D4.1 Video Surveillance Techniques: Initial Release

Work Package:	WP4
Lead partner:	CERTH
Author(s):	Ilias Gialampoukidis (CERTH), Gerasimos Antzoulatos (CERTH), Nikolaos Kopalidis (CERTH), Evaggelos Bektsis (CERTH), Ioannidis Konstantinos (CERTH), Stavros Paspalakis (CERTH), Antonis Chronakis (ACCELI), Spyros Velanas (ACCELI), Paris Oikonomou (ACCELI), Giannis Spyropoulos (ACCELI), Gilles LEHMANN (CSNov), Thomas ANDREJAK (CSNov)
Due date:	30/04/2021
Version number:	1.0 **Status:** Final
Dissemination level:	Public

Project Number:	883284 **Project Acronym:** 7SHIELD
Project Title:	Safety and Security Standards of Space Systems, ground Segments and Satellite data assets, via prevention, detection, response and mitigation of physical and cyber threats
Start date:	September 1st, 2020
Duration:	24 months
Call identifier:	H2020-SU-INFRA-2019
Topic:	SU-INFRA01-2018-2019-2020 Prevention, detection, response and mitigation of combined physical and cyber threats to critical infrastructure in Europe
Instrument:	IA

Horizon 2020 European Union Funding for Research & Innovation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 883284

Revision History

Revision	Date	Who	Description
0.1	10/03/2021	CERTH	First release of the template. ToC and assignments finalisation
0.2	17/03/2021	CERTH	First release of the v0.1
0.3	31/03/2021	CERTH	Contribution from CSNov and ACCELI. Update version and release of the v0.2
0.4	07/04/2021	CERTH	Contribution for Object detection and activity recognition section
0.5	08/04/2021	CERTH	Proofreading of the Object detection and activity recognition section
0.6	09/04/2021	CERTH	Added contribution for Face Detection and Face Recognition modules
0.7	12/04/2021	CERTH	Enriched conclusions, added T4.2 and KR06 description (Section 1.2), added scope of KR06 (Section 1.3)
0.8	13/04/2021	CERTH	Added WP4 context (Section 1.1), added Executive Summary
0.91	21/04/2021	CERTH	Addressed the first internal review comments for Object detection section
0.92	23/04/2021	CERTH	Proofreading, added complete list of definitions and acronyms
0.93	26/04/2021	CERTH	Addressed the first internal review comments for Face Detection and Recognition
0.94	26/04/2021	CERTH	Added Section 3.2.5. Updated table of contents, list of figures, list of tables.
0.95	27/04/2021	CERTH	Added complete list of authors and Quality Control information. Minor corrections in references
1.0	28/04/2021	CERTH, ENG	Release final version

Quality Control

Role	Date	Who	Approved/Comment
Internal review	21/04/2021	CS	Document accepted; no changes required.

Internal review	20/04/2021	DEIMOS	Document accepted; only minor changes suggested

Disclaimer

Executive Summary

The purpose of *D4.1: Initial Release of Video Surveillance Techniques* is to present the initial release face detection and recognition as well as object detection and activity recognition tasks within WP4 of 7SHIELD project. Following the submission of the End User Requirements report, the requirements which are related to face detection and recognition as well as object detection and activity recognition are identified and then discussed in the context of the objective of each respective task. A thorough review of the recent academic literature regarding related work in the aforementioned tasks follows. The initial release of the aforementioned 7SHIELD Video Surveillance Techniques is then presented, discussing the architecture of the developed modules and the motivation behind methodology choices which were inspired by state-of-the-art works. Evaluation of methods and experimental results are presented as well, that support the applicability of the methods and provide some early performance indicators. Moreover, some early output examples and visualizations were given and finally the outline of future steps is presented.

Table of Contents

List of figures

List of Tables

Definitions and acronyms

AP	Average Precision
BiFPN	Bidirectional Feature Pyramid Network
C/P	Cyber/Physical
CCTV	Closed-circuit Television
CNN	Convolutional Neural Networks
CPU	Central Processing Unit
DCEP	Data Collection and Edge Processing
DNN	Deep Neural Network
DoA	Description of Action
FC	Fully Connected
FCN	Fully Connected Network
FCNT	Fully Convolutional Networks Tracker
FD	Face Detection
FDDB	Face Detection Data Set and Benchmark
FDR	Face Detection and Recognition
FN	False Negatives
FP	False Positives
fps	Frames per Second
FR	Face Recognition
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
HCC	Hyper Combined Correlator
HOG	Histograms of Oriented Gradients
IoU	Intersection over Union
KCF	Kernelized Correlation Filters Tracker
KR	Key Result
LBP	Local Binary Patterns
LFW	Labelled Faces in the Wild
PFE	Probabilistic Face Embeddings
PUC	Pilot Use Case
RPN	Region Proposal Network
SGS	Satellite Ground Station
SoA	State-of-the-Art
SSD	Single Shot Detector
SSH	Single-Shot Headless
SURF	Speeded-Up Robust Features
SVM	Support Vector Machine
TN	True Negatives
TP	True Positives

UAV	Unmanned Aerial Vehicle
UR	User Requirement
WP	Work Package
YOLO	You Only Look Once

1. Introduction

1.1. Context

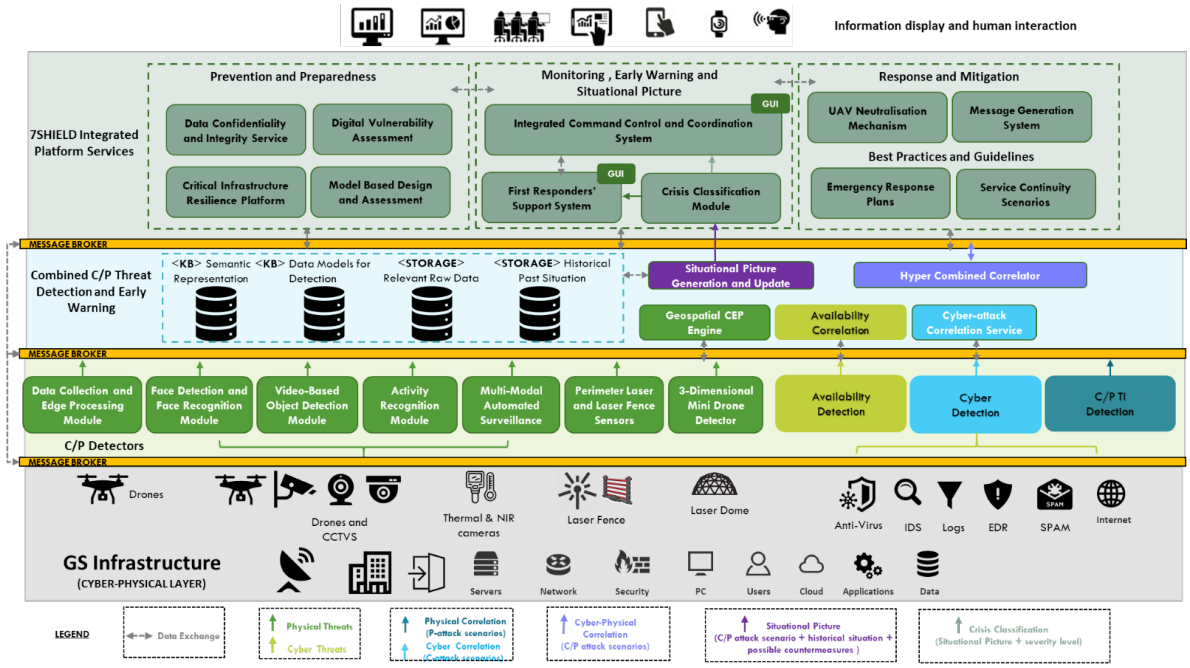

Figure 1.1 - General architecture of 7SHIELD

7SHIELD aims to be an integrated framework for the deployment of cyber-physical protection services of ground segments. The protection capabilities will be integrated or interoperating with existing protection solutions already deployed at their installations. The overall objective of Work Package 4 (WP4) is to provide crisis management for physical and cyber threats. Physical and cyber threats will be detected from an array of cutting-edge solutions, each one targeted exclusively to cover specific potential vulnerabilities in the protected facilities. The general architecture of 7SHIELD is shown in Figure 1.1. WP4 modules are located in the Cyber-Physical detection layer which will facilitate technologies such as, UAV monitoring and processing at the edge, face detection and recognition from CCTV cameras, object recognition and activity recognition from CCTV cameras, cyber-attack detection methods, infrared and thermal image processing for the detection of man-made disasters, laser based technologies for the detection of ground-based and aerial threat detection, and a combined physical and cyber threat early warning mechanism.

The purpose of this Deliverable is to present the initial release of the video surveillance techniques and will specifically focus on face detection and recognition as well as object detection and activity recognition tasks, namely T4.2 and T4.3. A short description for each task is provided bellow:

T4.2 Face Detection and Recognition

Task 4.2 will produce Key Result KR06 which states: "The module will be able to process still frames or video streams, in order to export the detected and recognized faces that may belong to suspicious individuals. The framework will be linked with a criminal database, cloud or local, which will contain the list of suspects, as well as image data on which they should be clearly depicted. The main expected result is the production of alarms whenever a person is found to closely match one of the suspects." Note that although only the recognition of suspicious individuals is referred in the KR description, from the User Requirements assigned to KR06 it is concluded that the recognition of authorized individuals is deemed just as important.

T4.3 Object Detection and Activity Recognition

Task 4.3 will produce Key Result KR07 which states: "This module will process video streams or still images in order to locate and recognize objects of interest in the provided sources. Additionally, after detecting any human presence in the scene the corresponding results of object detection will be propagated to the activity recognition sub-module to identify suspicious and harmful activities. The main purpose of the module is the accurate and efficient visual interpretation of the surroundings of the surveillance area."

The introduction is concluded with a summary of the adjacent WP4 tasks, to provide broader context as well as the progress on the corresponding activities.

1.2. Summary of related WP4 tasks

T4.1 Data collection from UAVs and processing at the edge

The Data Collection and Edge Processing (DCEP) Module vision is to introduce, develop, test and evaluate a UAV agent, equipped with advanced sensing and cognitive capabilities that can support missions at tactical level and replace the necessity for human-operator. As such, DCEP aspires to exploit innovative technologies and tools that will reach adequate maturity during projects lifetime, towards offering a complex interoperable surveillance system capable of supporting a wide set of missions in diverse environmental conditions.

The Data Collection and Edge Processing (DCEP) Module will be responsible for the collection of videos and a number of sensing data and will be able to host and run (edge processing) the object detection and identification algorithms. The proposed module will be offloaded on 7SHIELD UAV. ACCELI will provide a fully customized UAV, able to accommodate the various hardware components (e.g., height/distance sensors, cameras) in order to execute smart algorithms (e.g., visual object detection and collision avoidance services, algorithms for swarming), and generally to be easily adapted to the current operation by the user.

Technologies and solutions of DCEP platform can be split into the following three categories:

- **Front-End equipment systems and collection tools:** based on technologies employed in the surveillance and monitoring scenarios so as to acquire relevant and sensible information in real time (e.g., drone equipped with multiple sensors);
- **Data Fusion and Mediation Systems:** fuse the information from heterogeneous sources (i.e., drones, sensors etc.) and will provide a detailed and accurate situational/ context awareness to the end users.
- **Back-End applications, services and portals:** provide decision support capabilities to aviation stakeholders and vertical experts in the areas of critical infrastructures, transportation and public safety.

DCEP platform will contain all features that end-users identified and classified as highly important during the piloting and enhance them with the latest developments in drone related technologies. As various systems will be interacting having various operating systems, applications and technologies leading to a heterogeneous network of interacting devices, sensors and applications, a communication layer able to fast access, transmit and convert data in common understandable formats is required. As data is centrepiece of maintenance and monitoring activities it is required to make data fast accessible and stored in a secure decentralized manner. As technologies evolve and new applications and next generation sensors also, a new device should be able to connect to the network having a flexible and autonomous permission system that will allow data exchange immediately without exposing sensitive information to the public.

Heterogeneous data gathering sensors: A dedicated Payload Management system allows quick plug-and-play hardware reconfiguration, so that different sensor payloads can be installed on-board the UVs depending on the specific mission needs. Raw sensor data is both stored and processed locally on-board UVs and streamed to other drones or end-users when needed. Sensors may include acoustic sensors, video sensors, GPS receivers, sonars, laser scanner, 3D cameras etc. Sensors will be classified according to their autonomy and will be used either for constant monitoring or "need to" intervention, annotated with the current position and UV status information, whereas relevant (e.g., altitude, speed, etc.).

Event detection capabilities: The data gathered from the available sensors will be processed for event detection in a hierarchical way, both locally at the agents (when possible, in a cooperative manner) and centrally at the command centre. Results of collaborative data processing and fusion tasks can be directly exploited by the UAV as well as transmitted to end-users for further analysis and processing. Early event detection will trigger alerts that will prompt the operator for further action.

ACCELI UAV will be equipped with on-board sensors capable for surveillance operations, such as visible light sensors, hyperspectral sensors, and thermal sensors. The role of embedded sensors will be to acquire high-spatial and temporal images that can facilitate

the surveillance of a specific area (covering a predetermined distance/radius around the ground stations).

Computations will be executed on the on-board supercomputer NVIDIA JETSON, which delivers improved AI performance at a small size, making it ideal for mobile robotic applications. 7SHIELD UAV will be capable to perform on-board image processing, making use of machine learning-based techniques, and more precisely Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) exploiting the data (e.g., 2D/3D images/point clouds) obtained from the various aforementioned sensory inputs. **More detailed description will be presented in D4.3 due in month 14 by ACCELI.**

Secure communications framework: A Communication Manager will be installed on 7SHIELD UAV allowing it to be provided with different, inter-changeable communication front-ends, supporting a multitude of different tactical links enabling direct communication between UAV and the main Command and Control System (C2).

Advanced cooperative navigation capabilities: Platform will feature advanced methodologies to allow cooperative navigation for UAV agent. Once a mission is assigned to selected UAV on the field, the UAV typically operates in semi-autonomous, distributed, cooperative and coordinated fashion. A manual Remote-Control functionality is provided exclusively when direct human control is explicitly required. Cooperative localization features, relying on an integration of GNSS-based and shorter-range ranging technologies, will be used to ensure precise absolute and relative positioning of all UAVs at all-time, to enable group navigation and precise geo-localization of the collected data.

T4.7 Combined Physical and Cyber Threat Detection and Early Warning

The main objectives of the task 4.7 are combining cyber, physical and monitoring detection in a single architecture and connect this architecture with WP4 detectors. To do that a unified format for data detection and correlation will be needed (Cyber, Physical and Availability and Combined). For now, the architecture for collection and correlation is designed, and all responsibilities are dispatched.

Sensors and detectors of 7SHIELD are represented by tasks 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6. They have filled a survey detailing who they are, what they do, on which operating system they are running, what alerts they can raise, etc.

Workflow for processing events:

- Sensors and detectors of task 4.2 and task 4.3 will send UAF alerts over the 7SHIELD Message Broker
- The Correlation Availability module and the Correlation Physical module will do the correlation depending on the use cases detailed by WP2
- Alerts will be correlated in a combined way by the HCC
- Alerts are sent to WP5.

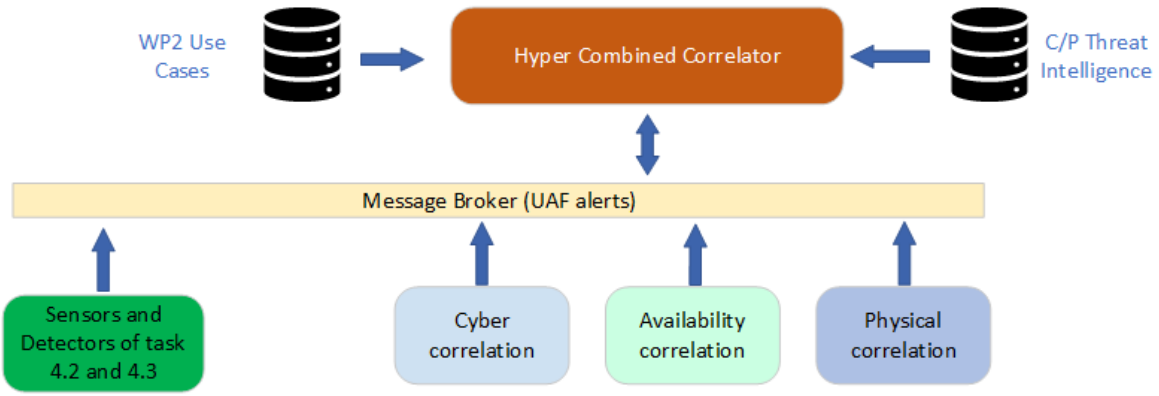

Figure 1.2 - Combined Physical and Cyber Threat detection

Next deliverable of task 4.7 is D4.2 Combined Physical and Cyber Threat detection (M9, May 2021).

1.3. Scope

Regarding KR06, the related User Requirements are first discussed in Section 2.1.1 and are discussed in the context of the task objectives in Section 2.1.2. The Introduction concludes a discussion about the first architectural schema of Task 4.2 and the module's connectivity inside 7SHIELD. Then, an extensive review of related work regarding face detection and face recognition is presented in Section 2.21 and 2.2.2 respectively. Face detection and recognition datasets and public benchmarks are presented as well in Section 2.2.3. Next, the adopted methodology details are discussed in Section 2.3 accompanied by preliminary experimental results in Section 2.4.

Regarding the KR07, all relevant technologies for video-based object and activity recognition tasks will be presented in this deliverable. Section 3.1 provides an introduction to the general framework. More specifically, Section 3.1.1 presents the User Requirements related to the task, Section 3.1.2 comments on the objectives of the module while Section 3.1.3 presents a high-level view of the architecture of the module in respect with its position in the project architecture. In Section 3.2 some related works are presented for each of the pillars of the module: Section 3.2.1 for Image Classification, Section 3.2.2 for object detection, Section 3.2.3 for Activity recognition and finally Section 3.2.4 for the tracking. The next Section regards the methodology being used in the Object detection (Section 3.3.1) and Activity recognition (Section 3.3.2) subtasks, focusing more on the specific models being used. Next, Section 3.4 include some initial experiments and results of the object detector module and Section 4 concludes the KR07 relevant information section. Finally, all references are grouped in Section 5.

2. Face Detection and Recognition

2.1. Introduction

2.1.1. User & Functional Requirements

Following the submission of the End User Requirements (URs) report several requirements were examined regarding their relevance to KR06 which is the main KR produced by Task 4.2. Those are presented in the Table below.

ID	Type	Priority	Description	PUC	KR
FR_SCE_04	Functional	MUST	7SHIELD must use facial recognition to identify target as authorized or unauthorized person.	PUC1(a)	KR06
FR_SCE_11	Functional	COULD	Dashboard should show live image of the target from cameras and UAV, annotating the detected object to help user immediately see why there is live image.	PUC1(c)	KR20, KR05, KR07
FR_SW_11	Functional	MUST	7SHIELD must indicate the confidence of the alert (e.g., 60% confident that the detected location is correct).	PUC3	KR21, KR11, KR07

Table 2.1 - User Requirements related to KR06

2.1.2. Objectives

All the requirements are functional with two of them being of highest priority, while the third was given a lower priority. Starting from the highest priority the related URs are discussed in the context of the KR06 objective:

- Functional requirement FR_SCE_04 states "7SHIELD must use facial recognition to identify target as authorized or unauthorized person" which is well aligned with the description of KR06, thus, it was recognized as the main UR which will be fully supported by KR06. Its description is straight forward setting the ultimate goal of the module which is to detect and characterize targets that appear in CCTV streams as authorized or unauthorized.
- Functional requirement FR_SW_11 is related to output of KR06 which will be reused by other 7SHIELD modules, specifically the higher-level correlators. The UR is with

regard to the need for a measure of confidence for each alert, which in the scope of KR06 refers to the confidence that a detected face is characterized authorized or unauthorized. The UR was not initially planned to be addressed by KR06, but a second examination revealed the importance of including model confidence in the module's reports.

- Functional requirement **FR_SCE_11** refers to the visualization capabilities of the dashboard which should be supported by KR06 regarding the bounding boxes of detected faces and recognized individuals. The UR was not initially planned to be addressed by KR06, but a second examination revealed that the dashboard cannot produce such outcomes if such information does not exist in the module's output.

2.1.3. Architecture

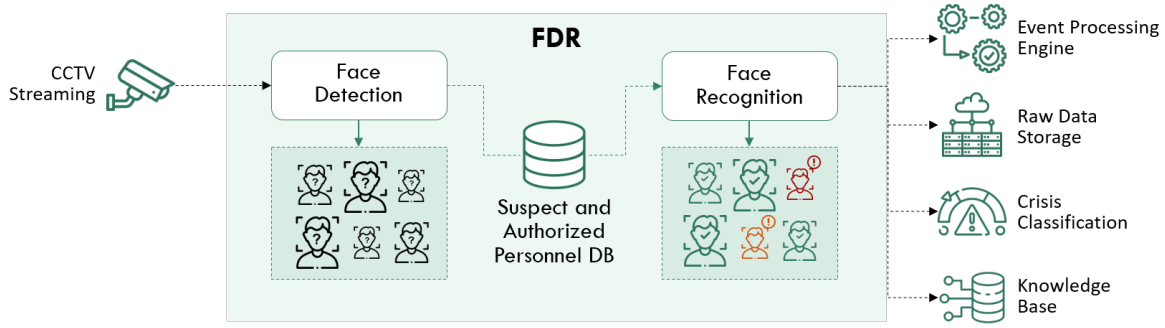

Figure 2.1 - 7SHIELD FDR architecture and connectivity diagram

The overall approach of Face Detection and Recognition (FDR) is naturally split into its two cooperating tasks and is shown in Figure 2.1. In general, such a system is initialized with an image file, a video file, or a video stream in the case of 7SHIELD. In any case, the module is designed to handle single frames in a serial processing pipeline (one after another), but it can be extended to handle batches of images if necessary, given the required hardware resources are available.

Processing begins on the Face Detection (FD) component. FD is responsible to detect patches inside the input frame where faces are tightly enclosed. The acquired face patches are instantly characterized as *unknown* and are immediately provided to the Face Recognition (FR) component for further processing. FR cooperates with a pre-existing gallery of *known* faces and is responsible to decide which of the *unknown* faces can be matched and with what certainty with any *known* face from the gallery. *Known* individuals can be either (a) authorized personnel expected to appear in the area covered by the CCTV camera, or (b) unauthorized individuals which are not allowed to be in this location. This information, along with photos depicting the *known* faces must be previously defined during the initialization of the FDR module and the gallery creation.

After the recognition process, a detailed report can be produced with the detection and recognition details, e.g., bounding boxes showing detected faces, recognized known

individuals, alarms raised for unknown individuals etc. Those results are immediately forwarded to the 7SHIELD Event Processing Engine, the 7SHIELD Crisis Classification module, the 7SHIELD Knowledge Base and are also stored in the 7SHIELD Raw Data Storage.

2.2. Related Work

In this section, the related works in face detection and face recognition domains are presented. Focus is given on more recent approaches as well as works that have inspired the 7SHIELD FDR methodology.

2.2.1. Face Detection

Face detection has always been a widely researched computer vision task. In early face detection techniques, shallow representations, like Haar cascades [26], SURF (Speeded-Up Robust Features) [27], Histograms of Oriented Gradients (HOG) [28] and Local Binary Patterns (LBP) [29] were manually designed in order to detect faces. However, the exploitation of handcrafted features entailed the deployment of exhaustive sliding window search techniques so as to discover candidate boxes of faces. This requirement made them extremely demanding in computational resources. Thus, later works focus on faster and more accurate approaches to this problem. To this end, efficient facial feature localization methods were proposed, such as the mixtures-of-trees [30] and consensus of exemplars [31] which required much less computational time. Those methods work by deploying dedicated local detectors for different parts of the face or key-points, and the final output is produced by fusion of the individual detection results.

Soon thereafter, with the rise of deep neural networks, and especially in the tasks of image classification and object detection, a breakthrough in performance can be observed. In [32] a deep convolutional network cascade was proposed, which achieved high detection rates, on some popular challenging face detection benchmarks like LFW [31] and YouTube faces [33]. Using facial key-point detection as a primary goal, the cascade of networks refined facial key-point location in each level, fusing the output of multiple networks, whilst implicitly encoding geometric constraints between the points. Furthermore, other methods tackled face detection like every other generic object detection task, like the ones proposed in [34] and [35]. These methods adopted high-accuracy generic object detection CNNs and fed them with annotated face boxes for training, while at the same time, applied techniques like hard negative mining, feature concatenation and careful fine-tuning to improve their results.

More sophisticated works considered the spatial structure and arrangement of facial parts [36]. Several CNNs were trained, each one dedicated to the detection of a single face part, while early convolution features maps were shared to improve the efficiency of the method. Moreover, the "faceness" score was introduced, which was a measure of "what constitutes

a face". An object proposal ranking stage was calculated using "faceness" by determining how well each proposal met the structural constraints posed by the detected facial parts. Later, the framework proposed by [37], leveraged a cascaded architecture with three stages, each one deploying a deep CNN. Object proposals were extracted during the first stage using a Fully Connected Network (FCN), false positive candidates were filtered using another CNN in the second stage, and further refinements were performed based on facial landmarks during the final stage.

Simultaneously with the object detection works found in the literature at that time, single-shot architectures used in generic object detection found their way into the face detection research interests. For instance, the SSH (Single-Shot Headless) face detector [38], alleviated the need for a face bounding box proposal generation step. Different layers of an FCN were deployed, so as to predict various scales of faces in a single forward pass. The work in [39] proposed a framework to deal with relevant tasks simultaneously, i.e., automatic facial landmark detection, pose estimation, gender recognition and face detection. This method was categorized as region-based, in essence working on patches of the image (candidates). Deep convolutional features taken from different layers were first fused and then fed to a five-headed output, where each "head" was a Fully Connected (FC) network dedicated to a task. The problem of detecting small objects, and thus faces as well, was the focus of [40]. Several key insights for small scale face detection, like the information in surrounding context of boxes and the use of large receptive fields, were established. They found that training multiple detectors for various scales produced state-of-the-art (SoA) results, while applying feature sharing between layers of the CNN hierarchy allowed the efficiency to be maintained to acceptable levels. More recently, inspired by Feature Pyramid Networks, multi-scale features were extracted, and high-level feature maps of various scales were aggregated, to argument low level feature maps as contextual cues in an agglomeration manner, with a hierarchical loss to train the pipeline, in the work of [41].

2.2.2. Face Recognition

Face recognition (FR) has always been an extremely active topic in the computer vision domain. FR includes all the algorithms and techniques designed for face identification or verification. Face identification refers to the problem of classifying a face to a certain identity. Face verification, on the other hand, is the problem of determining whether or not a pair of faces belong to the same identity. Generally, the first step of FR is face feature extraction and representation. Then, the resulting vector is either classified to an identity, or the minimum distance to the gallery vectors is found and a match is confirmed. Thus, FR can be viewed as a classification problem or a metric learning problem depending on the testing settings. Specifically, in an open-set setting, test identities might not appear in the training set, therefore, FR resolves to metric learning. On the contrary, in a closed-set setting, where test identities exist in the training set, a classifier is the proper way to solve the task.

Early works built low-level descriptors, such as LBP, SIFT, or CMD, which were then combined with a shallow model for identification, such as SVMs, discriminative dimensionality reduction or Fisher encoding [42, 43, 44, 45]. Later works depend heavily on deep learning methodologies to achieve significant boost in performance. In this class of algorithms, deep feature extractors are used to generate face representations, tuned for pose and illumination invariance, from the plethora of the available training data rather than from low-level hand-crafted features.

Siamese networks for deep metric learning were proposed in the work of [46], which was one of the initial attempts to leverage deep learning. A Siamese network works by extracting features separately from two modes (inputs), with two identical CNNs, taking the distance between the outputs of the two CNNs as dissimilarity. In the work of [47], the warping of faces from arbitrary poses and their transformation to frontal view with normal illumination was proposed using a trained deep neural network. Then, face representations were extracted from the last hidden layer, a practice often seen in deep learning approaches. Other works, involving multi-stage networks, proposed to align the faces using 3D modelling and to then feed them to a multi-class network for identification [48].

In a similar fashion with face detection methods, facial parts were processed separately in cascade networks, as in the work of [49]. Soon after, the focus shifted heavily towards improving the deep metric learning methodology, which led to significant performance improvements [50, 51, 52]. Experimentation with novel face similarity measures dominates the undertaken effort in these works. Moreover, discriminant face representations are characterized by smaller maximal intra-class distance and minimal inter-class distance in the embedding space, thus, novel CNN loss functions are meticulously explored as well, in order to find the most appropriate for the task.

2.2.3. Datasets

As far as existing benchmarks, there have been several datasets for face detection and face recognition made publicly available. Some of them are dedicated to the detection or the recognition task, while others offer images with metadata suitable for both tasks. Here is a non-exhaustive list of the most prominent ones:

- The **Labelled Faces in the Wild (LFW)** [31] dataset is a database of face photographs designed for studying the problem of unconstrained face recognition. The data set contains more than 13,000 images of faces collected from the web. The people that appear in this dataset are known public figures like politicians, athletes, actors, musicians and other various celebrities.
- **WIDER FACE** [53] dataset is a face detection benchmark dataset. It contains over 30000 images which mostly show people participating in various activities of everyday life based on 61 event classes. The human faces appear with a high degree of variability in scale, pose and occlusion. For each event class, predefined splits

consisting of 40%/10%/50% of the total amount of data exist as training, validation and testing sets respectively.

- The **Face Detection Data Set and Benchmark (FDDB)** [54] contains face regions designed for studying the problem of unconstrained face detection. This data set contains the annotations for 5171 faces in a set of 2845 images taken from the Faces in the Wild data set. The faces appear within a wide range of challenging scenarios including occlusions, difficult poses, low resolution and out-of-focus. In addition, annotation for face regions exist in the form of elliptical regions.
- The **CASIA-WebFace** dataset [55] is used for face verification and face identification tasks. The dataset contains 494,414 face images of 10,575 real identities collected from the web. The identities were extracted from the IMDB website.
- The **IARPA Janus Benchmark-C (IJB-C)** [56] face dataset is a challenging benchmark for unconstrained face detection and recognition. With its increased size (around 140000 faces) and variability, given emphasis on occlusion and diversity of subject occupation and geographic origin, it aims towards the improvement of the representation of the global population.
- The **YouTube Faces** Database [33] is a database of face videos designed for studying the problem of unconstrained face recognition in videos. The data set contains 3,425 videos of 1,595 different people. All the videos were downloaded from YouTube. An average of 2.15 videos are available for each subject. The shortest clip duration is 48 frames, the longest clip is 6,070 frames, and the average length of a video clip is 181.3 frames.

Table 2.2 provides a summary of the face detection and recognition datasets. For a dataset made for face recognition (or face verification), the number of face instances which appear in the images as well as the number of people that make up that dataset is given. In the case of face detection, the number of faces means the number of annotated bounding boxes in the entire collection of images (more than one face may exist in a single image), while the number of people is not given since it is irrelevant.

Dataset	Task	# faces	# people
LFW	Face Recognition	13233	5749
WIDER FACE	Face Detection	393703	-
FDDB	Face Detection	5171	-
CASIA-WebFace	Face Recognition	494414	10575
IJB-C	Face Detection and Recognition	140623	3531
YouTube Faces	Face Detection and Recognition	500000	1595

Table 2.2 - Summary of face detection and recognition datasets. The number of annotated face instances as well as the number of individuals depicted are reported.

Figure 2.2 shows the main factors that can pose significant challenges to the face detection or recognition solutions. As far as scale, the most challenging scenario is the detection and recognition of small faces and very distant ones. This challenge exists broadly in the generic object detection literature where it has been targeted and analysed as an isolated problem [61, 62, 63]. Facial parts may be severely occluded when arbitrary poses are considered, making it harder to rely on a consistent facial structure model. Arbitrary occlusions, like face masks, may also pose the same challenge, as well as extreme facial expressions. On the other hand, non-structural elements, like skin tone, may be heavily altered by extreme illumination conditions or makeup. Importantly, all the established face detection and recognition benchmarks, have made considerable efforts to offer some degree of variability in all the aforementioned challenges.

Figure 2.2 - Major challenges in unconstrained face detection and recognition. The established benchmarks offer high variability in scale, pose, occlusion, expression, appearance and Illumination.

2.3. Methodology

In this section, a closer look on the internal mechanisms of the FDR components is first given, and the synchronised operations between them that make up the FDR framework are discussed. Figure 2.3 shows the overall FDR framework architecture to be described.

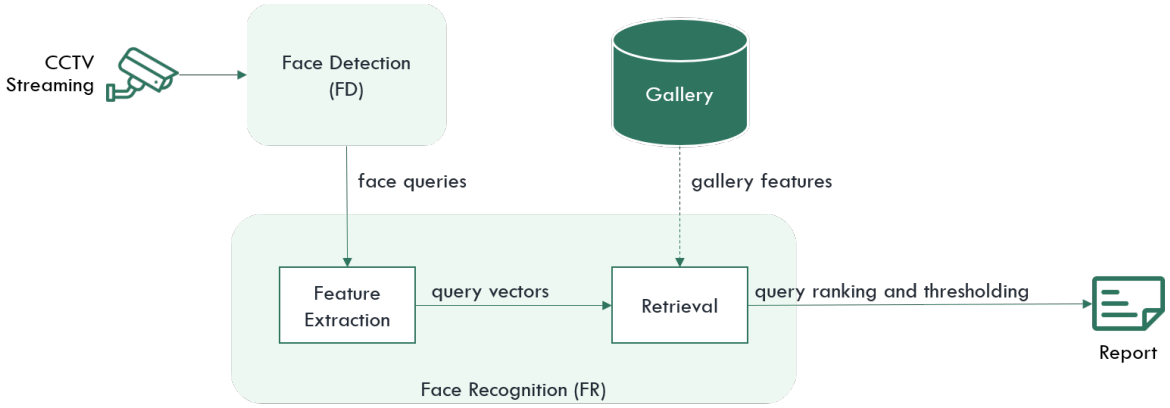

Figure 2.3 - 7SHIELD FDR framework

2.3.1. Face Detection

The FD component is responsible to detect all the faces in a given image or video frame, which in the scope of the 7SHIELD project are intended to be later categorised as known (authorized), or unknown (unauthorized) individuals. Naturally, a deep CNN architecture, like the ones presented in recent works discussed in Section 2.2.1, is adopted to serve this purpose.

As it is apparent in the presented literature and datasets related to face detection, the availability of massive amounts of training data are required to learn meaningful representations of faces, especially if the power of a state-of-the-art deep architecture is to be fully exploited. The acquisition of such amount of data is a challenging task and involves not only the task of gathering a large quantity of images with the objects of interest, but also painstakingly annotating the objects with bounding boxes, which both demand a lot of manual labour.

Fortunately, along with several datasets that have been made publicly available, some of the deep architectures pre-trained in these massive pools of data have also been published as "off-the-shelf" object detectors. State-of-the-art face detectors have already surpassed high accuracies in the area near 95% [58, 64, 65] in the respective benchmarks [53]. As such, the 7SHIELD FD component can adopt the acquired discriminative power of the available pre-trained models.

One major advantage of the current popular development frameworks that exist for deep learning, such as PyTorch or TensorFlow, is that they offer the required flexibility to use such models in a system interchangeably. This can be done by making the input and output of the model have a predefined structure. Then, the deployed model itself can be replaced easily with another, in accordance with the requirements at hand. In the experimental section that follows, any one state-of-the-art pre-trained face detectors, from the assortment that is presented, can be directly adopted in the 7SHIELD FDR module.

2.3.2. Face Recognition

FR is responsible to take input from FD in the form of cropped portions of images or video frames, each one containing a single face, and characterise them as being known (authorized) or unknown (unauthorized). This process entails that a gallery of the recognizable identities is first constructed. The gallery contains photos from the individuals that should be recognized by the FR component. Ideally, it contains multiple photos per person, in various conditions (pose, illumination, scale, etc.) and the photos are captured in different times of the person's life, providing samples of age progression or small variations in facial appearance that happens naturally on a daily basis (hair, accessories, etc.). Each time a new face is captured by FD, the FR component will search through the gallery to find appropriate matches.

The 7SHIELD FR incorporates a retrieval scheme. The retrieval process in abstract terms is a two-step communication. First, a **query** is entered to the system. The query is a request for information made by some user or other system which has a specific format and can be read by the system. The system then has to identify what information is in its reach and is relevant to the query and return it to the user, usually in the form of a ranked list of **objects** starting from the most relevant object. The objects are entities that are represented by information within a **collection** or a database. The system must decide which information from the database matches to the query contents and return the **top-ranked** objects linked to that information. The ranking of results is just as important as every other characteristic on the retrieval scheme.

When this abstract description is applied to the FR component, the query becomes a detected face "requesting to be recognized". The queries are all initialized with the status of *unknown (unauthorized)* faces. In our case, the FD component is the entity that generates the queries, the objects are the known individuals from the gallery, and they are represented by their photos. Therefore, the query face (image) is matched against all the faces (images) of the gallery. This process results in a ranked list of all the images in the gallery. Thus, the corresponding individuals are ranked accordingly from the most similar to the least similar to the query. Note, that this approach is fully dynamic and can work with any assembled gallery of faces (images). This means that the actual photos used for comparison can be altered or removed, or new ones can be added at any given time. It is also agnostic of the individuals' actual names and the gallery photos can be linked with anonymized IDs.

The ranking is based on some distance or similarity metric, e.g., Euclidean or Cosine. Specifically, during system initialization, the desired gallery is first processed. All the face images in the collection are fed to the FR component which then extracts their unique representations. The feature vectors are stored and are linked to their respective anonymized identities. When a query face arrives, its representation is also calculated from the FR component. Then, the distance metric is performed between the query feature

vector and all the gallery feature vectors. The smallest distance indicates highest similarity and is ranked at the top, the next smallest distance is ranked second and so on and so forth. This way a ranked list of feature vector distances is produced along with the corresponding ranked list of suspects. From then on, the top ranked distance value is examined. The gallery identities in the 7SHIELD project are intended to be solely *authorized* persons. Thus, if the value is smaller than a predetermined threshold, the query's status is altered as a possible authorized match. Otherwise, if no match is declared, the query's status remains as *unauthorized,* which produces the appropriate alert in the final report.

2.4. Experiments and Results

The experiments in this section were conducted with the aim to (a) deploy deep learning face detection and recognition models as a means of testing the development platform, (b) replicate and confirm the published evaluation results on public benchmarks and (c) make performance comparisons and draw conclusions about the state-of-the-art. For each task, three approaches were selected to represent the current state-of-the-art landscape. Each one was evaluated in a benchmark dataset, appropriate for the task, from the ones described in Section 2.2.3. Specifically, for face detection the WIDER FACE benchmark was selected, and for face recognition the LFW.

The evaluation metric for face detection is **Average Precision**. It is taken by calculating the area under the Precision-Recall curve. **Precision** is defined as the proportion of **true positives (TP)** out of all the *detected* faces and **Recall** as the proposition of true positives out of all the *annotated* faces. In other words, precision measures the accuracy of the detector and recall measures its ability to retrieve the existing faces. Whether a bounding box detection counts as a TP is decided based on its overlap with a ground truth box. The overlap is measured by the **Intersection over Union (IoU)** threshold. Thus, detected faces must have a good alignment with true faces in order to be considered correct. The Precision and Recall metrics are calculated for every alignment threshold (from most relaxed to most strict) to draw the Precision-Recall curve.

The evaluation metric for face recognition is **Accuracy**. The dataset is split to 10 equal parts, where the first 9 are used for cross validation in order to select the optimal distance threshold to achieve top accuracy. The 10^{th} part is the test set from which pairs of queries are given and the model decides if they belong to the same person or not. This is a standard strategy for evaluating face verification models, but it is also a good indicator of face recognition performance as well.

2.4.1. Face Detection

A brief overview of the evaluated face detection methods is given in this section.

TinyFaces [40]: TinyFaces is tuned to solve a binary, multi-channel heatmap prediction problem, utilizing a Fully Connected Network (FCN), where the predicted heatmap at a

single pixel location resembles the confidence of a fixed size detection window centred at that pixel. Separate scale-specific detectors were designed and trained in order to maximize performance for each face scale with respect to the characteristics of a detector's template size and resolution. The design is based on two very important conclusions that were verified in [40] involving the surrounding context and resolution of face boxes. It was proven that using deep CNN features of multiple layers (receptive fields), is of extreme importance especially for finding small faces.

PyramidBox [57]: This method is also motivated by the results in [40] regarding to the role of context in detecting hard samples. A novel context anchor was designed for the supervision of high-level contextual feature learning. High-level contextual features were combined with the low-level one by a Feature Pyramid Network. The prediction branch, which is the final stage of detection, was also specifically designed to be context-sensitive.

DSFD [58]: The main contributions of this method include a novel Feature Enhance Module to utilize different level information and extend discriminability and robustness of features, the insertion of auxiliary supervisions in early layers, via a set of smaller anchors and the improvement of anchor matching strategy to provide better initialization for the box coordinate regressor.

2.4.2. Face Recognition

A brief overview of the evaluated face recognition methods is given in this section.

Facenet [60]: The authors propose a deep CNN feature extractor, coupled with L2 normalization, leading to an embedding which is trained in an end-to-end manner by a triplet loss function. Triplet loss works by minimizing the distance between queries and their corresponding positive samples, (i.e., same identities), while maximizing the distance between queries and their corresponding negative samples (i.e., different identities).

PFE [59]: The motivation behind Probabilistic Face Embeddings is to deal with data uncertainty. Deterministic embeddings can distort the distance metric by *forcing* the estimation of features of ambiguous faces to become points in the feature space. A solution is proposed in which the uncertainty is encoded within the face representations and is taken into account during matching. Each face image is represented as a Gaussian distribution in the latent space instead of points, where the mean of the distribution estimates the most likely facial features while the variance can be interpreted as the uncertainty in the feature values.

Arcface [50]: Arcface's focus is on the design of an appropriate loss function for large scale face recognition. The Additive Angular Margin Loss is proposed in this direction which works by inserting a geodesic distance margin between the sample and corresponding class centre. The approach was compared to Intra-Loss, which decreases the geodesic distance between the sample and the corresponding centre, Inter-Loss, which increases the

geodesic distance between different centres and Triplet-Loss, which insert a geodesic distance margin between triplet samples and was found superior.

2.4.3. Results

Face Detection Method	WIDER FACE AP (%)
TinyFaces (2017) [40]	90.7
PyramidBox (2018) [57]	94.3
DSFD (2019) [58]	95.5

Face Recognition Method	LFW Accuracy (%)
Facenet (2015) [60]	99.4
PFE (2019) [59]	99.6
Arcface (2019) [50]	99.7

Table 2.3 - Face Detection and Face Recognition SoA performance

Table 2.3 shows the performance comparison of face detection and recognition SoA methods. There is an overall good agreement with published results, with maximum deviation at 0.3%. Regarding face detection performance in WIDER FACE, the three methods achieve high average precision, especially the more recent approaches. From the ones that focus on leveraging surrounding face context, the PyramidBox is the most superior. Regarding face recognition performance in LFW, all methods perform extremely well, which may indicate both superior performance of SoA and dataset saturation.

3. Object Detection and Activity Recognition

3.1. Introduction

3.1.1. User & Functional Requirements

There are a number of user requirements which are related to this task and are being taken into consideration. In the following Table a list of all related user requirement to the Object Detection and Activity Recognition module is presented.

ID	Type	Priority	Description	PUC	KR
FR_SCE_02	Functional	MUST	7SHIELD must offer the possibility to track the movement of the physical intruder during the intrusion and afterwards. Requirement details: Passive tracking must be used (following the object inside the frame) - active tracking should be used when the camera type supports it. The UAV must be able to follow the intruder inside the Ground Segment perimeter.	PUC1(a)	KR10, KR05
FR_SCE_08	Functional	SHOULD	Visual signal when a camera (IR or normal) detects movement or moving objects (show the camera feed on main screen). Requirement details: Provide annotated video or image(s) of the detected object.	PUC1(a)	KR10, KR07
FR_SCE_11	Functional	COULD	Dashboard should show live image of the target from cameras and UAV, annotating the detected object to help user	PUC1(c)	KR20, KR07, KR05

			immediately see why there is live image.		
FR_SCE_14	Functional	SHOULD	Visual connection to area, where movement is detected through video-based recognition module. Requirement details: This should not be all movements but movements that are over some value that is hard-coded or can be changed. Should be able to change properties for different cameras, e.g., camera outside sees lots of movement done by wind etc, but camera inside server room sees only movement when someone is working there or something unusual happens. Visual connection refers to the Control Room's main screens.	PUC1(a)	KR07, KR20
FR_SCE_76	Functional	COULD	Video analytics rules: i.e., zone/line crossing detection, intrusion detection, unattended item detection.	PUC3(c)	KR07
FR_SW_22	Functional	COULD	Video image verification could be possible also under harsh weather conditions (e.g., rain, snowstorm) and in darkness.	PUC3	KR09
NFR_PERF_23	Non-Functional	MUST	Multi-tracking targets and auto target switch, with less than 1s response time. More than 30 targets simultaneously tracked.	ALL	KR07

			Requirement details: passive tracking		

Table 3.1 - User Requirements relevant to KR07

3.1.2. Objectives

This module includes two separate subtasks: a) object detection and b) activity recognition which are however closely connected to each other. Combined those two will provide the information needed to help to increase the awareness of the system.

The general objective of the first subtask of this module is to develop a framework for identifying and localizing object of interest in the provided video streams. Additionally, if the detected objects include specific objects like humans or other critical objects (e.g., cars) the results will be forwarded to the activity recognition subtask which will provide information for any suspicious or potentially harmful activity being observed.

The User Requirements created by the end users focus primarily on detecting of some sort of object's movement as in UR FR_SCE_02, FR_SCE_08, FR_SCE_14 and FR_SCE_76. All these URs require some detection of motion from objects of interest. Thus, the first step is to identify that actually objects of interest are present in the monitored scene and secondly that they are actually moving. Of course, not every movement is of the same gravity and importance, so, depending on the conditions a further analysis of the movement should follow. The users do not specifically, clarify a final list of objects of interest but only implicitly refer to some of them. An initial object list has been created which include a human general class along with the most common vehicle of transportation like cars, trucks, buses, motorcycle etc. By using this list, the system can be informed of any presence of these objects in the scenery and then further monitor their activity. Also, when a visual detection is present it can be demonstrated to the dashboard if that is needed.

After the detection of the objects of interest URs mention necessity to track the detected instances and follow them while they move around the scene. These tracking capabilities are mentioned in FR_SCE_02 and NFR_PERF_23. This requirement will be fulfilled by using a tracker which will connect the detected objects between consecutive frames so as to be able for the framework to follow the objects and analyse their movement.

3.1.3. Architecture

This module is crucial in the whole 7SHIELD architecture as its role is to provide detection information in a rather low level. The output of the module has a dual role as it is intended to be used directly by the end user as well as an intermediate output, with main purpose to be further processed by other modules.

This module takes input from cameras either fixed at some position or moving. In the first case the cameras could be some CCTV surveillance cameras which are used to monitor the

sensitive infrastructure and cover it from different point of view, angles and positions while on the second case this include UAV vehicles which will be used to patrol, follow and identify objects of interest in areas not fully covered by the aforementioned fixed position cameras.

The output of the module will be used in various other components of the whole 7SHIELD architecture. Namely, the modules that are going to utilize the results of the module are:

1. The Geospatial Complex Event Processing Engine, which is responsible for finding the correlation between geospatial events.
2. The Crisis Classification Module, which is used to determine if a crisis situation is present by analysing the available information.
3. The 7SHIELD Knowledge Base whose role is to store information to be used by other modules.
4. And the Raw Data Storage which will also store information deriving from Object Detection and Activity Recognition module but will mainly regards the ones intended for visualization purposes.

In general, as mentioned already, the output of the module is two-fold. This can be seen to the Figure 3.1. The one containing the raw output of the processing of the input data by the module and includes any inferring bounding boxes, relevant confidence scores and classes labels for objects or activities accompanied by any additional information that could be useful in order to clarify which source, geolocation and temporal time it refers to. For example, for object detector the latter additional information could include the camera the video stream derived from, the geolocation of this camera, the frame id to clarify the specific frame that was processed and the time of the detection or message sending etc. The second output include a video stream or video snapshots that will annotate the main results of the module overlayed on the original image. The latter output is intended to be used for visualization reasons mainly.

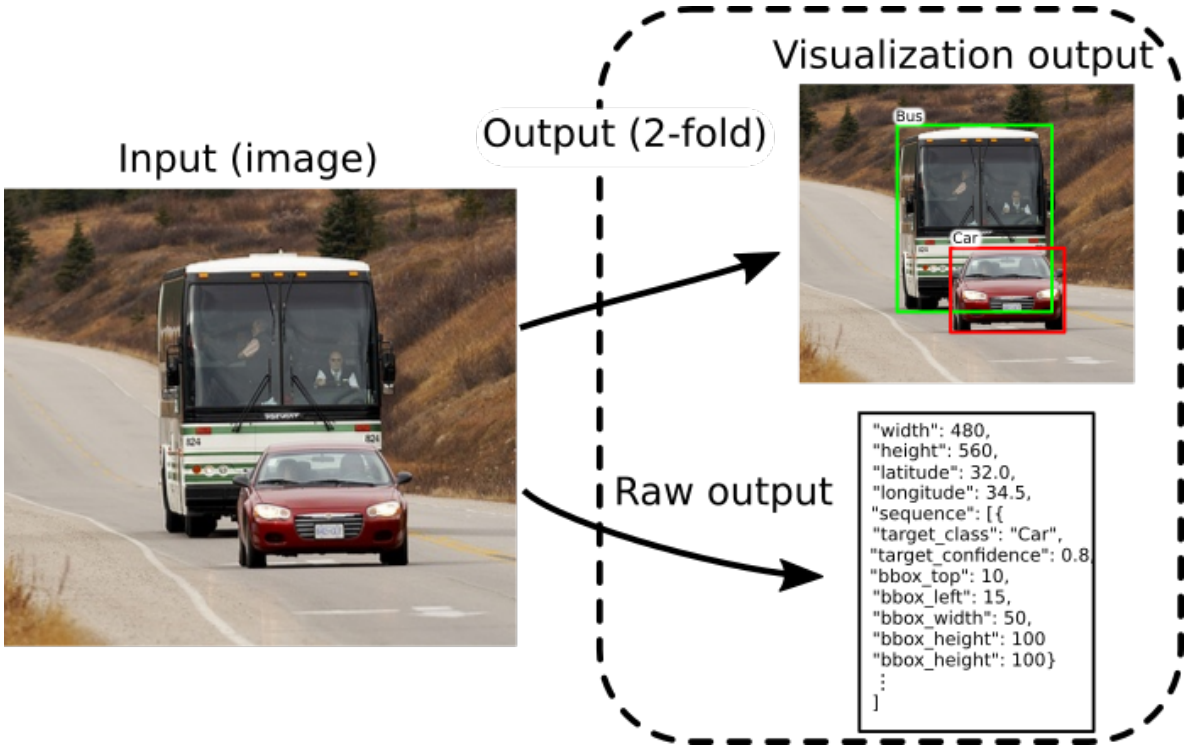

Figure 3.1 - Two-fold object detection output

3.2. Related Work

3.2.1. Image Classification

Recent achievements in machine learning have given the opportunity to achieve remarkable results in difficult [1] and abstract tasks [2] using the power of deep learning [3] which provided the mean to some breakthrough technologies. The main factor which allowed for these achievements was the introduction of CNNs. The primary task which is related to the CNNs is image classification (or recognition) which attempts to assign a label(s) which will characterize the whole image. Image classification architecture has covered a long way and includes simple (in concept) implementations like in VGG [4] or more complex ones like in ResNet family [5].

Typical and critical dataset for Image Classification is ImageNet [18] which has the intention to include so many images that could cover a huge portion of the possible real-life images. This way it was thought that it could resemble any visual task that could be taken up. As a dataset it is often used to pretrained the image classification models (or any relevant model that it relies on image classification like object detection etc.).

3.2.2. Object Detection

Closely connected to the previous task is the object detection which is used when there is also necessity for localization of the objects inside the image. As it can be seen in Figure 3.1 Image classification is useful when a) the object covers a large portion of the whole image so that the label could be considered representative of the entire image and b) when

there aren't multiple objects (of the same or even different class) inside the image. In the second subfigure of the Figure 3.1 there are 4 persons depicted which make it difficult to find a label for the entire image. Also, it is clear that besides the actual labels a different output seems reasonable: the localization of the object inside the image.

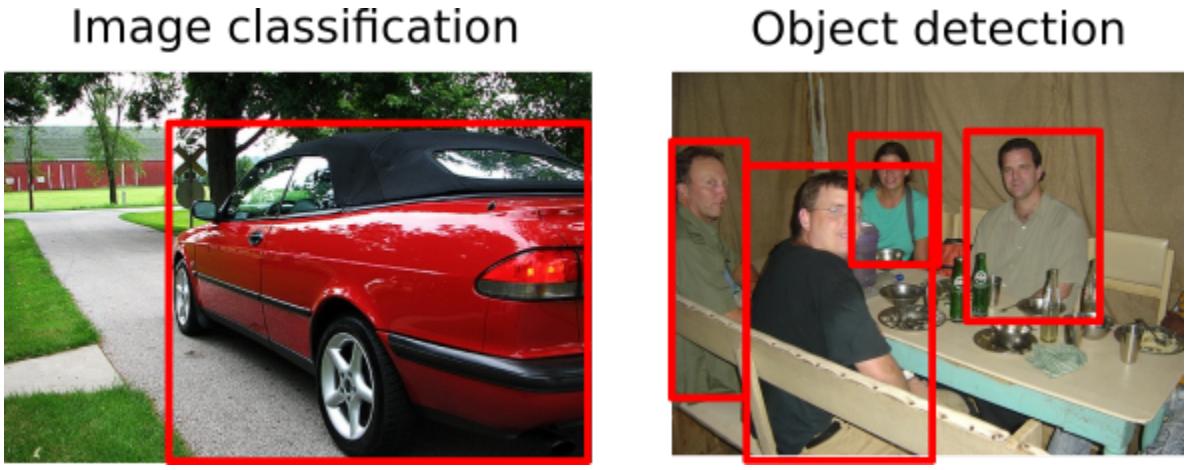

Figure 3.2 - Image classification vs Object detection

The depiction of the locations of object inside the image is performed by using coordinates, relative or absolute. The most typical approach is to use a tight bounding box around each object (as shown in Figure 3.2) and report its upper left and lower right corner coordinates. Other approaches also are valid (and may be used interchangeably with the latter ones if the necessities are better fulfilled using either type of coordinates). These include a centroid coordination system where only one absolute coordinate (the centre of the bounding box) is reported along with the dimension of the box (width and height).

The number of bounding boxes produces for each image is determined by each model and typically it cannot be easily changed afterwards. This number is around 100-300 boxes for each image with the grand majority of these prediction to not point to an actual object. So, in order to being able to judge if a prediction is probable to contain an actual object a corresponding score is also reported. This score is a floating-point number in the range 0.0-1.0, with the lower boundary (0.0) depicting a surely non-object area while the upper boundary (1.0) signifying a 100% model certainty to contain an object. This score is referred to as confidence score (among other names) and can be used to filter out useless prediction in the final output.

The object detectors can be divided into two categories judging from the use of a region proposal network (RPN). An RPN is a network that its role is to provide the system with solid bounding box propositions so as to increase the effectiveness of the system. In other words, it filters the majority of irrelevant information found inside an image and keeps only a portion of good candidate areas to feed to the rest of the model.

The earlier model approaches used the RPN since it came naturally as an extension to previous frameworks. One of the earlier solutions for object detection used a region

proposal network combined with CNNs as in Fast RCNN [6] but because of efficiency issues being witnessed, a newer end-to-end approach was shortly proposed to mitigate these problems which resulted in Faster R-CNN [7]. Another work that uses RPN is R-FCN [8] which was introduces later and tried to mitigate some of the mainly efficiency issues apparent in the first approaches. The models of this category are collectively known as two phase detectors while the models belonging to other category are known as single-phase detectors.

The second category was proposed later and was introduced in order to increase the speed of the model. The simultaneously calculation of both bounding boxes labels and coordinates seem reasonable but in practice introduced some new challenges in the accuracy field. Some fundamental works of this category include the You Only Look Once (YOLO) [9] (with its various updates) and Single Shot Detector (SSD) [10]. Although, both are single-phase detectors they primary focus on slightly different goals. The first one is famous for its speed and efficiency while not focusing to the effectiveness of the model. SSD on the other have is a more balanced approach between efficiency and effectiveness. There have been many works that further expand the previous works. For example, EfficientDet [13] uses a combination of residual networks along with bidirectional feature pyramid network (BiFPN) which aims to fuse the information from each decision feature map to all other feature map in different levels.

The presence of RPN provides two-phase detectors with a robustness and allow them to be less demanding in data augmentation in order to achieve their top performance. On the other hand, this has some effect on the efficiency the specific models can achieve. Typically, they are considered more effective but slower models in comparison with the single-phase ones.

There are several datasets used for object detection. A first division is by their initial purpose they were created for. Two of the most commonly used general purpose object detection datasets are Pascal VOC [19] and MSCOCO [20]. These datasets aim at covering common use objects from a typical point of view (although there is some diversity in both of these options inside the dataset). There are other datasets that are focusing on more specific targets, due to one or more of the following reasons:

A) The objects included in the dataset are more uncommon, e.g., weapons, boxes, specific animals etc.

B) The point of view of the dataset is not the typical first-person perspective. Examples include satellite images, drone perspective images etc.

C) The conditions of the capturing are different than daylight fair weather images. Examples in this case are low light images, foggy or misty scenes and even infra-red imagery.

A dataset that focuses on realistic scenery and diverse images is Virat [21]. The perspective is also varying from surveillance cameras to first person perspective. Another dataset that is special in its point of view is UAV123 [22] which is not an object detection dataset, but its task is tracking. Nevertheless, we have sampled the set and annotated the chosen frames in order to create a usable object detection dataset.

3.2.3. Activity Recognition

Activity recognition is the task to determine the action being executed in the specified number of frames. This means that the temporal boundaries of the activity are known (this could the first and last frame provided if the videos are cropped or the actual spatial boundaries otherwise) and the desired output is the type of action being performed. This is different than activity detection which requires to retrieve the temporal boundaries and possibly the spatial ones. The spatial ones refer to the actual coordinates of the object being involved in the activity or alternatively to the bounding box that include the area of activity.

There are several works that attempt to detect and recognize activities. On the major challenges is the difficulty to decide the temporal boundaries of the activity. This is the reason why activity recognition is considered much easier than the activity detection. One of the earlier breakthrough work works was the Two-Stream [12] where two distinct streams were used to increase the performance of the model. A different approach was used when ResNet were ported to 3D dimension to produce the 3D ResNet [14].

There are several activity recognition datasets which include miscellaneous activities. Typically, the included activities cannot be used outside their initial scope. For example, a dataset which contains activities about swimming and climbing cannot be used for recognizing running. A typical large-scale dataset is UCF101 [23] which as the name implies it contains 101 human activities. Nevertheless, it does not contain any relevant activity as walking, running or moving which is the goal for 7SHIELD project. Another dataset is HMDB-51 [24] which contains 51 actions. This one contains both walking and running but it's restricted to humans only while our interest expands to other object such as cars also. In this aspect we aim to use the object detector in conjunction with the tracker to recognize the motion of our objects of interest and implement the required activity recognition task.

3.2.4. Tracking

Between the notions of object detection and activity recognition stands the function of tracking. Indeed, an object detector typically is applied to still images and does not carry any information between frames by default. On the other hand, activity recognition by default covers the temporal nature of the video and possibly the spatial aspect also (i.e., object coordinates). A tracker has as a purpose to follow an object (or multiple objects if necessary) along the time dimension, meaning from a frame to the next one(s). It is distinct

by object detection because it does not detect any object (depicted by the bounding box) but rather simply updates the bounding box(es) it has been fed with the new coordinates they have in every frame. So, its role is complementary to the object detector, as each one has different purpose: object detector functions on still images and can produce new bounding boxes by detecting objects of interest, while the tracker can consume these initial detections and follow them on the following frames, updating their respective coordinates.

The introduction of deep learning has affected also the tracking algorithms. Nowadays the state-of-the-art for best performance is maintained by deep learning models such as GOTURN [15] which can achieve 100 fps on GPU or Fully Convolutional Networks Tracker (FCNT) [16] which can achieve 51.8 fps. The problem is that these tracking models uses the same resources as the object detector since a dedicated GPU is required to make them function to their maximum potential. For example, when deployed only on CPU the GOTURN drops to 3.9 fps and FCNT to 3.2 fps. So, we opt for a machine learning tracking algorithm which does not pose such resource demands, like Kernelized Correlation Filters Tracker (KCF) [17]. The KCF tracker can achieve up to 170 fps without the use of an GPU so it can be used complementary with the other GPU driven modules. Since, we are not focusing on training a tracker we do not intend to use any specific tracking dataset in our module.

3.2.5. Datasets

For the purposes of the project a collection of datasets has been accumulated. Considering the height, position of the fixed cameras and the expected use of UAV cameras only relevant images have been selected for the training of the algorithms. In

Dataset	Task	# images
VisDrone train set	Object detection from drone point of view	6471
VisDrone test set	Object detection from drone point of view	548
UAV123 (car-truck etc.)	Tracking (modified for object detection)	1137
UCF-Lockheed-Martin UAV Data Set	Activity recognition from drone point of view	407
Person datasets (UAV123, VIRATv1 etc.)	Person instances from drone point of view	9478

Table 3.2 a list of these accumulated images is presented.

- The **VisDrone** dataset [66] comprised of images collected from UAV perspective. It contains in total of **10,209** static images. The initial annotation included some classes that was not relevant with the tasks of 7SHIELD and thus an adjustment of the images and annotations was performed to make it available for the project's purposes.

- As mentioned before **UAV123** is a dataset focusing on tracking. Nevertheless, is contains objects captured from a low-altitude point of view and it is suitable for our object detection task.
- The **UCF-Lockheed-Martin UAV Data Set** [67] is one of the first UAV datasets created. It contains various actions captured from a 400-450 feet height. We sampled the actual videos and manually adjusted the annotated objects depicted in them.
- The **VIRATv1** is a large dataset focusing on real life videos. We have used sampled annotated images from the videos provided.

Dataset	Task	# images
VisDrone train set	Object detection from drone point of view	6471
VisDrone test set	Object detection from drone point of view	548
UAV123 (car-truck etc.)	Tracking (modified for object detection)	1137
UCF-Lockheed-Martin UAV Data Set	Activity recognition from drone point of view	407
Person datasets (UAV123, VIRATv1 etc.)	Person instances from drone point of view	9478

Table 3.2 - Object detection datasets

3.3. Methodology

3.3.1. Object Detection

For the purposes of 7SHIELD 2 models are going to be implemented and evaluated. The first one, is a robust Faster RCNN model which can provide an accurate model for object detection while the second choice is a newer EfficientDet which is a much lighter and efficient model and it's going to be used in situations where the efficiency is of the ultimate value.

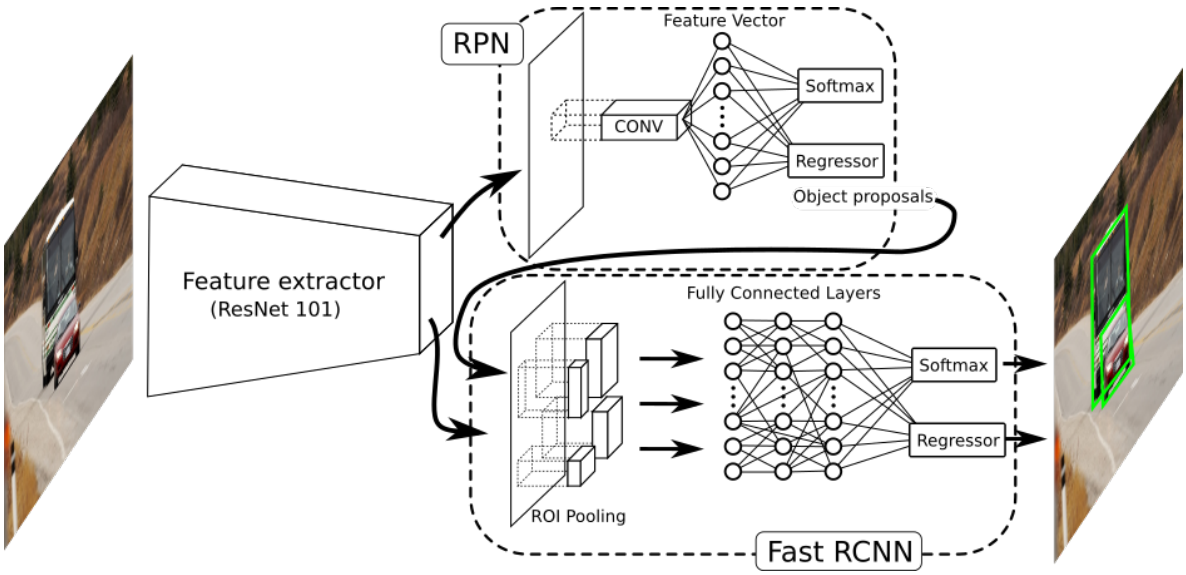

Figure 3.3 - Faster R-CNN architecture

As it has already been mentioned Faster R-CNN is a two-stage object detector which incorporates a Region Proposal Network in its framework. This additional submodel provides the whole framework with good candidates but on the other hand adds some efficiency bottlenecks.

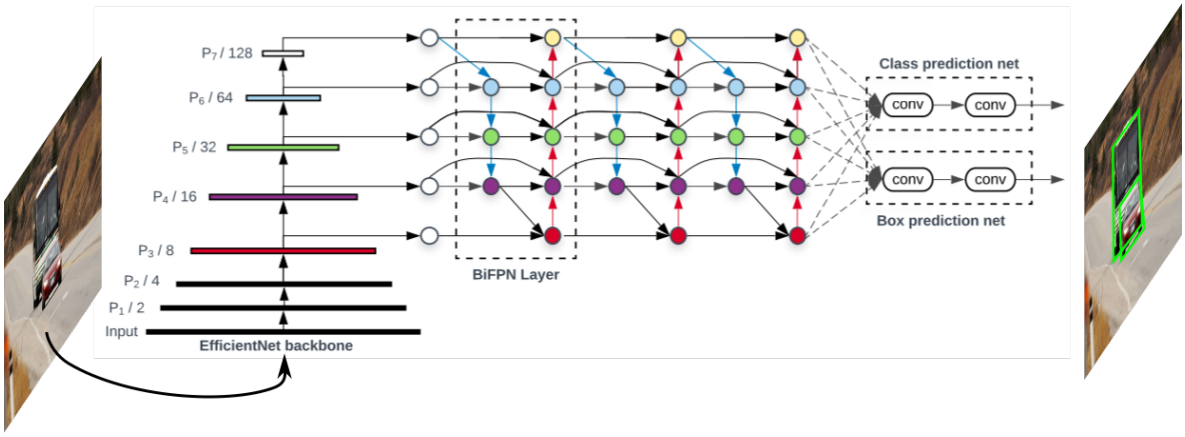

Figure 3.4 - EfficientDet architecture

For both the models the input is going to be a visual 3D image, namely WxHx3, where W and H are the width and height of the image (video) respectively. The backbone feature extractor for Faster R-CNN can be changed from the original (VGG). In our implementation we use a ResNet 101 network which is more powerful than VGG since it uses residual mechanisms to increase the discriminality of the network. As it can be seen in Figure 3.3 in Faster R-CNN there is a second subnetwork named Region Proposal Network (RPN) which is responsible for proposing bounding boxes to the model. Then essentially a Fast R-CNN [25] model is being used in order for the regression and classification for the image patches to take place. Faster R-CNN uses fully connected layers in its architecture which also affect its efficiency but produces an otherwise robust and effective model.

EfficientDet on the other hand uses only convolutional layers. It uses multiple feature maps from the backbone network (in this case is EfficientNet [26]) which fuses together in multiple passes using a bidirectional Feature Pyramid Network (Bi-FPN). After that it promotes the output to the class prediction and bounding box regression module.

3.3.2. Activity Recognition

In the aspect of 7SHIELD an object detector is already being used and the addition of an activity recognition module provides a kind of activity detection module (meaning it can detect the temporal boundaries of the activity and not only its label). Since, the assess of user requirements provided a rather limited list of activities to be tracked, namely only activities based around object movement, a solution based on object detector has been chosen which will provide the necessary information to the activity recognition module.

3.4. Experiments and Results

In order to evaluate the module some evaluation experiments have been set up. Since this module comprises of an object detector, a tracker and an activity recognizer we opt to mainly evaluate the performance of object detector as the basis of the module.

We have conducted a couple of experiments for evaluating the object detector we have implemented. This detector includes 6 classes which include the more typical vehicle of transportation for land, a general person class and a UAV class.

Object detection evaluation experiments Average Precision (AP)		
0.75330 (UAV)	0.57315 (bus)	0.75726 (car)
0.73409 (moto-bike)	0.82152 (person)	0.53351 (truck)
mAP: 0.6954		

Table 3.3 - Initial object detection results

In general, the mean AP is a little less than 70% with the lower being the truck class with 53.35% and the higher the person with 82.15%.

The tracking algorithm as mentioned will not be evaluated since no development or training is scheduled. Also, the evaluation of activity recognition has not yet been implemented since this submodule is closely related to the object detection submodule. We intend to create an evaluation tool as soon as possible though.

4. Conclusions and Future Outlook

This concludes the Deliverable 4.1 which reported on the topics related to Tasks 4.2 and 4.3, their respective Key Results, KR06 and KR07, as well as all the activities involved from the start of the project until now (M8).

Following the User Requirements report, the requirements which are related to KR06 and KR07 were first identified and then discussed in the context of the objective of each respective Task. A thorough investigation was performed and presented in the recent literature regarding related work in the tasks of Face Detection, Face Recognition, Object Detection and Activity Recognition. The initial release of the aforementioned 7SHIELD Video Surveillance Techniques was then presented, discussing the architecture of the developed modules and the motivation behind methodology choices inspired by state-of-the-art works. More specifically, the combination of DSFD [60] (for face detection) and Facenet [58] (for face recognition) were adopted for the first version of the FDR framework, due to their performance regarding accuracy and speed as well as ease of deployment. For the object detection algorithms, a robust yet heavier solution, namely, Faster R-CNN was chosen along with a lighter and more efficient one: EfficientDet. The combination of the two models seems to cover the project's requirements regarding this task. Evaluation of methods and experimental results were presented that support the applicability of the methods and provide some early performance indicators. Moreover, some early output examples and visualizations were given.

Since this is the initial release of the 7SHIELD Video Surveillance Techniques there are various issues that have been identified and will be addressed in the next iteration of this Deliverable (M18):

- Integration of video stream processing will be thoroughly investigated in addition to processing videos or images on demand. Video streams differ from offline processing and pose significant challenges. Parameters like processing speeds, frame dropping, etc. need to be assessed with caution.
- Regarding face detection and recognition, focus will be given on exploring the impact of additional parameters like the gallery size, the number of photos per person, the distance metric and the distance threshold optimization. Extensive evaluation on multiple public datasets as well as data from pilot runs will be performed to explore those topics.
- Regarding the object detection, EfficientDet is going to be trained, evaluated and examined in more real simulating situations in order to examine specific issues like the robustness of the model etc.

- Regarding the activity recognition submodule, a whole framework that is based on object detection will be also implemented and evaluated. The cooperation of the two submodules is also going to be examined extensively.

5. References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," arXiv preprint arXiv:1312.5602, 2013.

[2] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," Science, vol. 362, no.6419, pp. 1140–1144, 2018.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, pp. 436–444, 2015

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

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[6] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).

[7] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[8] Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. In Advances in neural information processing systems (pp. 379-387).

[9] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[10] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

[11] Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10781-10790).

[12] Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. arXiv preprint arXiv:1406.2199.

[13] Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning (pp. 6105-6114). PMLR.

[14] Hara, K., Kataoka, H., & Satoh, Y. (2018). Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 6546-6555).

[15] Held, D., Thrun, S., & Savarese, S. (2016, October). Learning to track at 100 fps with deep regression networks. In European conference on computer vision (pp. 749-765). Springer, Cham.

[16] Wang, L., Ouyang, W., Wang, X., & Lu, H. (2015). Visual tracking with fully convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 3119-3127).

[17] Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. (2014). High-speed tracking with kernelized correlation filters. IEEE transactions on pattern analysis and machine intelligence, 37(3), 583-596.

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

[19] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2), 303-338.

[20] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

[21] Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C. C., Lee, J. T., ... & Desai, M. (2011, June). A large-scale benchmark dataset for event recognition in surveillance video. In CVPR 2011 (pp. 3153-3160). IEEE.

[22] Mueller, M., Smith, N., & Ghanem, B. (2016, October). A benchmark and simulator for uav tracking. In European conference on computer vision (pp. 445-461). Springer, Cham.

[23] Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402.

[24] Jhuang H, Garrote H, Poggio E, Serre T, Hmdb T (2011) A large video database for human motion recognition. In: Proceedings of IEEE international conference on computer vision

[25] Wang, X., Shrivastava, A., & Gupta, A. (2017). A-fast-rcnn: Hard positive generation via adversary for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2606-2615).

[26] Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001 (Vol. 1, pp. I-I). IEEE.

[27] Li, J., Wang, T., & Zhang, Y. (2011). Face detection using surf cascade. Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference On, 2183–2190.

[28] Shu, C., Ding, X., & Fang, C. (2011). Histogram of the oriented gradient for face recognition. Tsinghua Science and Technology, 16(2), 216–224.

[29] Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence, (12), 2037–2041.

[30] Zhu, X., & Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference On, 2879–2886.

[31] Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., & Kumar, N. (2013). Localizing parts of faces using a consensus of exemplars. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(12), 2930–2940.

[32] Sun, Y., Wang, X., & Tang, X. (2013). Deep convolutional network cascade for facial point detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3476–3483.

[33] Wolf, L., Hassner, T., & Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference On, 529–534.

[34] Jiang, H., & Learned-Miller, E. (2017). Face detection with the faster R-CNN. Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference On, 650–657.

[35] Sun, X., Wu, P., & Hoi, S. C. H. (2018). Face detection using deep learning: An improved faster RCNN approach. Neurocomputing, 299, 42–50.

[36] Yang, S., Luo, P., Loy, C.-C., & Tang, X. (2015). From facial parts responses to face detection: A deep learning approach. Proceedings of the IEEE International Conference on Computer Vision, 3676–3684.

[37] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499–1503.

[38] Najibi, M., Samangouei, P., Chellappa, R., & Davis, L. S. (2017). SSH: Single Stage Headless Face Detector. ICCV, 4885–4894.

[39] Ranjan, R., Patel, V. M., & Chellappa, R. (2017). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[40] Hu, P., & Ramanan, D. (2017). Finding tiny faces. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 951–959.

[41] Zhang, J., Wu, X., Hoi, S. C. H., & Zhu, J. (2020). Feature agglomeration networks for single stage face detection. Neurocomputing, 380, 180–189.

[42] Cao, Z., Yin, Q., Tang, X., & Sun, J. (2010). Face recognition with learning-based descriptor. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2707–2714.

[43] Chen, D., Cao, X., Wen, F., & Sun, J. (2013). Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3025–3032.

[44] Huang, C., Zhu, S., & Yu, K. (2012). Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval. ArXiv Preprint ArXiv:1212.6094.

[45] Simonyan, K., Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2013). Fisher vector faces in the wild. BMVC, 2(3), 4.

[46] Chopra, S., Hadsell, R., LeCun, Y., & others. (2005). Learning a similarity metric discriminatively, with application to face verification. CVPR (1), 539–546.

[47] Zhu, Z., Luo, P., Wang, X., & Tang, X. (2014). Recover canonical-view faces in the wild with deep neural networks. ArXiv Preprint ArXiv:1404.3543.

[48] Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1701–1708.

[49] Sun, Y., Wang, X., & Tang, X. (2015). Deeply learned face representations are sparse, selective, and robust. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2892–2900.

[50] Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4690–4699.

[51] Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., & Song, L. (2017). Sphereface: Deep hypersphere embedding for face recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 212–220.

[52] Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., … Liu, W. (2018). Cosface: Large margin cosine loss for deep face recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5265–5274.

[53] Yang, S., Luo, P., Loy, C. C., & Tang, X. (2016). Wider face: A face detection benchmark. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5525-5533).

[54] Jain, V., & Learned-Miller, E. (2010). Fddb: A benchmark for face detection in unconstrained settings (Vol. 2, No. 4, p. 5). UMass Amherst technical report.

[55] Yi, D., Lei, Z., Liao, S., & Li, S. Z. (2014). Learning face representation from scratch. arXiv preprint arXiv:1411.7923.

[56] Maze, B., Adams, J., Duncan, J. A., Kalka, N., Miller, T., Otto, C., ... & Grother, P. (2018, February). Iarpa janus benchmark-c: Face dataset and protocol. In 2018 International Conference on Biometrics (ICB) (pp. 158-165). IEEE.

[57] Tang, X., Du, D. K., He, Z., & Liu, J. (2018). Pyramidbox: A context-assisted single shot face detector. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 797-813).

[58] Li, J., Wang, Y., Wang, C., Tai, Y., Qian, J., Yang, J., ... & Huang, F. (2019). DSFD: dual shot face detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5060-5069).

[59] Shi, Y., & Jain, A. K. (2019). Probabilistic face embeddings. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 6902-6911).

[60] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).

[61] Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., & Yan, S. (2017). Perceptual generative adversarial networks for small object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1222-1230).

[62] Ozge Unel, F., Ozkalayci, B. O., & Cigla, C. (2019). The power of tiling for small object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 0-0).

[63] Tong, K., Wu, Y., & Zhou, F. (2020). Recent advances in small object detection based on deep learning: A review. Image and Vision Computing, 97, 103910.

[64] Zhang, F., Fan, X., Ai, G., Song, J., Qin, Y., & Wu, J. (2019). Accurate face detection for high performance. arXiv preprint arXiv:1905.01585.

[65] Chi, C., Zhang, S., Xing, J., Lei, Z., Li, S. Z., & Zou, X. (2019, July). Selective refinement network for high performance face detection. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 8231-8238).

[66] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Qinghua Hu, Haibin Ling. Vision Meets Drones: Past, Present and Future. arXiv preprint arXiv:2001.06303 (2020).

[67] https://www.crcv.ucf.edu/data/UCF_Aerial_Action.php

Horizon 2020
European Union Funding
for Research & Innovation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 883284