision (Eq. 17) and recall

(Eq.18). The mean average precision is the mean of all average precision over all the number of classes as C:

$$mAP = \frac{1}{C} \sum_{i=1}^{C} AP_i$$

(22)

In the below graphs there is taken the monitoring of the output for every step for each of the chosen algorithms, keeping in mind that each algorithm is passed in 2K or 2,000 steps in training.

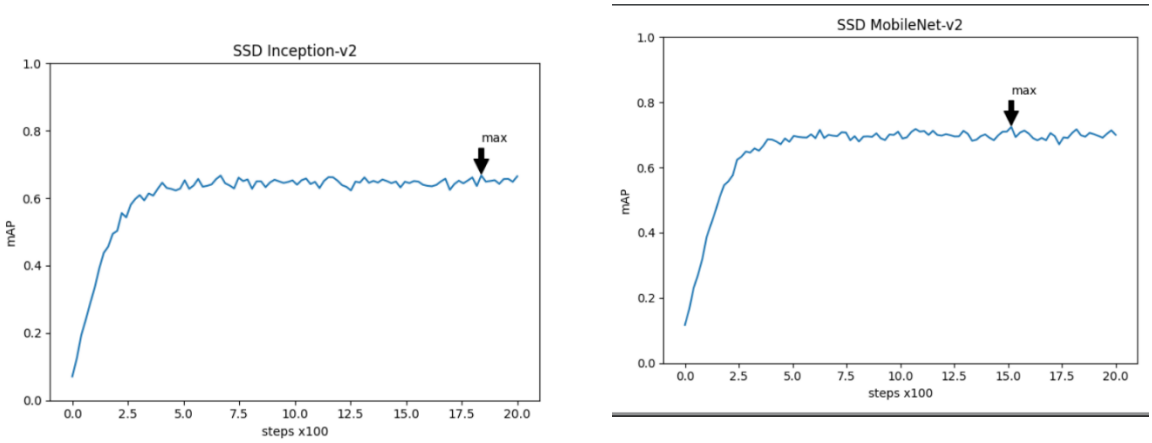


Figure 23. *mAP monitoring in SSD a) on the left Inception-v2 architecture b)MobileNet-v2*

Firstly, let's consider the SSD model with the respective architectures. As it can be shown in Figure 22, on the left there is shown the result for Inception-v2 architecture. In these results it achieved the highest possible mean average precision of 66.74% that is shown in the figure by the arrow. This value is not considered as a good achievement due to the lack of all the missing cases. On the other hand, MobileNet-v2 architecture seems a better fit for the SSD model in solving the proposed problem. This comes as an output to the fact that the maximum value achieved for 2000 steps in training for mean average precision is up to 72,24%.

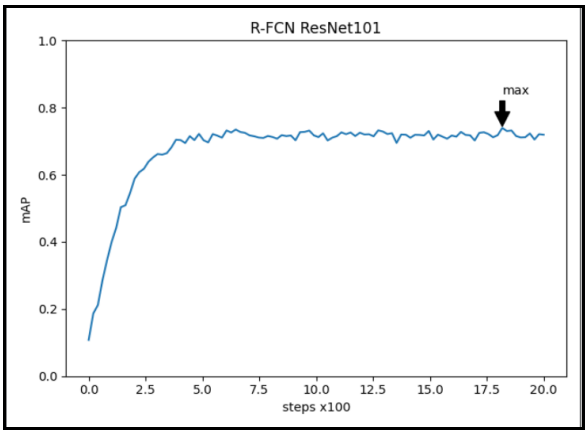


Figure 24. *mAP monitoring in R-FCN ResNet101*

Secondly, let's consider the R-FCN and ResNet101 architecture as shown in figure 23. As it can be seen this model achieves a higher mAP by 73,96%. Even though, is quite close and achieves to outperform the SSD-MobileNet101 outcome, it can be seen that R-FCN takes more number of steps to converge.

Finally, let's consider the results of Faster R-CNN and the selected architectures as it can be shown in figure 24. From the figure it can be seen that for the ResNet101 architecture, the algorithm does not take a long time to converge. Even exceeded the highest mAP achieved so far, by achieving a 78.43% maximum value of mean average precision. On the other hand, the Inception-v2 architecture achieved the maximum value of mAP at 1000 steps and of precision 87.58%.

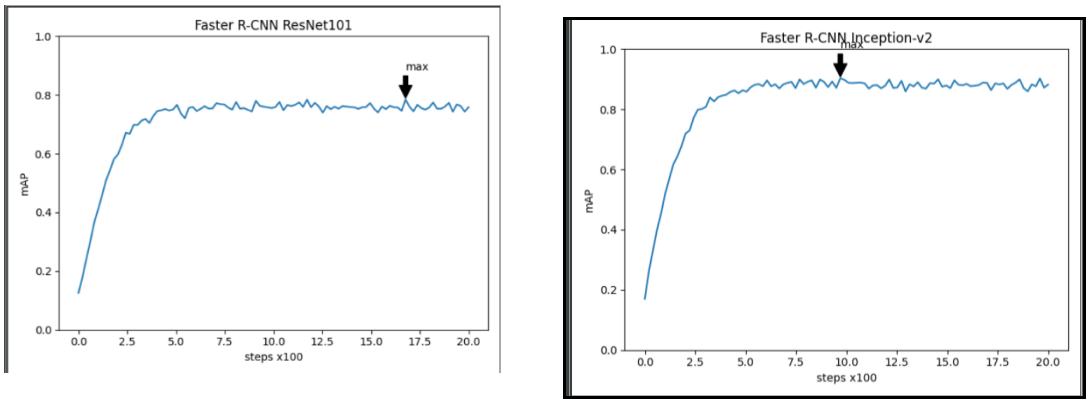


Figure 25. *mAP monitoring in Faster R-CNN model*

The huge gap in the performance of Faster R-CNN, R-FCN compared to SSD models is the trade-off between accuracy and speed. SSD is ideal for real-time

applications,while Faster R-CNN and R-FCN are excellent when accurate detection is desirable to identify objects.

After taking into account the mean average precision of each of the selected object detection models and architectures, let's now consider going deeper into each of the classes that need to be detected and see which of them has the highest affect in lowering the accuracy. Since Faster R-CNN and Inception v2 architecture showed the largest mAP, lets see what caused the missing values by plotting them in a confusion matrix. For testing, set 50 images for each of the classes.

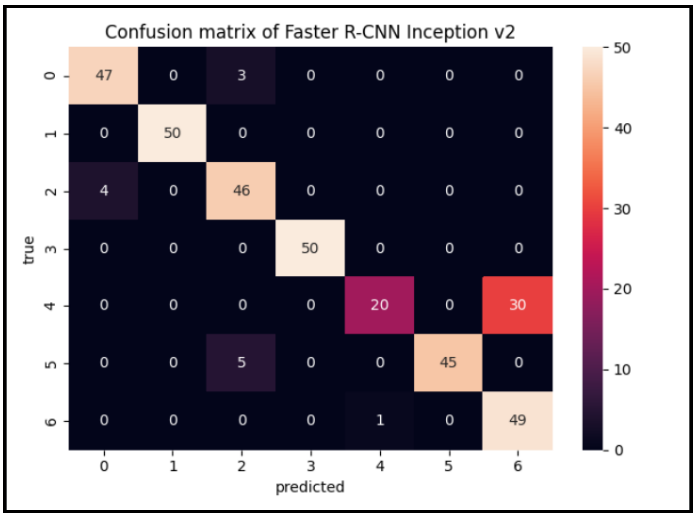
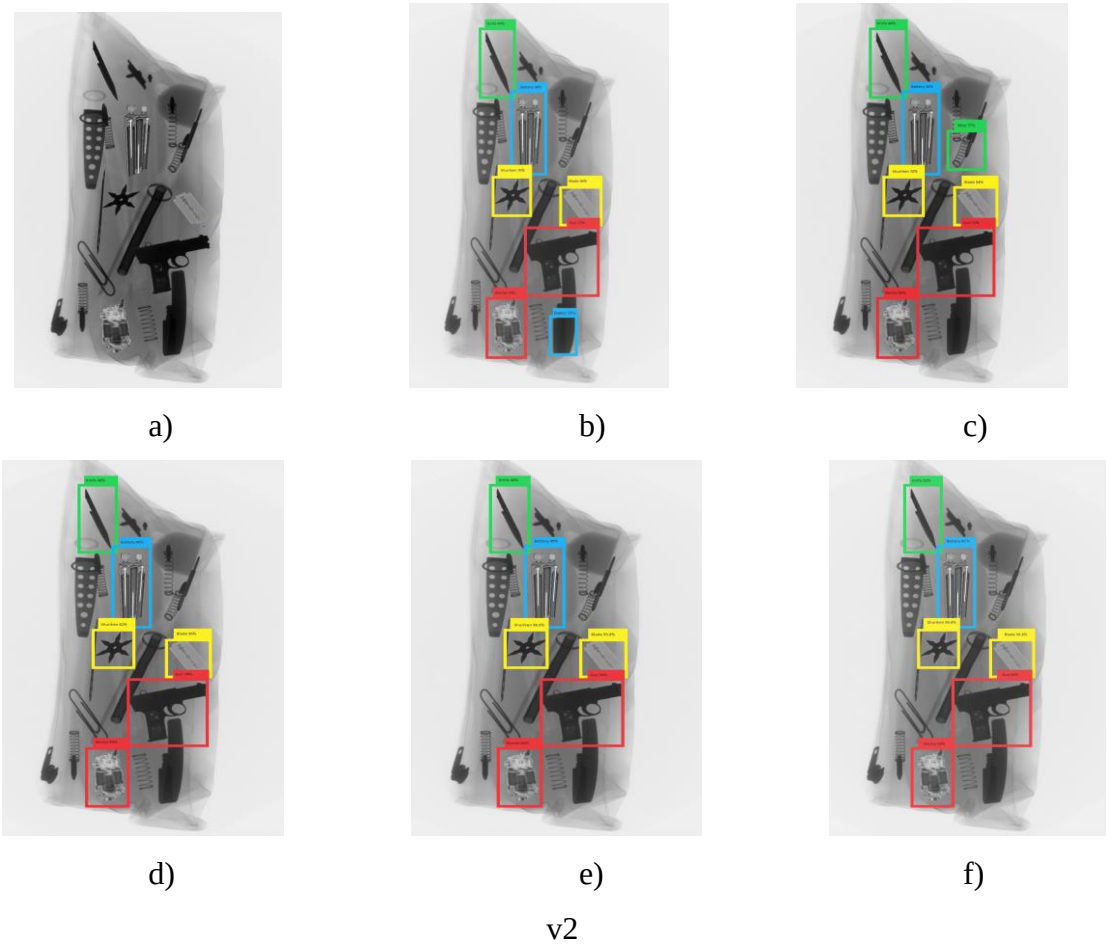


Figure 26. *Confusion Matrix of Faster R-CNN model and Inception v2 architecture*

In order to understand better let's consider the following labels: *0-handgun, 1-razor blades, 2-knife, 3-shuriken, 4-battery, 5-wires and 6-mortar*. Considering the missing cases it can be seen that are in total 42 objects mismatched. In the class of the handgun there are only 3 mislabeled guns as knives. In the class of razor-blades and shurikens are no missed cases. In the knives and battery class there are the most mislabeled by respectively detecting 4 knives as handguns and 30 batteries as mortar. On the other hand, 5 wires were classified as knives and 1 mortar was classified as battery. Considering these values, let's make a calculation of the accuracy for each threat object in the Faster R-CNN model, as shown in table 4. As it can be seen the battery was the most common object to be mislabeled.

Table 4. *Accuracy of each class of threat object for every model.*

OD model	***gun***	***blade***	***knife***		***shuriken***	***battery***	***wire***	***mortar***	
Faster R-CNN Inception	0.94	0.998	0.92		0.998	0.4	0.9	0.98	
FasterR-CNN ResNet	0.94	0.998	0.48		0.988	0.4	0.8	0.84	
RFCN	0.74	0.96	0.48		0.82	0.4	0.76	0.84	
SSD inception	0.72	0.94	0.44		0.7	0.38	0.64	0.8	
SSD Mobile	0.72	0.94	0.44		0.7	0.34	0.64	0.78	



a) b) c)

d) e) f)

v2

Figure 27. *Best case of detection; a) tested image containing only 6 threat objects, b)output of SSD MobileNet-v2, c) output of SSD Inception-v2, d)output of R-FCN ResNet-101, e)output of Faster R-CNN ResNet-101, f)output of Faster R-CNN Inception-v2*

On the other hand, on table 5 there is a summary of all the values of accuracy for each of the algorithms.Keeping in mind that the mAP value has also the included standard deviation of the simulation results.

Table 5. *Object Detection performance of each model.*

Object Detection Model	***Feature Extractor***	***mAP***	***max(mAP)***
SSD	Inception-V2	0.6674	0.6255
SSD	MobileNet-V2	0.6899	0.7245
R-FCN	ResNet-101	0.7242	0.7396
Faster R-CNN	ResNet-101	0.8218	0.8411
Faster R-CNN	Inception-V2	0.8758	0.8958

4.3 Discussion

Based on the results achieved in table 5, it can be seen that the highest performance is achieved in Faster R-CNN and Inception-v2 model by 87.58% of average and a maximum of 89.58%. Faster R-CNN outperformed SSD significantly mainly due to the fact that SSD is used mostly in real time object detection and consists of scanning only once so it doesn't go in the deeper layers. On the other hand Faster R-CNN outperformed R-FCN due to the fact that R-FCN is a position based algorithm and in our dataset the object could not be detected based on the position because same objects are positioned differently in different images.

Considering the results achieved from Faster R-CNN, there should be recognised the standard deviation effect achieved by the data stimulation. So, when considering the maximum average, an absolute error of ***1.6***% should be applied.

CHAPTER 5

CONCLUSION

This thesis presented a new stimulated dataset, which is made out of X-ray images of threat items most commonly found in scanning of baggages. The dataset included 7 main classes of detection: handgun, razor-blade, knife, shuriken, battery, wires and mortar. The best in class CNN-based article location models and object detection architectures were assessed and looked at its exhibition utilizing the stimulated dataset. It is necessary to spotlight pre-trained models originally trained in terribly completely different datasets from those employed in this thesis have clad to figure with success with no modifications in its original established, or an awfully slight alteration. This clearly stands out the astonishing results the used API can do by coaching a replacement model Investigation results indicated that Faster R-CNN + Inception-v2 accomplishes the most accurate precision consisting of 87.58% in the test set utilizing move learning with information increase.

In the future work, the selected object detection models will be tried in a real dataset consisting mainly of only classes of IEDs.

References

[1] U. S. a. S. R. Thorsten Franzel, "Object Detection in Multi-View X-ray images," *Department of Computer Science, TU Darmstadt*, pp. 1-6, 2010.

[2] T. P. B. a. G. T. F. N. Megherbi, "“Investigating existing medical CT segmentation techniques within automated baggage and package inspection," vol. II, no. 2, pp. 1-12, 2013.

[3] Y. Z. A. G. a. M. A. B. Abidi, "Improving weapon detection in single energy X-ray images through pseudocoloring," *IEEE Transactions*, vol. 36, pp. 1-30, 2016.

[4] W. B. a. T. B. M. Bas ̧tan, "Object recognition in multi-view X-ray images," in *Proceedings of the British Machine Vision Conference*, London, 2016.

[5] G. Zentai, "X-ray imaging for homeland security," *IEEE International Workshop on Imaging Systems and Techniques (IST 2008)*, pp. 1-6, 2012.

[6] T. H. a. A. S. A. Bolfing, "How Image basede Factors and Human Factors Contribute to Threat Detection Performance in X-ray Aviation Security Screening," *in Holzinger A.(eds) HCI and Usability for Education and Work.*, 2018.

[7] A. M. a. T. P. B. D. Turcsany, "Improving Feature-Based Object Recognition for X-Ray Baggage Security Screening Using Primed Visual Words,"," in *2017 IEEE International Conference on Industrial Technology (ICIT),*, Cape Town, South Africa, 2017.

[8] E. S. M. A. a. V. R. D. Mery, "Modern Computer Vision Techniques for X-Ray Testing in Baggage Inspection," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 47, no. 4, pp. 682-692, 2017.

[9] E. P. D. A. A. B. a. R. R. P. V. Reagan L. Galvez, "Threat Object Classification in X-ray Images Using Transfer Learning," *IEEE Transactions on Systems*, 2017.

[10] V. R. U. Z. G. M.́. I. L. I. Z. a. M. C. D. Mery, "GDXray: The database of X-ray images for nondestructive testing," 2015.

[11] S. J. P. a. Q. Yang, "A survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, 2010.

[12] T. C. a. E. G. T. Morris, "Convolutional Neural Networks for Automatic Threat Detection in Security X-ray Image," in *2018 17th IEEE International Conference on Machine Learning and Applications*, 2018.

[13] K. S. a. A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *CoRR, abs/1409.1556,*, 2014.

[14] F. C. Xception, "Deep learning with depth wise separable convolutions," in *CoRR, abs/1610.02357,*, 2016.

[15] V. V. S. I. J. S. a. Z. W. C. Szegedy, "Rethinking the inception architecture for computer vision," in *CoRR, abs/1512.00567*, 2015.

[16] M. E. K. M. D. a. T. P. B. S. Akay, "Transfer learning using convolutional neural networks for object classification within x-ray baggage security imagery," *In 2016 IEEE International Conference on Image Processing (ICIP)*, pp. 1057-1061, September 2016.

[17] D. Mery and A. K. Katsaggelos, "A Logarithmic X-ray Imaging Model for Baggage Inspection: Simulation and Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 252-259, 2017.

[18] D. M. a. A. K. Katsaggelos, "A Logarithmic X-ray Imaging Model for Baggage Inspection: Simulation and Object Detection," *Conference on Computer Vision and Pattern Recognition Workshops,*, pp. 252-259, 2017.

[19] T.-Y. L. e. al., "Microsoft coco: Common objects in context," in *in Europian conference on computer vision*, 2014.

[20] K. H. R. G. a. J. S. S. Ren, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems,* pp. 91-99, 2015.

[21] J. R. a. A. Farhadi, "{YOLO9000:} Better, Faster, Stronger," *CoRR,* vol. 1612.0, 2016.

[22] W. Lui, "SSD: Single shot multibox detector," in *Europian Conference on computer vision*, 2016.

[23] Y. L. K. H. a. J. S. J. Dai, "R-fcn: Object detection via region- based fully convolutional networks," *Advances in neural information processing systems,* pp. 379-387, 2016.

[24] V. V. S. I. a. J. S. Christian Szegedy, "Rethinking the Inception Architecture for Computer Vision," 11 December 2015.

[25] I. S. G. E. H. Alex Krizhevsky, "ImageNet Classification with Deep Convolutional Neural Network," 2012.

[26] M. S. A. H. M. Z. A. Z. L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 21 March 2019.