UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

**DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE**

**CORSO DI LAUREA MAGISTRALE IN INGEGNERIA DELL'AUTOMAZIONE**

# "MARKERLESS MOTION CAPTURE VIA CONVOLUTIONAL NEURAL NETWORK"

**Relatore: Prof. Alessandro Chiuso**
**Correlatore: Prof.ssa Zimi Sawacha**

**Laureando: Niccolò Monaco**

**ANNO ACCADEMICO 2021 – 2022**

**Data di laurea: 5 ottobre 2022**

# Abstract

A human motion capture system can be defined as a process that digitally records the movements of a person and then translates them into computer-animated images.
To achieve this goal, motion capture systems usually exploit different types of algorithms, which include techniques such as pose estimation or background subtraction: this latter aims at segmenting moving objects from the background under multiple challenging scenarios.
Recently, encoder-decoder-type deep neural networks designed to accomplish this task have reached impressive results, outperforming classical approaches.
The aim of this thesis is to evaluate and discuss the predictions provided by the multi-scale convolutional neural network FgSegNet_v2, a deep learning-based method which represents the current state-of-the-art for implementing scene-specific background subtraction.
In this work, FgSegNet_v2 is trained and tested on BBSoF S.r.l. dataset, extending its scene-specific use to a more general application in several environments.

# Sommario

Un sistema di motion capture per soggetti umani può essere definito come un processo digitale che registra i movimenti di tali soggetti e li converte in animazioni computerizzate. Per raggiungere tale scopo, i sistemi di motion capture generalmente utilizzano diversi tipi di algoritmi che includono tecniche quali pose estimation o background subtraction: quest'ultima mira a segmentare e separare gli oggetti in movimento dallo sfondo in diversi scenari difficili.
Recentemente, le reti neurali di tipo encoder-decoder progettate per questo obiettivo hanno raggiunto risultati impressionanti, surclassando gli approcci classici.
Lo scopo di questa tesi è di analizzare e valutare le previsioni fornite dalla rete neurale convoluzionale FgSegNet_v2, un metodo basato sul deep learning che costituisce l'attuale stato dell'arte per l'implementazione della tecnica di background subtraction per scene specifiche.
Il modello è stato allenato e testato sul dataset di BBSoF S.r.l., estendendo il suo utilizzo per scene specifiche ad un'applicazione più generale in diversi ambienti.

# 1. Motion Capture

Motion capture is defined as the process of digitally tracking and recording the movements of objects or living beings in space.
Thanks to their versatility, motion capture systems are implemented in a wide range of applications, such as healthcare and clinical settings, sports (postural efficiency analysis for improving athletes' performance), smart surveillance systems, industrial settings like entertainment, gaming industry, sectors of robotics and automotive applications [19].
Human motion tracking can be formulated as a keypoint-based pose estimation problem (especially for biomechanical applications), since several studies have shown that human motion can be defined by a few coherently moving points [14][15], or as a silhouette-based background subtraction method.
The aim of this work is to evaluate an existent robust deep learning-based model for background subtraction to be integrated with BBSoF S.r.l. markerless motion capture system, whose application involves two main scenarios [9]:

- scheduling customized training sessions depending on motion needs detected on tested athletes
- optimizing rehabilitation process, establishing recovery time, and guaranteeing to the athlete a safe return into practicing competitive activity in case of lower limbs injuries

The thesis is divided into four chapters, where the first one describes the literature of human motion capture in the context of biomechanics applications, including marker-based and markerless methods for pose estimation and silhouette-based methods for background subtraction. Furthermore, the focus is put on the variety of deep learning-based methods expressly designed for background subtraction.
The second chapter describes the architecture of FgSegNet_v2 and reports the details of its training phase.
The third chapter includes the definition of the experimental setup (i.e., test frame dataset and chosen evaluation metrics) and some intermediate (i.e., feature maps) and final (i.e., foreground masks) results.
Finally, a brief discussion and some theoretical conclusions are included in the last section of this work.

## 1.1 Human Pose Estimation

A human pose estimation algorithm is a function able to map video frames into 2D or 3D coordinates of body parts.
Objects of interest are represented by collections of pixels, which include keypoints that are associated to semantic meaning (such as joints or other body parts): the connection of these pixels provides visual data such as a skeleton model or a graph.
As an additional component, human pose estimation algorithms can integrate temporal information into processed data.

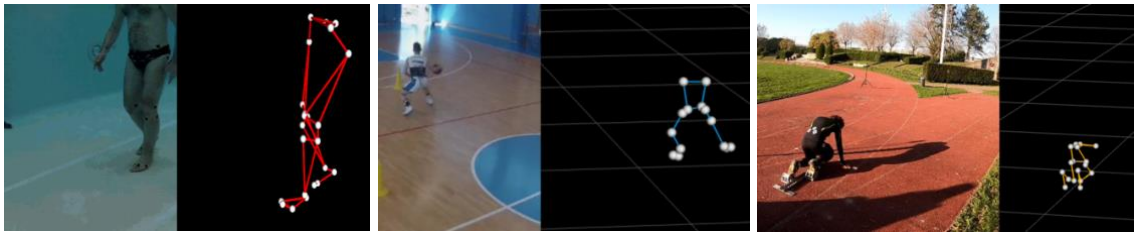

**Fig. 1.1**: Examples of human pose estimation algorithm implementations

However, human pose estimation systems may face several non-trivial challenges, such as variations in body shape, clothing, lighting issues, and partial or full joint occlusions, which may lead to inaccurate predictions.
Most importantly, the research in deep learning-based methods for this application has been motivated by the need to explore and detect all possible natural poses, a task that previous approaches and even deep learning models cannot fully accomplish because of failure in atypical postures generalization (such as yoga poses) with respect to the training set.

**Fig. 1.2**: Examples of challenging scenarios for human pose estimation task: in the left image, the right arm is occluded, while in the right image the left body half is not visible

During the last 10 years, deep learning-based methods have approached this task by learning low and high-level human body features, developing the ability to capture the full context of each joint and becoming more tolerant to variations in the training set with respect to classical approaches. In addition to that, these methods can follow holistic reasonings, and therefore predict the location of an occluded joint or limb by exploiting the visible parts of the pose and anticipating subject's motion [36].

On the other hand, incorporating priors about human body structure in deep learning models is a difficult task since their low-level mechanisms are often hard to interpret [34].

A huge advantage in using deep learning-based methods for motion capture is given by the great flexibility of these ones, which allow the user to define the points of interest that should be tracked: features are directly extracted from the original image and passed through layers to obtain high-dimensional information of the processed frame [35].

Moreover, the dataset used for training the model defines the input-output relationships that this one should learn: the learning process consists in the iterative update of model's parameters (called "weights") for the minimization of the chosen loss function.

The keypoints estimation task can be achieved following two completely different approaches: it can be formulated as a regression problem, where the deep learning model is trained to track body parts (whose coordinates are the targets to predict) or, alternatively, it can be treated as a classification problem, where the model predicts a heatmap (scoremap) of location probabilities for each body part rather than key joint locations on the human skeleton [14].

**Fig. 1.3**: Examples of heatmaps for joint localization

### 1.1.1 Marker-based and markerless approaches

Deep learning-based methods for motion capture can be divided into two categories: model-based tracking algorithms and feature-based tracking algorithms [38]. The first ones perform the tracking task using as reference a 3D model of the object to be tracked, while the latter use points of interest in the frames for object(s) tracking.

Furthermore, feature-based tracking algorithms can be distinguished into two different approaches, namely marker-based and markerless motion capture systems: the first one requires the attachment of markers to the body's segments, while the second one directly maps raw video frames to coordinates.
Both approaches aim to identify approximately well 2D or 3D human joint locations using 2D frames coming from a monocular camera or a synchronized multi-view camera system [22]: the 3D segmented model is constructed exploiting the projection of subject's detected pose from each camera.

**Fig. 1.4**: Examples of marker-based (first image) and markerless motion capture approaches

Usually, marker-based systems are more accurate than markerless ones, and markers represent useful tools for visual detection algorithms: they are manually placed by expert operators on anatomical points of interest for visually enhancing them and tracking their motion while processing sequencies of frames.
Position and movement of markers are used to infer the relative movement between two adjacent segments (e.g., anterior cruciate ligament, ACL), with the goal of accurately defining the joint movement.
On the other hand, markers could modify the naturalness of a subject's movement and, more importantly, skin deformation and displacements may cause dislocations between a marker and the underlying bone it is labeling: this latter issue, known as skin artifact or soft tissue artifact, represents the major obstacle to obtain accurate joint kinematics estimations and may potentially invalidate injury risk detection [18].
Moreover, it is not possible for marker-based methods to extract additional keypoints at a later stage, which is instead achievable by markerless motion capture methods [14].
Another limitation related to marker-based methods application is the necessity of subjects to wear skin-tight clothing [10].

The use of markerless motion capture methods, particularly for clinical applications, has been limited by the complexity of acquiring accurate 3D kinematics: without the use of markers, the general problem of motion estimation has less constraints, leading to the construction of a model with a high number of parameters or degrees of freedom (DOF), increasing the risk of overfitting.

However, this issue can be tackled by introducing an a priori model of the subject that incorporates limb orientations and shape, providing therefore anthropomorphic constraints that are automatically satisfied when processing input data. Analogously, the number of recording cameras can be increased and the space of possible poses can be limited to the anatomically appropriate ones.

Figure 1.6 reports an example of model designed for studying musculoskeletal biomechanics via markerless motion capture [40]: it is composed of 12 main anatomical segments (i.e., feet, shanks, thighs, pelvis, combined torso and head, arms, and forearms) and characterized by 33 degrees of freedom, divided into:

- 3 DOF in rotation for hip and knee joints (i.e., flexion-extension, adduction-abduction, internal-external rotation)
- 2 DOF in rotation for ankles (plantar-dorsi flexion, in-eversion)
- 1 DOF in rotation for movement between torso and pelvis (flexion at the 5$^{th}$ lumbar)
- 3 DOF in rotation for shoulders (flexion-extension, adduction-abduction, internal-external rotation)
- 2 DOF for elbows (flexion-extension, pronation-supination)
- 6 DOF for pelvis translation and rotation in space

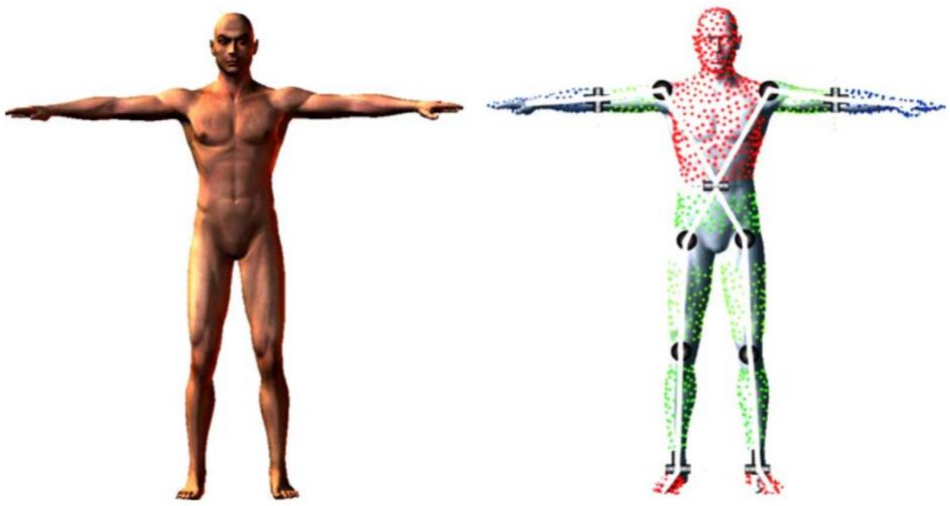

**Fig. 1.5**: Model in reference pose and 33 DOF full body model

Neural networks for articulated human pose estimation are typically designed as encoder-decoder architectures: a common strategy consists in integrating encoder models which were pre-trained on one or more huge image datasets (such as ImageNet).
This design choice, which dramatically reduces the required data amount for the training phase, comes from the principles of transfer learning, a machine learning branch which focuses on gaining information while solving a specific task and then applying the stored knowledge to a related (but different) problem [37].
In the deep learning-based methods literature, there have been identified two main approaches to accomplish the markerless pose estimation task: the first one localizes one or more individuals and then performs pose estimation per detected individual (top-down method), while the other one localizes body parts and uses a pre-trained network to predict their connections within individuals (bottom-up method) [22].
As well as keypoint-based human pose estimation, markerless motion capture can also be achieved by tracking the silhouette, namely separating the subject(s) of interest from the background.
A common method for accomplishing this task is by applying the technique of background subtraction, whose aim, methods in literature and implementations are described in Section 1.2.

### 1.1.2 Applications in biomechanics

In the biomechanics field, pose estimation algorithms represent optimal tools for the investigation and detailed analysis of big topics such as functional mechanisms, injury prevention, rehabilitation, and motor control of human movements [22].
The backbone of several applications implemented in these areas is the classification of movement patterns, which allows researchers to focus onto the distinction of pathological kinematics from normal ones.
More specifically, these algorithms are mainly used for the diagnosis of pathomechanics (i.e., kinematics of misplaced or damaged bones) due to musculoskeletal diseases, preventive interventions for musculoskeletal diseases, and development and evaluation of rehabilitative treatments [16].
A common method for solving tasks in clinical context is gait analysis, namely the systematic visual study of human walking, augmented by the use of instrumentation for measuring body mechanics to assess and compare human movement patterns into a variety of health factors [17]. This method allows to extract spatiotemporal parameters (e.g., gait speed, step length,

stride width, etc.), which represent useful clinical measures for detecting either negative (due to pathologies) or positive (thanks to rehabilitation processes) changes in subjects' gait patterns [33].

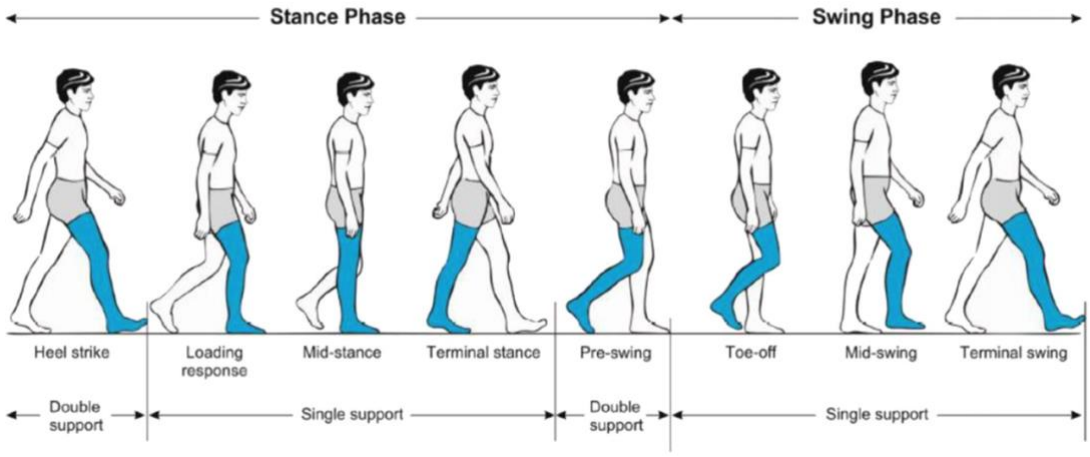


**Fig. 1.6**: Gait cycle phases

The current gold standard methods for gait analysis are marker-based motion capture systems, which can be divided into two categories: active marker systems and passive marker systems. The first ones employ markers that contain a source of light (generally infrared rays) which can be detected by sensors, while the latter use markers that reflect light back to sensors. Despite their accuracy, marker systems use in clinical routines is limited due to several reasons, such as labor costs for data acquisition and processing, equipment costs and the need for a controlled laboratory environment [21].

A more difficult scenario for motion capture systems is represented by sports applications, where data analysis often has to deal with highly dynamic motions, that are more difficult to capture than slow movements (such as gait analysis). Moreover, selecting the most suitable system for the given experimental setup can be a labor-intensive and difficult task: for example, the necessity of sampling at high frequency may cause an excessive amount of data and information corruption due to high-frequency noise, and data acquisition may require expensive high-speed and high-quality cameras [38].

The development of machine learning and deep learning-based methods for biomechanical applications plays a fundamental role in advancing human movement research, improving clinical decision making and accelerating rehabilitation processes for patients affected by neuromuscular and musculoskeletal diseases [39].

## 1.2 Background Subtraction

Background subtraction, also known as foreground segmentation, is a common and widely used technique to implement motion detection algorithms.

During the last three decades, it has been one of the most active research topics in computer vision, owing to many applications such as pedestrian detection, traffic monitoring, action recognition and industrial machine vision [8].

As the name suggests, this technique performs a subtraction between the processed frame and the so-called background model (i.e., the static part of the scene or, more in general, everything that can be considered as part of the background) to generate a foreground mask (namely, a binary image containing the pixels that belong to the motion region) by using static cameras [7][23].

A complete foreground segmentation technique has therefore four main components: a background initialization process, a background modeling strategy, a model update mechanism (to add new static objects in the scene to the background model [28]) and a subtraction operation [11].

In addition, a well-performing background subtraction algorithm must be able to capture significant visual changes in subsequent video frames, while neglecting noise-produced disturbances [27].

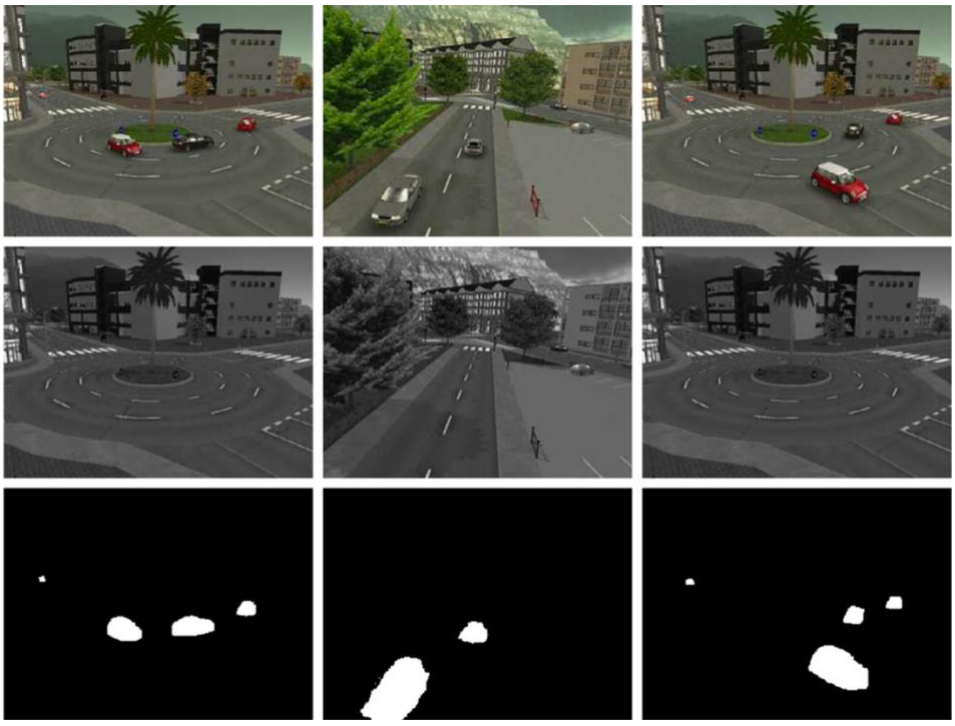

**Fig. 1.7**: Examples of background subtraction operations: the first row illustrates the original frames, the second one shows the background models and the third one reports the foreground masks

### 1.2.1 Challenging scenarios

Challenging scenarios for a background subtraction algorithm include illumination changes (either sudden or gradual), shadows labeled as part of the foreground object, dynamic background motion (due to atmospheric conditions such as rain, snow, or waving trees), camera motion (jittering and panning-tilting-zooming), and subtle regions (i.e., similarities between foreground pixels and background pixels).

Despite conventional approaches have been designed to eliminate or heavily mitigate all the mentioned issues, these methods perform well only on some specific scenarios and lack the problem in a general setting [2]. In addition to that, the constant improvement of camera technologies is leading to videos increase in complexity, size, and numbers [27]: this latter issue shows that it is necessary to build background subtraction models that can efficiently handle big data.

Not surprisingly, neural networks became very useful methods to address such challenges: models are trained on huge datasets of frames taken from specific-issued videos to effectively learn how to deal with the encountered challenges and then generalize to more complex scenarios.

### 1.2.2 Literature

Based on mathematical theories, the simplest way to obtain a background image is to compute the temporal average or the temporal median of a given set of frames belonging to the same scene. More practically, background subtraction implementations involved a big variety of models, going from signal processing to machine learning.

These models can be classified as:

- mathematical models, including statistical models (based on Gaussian mixtures, GMMs), fuzzy models and Dempster-Schafer models
- filter-based signal processing models (Wiener filter, Kalman filter, maximum correntropy Kalman filter and Chebycev filter)
- machine learning and deep learning-based models, such as subspace learning models (reconstructive, discriminative, or mixed), robust PCA subspace learning, support vector machines and neural networks

As previously described, conventional approaches rely on building a background model for a specific scene by using statistical or parametric methods, which implement mixtures of Gaussians (either with fixed or dynamical number of components) to model each pixel as a background or foreground pixel.

This strategy has been enhanced and improved by machine learning and deep learning-based methods: foreground segmentation is indeed formulated as a pixel-based binary classification problem, where background (and foreground) pixel representation can be learnt in a supervised or unsupervised manner.

Machine learning models developed for this specific task implement algorithms that are mainly based on kernel density estimates (KDE) or principal component analysis (PCA): the main advantage provided by these approaches consists in their ability to tackle the issue of parameters estimation and update in GMMs.

The first ones follow a probabilistic approach for scene modeling: it is estimated the probability density function of a pixel value by analyzing a set of consecutive frames and using the Normal distribution $N(0, \Sigma)$ as kernel estimator function. Efficient implementations of these methods include additional parameters such as adaptive kernel size and a decaying factor in the model update mechanism to reduce older samples influence.

On the contrary, PCA methods aim to build a background model by decomposing the input frames into a time-variant low-rank subspace: the distinctive characteristic of these methods is the significant reduction of the space dimensionality, which allows to speed up computations while achieving satisfying accuracy values.

However, these approaches do not perform well in scenarios where background pixels constantly change (due to non-static lightning conditions or changes of camera angle) or where foreground pixels remain unchanged (problem known as intermittent object motion).

The great impact that deep learning-based methods had in solving background subtraction tasks has been favored by the possibility to develop end-to-end systems for image segmentation: not surprisingly, deep neural networks (DNNs) and deep convolutional neural networks (CNNs) have achieved impressive results, outperforming classical approaches.

Still nowadays, CNNs are considered the best tools for solving computer vision tasks such as object detection, feature extraction, and scene classification. As aforementioned, these models are fed with 2D data, which are processed and gradually downsampled while the number of learnt features increases: this learning mechanism (that characterizes CNNs) is based on the implementation of strided-pooling layers (like max-pooling or average-pooling layers), which reduce the spatial resolution by computing a summary statistic over a local spatial region. The main purpose behind the use of these modules is to promote invariance to local input transformations [6].

Despite CNNs have been a research topic for a long time, their use for accomplishing computer vision tasks has been initially limited by factors such as the size of available labeled training datasets or the models' computational capacity [8][13]. More recently, these models

have reached great success in large-scale image and video recognition thanks to the implementation of huge public image repositories (e.g., ImageNet [25]) and high-performance computing systems like GPUs [3].

### 1.2.3 ConvNet

The first convolutional neural network specifically designed for background subtraction, which led to a turning point for the solution of this task, was realized in 2016 and took the name of ConvNet [11].

The core of this work, which is still considered a cornerstone, was to demonstrate that the complexity of foreground segmentation task could be addressed during subtraction operation, instead of requiring a complex background modeling strategy. More in detail, the background is obtained by computing the temporal median of few video frames and is further modeled within a single grayscale image, while the task of subtracting background image from input frame is delegated to the neural network.

The main benefit of this method consists in the ability of the network to learn deep and hierarchical features, which turns out to be a more efficient strategy than the extraction of hand-crafted features for comparing image patches (strategy implemented by conventional approaches).

This model has similar architecture to LeNet5 [12], a popular neural network used for handwritten digit classification: ConvNet is composed of two feature stages followed by a classical two-layer fully connected feed-forward neural network, where each feature stage consists of a convolutional layer followed by a max-pooling module.

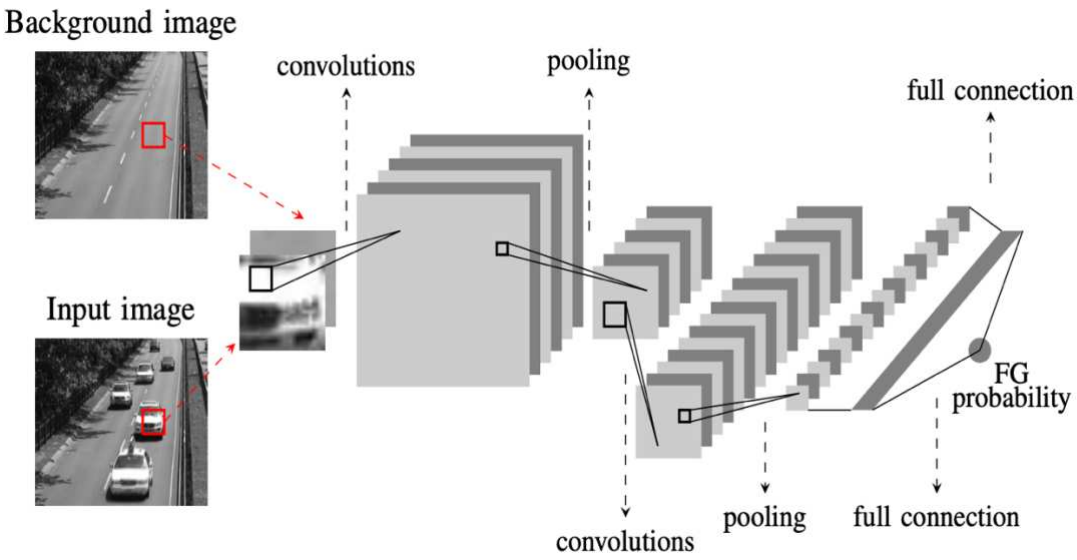


**Fig. 1.8**: FgSegNet_v2 architecture workflow

This model involves four stages in foreground masks generation:

1) background image extraction (by using a temporal gray-scale median)
2) specific-scene dataset generation
3) network training
4) background subtraction

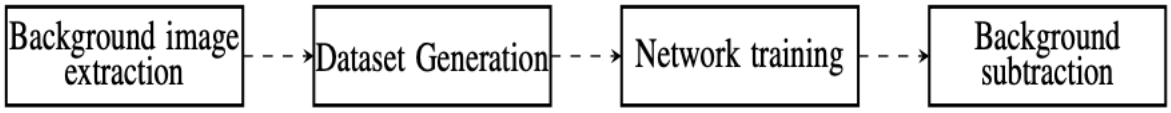


**Fig. 1.9**: ConvNet algorithm pipeline

More specifically, this model uses a patch-wise training strategy: for each frame in a specific scene, image patches are extracted and combined with the corresponding background patches. Then, combined patches are fed to the neural network to predict the foreground probabilities of their center pixels.

Despite its meaningful impact, this first deep learning application for foreground segmentation has several limitations, such as computational inefficiency and possible overfitting due to highly redundant data. In addition to that, foreground masks may contain isolated false positives and false negatives since each pixel is processed independently.

From the publication of this work, many studies have been conducted in the field of deep learning applications to background subtraction, demonstrating the efficiency of these methods. Moreover, DNNs have also been employed in complementary tasks, such as background initialization and foreground detection enhancement [1][8].

### 1.2.4 Deep learning-based approaches

Further works developed CNNs following different types of approach to solve the background subtraction task (e.g., by incorporating temporal data and building a background dynamical model, or by classifying pixels as part of the foreground mask or as background components). Therefore, a big variety of a models have been designed and tested, allowing comparisons between divergent approaches, and highlighting the most relevant characteristics for designing a task-specific neural network and improving several aspects of its performance. These models' architectures include multi-scale and cascaded CNNs, fully connected CNNs, deep CNNs, double encoding CNNs and even generative adversarial networks (GANs).

More in detail, one of the first approaches (following the guideline outlined by ConvNet) consists of a CNN designed for vehicle detection and classification in low resolution traffic

videos [30], able to detect vehicles and distinguish their type among 6 different categories (i.e., jeep, sedan, truck, bus, SUV, and van).

Another work consists of and encoder-decoder structured CNN [26] which takes as input the concatenation of the target frame, its previous frame, and the correspondent background model: then, the encoder generates a feature vector of the given images, and finally the decoder converts this vector into a segmentation map.

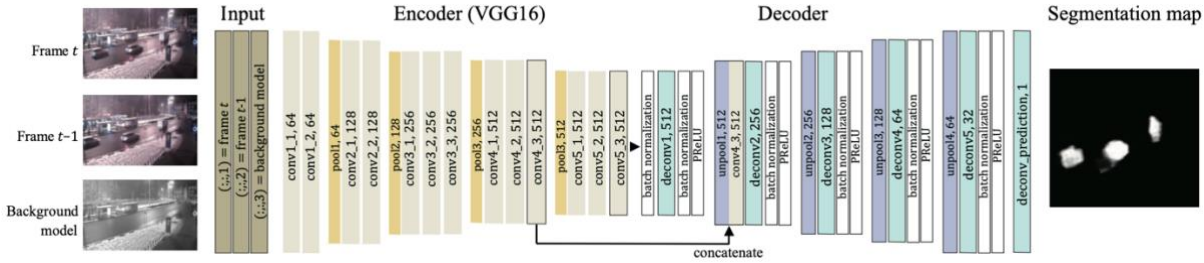


**Fig. 1.10**: Background subtraction encoder-decoder structured CNN architecture

A different temporal-based strategy involves the implementation of a multi-scale CNN [27] for tracking visual changes in a video sequence by applying convolutions to the most recent frames. This network can handle multiple scenarios without performing scene-specific fine-tuning and does not need to build and update a background model.

An important study includes the use of a highly accurate semi-automatic method for segmenting foreground moving objects pictured in surveillance systems [29]; this work aims to pursue two main objectives: produce foreground masks sufficiently accurate to be used as ground truth for other models and minimize user interventions.

Moreover, this research includes a set of metrics values that define human annotation error margin in foreground objects segmentation, i.e., a F-measure of 0.94, a percentage of wrong classifications (PWC) of 0.9 and an error distance of 3.6 pixels.

Another important conclusion that this study provides is that multi-scale CNNs with a cascaded architecture are the best performing models for background subtraction task.

Finally, it is necessary to mention BScGAN [31], a deep foreground segmentation method based on a conditional generative adversarial network (cGAN): this type of neural network learns statistical invariant features of input frames to generate similar images.

The model consists of two subsequent networks, namely generator and discriminator, which compete in a game theoretic scenario. More in detail, the generator's aim is to learn the mapping function that associates inputs (i.e., target frame and background) to the output (i.e., foreground mask) to produce predictions that cannot be distinguished from ground truth images.

Then, the discriminator learns a loss function to train this mapping mechanism by comparing generator's prediction (fake foreground) with ground truth mask (real foreground).
As noticeable, this approach addresses foreground detection as a segmentation problem (and not as a classification task), where the segmentation operation is carried out by the generator.

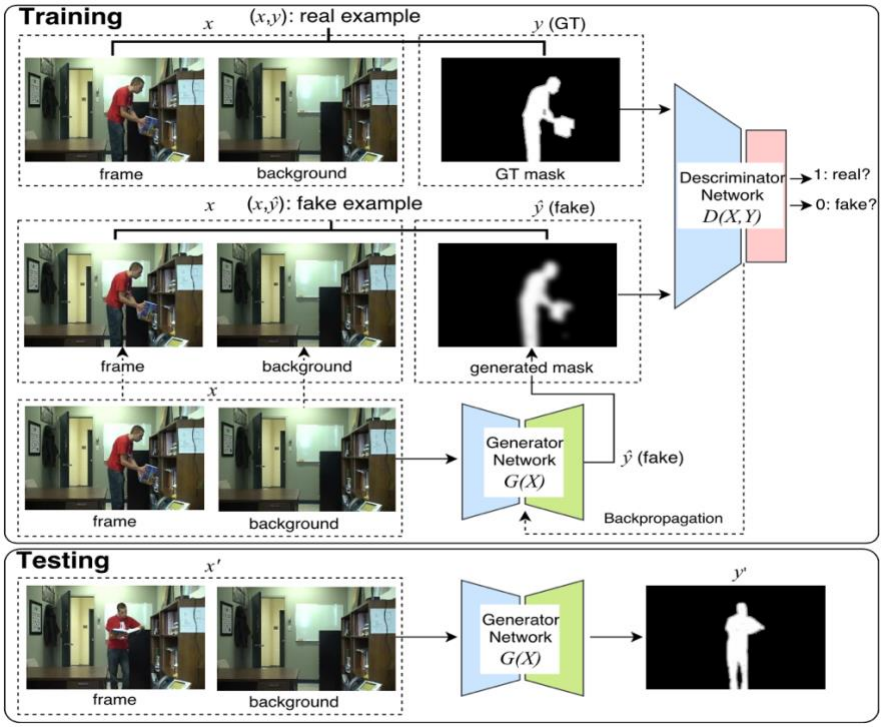


**Fig. 1.11**: Background subtraction conditional generative adversarial network

# 2. Multi-scale Convolutional Neural Network for Motion Capture

In this section there are reported the architecture and the training phase details of the implemented tool used for achieving background subtraction: the multi-scale convolutional neural network FgSegNet_v2 [1].

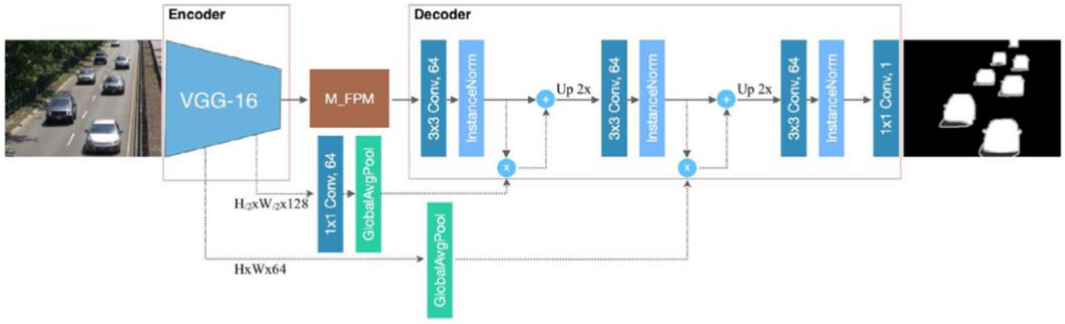


**Fig. 2.1**: FgSegNet_v2 architecture workflow

This model addresses the foreground segmentation task as a supervised pixel binary classification problem (i.e., pixels belong to foreground mask or to background).
The choice of implementing this neural network comes from the fact that it represents the state-of-the-art method for the background subtraction task, showing impressive results in terms of classification accuracy (i.e., an average overall F-measure of 0.9874 on CDnet2014 [32] datasets) without the need to incorporate temporal data.
As a deep learning-based method for solving background subtraction, FgSegNet_v2 does not rely on building a stationary background model, since this method has been shown to be not very effective in adapting to difficult scenarios such as rapid illumination changes, shadow detection, dynamic background motion and/or camera motion.
On the contrary, this specific network and other CNNs based on gradient learning demonstrated to be very powerful tools in extracting useful features representations from data.

## 2.1 Architecture

The network is constructed following an encoder-decoder architecture, trained end-to-end and characterized by three main components: an encoder, a feature pooling module (M-FPM, implemented to extract features at multiple scales) and a decoder with Global Average Pooling (GAP) modules [1].

The encoder consists of a modified version of VGG-16 Net [3], a popular neural network originally designed for object classification and then fine-tuned for other visual tasks, such as object detection or semantic segmentation [26]: this component has the main function of extracting visual features from the input image.

Then, M-FPM extracts features at multiple scales by applying parallel dilated convolutions.

Finally, the decoder takes downsampled features, gradually upsamples them until the desired resolution is reached, and predicts the foreground mask of the input frame.

As a multi-scale convolutional neural network, FgSegNet_v2 is characterized by the incorporation of features from shallow, mid-level and deep layers, and by the distinctive feature of processing multiple branches of the same 2D data at several resolution values [20][27]: this scale-aware mechanism enables the network to use image length and width information more effectively, improving its performance in many scenarios [41].

### 2.1.1 Encoder

As aforementioned, this component includes the first four blocks of pre-trained VGG-16 Net (the fifth block and the third max-pooling layers have been removed by the authors to obtain higher-resolution feature maps), where Dropout [4] layers (with dropout rate of 0.5) have been positioned between all the convolutional layers of the fourth block to improve generalization performance and to avoid overfitting (by preventing activations from becoming strongly correlated) [6]. Then, only this last block is fine-tuned, while other blocks keep the pre-trained coefficients of the original network [1][2].

More in detail, the first and the second block include two consecutive convolutional layers of, respectively, 64 and 128 hidden neurons. These first two blocks are both followed by a max-pooling module, which has the main function of downsampling the processed data, and which is characterized by a 2×2 filtering kernel.

The third and the fourth block are both composed of three convolutional layers of, respectively, 256 and 512 hidden neurons. All the convolutional layers are characterized by 3×3 kernels.

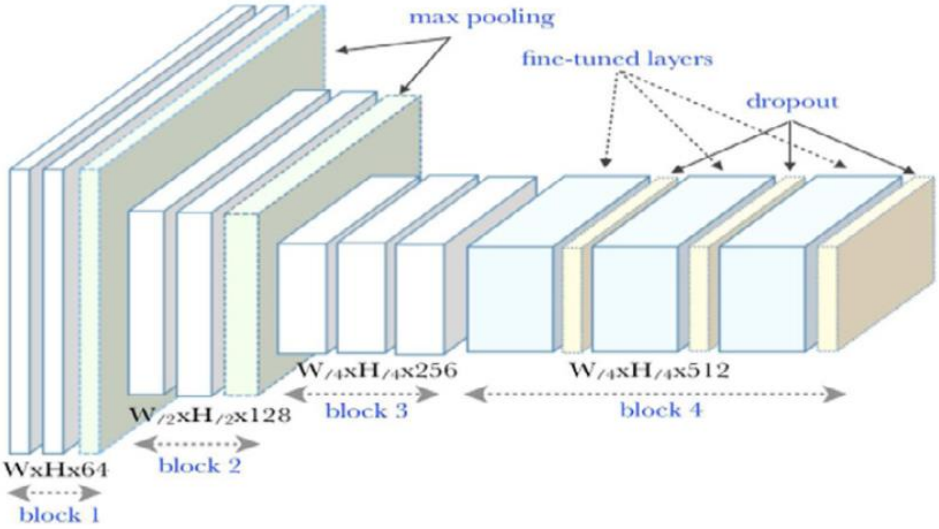


**Fig. 2.2**: Encoder structure (modified VGG-16)

### 2.1.2 Feature Pooling Module

The introduction of "Modified Feature Pooling Module" (referred to as M-FPM) comes from the idea of dilated convolution: this technique computes convolution by multiplying filter coefficients in a spatially sparse way.

Moreover, this design choice has been motivated by the promising results shown by dilated convolution in semantic segmentation domain, with the major effect of enlarging field of views in the neural network without learning extra parameters.

As aforementioned, this module operates on top of the final encoder layer to extract features at multiple scales by applying a 2×2 max-pooling layer (followed by a 1×1 convolution) and four parallel dilated convolutions. More in detail, these latter comprise a normal 3×3 convolution and three 3×3 dilated convolutions with dilation rates of 4, 8 and 16.

All the convolutional layers are composed of 64 hidden neurons.

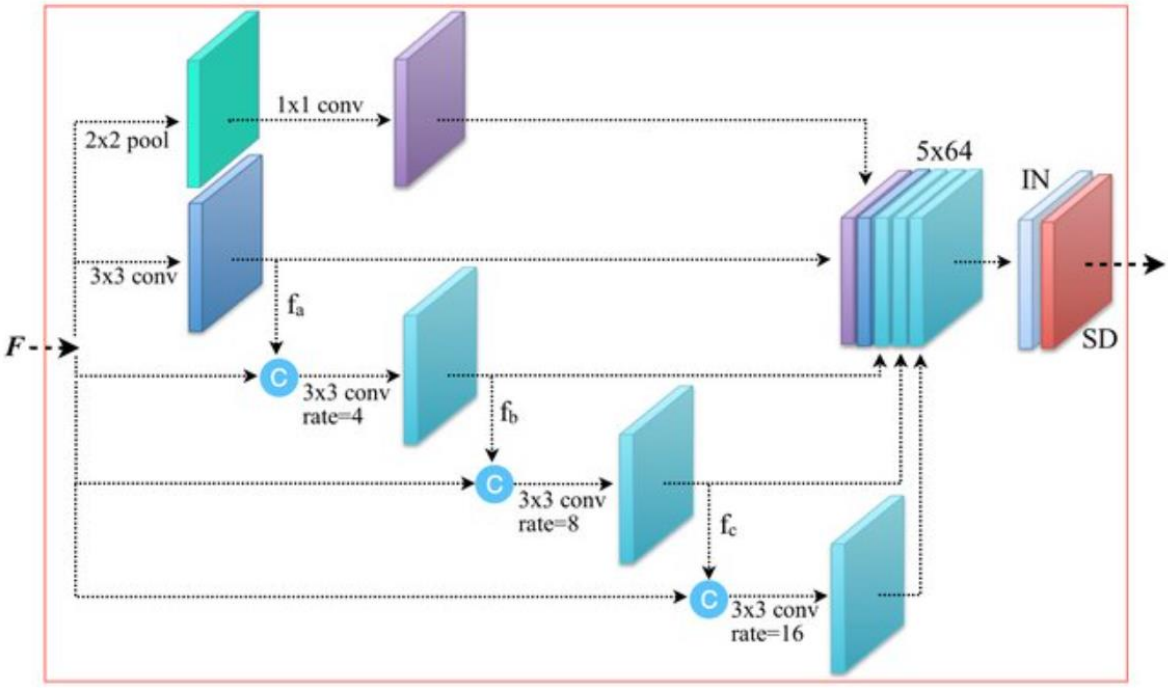


**Fig. 2.3**: M-FPM module

According to the figure above, the resultant features $f_a$ from normal 3×3 convolution are concatenated with feature maps $F$ (obtained from the output of the encoder) and progressively pooled by 3×3 convolution with dilation rate of 4, resulting in features $f_b$.

Then, $F$ and $f_b$ are concatenated and fed to 3×3 convolution with dilation rate of 8, resulting in features $f_c$.

Again, $F$ and $f_c$ are concatenated and fed to 3×3 convolution with dilation rate of 16.

Finally, all features are concatenated to form 5×64 multi-scale depth features, which are passed through Instance Normalization [5], ReLU (Rectified Linear Unit) activation and Spatial Dropout [6] layers.

Instance Normalization is used to normalize the activations of previous layer at each step (i.e., mean close to 0 and standard deviation close to 1), while Spatial Dropout is used to drop the entire 2D feature maps by a chosen rate (0.25 in this implementation): since multiple pooling layers operate on the same features, the concatenated features are likely to be correlated as well as adjacent pixels within feature maps.

For this reason, Spatial Dropout is useful for improving the learning performance, preventing model overfitting and promoting independent features [1][2].

### 2.1.3 Decoder

The decoder with GAP modules is illustrated in Fig. 2.1: it includes the stack of three 3×3 convolutional layers, each one characterized by 64 feature maps and followed by an Instance Normalization layer, and a 1×1 convolutional layer, with a single feature map that embodies the projection from feature space to image space.

In addition to that, ReLU activation function is applied after each Instance Normalization layer, while sigmoid activation function is applied after the last 1×1 convolutional layer to provide the probability of each pixel to figure as component of the foreground mask (output value = 1) or as part of the background model (output value = 0).

Then, a binarization thresholding operation (i.e., pixels are classified as foreground if their value is greater than a given threshold, below which pixels are instead embodied in the background) is applied to the output result to obtain binary segmentation labels [1].

### 2.1.4 Global Average Pooling (GAP)

The main purpose of Global Average Pooling (GAP) layers implementation is to combine information from the low-level features of the encoder with the high-level features of the decoder: indeed, the two GAP modules operate on the second and on the fourth convolutional layers of modified VGG-16.

The result of this pooling operation provides two coefficient vectors $\alpha_i$, which are multiplied by the output features of the decoder's first and second convolutional layers ${f_j}^i$. Then, scaled features are added to the original ones, obtaining features

$$ {f_j}'^i = \alpha_i \cdot {f_j}^i + {f_j}^i $$

where $i \in [0, 63]$ is the index of each feature depth (i.e., the index of correspondent channel) and $j$ is the index of an element in each feature slice.

Finally, the concatenated features ${f_j}'^i$ are upsampled by 2× using bilinear interpolation and fed to the subsequent layers [1].

## 2.2 Network Training Phase

The model has been implemented using Tensorflow Keras framework; then, it has been trained and tested on a NVDIA Tesla T4 GPU.

The training loop involves the use of RMSProp as learning rate optimizer, with parameters $\rho = 0.9$, $\varepsilon = 10^{-8}$ and a batch size of one sample per batch. The learning rate is set to an initial value of $10^{-4}$, which is reduced by a factor of 10 if validation loss does not improve over 5 consecutive epochs.

Moreover, the maximum number of learning epochs is set to 100, but the training loop is stopped earlier if validation loss does not improve over 10 consecutive epochs.

The model is trained using 200 frames, which are randomly split into training and validation datasets in a 70-30 percent rate. Furthermore, frames are compressed by a factor of 3.75 (i.e., from a resolution of 1920×1080 pixels to 512×288) to speed up model's internal computations; ground truth labels have been manually generated for improving model's accuracy in detecting edges.

The chosen loss function for evaluating the difference between true pixel values and predicted ones (not binarized through thresholding during the training phase) of a single frame is binary cross-entropy loss:

$$L_{BCE} = -\frac{1}{n}\sum_{i=1}^{n} y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log(1 - \hat{y}_i)$$

where $y_i$ is the true label of the $i$-th pixel, $\hat{y}_i$ represents the model's prediction and $n$ is the total number of pixels in a single frame (i.e., width×height).

The scored average training time is of 160 ms/frame.

### 2.2.1 Database and training samples

The model has been trained on frames coming from approximately 150 videos belonging to BBSoF S.r.l. dataset: these samples involve, as main characteristics, 7 different environments (i.e., 4 outdoor football pitches, 2 indoor medical laboratories and an indoor basketball pitch) with subjects performing several tasks (i.e., gaits, sprints, cutting maneuvers, squats, pistol squats, lunges, jumps) or technical gestures (such as dunks for basketball players or pirouettes for skaters) in various lightning and atmospheric conditions.

The whole BBSoF database includes more than 350 videos, whose focus is mainly put on the technical gestures performed by athletes. The recording session setup consists of 4 synchronized GoPro cameras, with frame resolution of 1920×1080 pixels and sampling rate of 30 frames per second.

For the training phase, 110 samples have been taken from outdoor recorded videos, while the other 90 frames come from laboratory acquisitions.

Following a holistic reasoning for ground truth labeling, occluded body parts (e.g., covered by obstacles) are labeled (where possible) as part of the foreground mask: this choice is motivated by the fact that FgSegNet_v2 learns human features and whole-body shape, being able to reconstruct full subject figures even in presence of obstacles between subject(s) and camera.

Moreover, subjects whose figure is partially visible (i.e., less than 50%) or people in the background (except for operators near to the cameras setup) have been considered as part of the background, and so they have not been labeled as foreground objects: this choice comes from the fact that this neural network does not embody temporal information while doing the background subtraction operation (unlike classical methods with dynamic background modeling), and so it is crucial to highlight which subjects to detect.

**Fig. 2.4**: Examples of training samples (frame + ground truth)

# 3. Experimental Results

This section describes the experimental setup (i.e., test dataset and metrics) used for evaluating model's performance and reports some results generated by FgSegNet_v2 on BBSoF database frames.

## 3.1 Test Dataset and Evaluation Metrics

The dataset used for testing the model is structured in groups of 4 frames reporting the same recorded image from different angles.
Moreover, the test dataset is divided into 8 different subsets according to BBSoF database categorization: these subsets include athletes performing various tasks in 6 different environments (i.e., the same ones used for training the model), with the aim of testing the model on several human poses.
Since the problem is formulated as a pixel-wise binary classification task, each pixel can produce one of the following outcomes:

- true positive (TP), corresponding to correctly classified foreground pixel
- true negative (TN), corresponding to correctly classified background pixel
- false positive (FP), corresponding to incorrectly classified background pixel
- false negative (FN), corresponding to incorrectly classified foreground pixel

Here there are reported the formulas of metrics used for evaluating model's performance:

$$ACCURACY = \frac{TP+TN}{TP+FP+FN+TN} \qquad PRECISION = \frac{TP}{TP+FP}$$

$$RECALL = \frac{TP}{TP+FN} \qquad F1-SCORE = \frac{TP+TN}{TP+\frac{1}{2}(FP+FN)}$$

$$\kappa = \frac{2(TP \cdot TN - FN \cdot FP)}{(TP+FP)\cdot(FP+TN) + (TP+FN)\cdot(FN+TN)}$$

## 3.2 Feature Maps

For a convolutional neural network, the size of a hidden layer is given by the number of its neurons and the number of its channels.
Furthermore, each hidden neuron is characterized by a kernel (also named filter), namely a local receptive field encoding a specific feature. The combination between the kernels of a hidden layer provides a feature map of that layer, whose quantity is equal to the number of channels.
Feature maps are useful to study the model's behavior at each layer, obtaining visual feedbacks of the operations (e.g., edges detection, features extraction and encoding, dilated convolutions and foreground segmentation) that lead to neural network's predictions.
This section reports some feature maps (i.e., 16 per convolutional layer) of the reference frame below along with the correspondent layer, its total number of feature maps and the resolution of these ones.
As illustrated by the feature maps of the next subsections, the encoder operates an edge detection to obtain downsampled representations of human body parts or full figures (i.e., it encodes a compact representation of human body features, which are then detected and segmented by the decoder).
Then, the M-FPM applies parallel dilated convolutions to extract features at different scales, encouraging the isolation of the moving subject from the background.
Finally, the decoder combines the information received from the two previous components to segment moving subject's silhouette and therefore perform background subtraction operation.

**Fig. 3.1**: Reference frame

### 3.2.1 Encoder

**Fig. 3.2**: First convolutional layer of the first block (64 feature maps of resolution 512×288)

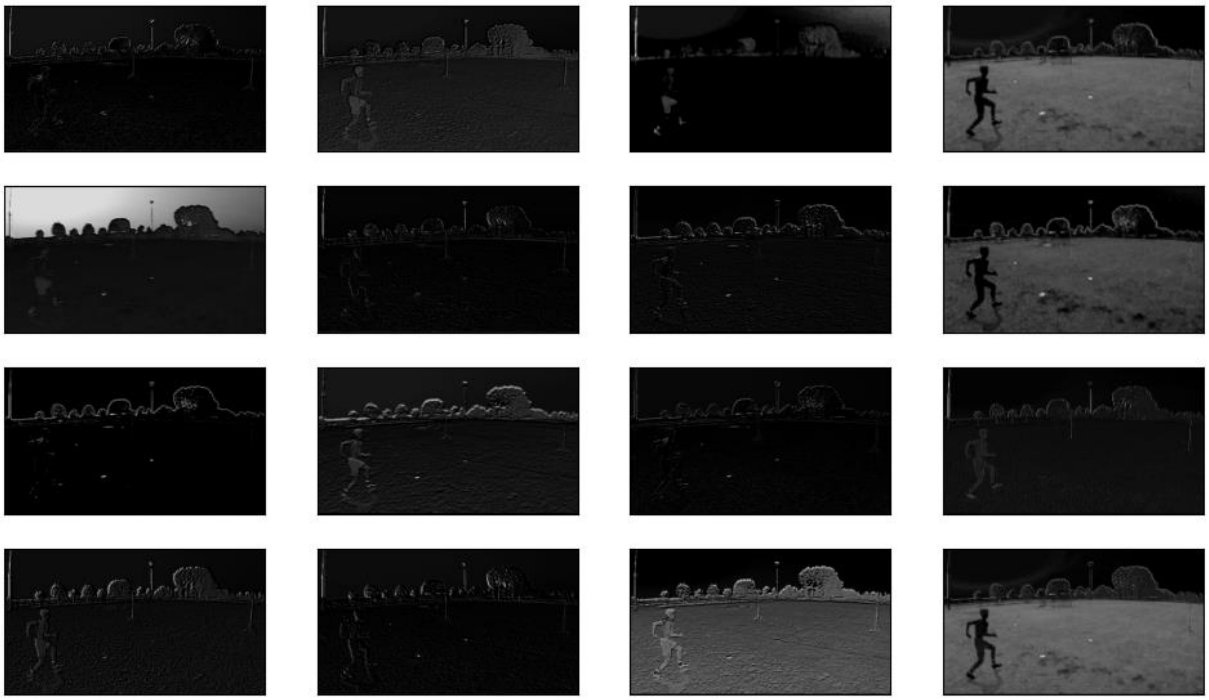

**Fig. 3.3**: Second convolutional layer of the first block (64 feature maps of resolution 512×288)

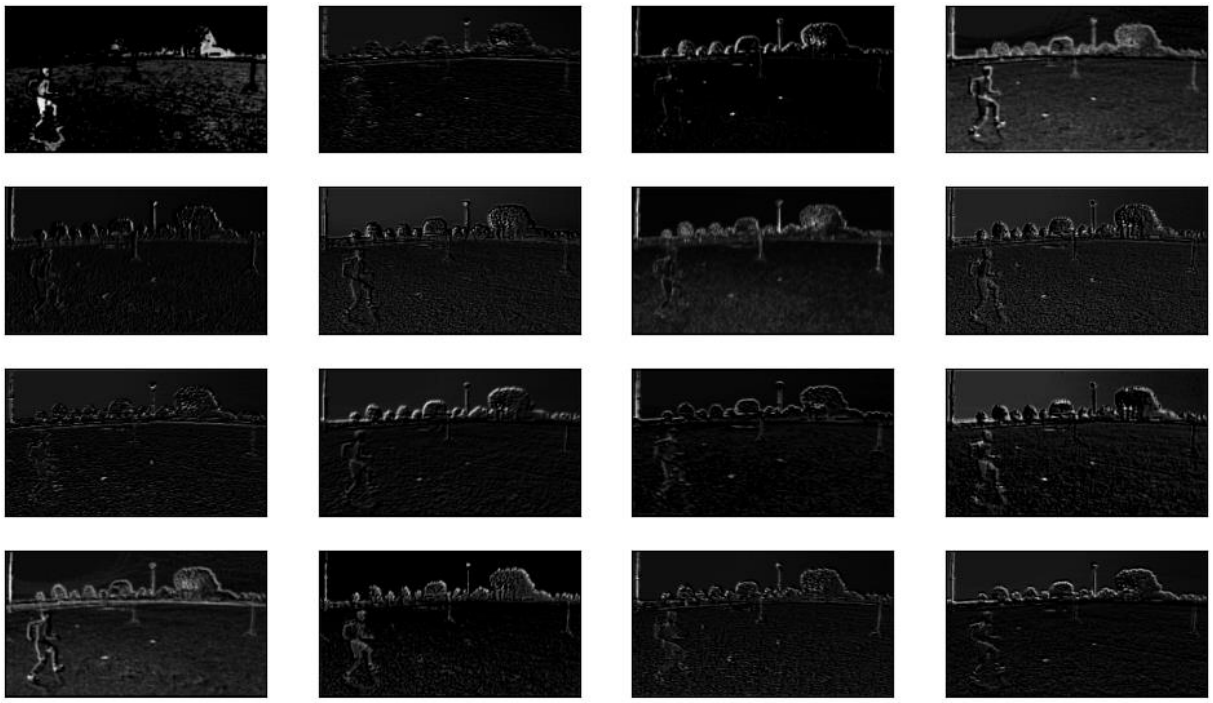

**Fig. 3.4**: First convolutional layer of the second block (128 feature maps of resolution 256×144)

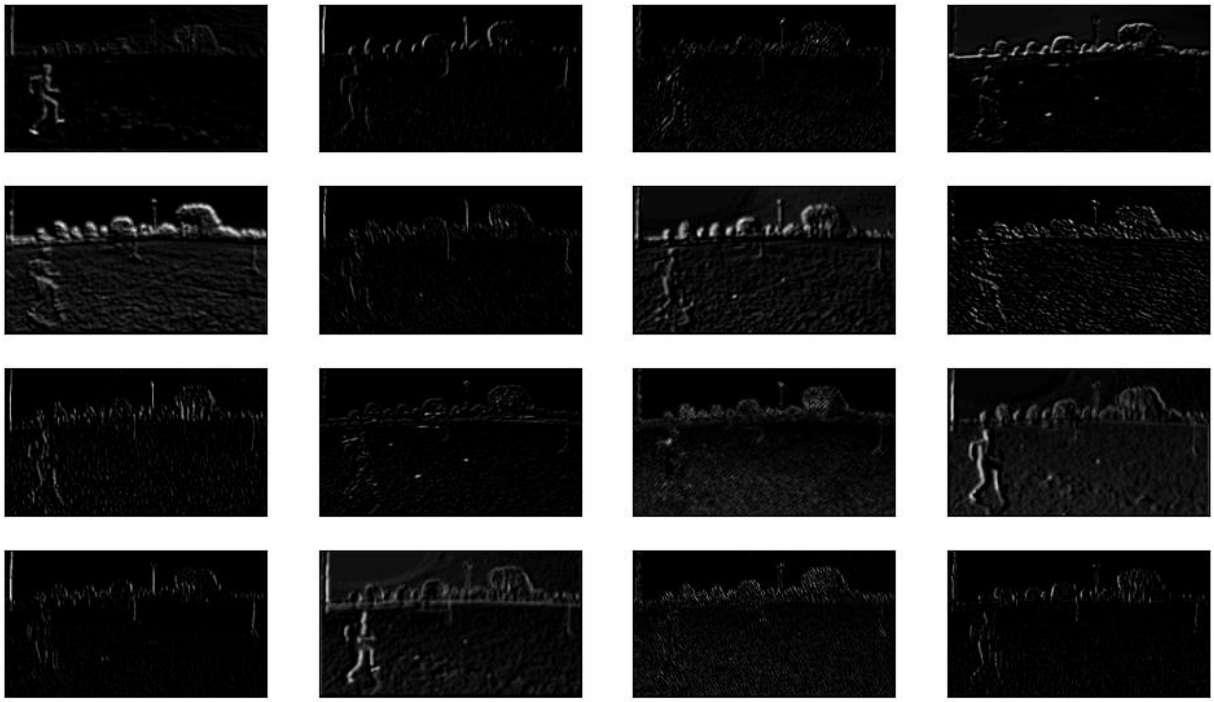

**Fig. 3.5**: Second convolutional layer of the second block (128 feature maps of resolution 256×144)

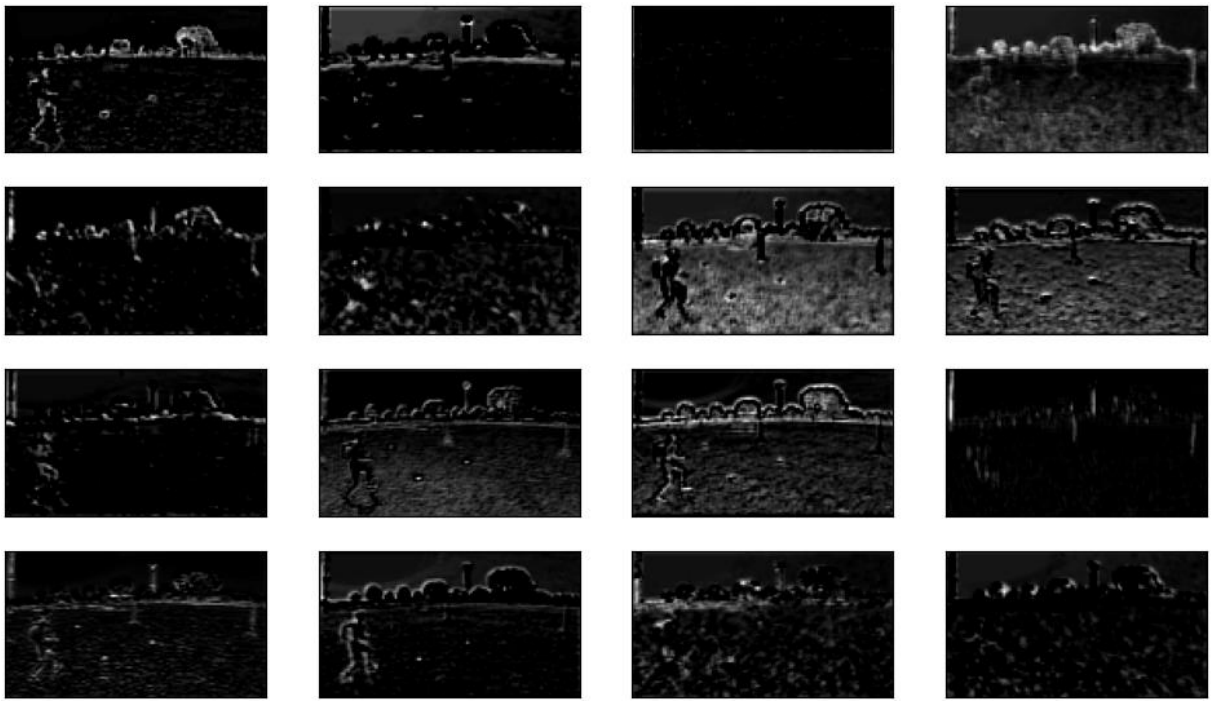

**Fig. 3.6**: First convolutional layer of the third block (256 feature maps of resolution 128×72)

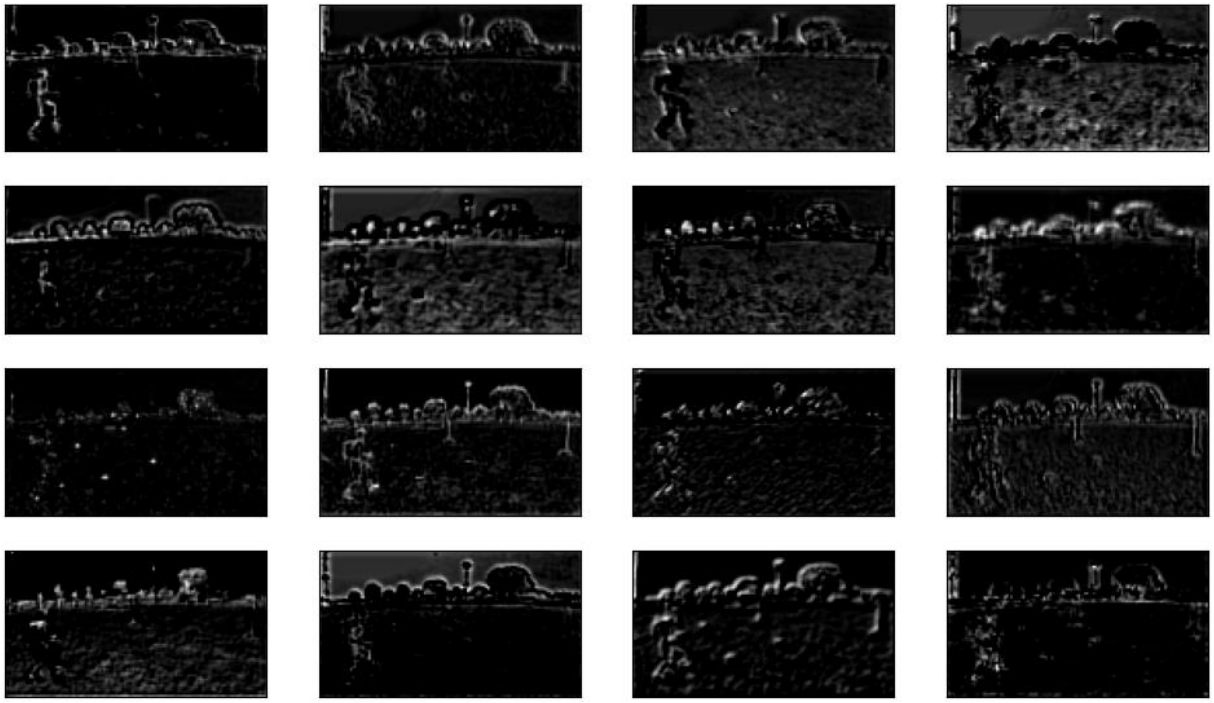

**Fig. 3.7**: Second convolutional layer of the third block (256 feature maps of resolution 128×72)

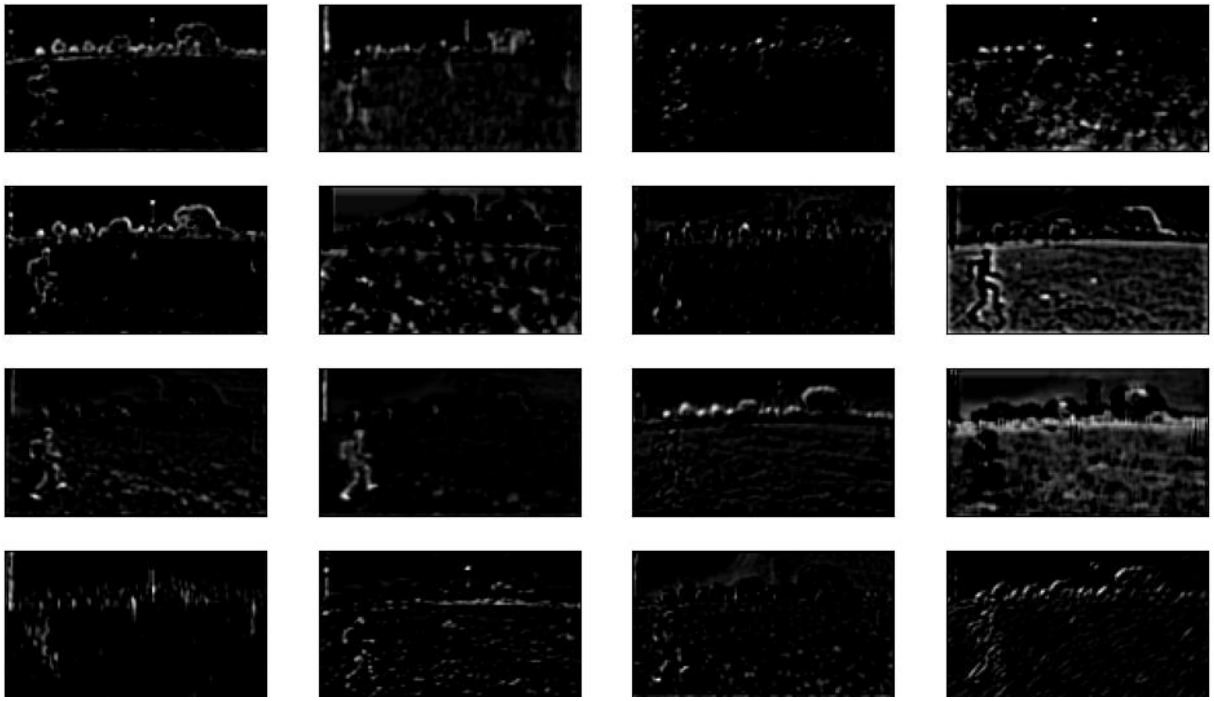

**Fig. 3.8**: Third convolutional layer of the third block (256 feature maps of resolution 128×72)

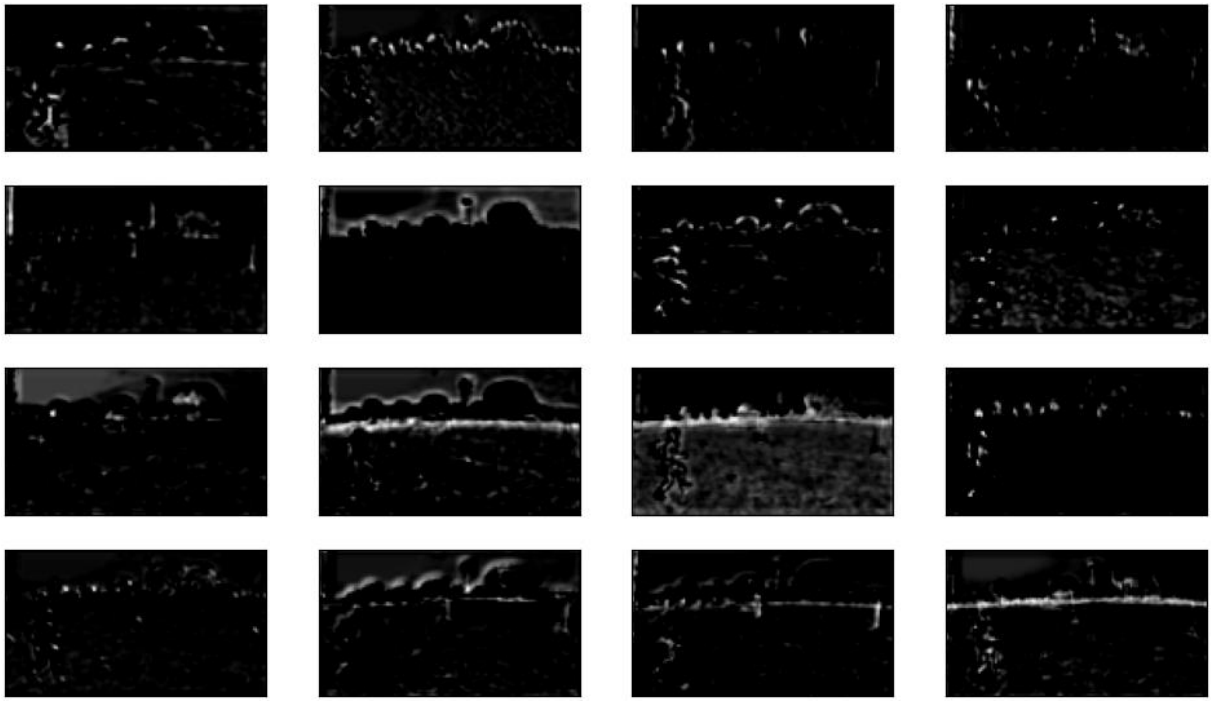

**Fig. 3.9**: First convolutional layer of the fourth block (512 feature maps of resolution 128×72)

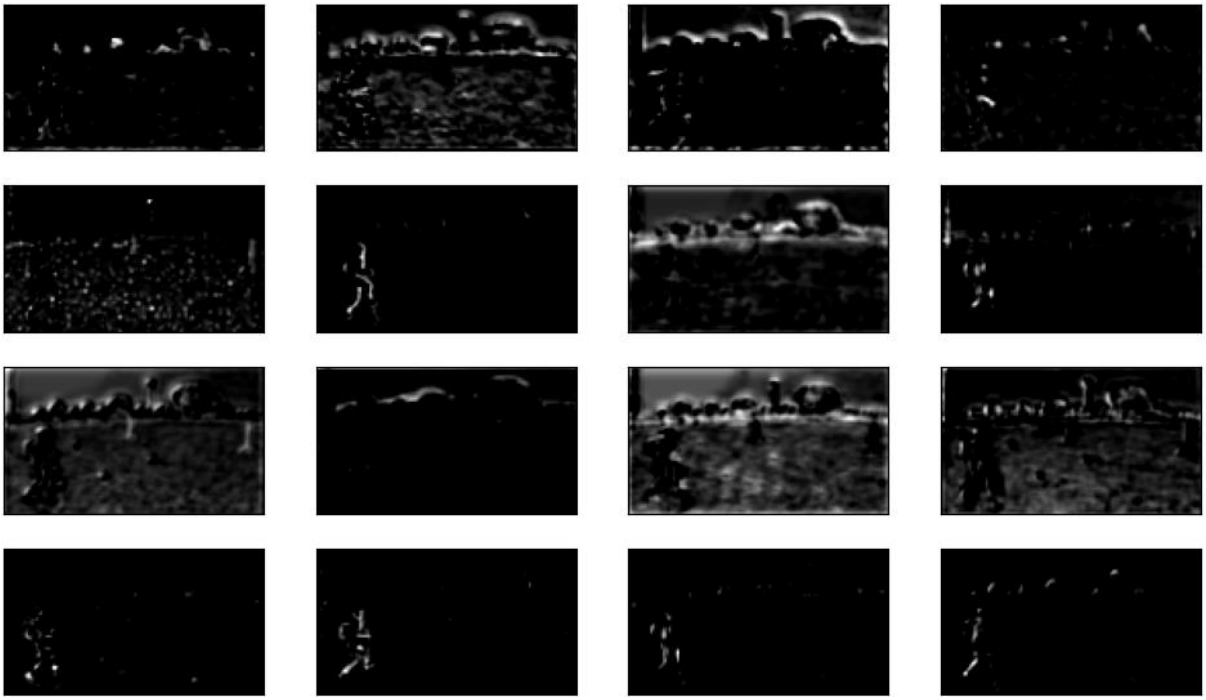

**Fig. 3.10**: Second convolutional layer of the fourth block (512 feature maps of resolution 128×72)

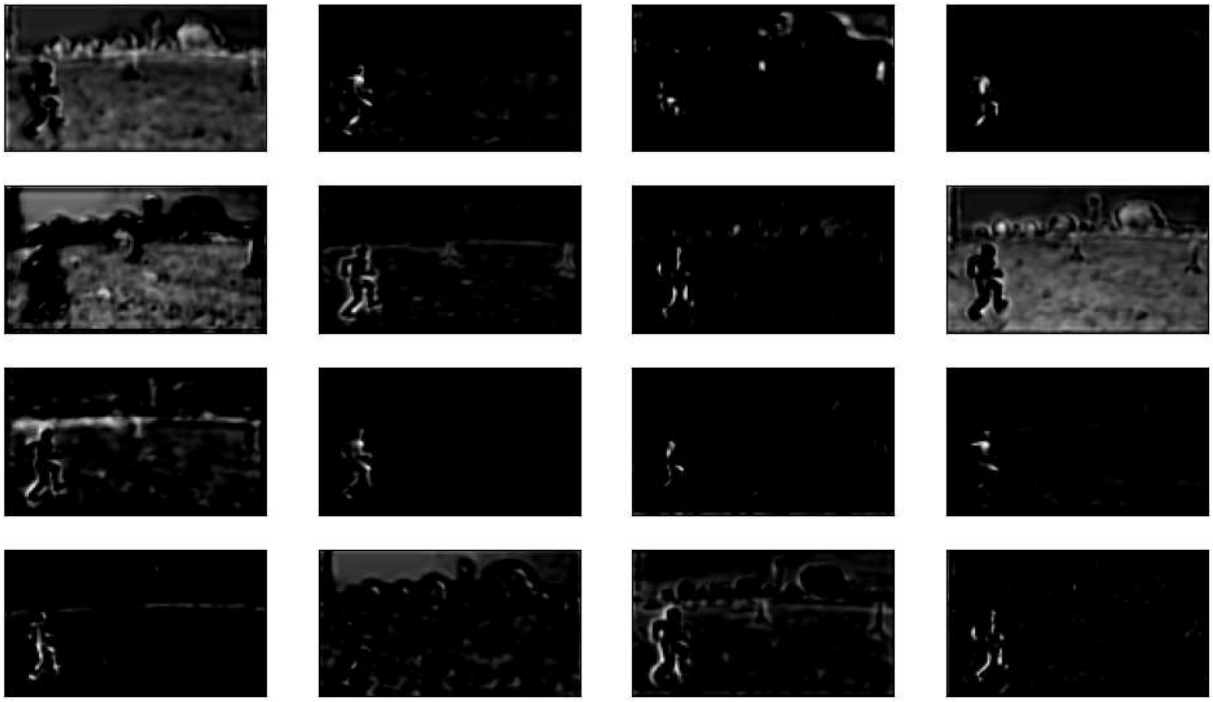

**Fig. 3.11**: Third convolutional layer of the fourth block (512 feature maps of resolution 128×72)

### 3.2.2 M-FPM

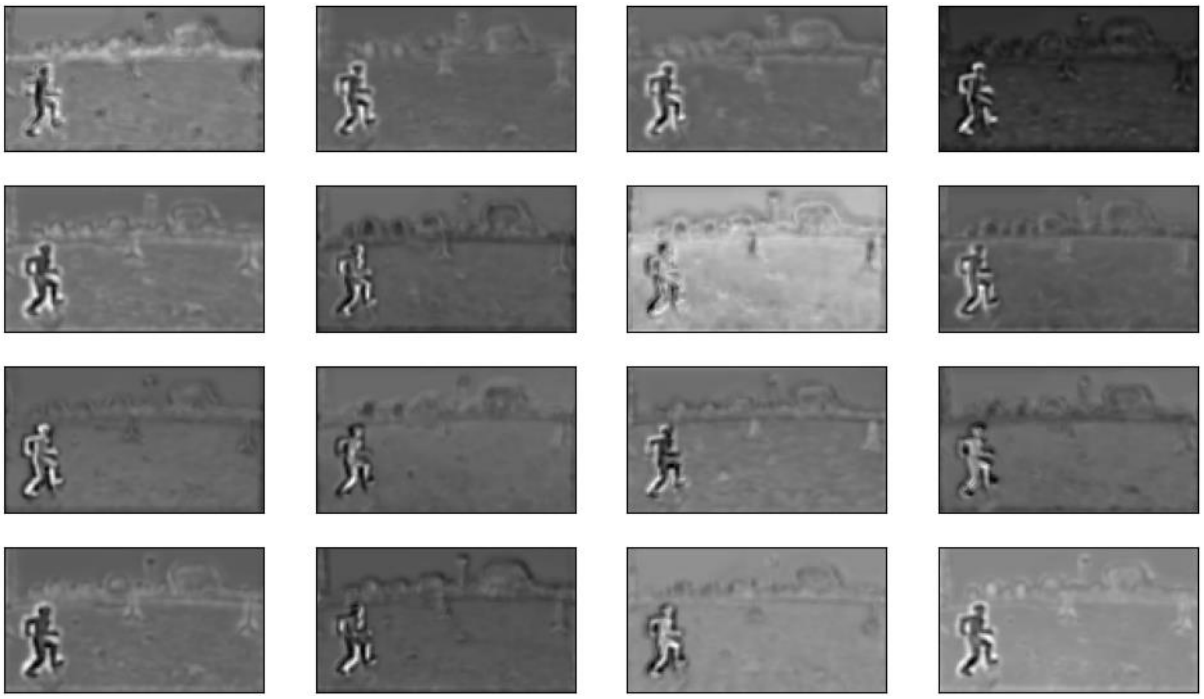

**Fig. 3.12**: 1×1 convolutional layer after max pooling (64 feature maps of resolution 128×72)

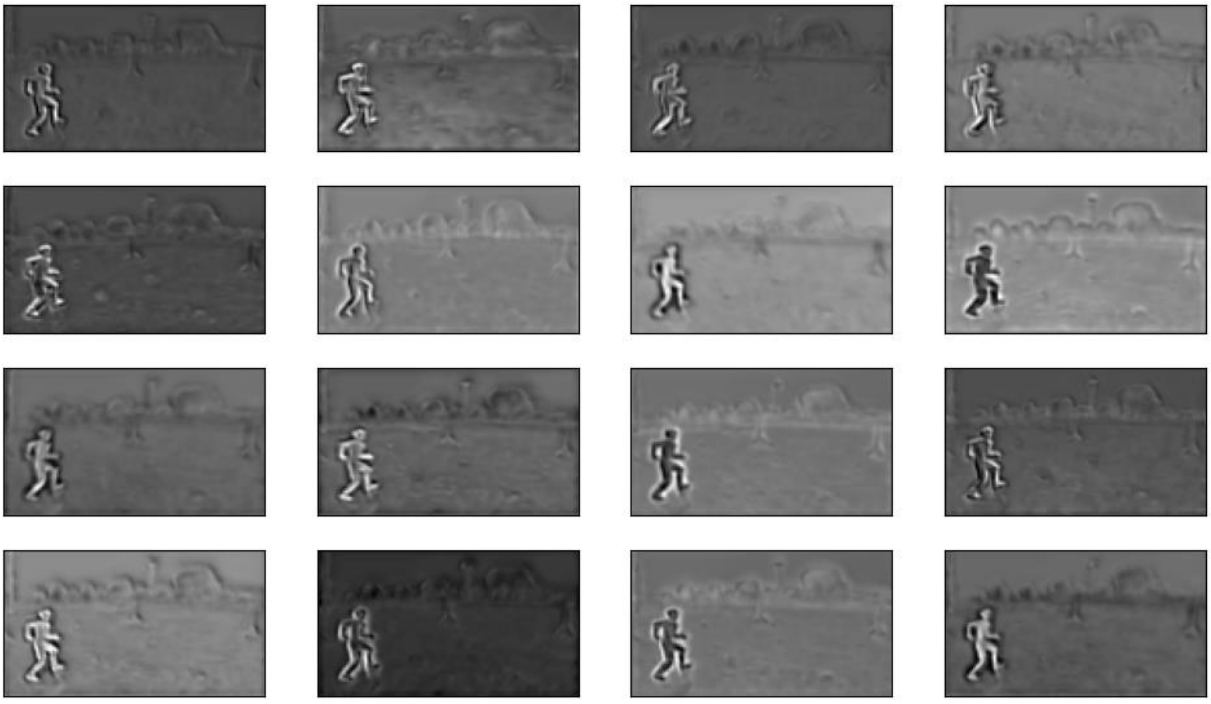

**Fig. 3.13**: 3×3 convolutional layer without dilation (64 feature maps of resolution 128×72)

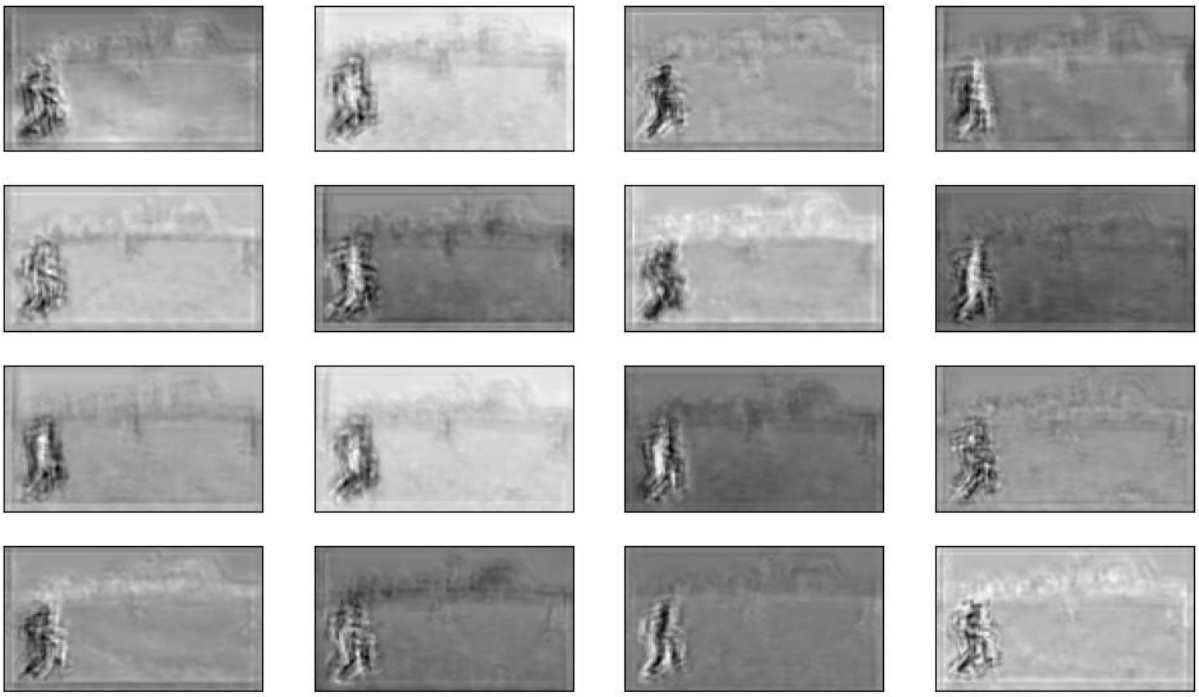

**Fig. 3.14**: 3×3 convolutional layer with dilation rate of 4 (64 feature maps of resolution 128×72)

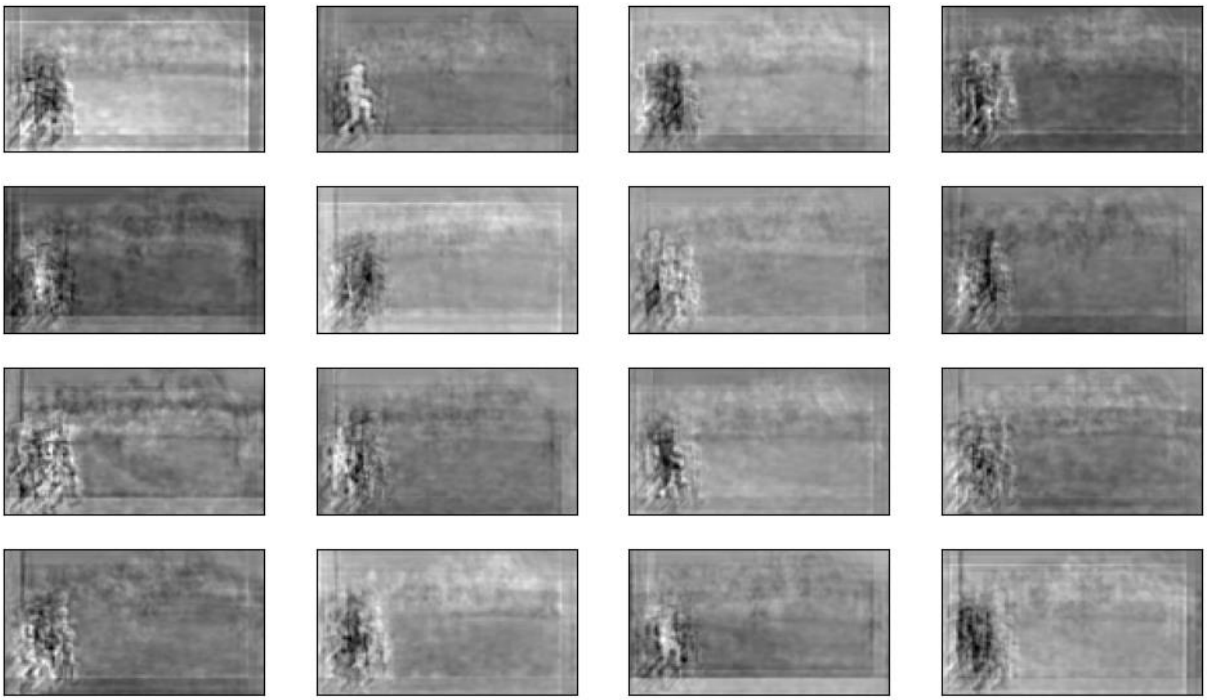

**Fig. 3.15**: 3×3 convolutional layer with dilation rate of 8 (64 feature maps of resolution 128×72)

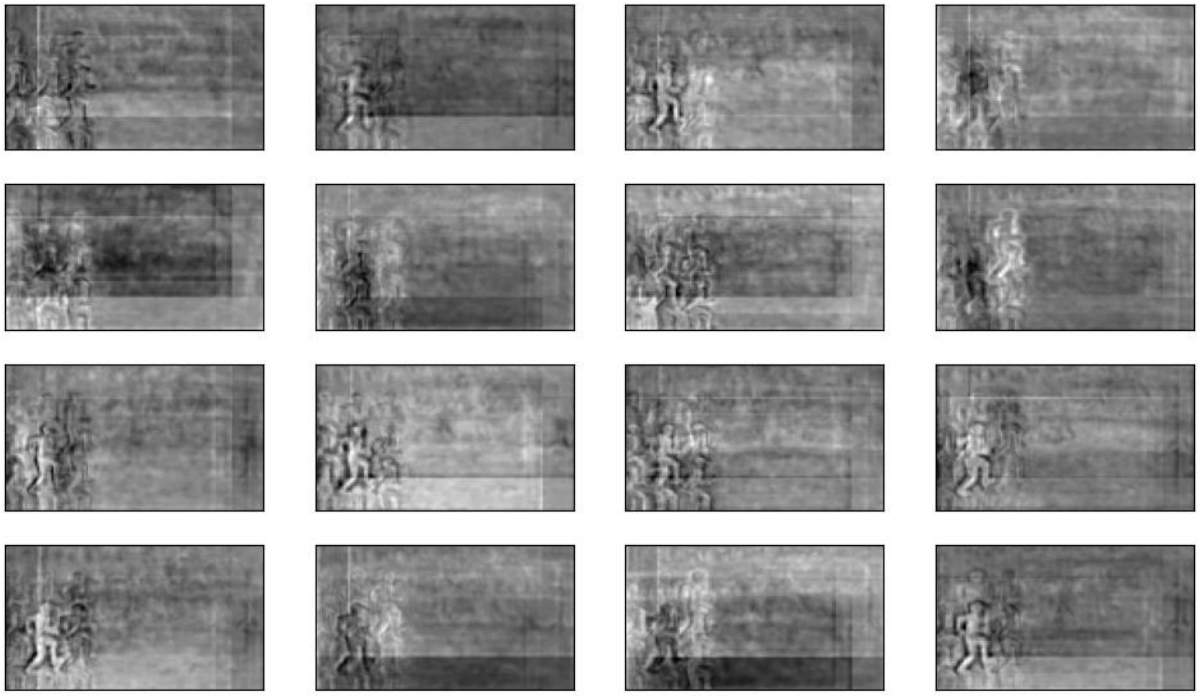

**Fig. 3.16**: 3×3 convolutional layer with dilation rate of 16 (64 feature maps of resolution 128×72)

### 3.2.3 Decoder

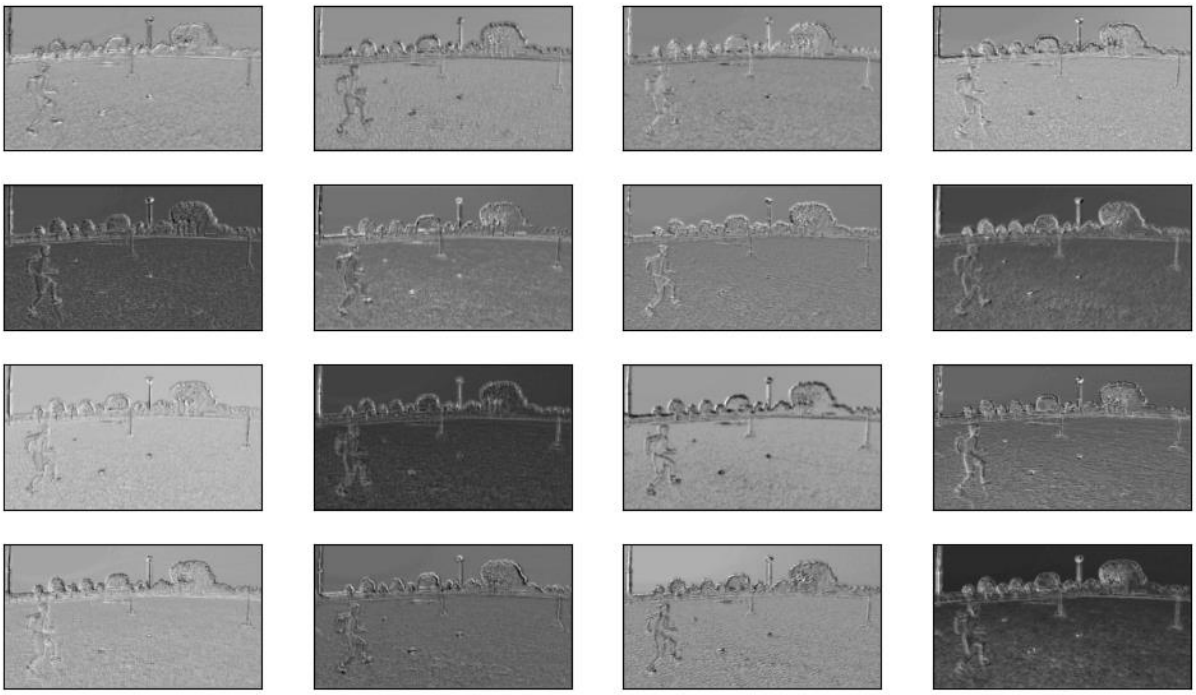

**Fig. 3.17**: 1×1 convolutional layer before Global Average Pooling (64 feature maps of resolution 256×144)

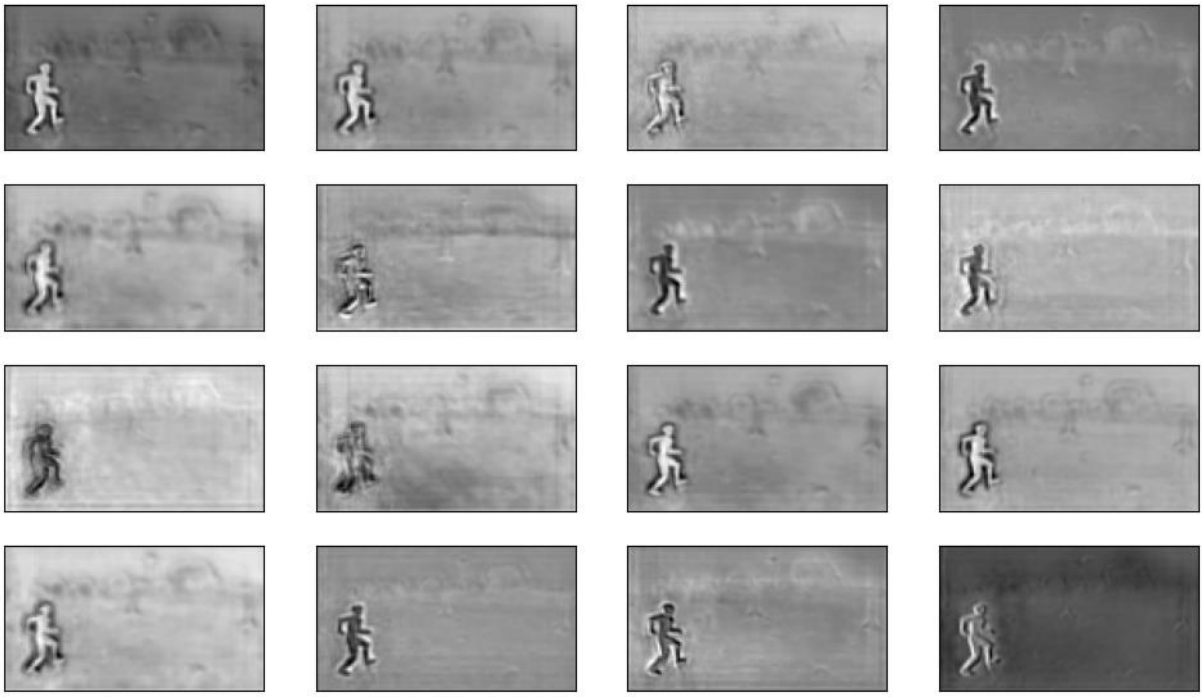

**Fig. 3.18**: 3×3 convolutional layer after Spatial Dropout (64 feature maps of resolution 128×72)

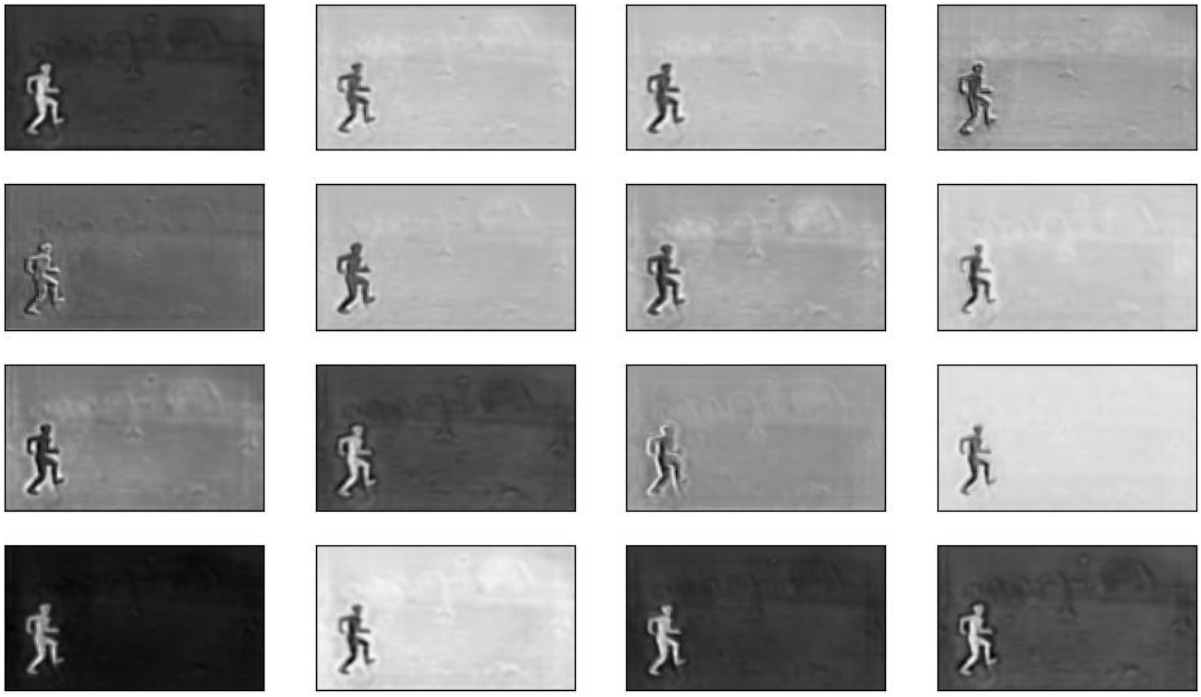

**Fig. 3.19**: 3×3 convolutional layer after first upsampling operation (64 feature maps of resolution 256×144)

**Fig. 3.20**: 3×3 convolutional layer after second upsampling operation (64 feature maps of resolution 512×288)

## 3.3 Foreground Masks

This section includes, in the following order, groups of 4 frames for 8 different categories reporting the same recorded image from different angles (i.e., input frames), the corresponding ground truth images, model's predictions, and binarized results.
Moreover, there are specified for each environment how many frames coming from it have been used during the training phase and the selected threshold value for optimizing binary predictions.
The results of this section show that FgSegNet_v2 performs well both in indoor labs and outdoor football pitches. Furthermore, the model has been able to tackle issues such as dynamic background motion, illumination changes and shadows labeled as part of the foreground object, which represent common difficult scenarios especially for outdoor acquisitions.
Many false positive pixels correspond to static human subjects (either fully visible or partially hidden), meaning that the model has been able to correctly encode human features.
In addition to that, the model struggled into detecting human body features under very similar color conditions between these ones and other objects (e.g., basketball players hands and the basketball), in frames with blurred edges, or in conditions where subjects were not close to the camera.
This lack of accuracy is also caused by the frames compression of a factor of 3.75: the model processes images with dimension 512×288, which are then expanded (once the elaboration is concluded) back to their original dimension of 1920×1080 pixels.

**Calciatori Iozzino @lab**

(72 training frames coming from 4 cameras, threshold value = 0.7)

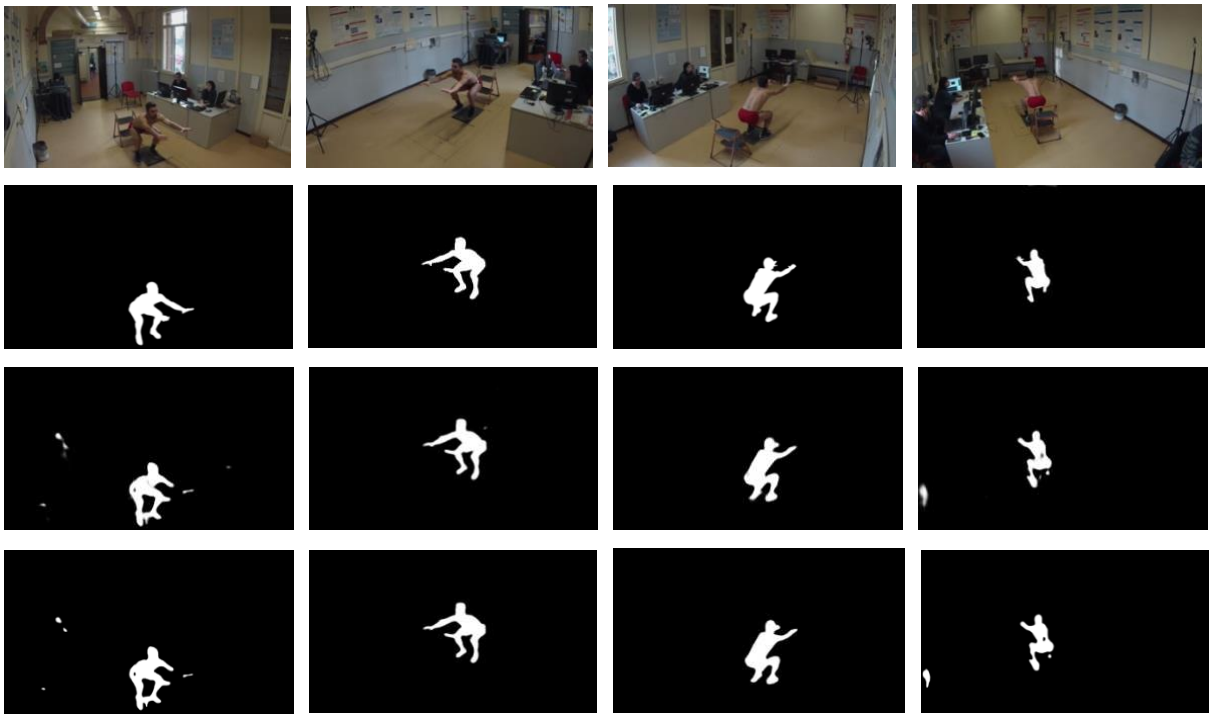

**Fig. 3.21**: Drop squat

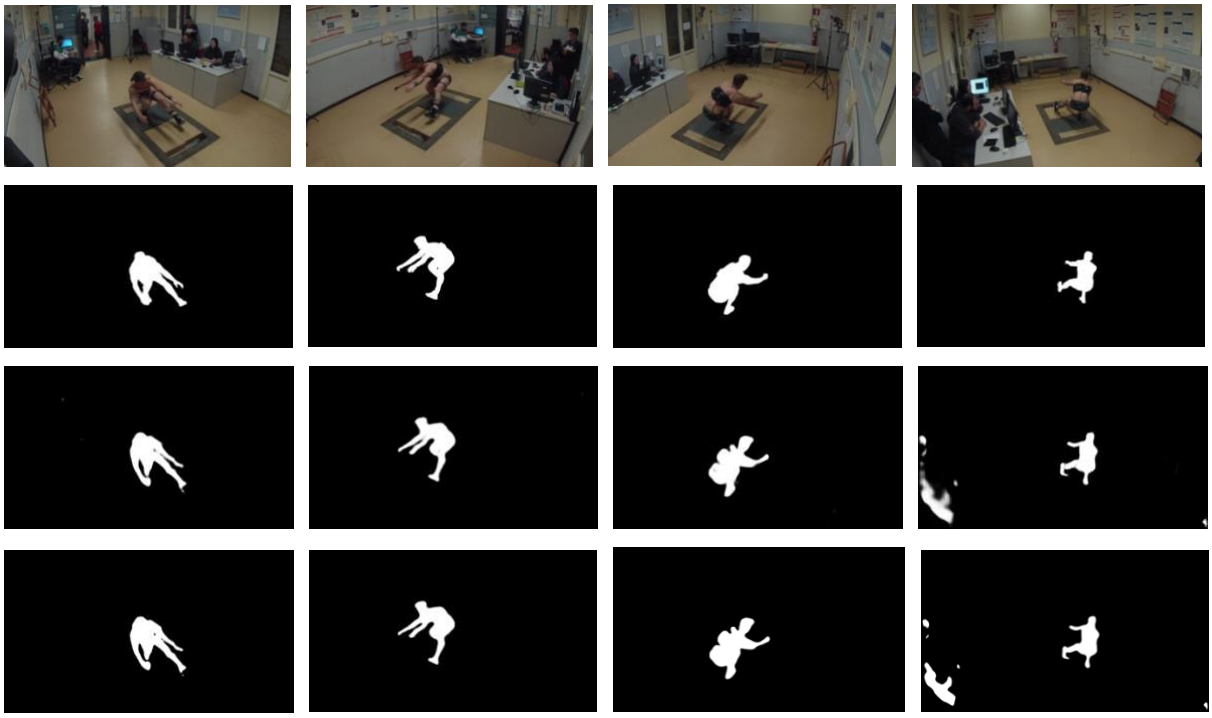

**Fig. 3.22**: Pistol squat

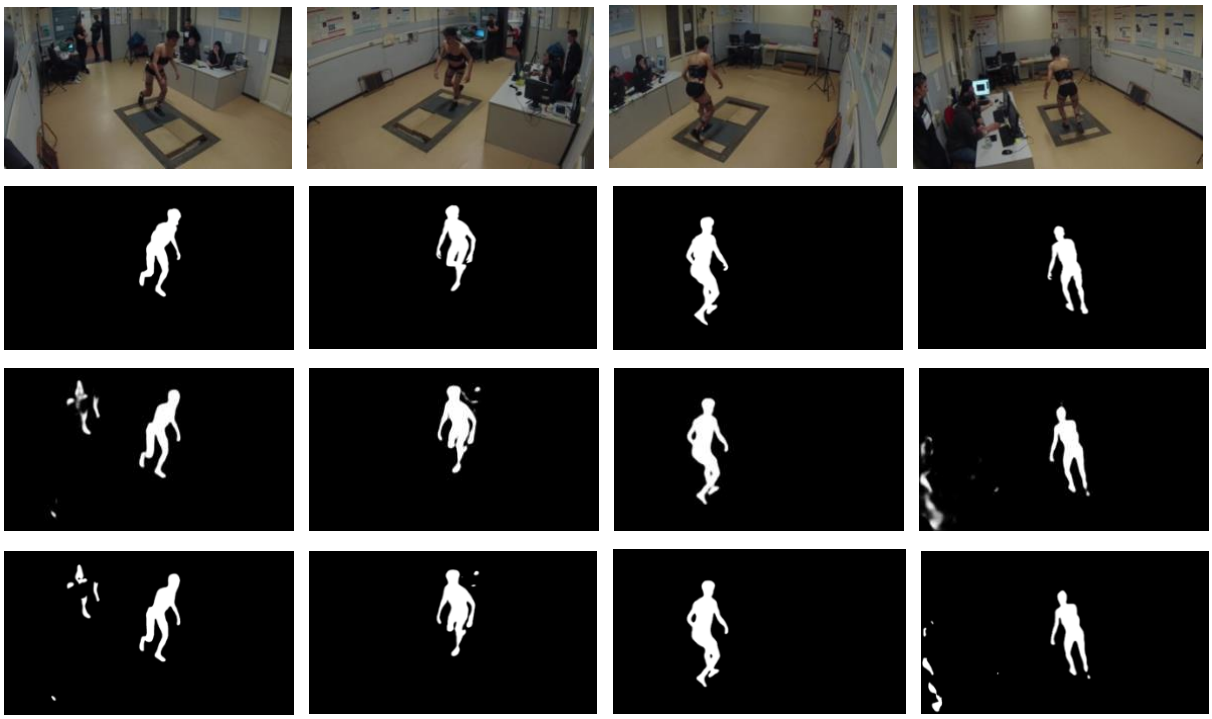

**Fig. 3.23**: Lunge

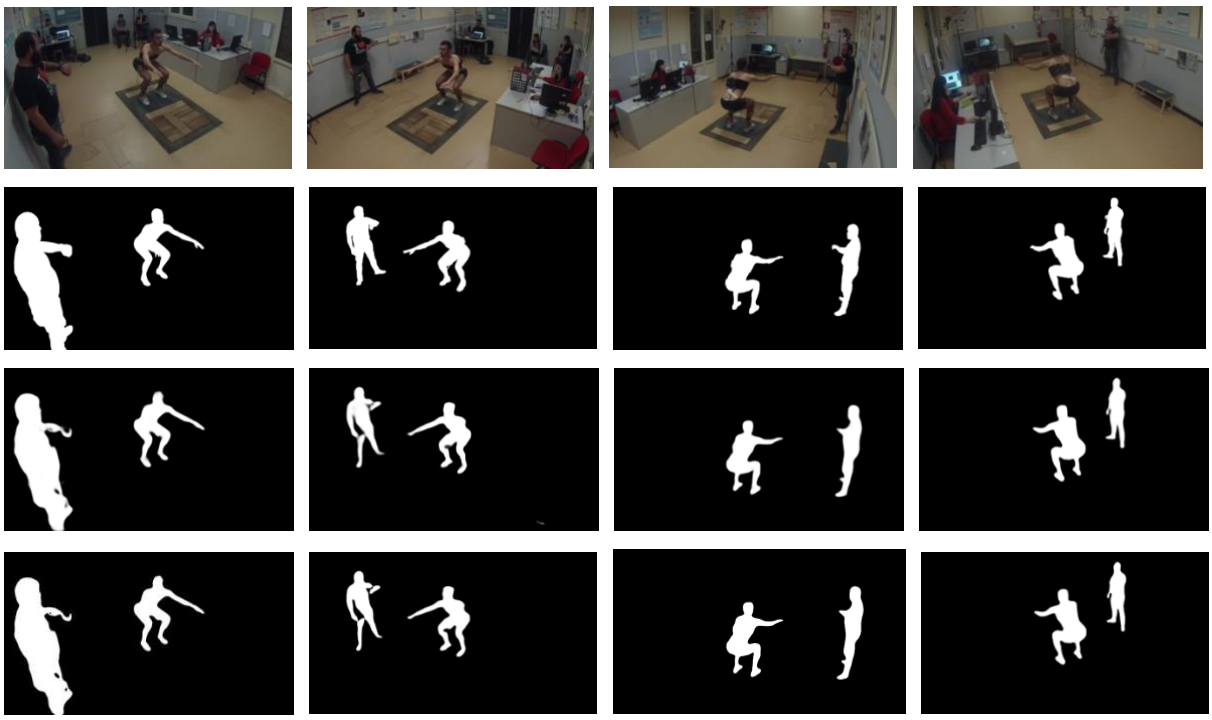

**Fig. 3.24**: Squat

**Fig. 3.25**: Gait

**Inter 22.05.2018 @Appiano Gentile**

(55 training frames coming from 3 cameras, threshold value = 0.5)

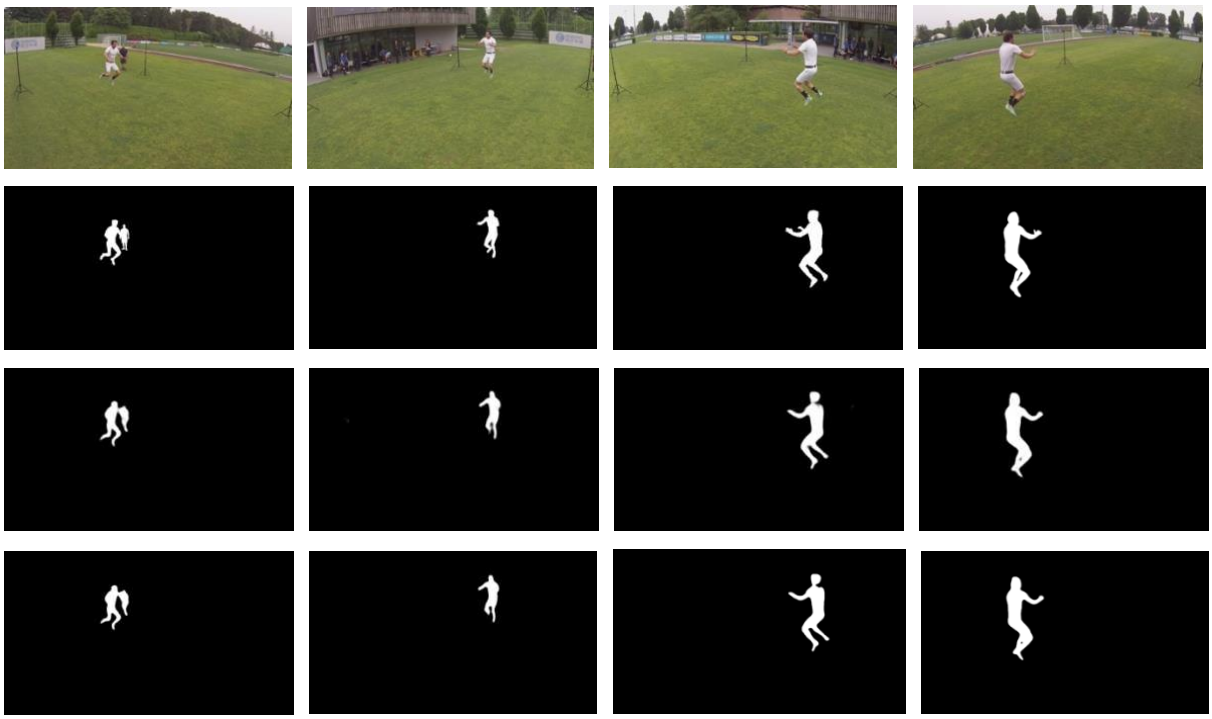

**Fig. 3.26**: Hop

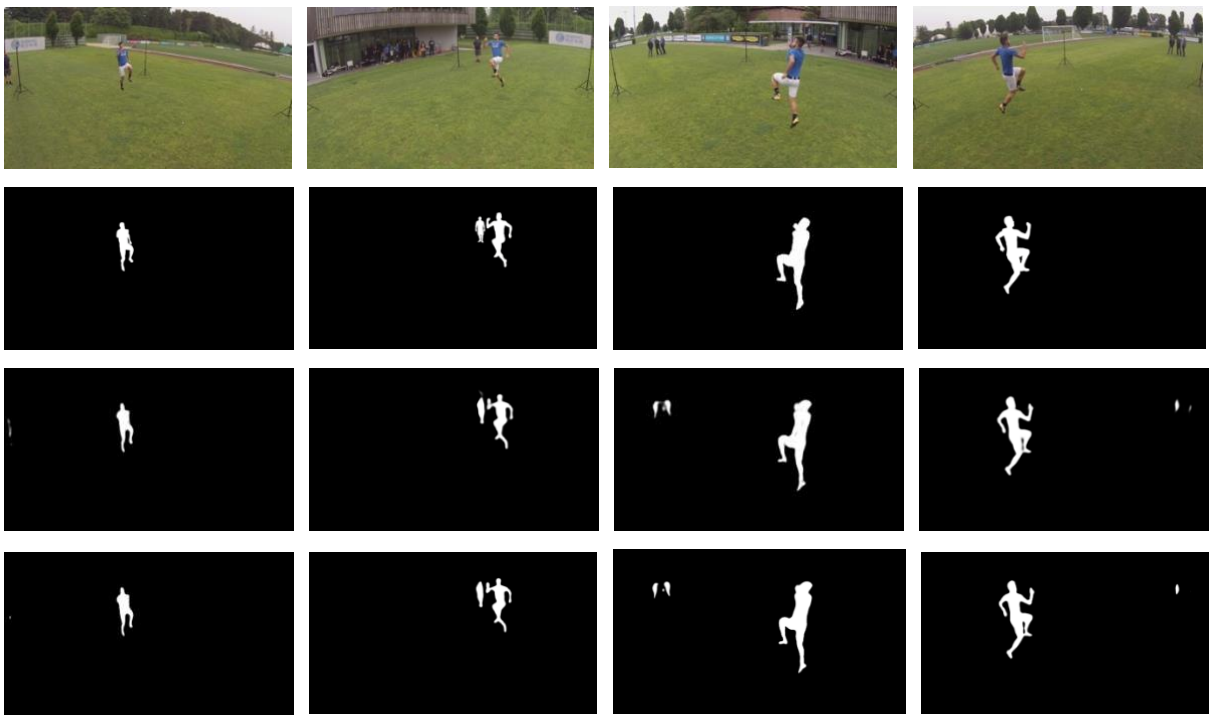

**Fig. 3.27**: Hop + Cutting maneuver

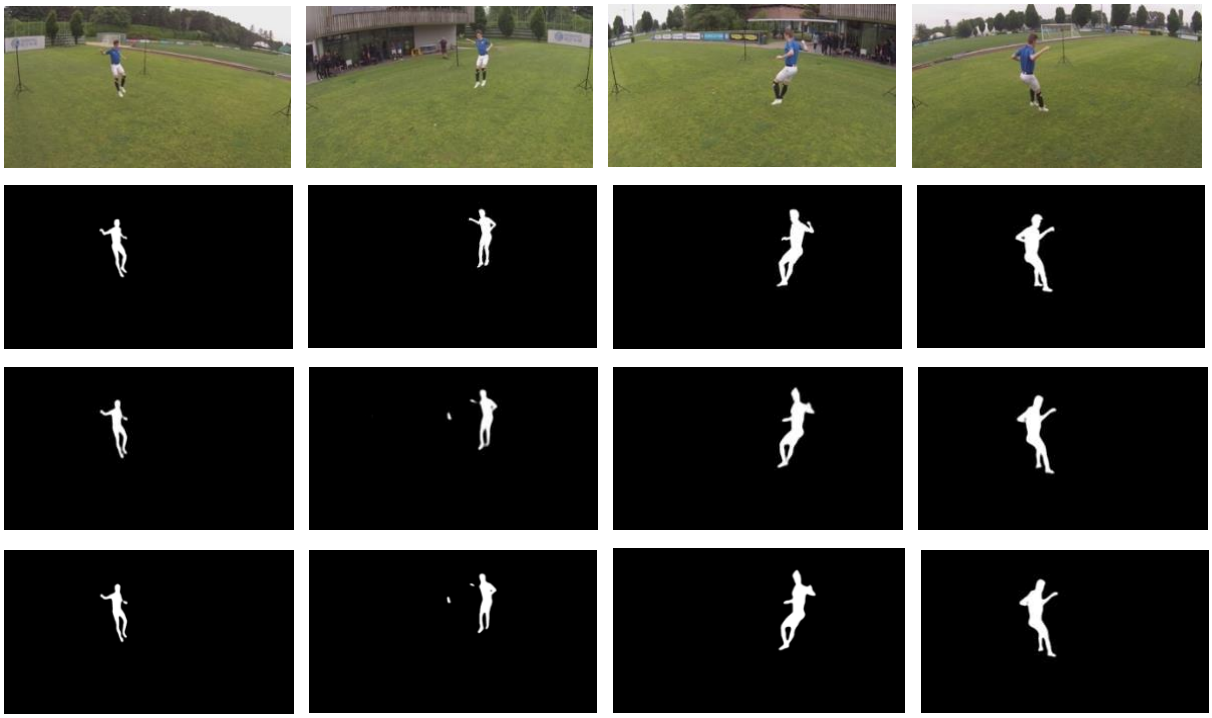

**Fig. 3.28**: Hop + Cutting maneuver

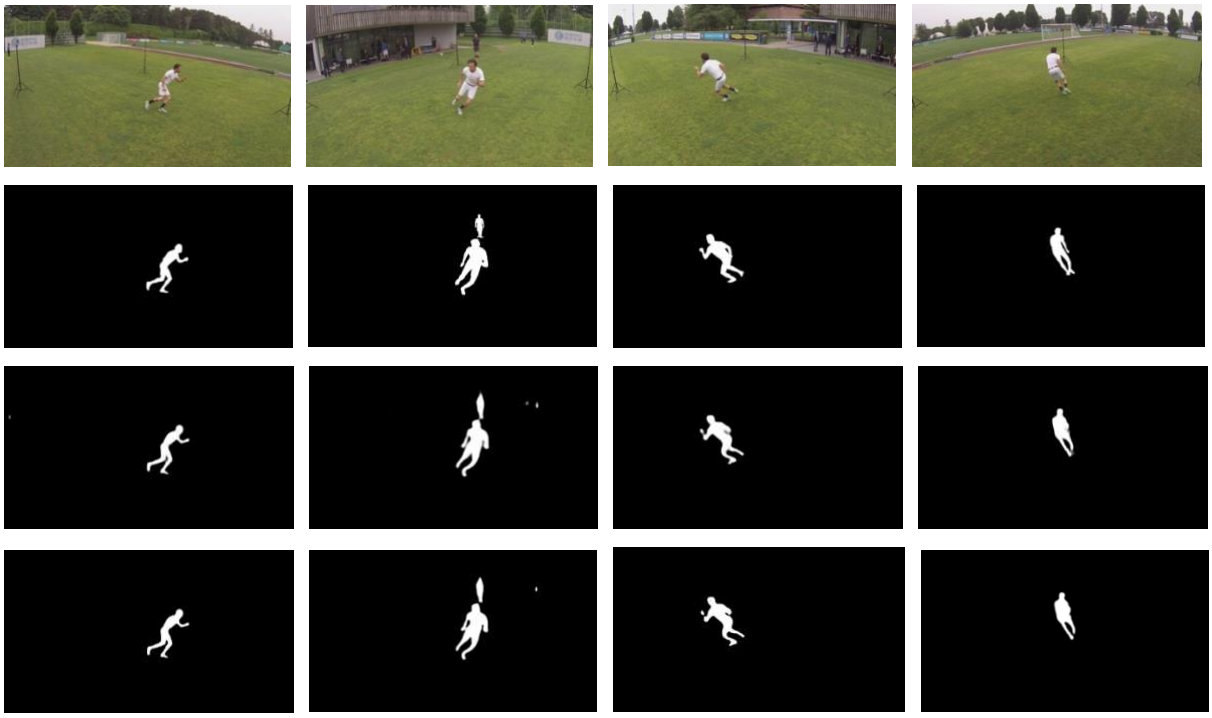

**Fig. 3.29**: Cutting maneuver

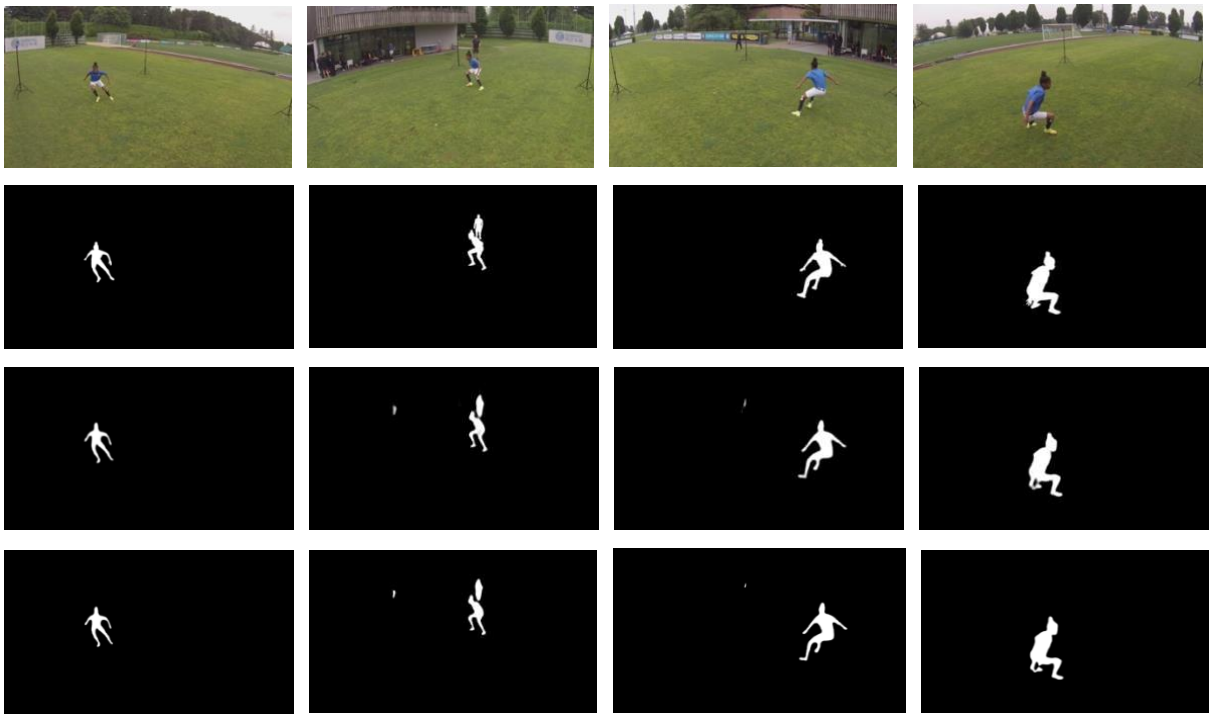

**Fig. 3.30**: Cutting maneuver

**2019-03-05 @Magic**

(5 training frames coming from one camera, threshold value = 0.6)

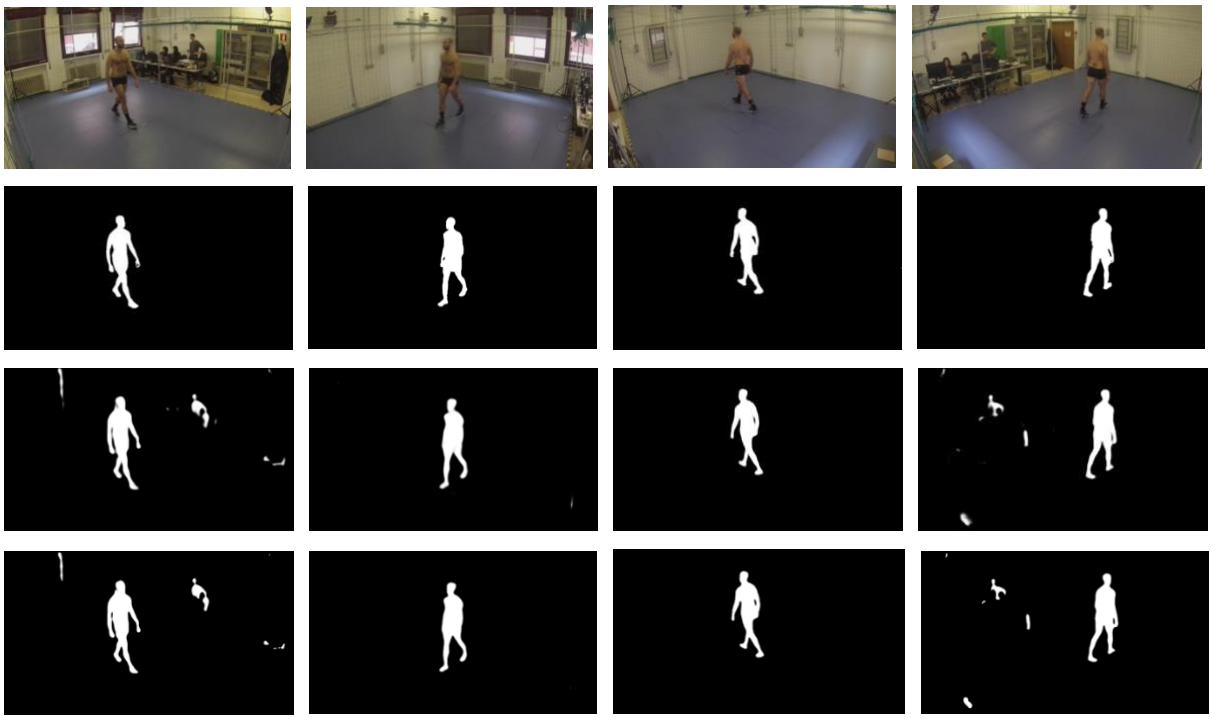

**Fig. 3.31**: Gait

**Fig. 3.32**: Drop lunge

**Fig. 3.33**: Drop squat

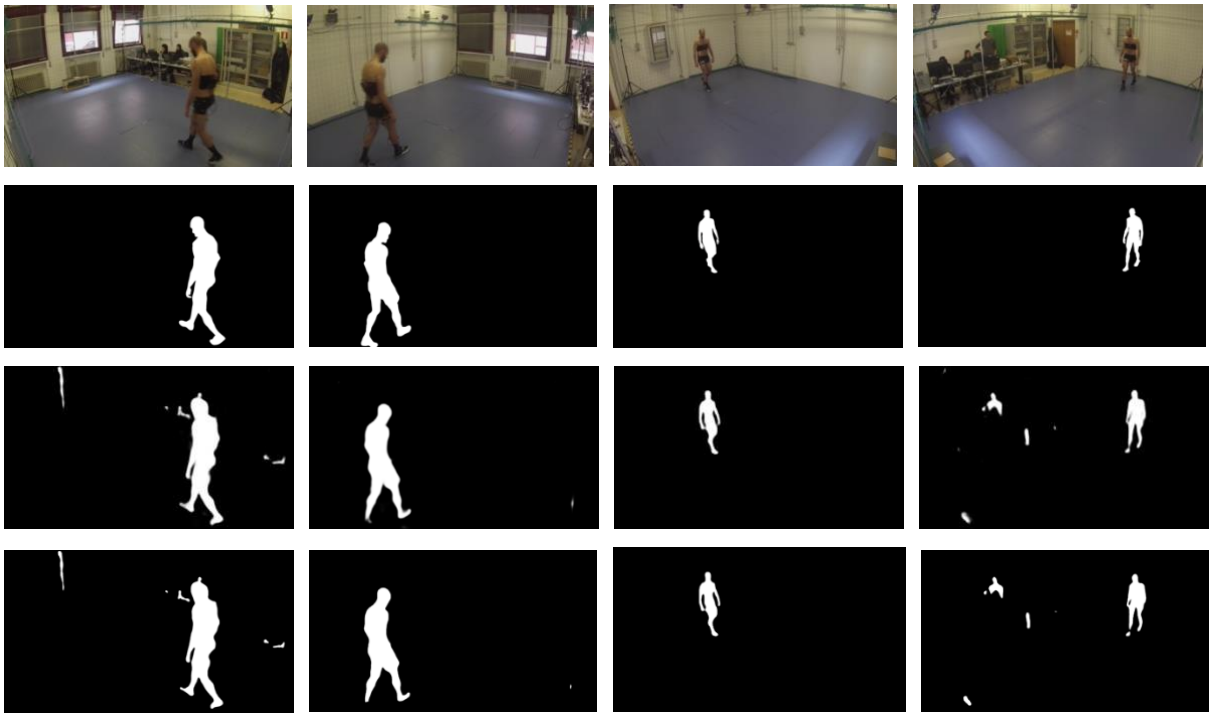

**Fig. 3.34**: Gait

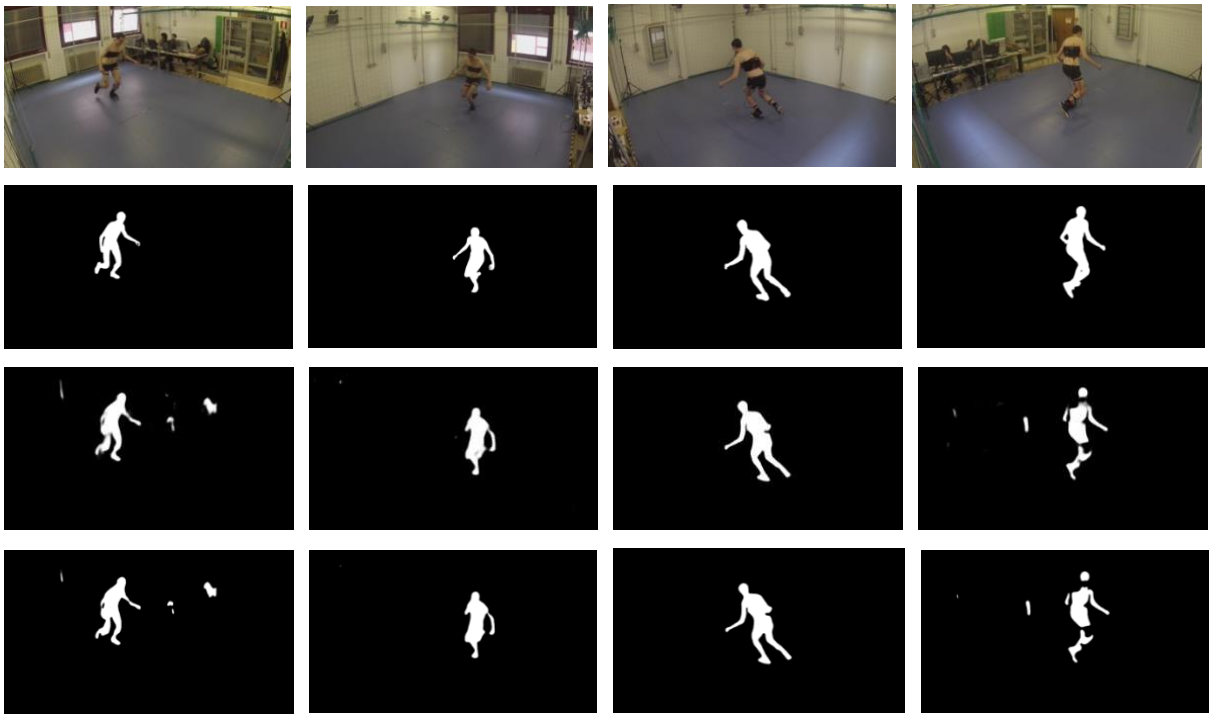

**Fig. 3.35**: Lunge

**Fig. 3.36**: Lunge

**csaPlantare @Vertigo**
(5 training frames coming from 2 cameras, threshold value = 0.55)

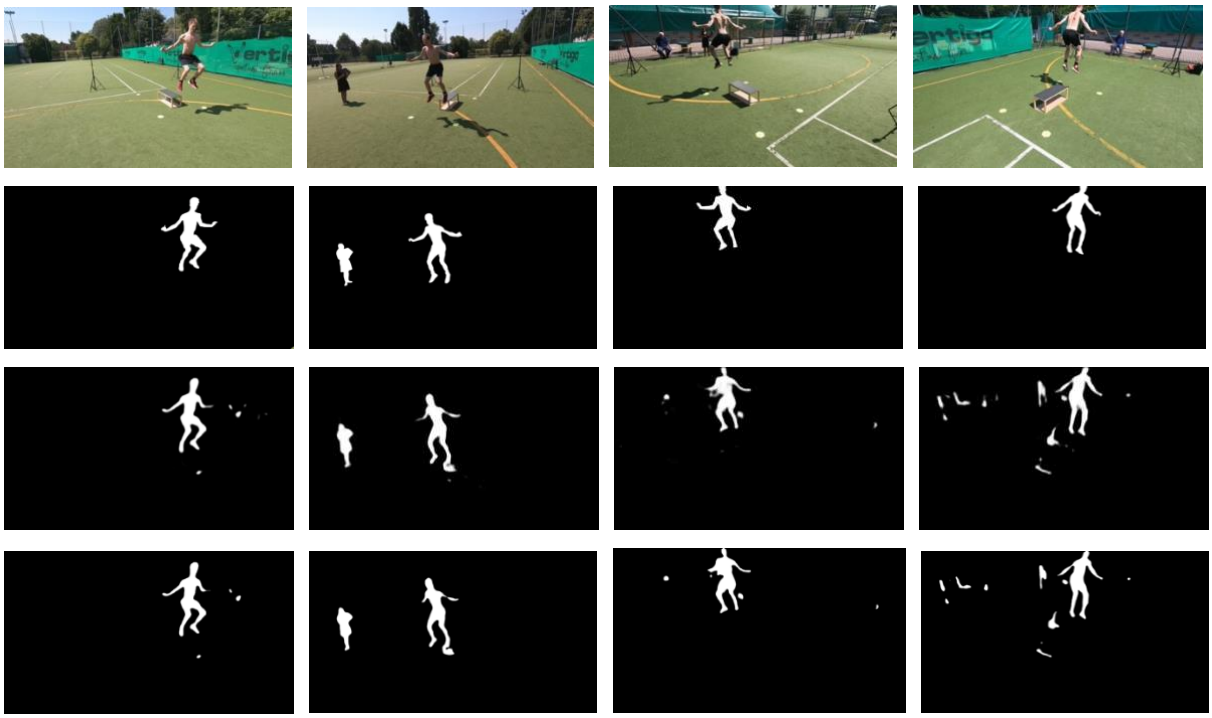

**Fig. 3.37**: Drop + Jump squat

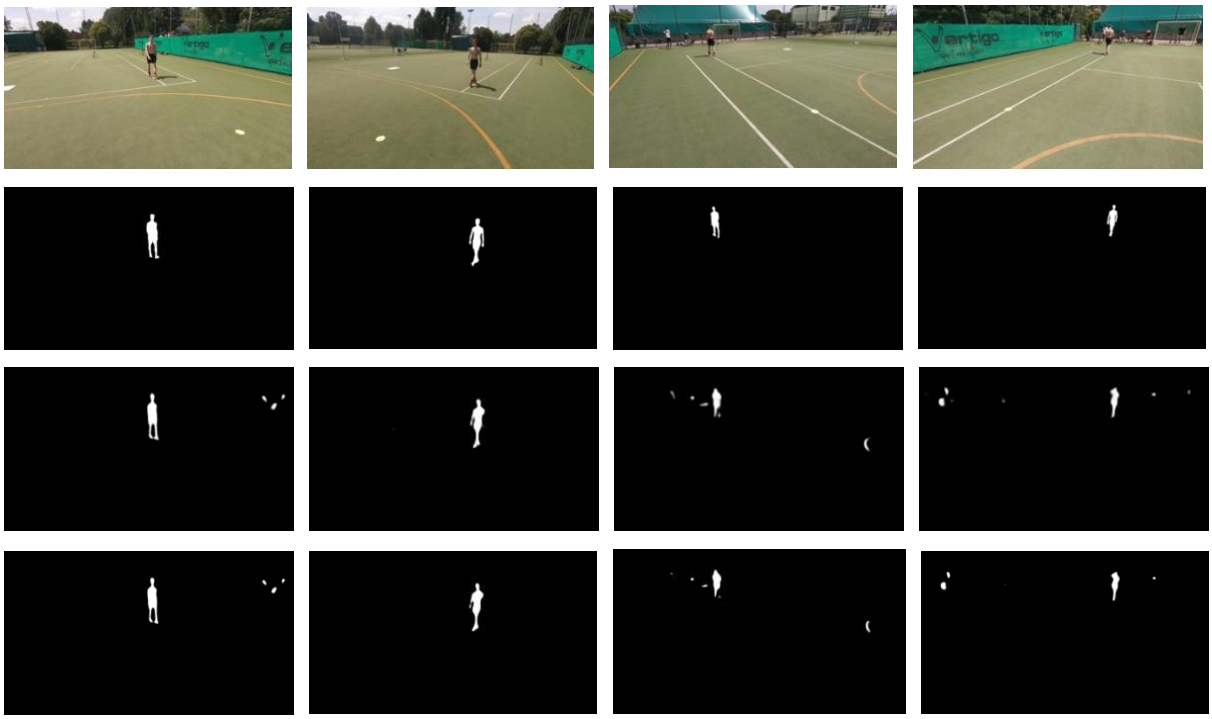

**Fig. 3.38**: Gait

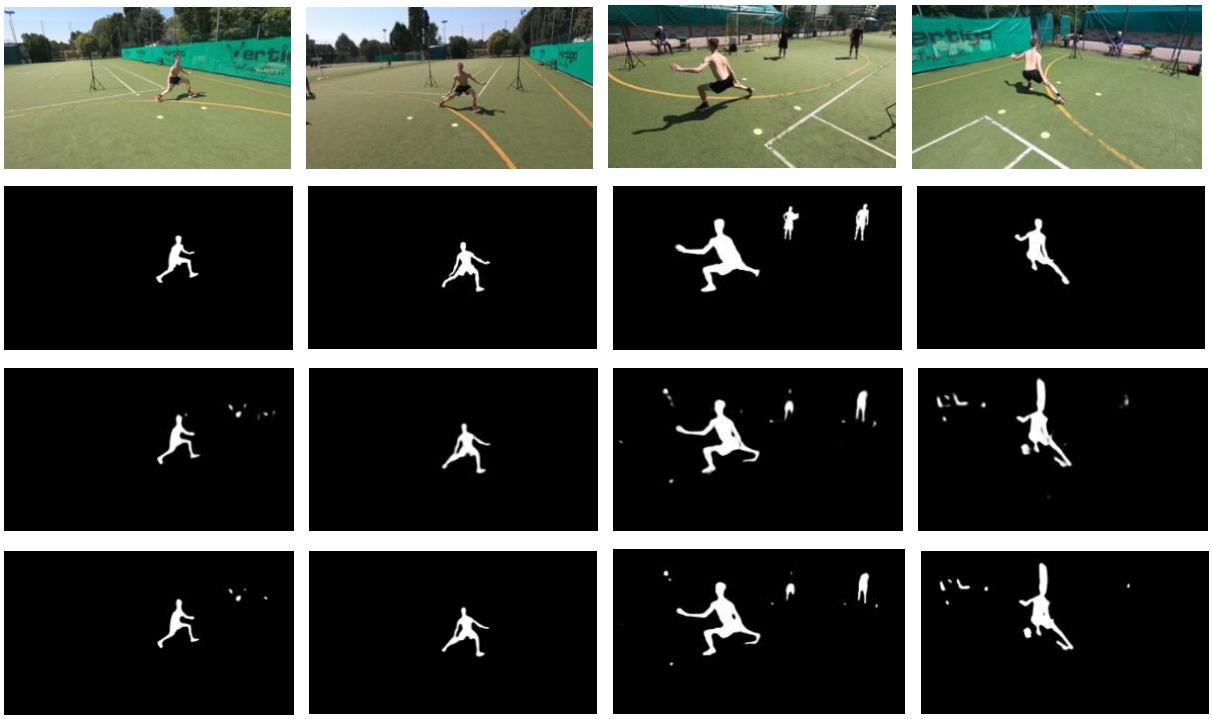

**Fig. 3.39**: Cutting maneuver

**Fig. 3.40**: Cutting maneuver

**2019-03-05 @CUS**

(7 training frames coming from 2 cameras, threshold value = 0.5)

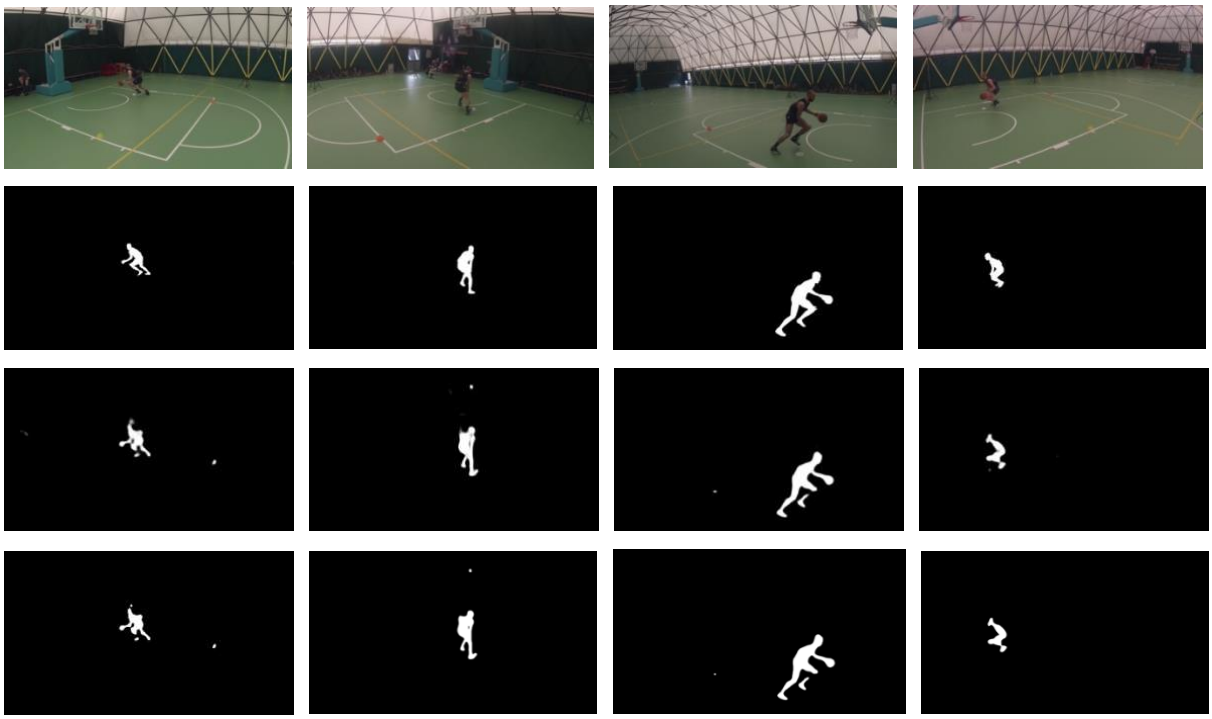

**Fig. 3.41**: Dribbling

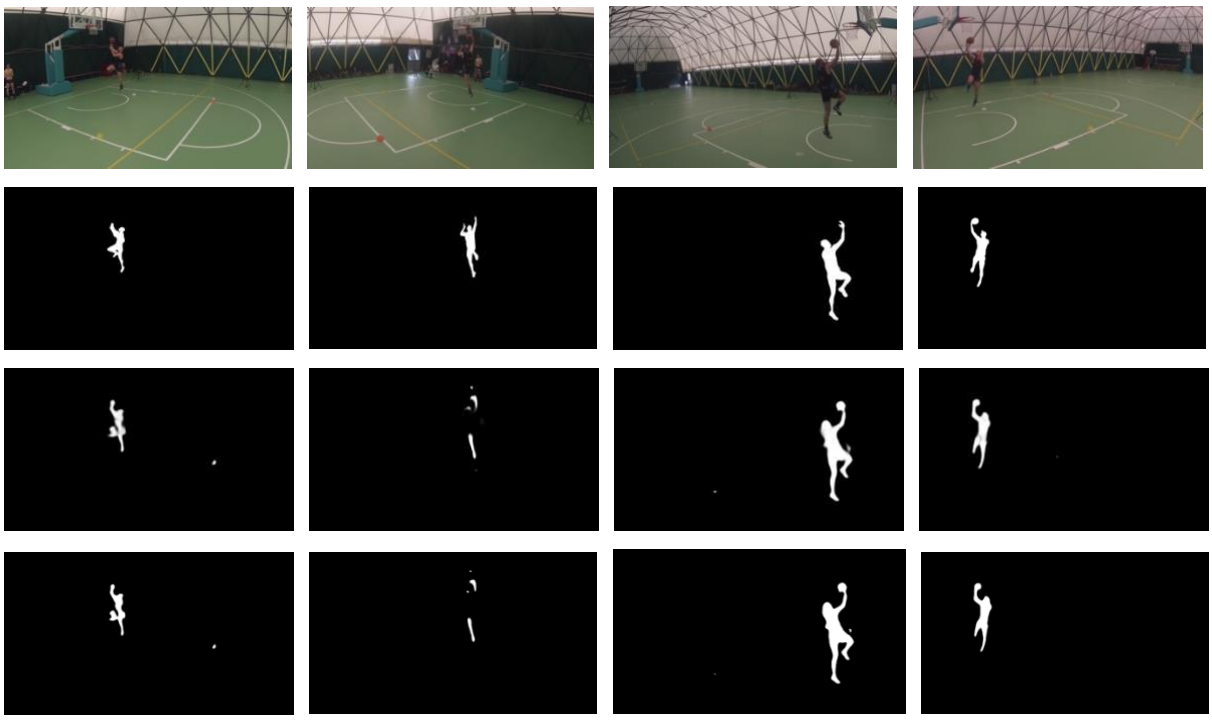

**Fig. 3.42**: Hoop

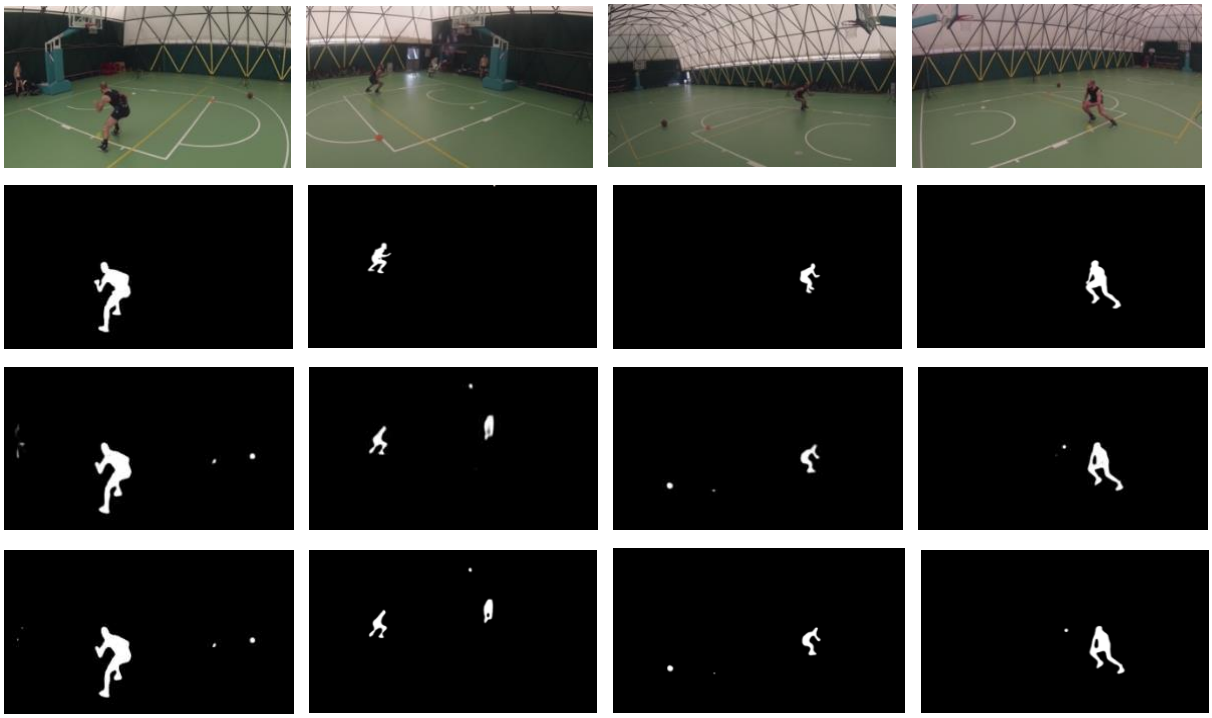

**Fig. 3.43**: Cutting maneuver

**Fig. 3.44**: Dribbling

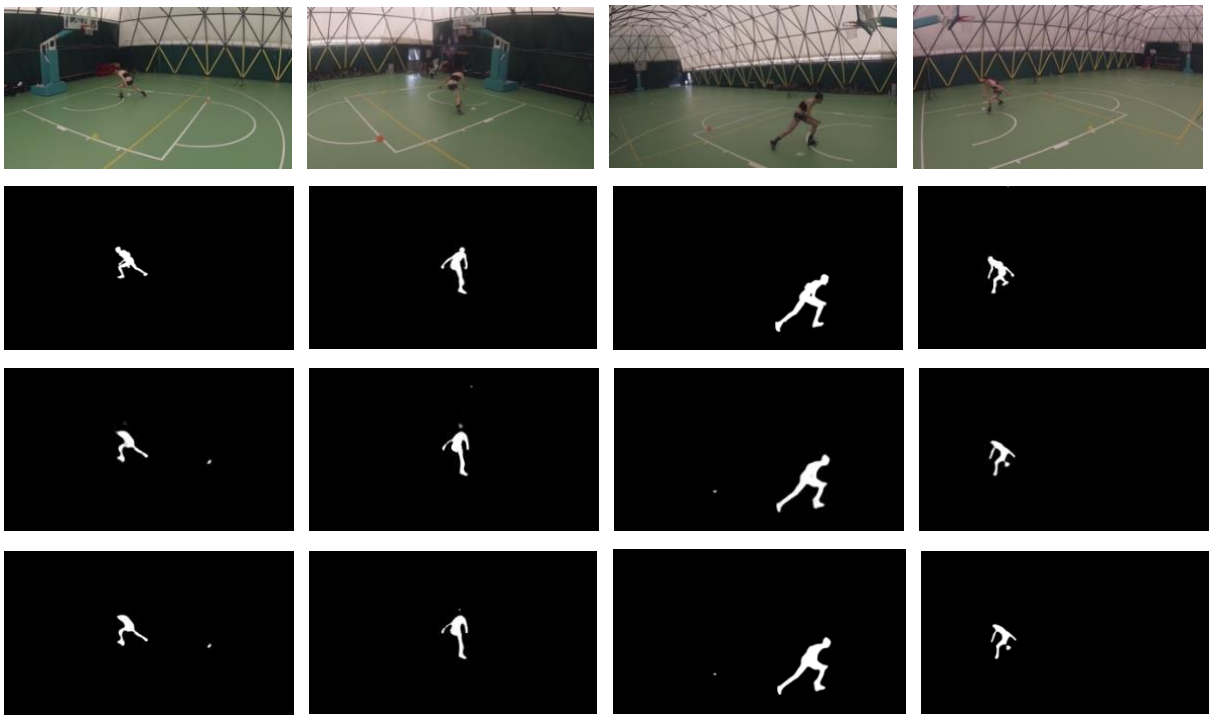

**Fig. 3.45**: Cutting maneuver

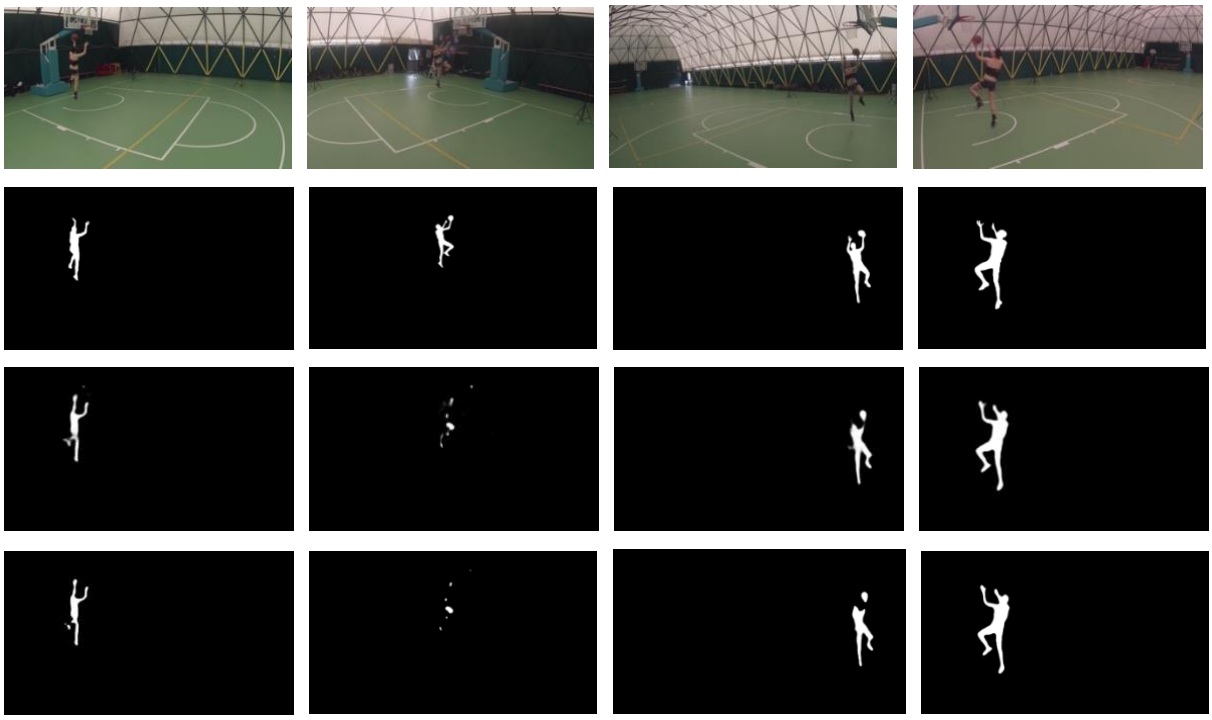

**Fig. 3.46**: Hoop

**cacb cam @Calcio Padova**

(41 training frames coming from 2 cameras, threshold value = 0.7)

**Fig. 3.47**: Cutting maneuver

**Fig. 3.48**: Cutting maneuver

**Calciatori Soldo Drop @lab**

(No training frames, threshold value = 0.8)

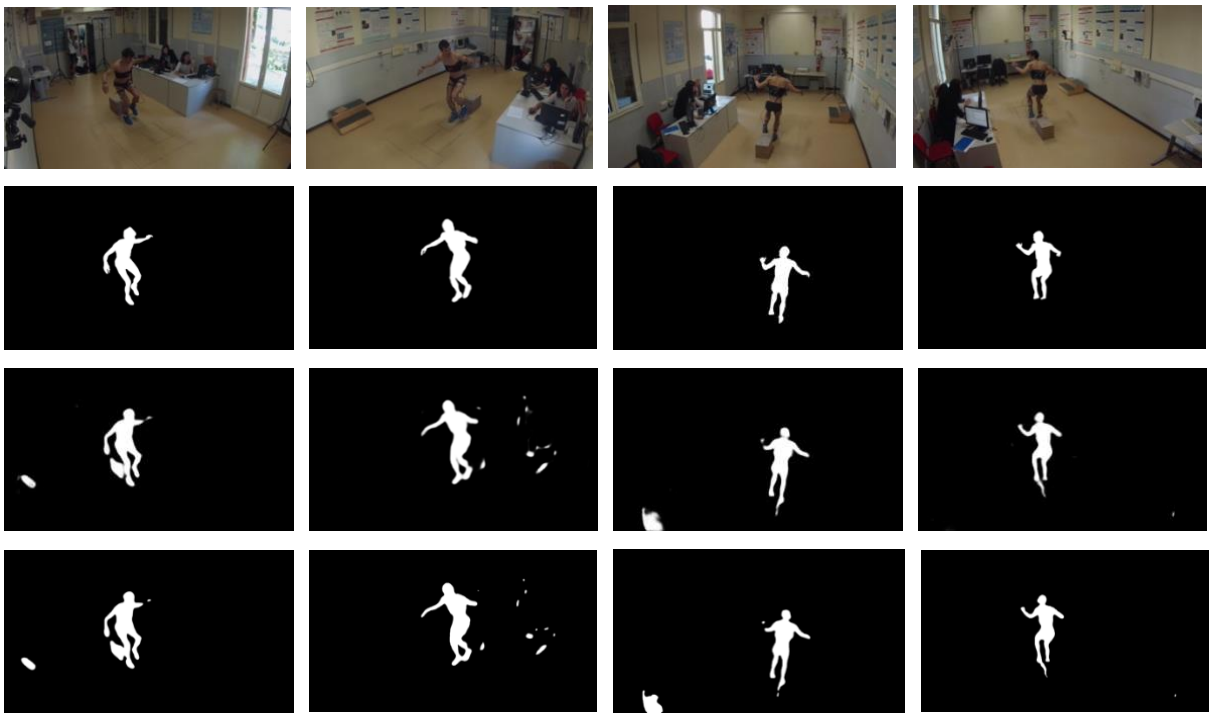

**Fig. 3.49**: Drop lunge

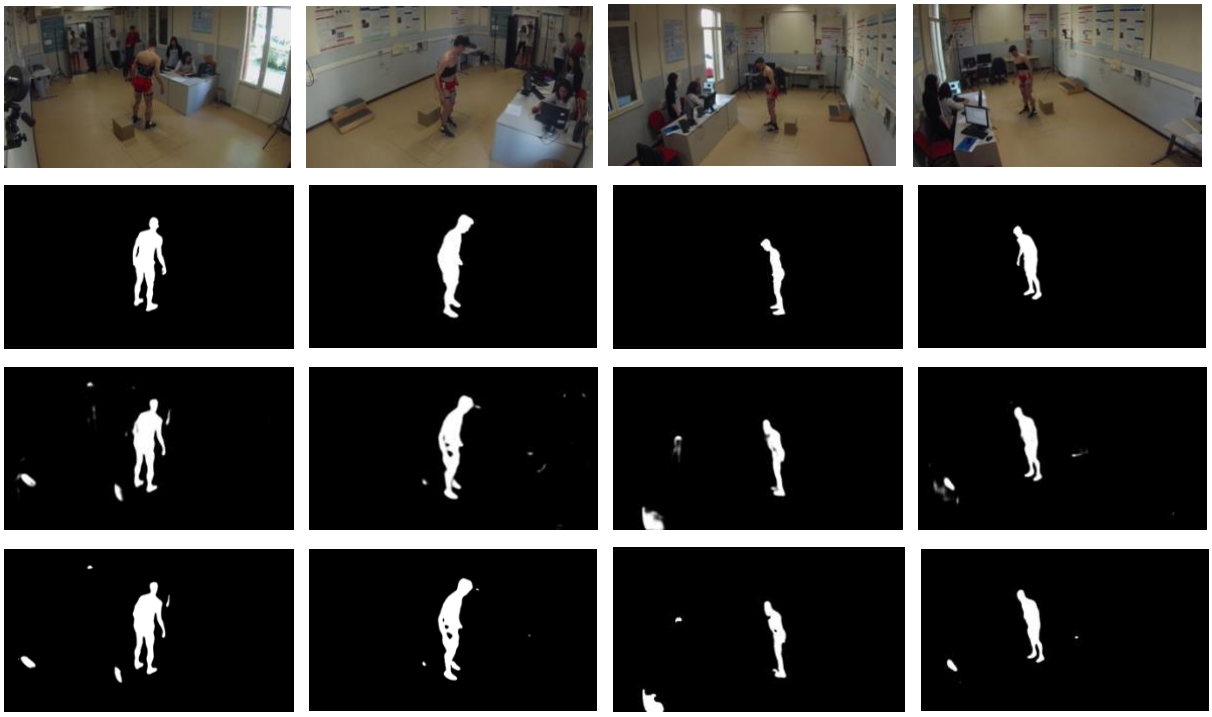

**Fig. 3.50**: Drop squat

**Pattinatrici @ MAGIC**

(4 training frames coming from one camera, threshold value = 0.8)

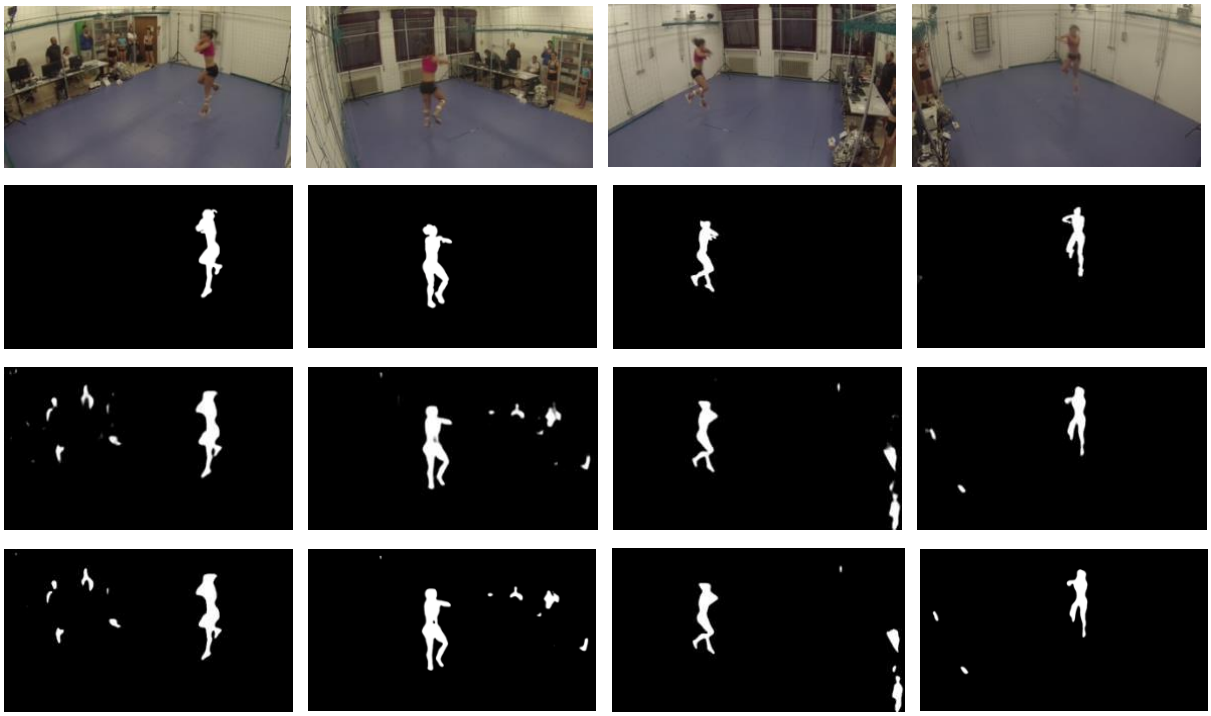

**Fig. 3.51**: Axel (pirouette)

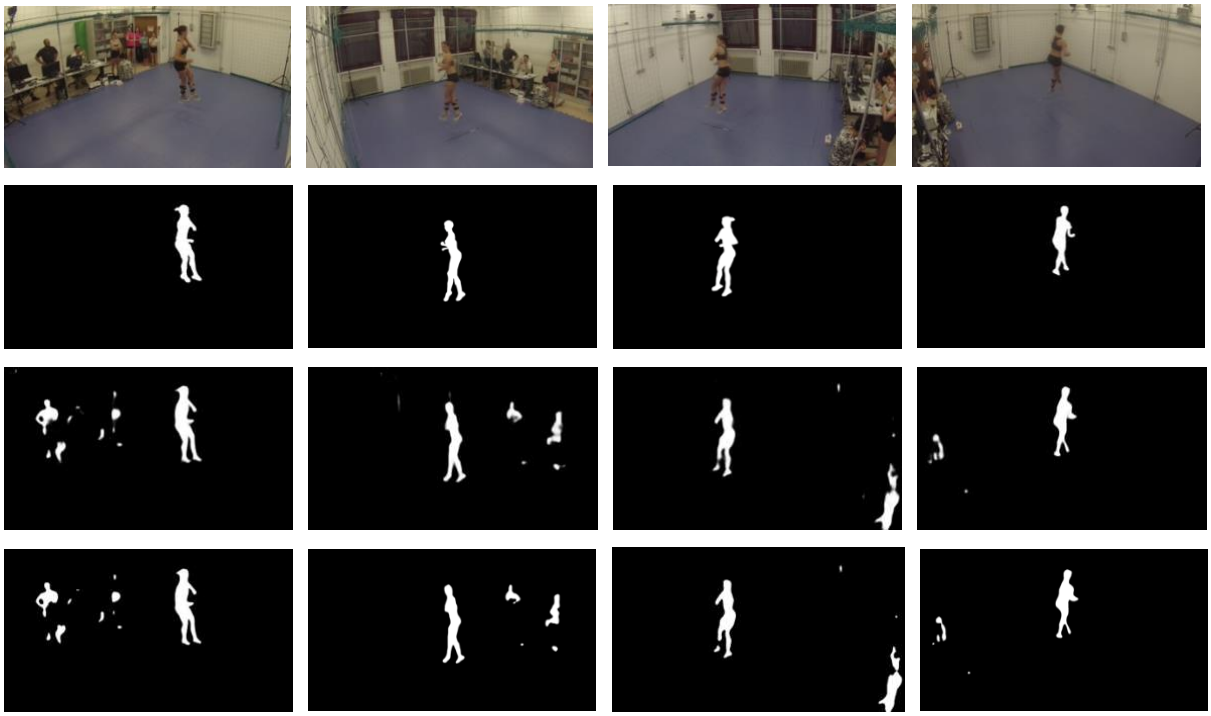

**Fig. 3.52**: Axel (pirouette)

## 3.4 Metrics Values

In addition to the mentioned metrics, another parameter used to evaluate FgSegNet_v2 performance is ROC AUC, namely the "Area Under the Receiver Operating Characteristic (ROC) Curve", a graph reporting the performance of a classification model at several classification thresholds.
This area may assume values, by definition, between 0 and 1, and the graph has as axes the true positive rate (TPR) and the false positive rate (FPR), whose formulas are written below:

$$TPR = \frac{TP}{TP+FN} \qquad FPR = \frac{FP}{FP+TN}$$

This section reports the metrics obtained for each dataset category, whose values come from the comparisons between ground truth images and binarized results.
The threshold values, which provided the highest F1 scores, correspond to the ones specified in the previous section and therefore have been omitted.

| | Accuracy | Precision | Recall | F1 Score | Cohen's Kappa | ROC AUC |
|---|---|---|---|---|---|---|
| Calciatori Iozzino | 0.995 | 0.924 | 0.917 | 0.918 | 0.915 | 0.999 |
| 2019-03-05 @Magic | 0.995 | 0.878 | 0.92 | 0.895 | 0.892 | 0.999 |
| 2019-03-05 @CUS | 0.997 | 0.841 | 0.819 | 0.815 | 0.813 | 0.998 |
| Pattinatrici | 0.991 | 0.719 | 0.849 | 0.775 | 0.771 | 0.996 |
| Calciatori Soldo Drop | 0.994 | 0.859 | 0.9 | 0.876 | 0.873 | 0.999 |

**Tab 3.1**: Indoor medical labs values

| | Accuracy | Precision | Recall | F1 Score | Cohen's Kappa | ROC AUC |
|---|---|---|---|---|---|---|
| Inter 22.05.2018 | 0.998 | 0.94 | 0.923 | 0.931 | 0.93 | 0.999 |
| csaPlantare | 0.995 | 0.828 | 0.865 | 0.841 | 0.839 | 0.999 |
| cacb cam | 0.997 | 0.847 | 0.89 | 0.866 | 0.865 | 0.999 |

**Tab 3.2**: Outdoor football pitches values

The following graphs report how F1 score changes depending on the applied threshold value per video category. As for the metrics tabs, environments have been divided into outdoor and indoor classes.

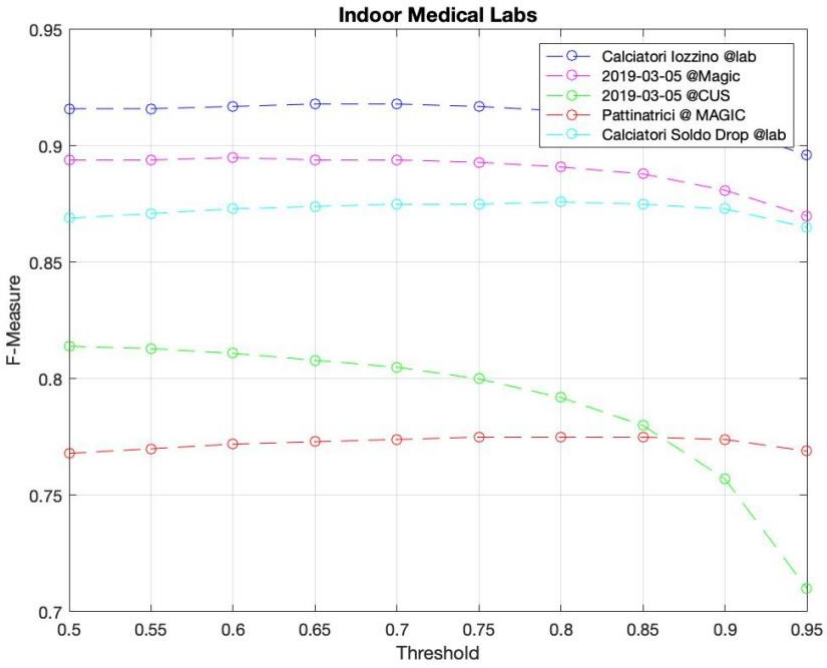


**Fig 3.53**: FgSegNet_v2 performance on indoor medical labs

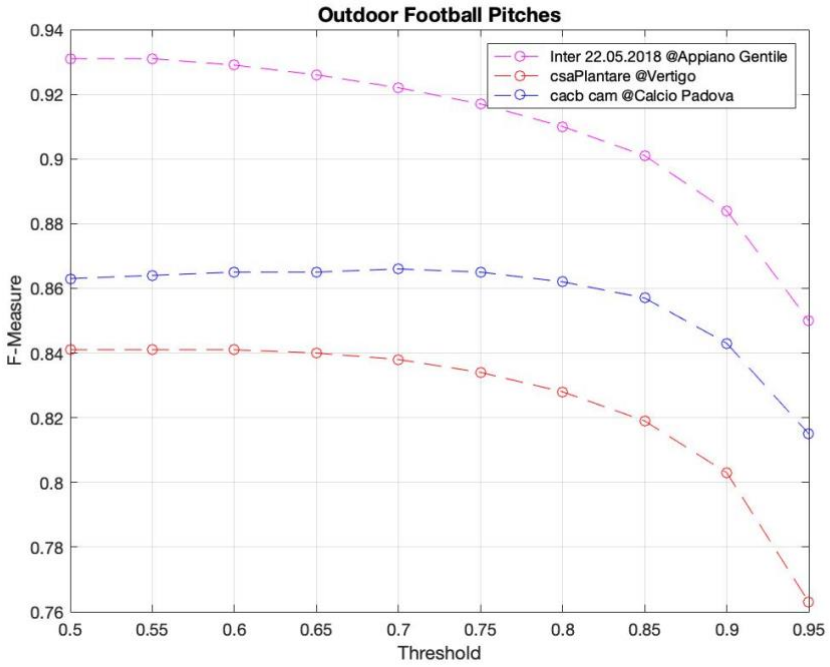


**Fig 3.54**: FgSegNet_v2 performance on outdoor football pitches

# 4. Discussion and Conclusions

This work consisted in exploring the use of the state-of-the art multi-scale CNN FgSegNet_v2 to achieve background subtraction in a non-scene-specific context.

Despite being used for handling a different task from the original experiment, the neural network reached satisfying results, reporting a F-measure score of 0.931 for the “Inter 22.05.2018 @ Appiano Gentile” dataset category.

As expected, the model was able to detect more easily targets in conditions of high contrast in colors between subjects and background; furthermore, higher scores in accuracy have been reached in the case of test samples that were very similar to the ones used during the training phase.

The neural network demonstrated to be able to tackle several issues related to background subtraction task, which predominantly involve dynamic background motion, illumination changes and shadows labeled as part of the foreground object.

The efficiency of this method mainly comes from its learning process, where the dataset used for training the model defined the input-output relationships.

Most of the false positive pixel masks correspond to objects that may resemble human body parts or static subjects (either fully visible or partially hidden) in the background. Among these latter, the ones which were closer to the camera and fully visible were considered as part of the foreground masks: this choice follows the model training strategy of including human figures at different scales to obtain a better performance in motion tracking.

Graphs have shown that F-measure starts to significantly decrease above a threshold value of ~ 0.7 for outdoor pitches, while it maintains similar values in the range [0.5, 0.85] for indoor environments.

Further developments may explore a different strategy in labeling (e.g., by removing all the subjects in the background), include the embodying of temporal information to remove static subjects, train the model on more environments to improve its adaptability to new contexts and its generalization capability, or develop two different models (i.e., one for the outdoor recording sessions and one for the indoor lab acquisition).

# Bibliography

[1] Lim L.A. & Keles H.Y., “Learning multi-scale features for foreground segmentation”, Springer, Pattern Analysis and Application (2020) 23:1369-1380

[2] Lim L.A. & Keles H.Y., “Foreground segmentation using convolutional neural networks for multiscale feature encoding”, Elsevier B.V., Pattern Recognition Letters 112 (2018) 256-262

[3] Simonyan K. & Zisserman A., “Very deep convolutional networks for large-scale image recognition”, arXiv:1409.1556v6 [cs.CV] 10 Apr 2015

[4] Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., “Dropout: a simple way to prevent neural networks from overfitting”, Journal of Machine Learning Research 15 (2014) 1929–1958

[5] Ulyanov D., Vedaldi A., Lempitsky V., “Instance Normalization: The Missing Ingredient for Fast Stylization”, arXiv:1607.08022v3 [cs.CV] 6 Nov 2017

[6] Tompson J., Goroshin R., Jain A., LeCun Y., Bregler C., “Efficient Object Localization Using Convolutional Networks”, arXiv:1411.4820v3 [cs.CV] 9 Jun 2015

[7] Zhang L. & Liang Y., “Motion human detection based on background subtraction”, 2010 Second International Workshop on Education Technology and Computer Science

[8] Bouwmans T., Javed S., Sultana M., Jung S.K., “Deep neural network concepts for background subtraction: A systematic review and comparative evaluation”, Elsevier Ltd., Neural Networks 117 (2019) 8-66

[9] https://www.bb-sof.com/

[10] Kanko R.M., Laende E., Selbie W.S., Deluzio K.J., “Inter-session repeatability of markerless motion capture gait kinematics”, Elsevier Ltd., Journal of Biomechanics 121 (2021) 110422

[11] Braham M. & Van Droogenbroeck M., “Deep Background Subtraction with Scene-Specific Convolutional Neural Networks”, IEEE International Conference on Systems, Signals and Image Processing (IWSSIP), Bratislava, Slovakia, pages 1-4, May 2016.

[12] Cun Y. L., Bottou L., Haffner P., “Gradient-based learning applied to document recognition”, Proceedings of IEEE, 86, 2278-2324, 1998

[13] Garcia-Garcia B., Bouwmans T., Rosales Silva A.J., “Background subtraction in real applications: Challenges, current models and future directions”, Computer Science Review, Volume 35, 2020, 100204, ISSN 1574-0137

[14] Mathis A., Schneider S., Lauer J., Mathis M.W., “A Primer on Motion Capture with Deep Learning: Principles, Pitfalls, and Perspectives”, Neuron, 108, October 14, 2020, Elsevier Inc.
[15] Johansson, G., “Visual perception of biological motion and a model for its analysis”, Percept. Psychophys, 14, 201-211, 1973
[16] Muendermann L., Corazza S., Andriacchi T.P., “The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications”, Journal of NeuroEngineering and Rehabilitation 2006, 3:6
[17] Ceseracciu E., Sawacha Z., Cobelli C., “Comparison of Markerless and Marker-Based Motion Capture Technologies through Simultaneous Data Collection during Gait: Proof of Concept”, PLOS ONE, March 2014, Volume 9, Issue 3, e87640
[18] Smale K.B., Potvin B.M., Shourijeh M.S., Benoit D.L., “Knee joint kinematics and kinetics during the hop and cut after soft tissue artifact suppression: Time to reconsider ALC injury mechanisms?”, J. Biomech. 62, 132-139
[19] Menolotto M., Komaris D., Tedesco S., O’Flynn B., Walsh M., “Motion Capture Technology in Industrial Applications: A Systematic Review”, Sensors 2020, 20, 5687
[20] Cui Z., Chen W., Chen Y., “Multi-Scale Convolutional Neural Networks for Time Series Classification”, arXiv:1603.06995v4 [cs.CV] 11 May 2016
[21] Vafadar S., Skalli W., Bonnet-Lebrun A., Khalifé M., Ranaudin M., Hamza A., Gajny L., “A novel dataset and deep learning-based approach for marker-less motion capture during gait”, Elsevier, Gait & Posture 86 (2021) 70-76
[22] Nakano N., Sakura T., Ueda K., Omura L., Kimura A., Iino Y., Fukashiro S., Yoshioka S., “Evaluation of 3D Markerless Motion Capture Accuracy Using OpePose With Multiple Video Cameras”, Front. Sports Act. Living 2:50 (2020)
[23] https://docs.opencv.org
[24] Kanko R.M., Laende E.K., Strutzenberger G., Brown M., Selbie W.S., DePaul V., Scott S.H., Deluzio K.J., “Assessment of spatiotemporal gait parameters using a deep learning algorithm-based markerless motion capture system”, Journal of Biomechanics 122 (2021), 110414, Elsevier Ltd.
[25] Deng J., Dong W., Socher R., Li L.-J, Kai Li, Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.
[26] Lim K., Jang W.-D., Kim C.-S., “Background Subtraction Using Encoder-Decoder Structured Convolutional Neural Network”, IEEE AVSS 2017, 987-1-5386-2939-0

[27] Sakkos D., Liu H., Han J., Shao L., “End-to-end video background subtraction with 3D convolutional neural networks”, Springer Nature, Multimed Tools Appl (2018) 77:23023-23041
[28] Mondéjar-Guerra V., Rouco J., Novo J., Ortega M., “An end-to-end deep learning approach for simultaneous background modeling and subtraction”, BMVC (2019)
[29] Wang Y., Luo Z., Jodoin P-M., “Interactive deep learning method for segmenting moving objects”, Pattern Recognition Letters 96 (2017) 66-75, Elsevier B.V.
[30] Bautista C., Dy C., Manalc M., Orbe R., Cordel M., “Convolutional neural network for vehicle detection in low resolution traffic videos”, TENCON 2016, 2016
[31] Bakkay M.C., Rashwan H.A., Salmane H., Khoudour L., Puig D., Ruichek Y., “BScGAN: Deep Background Subtraction with Conditional Generative Adversarial Networks”, ICIP 2018, 978-1-4799-7061-2, IEEE
[32] Y. Wang, P. -M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth and P. Ishwar, “CDnet 2014: An Expanded Change Detection Benchmark Dataset”, 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 393-400, doi:10.1109/CVPRW.2014.126.
[33] Low W.S., Chan C.K., Chuah J.H., Tee Y.K., Hum Y.C., Salim M.I.M., Lai K.W., “A Review of Machine Learning Network in Human Motion Biomechanics”, Springer Nature, J Grid Computing (2022) 20:4
[34] Tompson J., Jain A., LeCun Y., Bregler C., “Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation”, arXiv:1406.2984v2 [cs.CV] 17 Sep 2014
[35] Lan W., Dang L., Wang Y., Wang S., “Pedestrian Detection Based on YOLO Network Model”, Proceedings of 2018 IEEE, International Conference on Mechatronics and Automation August 5-8, Changchun, China
[36] Toshev A., Szegedy C., “DeepPose: Human Pose Estimation via Deep Neural Networks”, arXiv:1312.4659v3 [cs.CV] 20 Aug 2014
[37] Mathis A., Mamidanna P., Cury K.M., Abe T., Murthy V. N., Mathis M.W., Bethge M., “DeepLabCut: markerless pose estimation of user-defined body parts with deep learning”, Nature Neuroscience, Vol 21, September 2018, 1281-1289
[38] van der Kruk E., Reijne M.M., “Accuracy of human motion capture systems for sport applications; state-of-the-art review”, (2018) European Journal of Sport Science, 18:6, 806-819
[39] Halilaj E., Rajagopal A., Fiterau M., Hicks J.L., Hastie T.J., Delp S.L., “Machine learning in human movement biomechanics: Best practices, common pitfalls, and new opportunities”, Elsevier Ltd., Journal of Biomechanics 81 (2018) 1-11

[40] Corazza S., Muendermann L., Chaudari A.M., Demattio T., Cobelli C., Andriacchi T.P., “A Markerless Motion Capture System to Study Musculoskeletal Biomechanics: Visual Hull and Simulated Annealing Approach”, Biomedical Engineering Society, Annals of Biomedical Engineering, Vol. 34, No.6, June 2006, pp. 1019-1029
[41] Hsu W.-Y., Lin W.-Y., “Ratio-and-Scale-Aware YOLO for Pedestrian Detection”, IEEE Transactions on Image Processing, Vol. 30, 2021