الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Khider – Biskra
Faculté des Sciences et de la Technologie
Département : Génie Electrique
Ref :………………

جامعة محمد خيضر بسكرة
كلية العلوم و التكنولوجيا
قسم: الهندسة الكهربائية
المرجع:……….

Thèse présentée en vue de l'obtention
du diplôme de

Doctorat en sciences en : Automatique

Approche de suivi visuel d'objet basée sur la Transformée de Cosinus Discret DCT et les reseaux de neurones convolutionnels CNN

Présentée par :
NEBBAR HANANE

Soutenue publiquement le ../../2024

Devant le jury composé de :

Pr.salim SBAA	**Professeur**	**Président**	**Université de Biskra**
Pr. Nadjiba TERKI	**Professeur**	**Rapporteur**	**Université de Biskra**
Pr. Yahia KOURD	**Professeur**	**Examinateur**	**Université de Souk Ahras**
Pr. Kheireddine CHAFAA	**Professeur**	**Examinateur**	**Université de Batna**

الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research

MOHAMED KHIDER UNIVERSITY - BISKRA

Thesis submitted to the department of electrical engineering in candidacy for the Degree of Doctorate in Electrical Engineering

Specialty: Automatic

Option: Modeling and Control of Dynamic Systems

Visual object tracking approach based on Discrete Cosine Transform DCT and Convolutional Neural Networks CNN

Presented by:
NEBBAR HANANE
Discussed publicly :

In front the jury consists of:

President:	**Pr. salim SBAA**	**Prof**	**University of Biskra**
Supervisor:	**Pr. Nadjiba TERKI**	**Prof**	**University of Biskra**
Examiner:	**Pr. Yahia KOURD**	**Prof**	**University of Souk Ahras**
Examiner:	**Pr. Kheireddine CHAFAA**	**Prof**	**University of Batna**

Abstract

This thesis presents a groundbreaking Visual Object Tracking (VOT) approach designed to tackle a prevalent challenge in existing methods: the considerable alteration in object appearance, primarily stemming from extensive occlusion and varying illumination conditions.

The proposed method integrates several key components, including Deep Convolutional Neural Networks (DCNN), Discrete Cosine Transform (DCT), Histograms of Oriented Gradients (HOG) features, and an HSV-based energy condition. Initially, an HSV-based energy condition enriches the learning process by merging both RGB and HSV color bases, enhancing the model's adaptability. Rather than relying on the image template, this technique utilizes the coefficients derived from the image's DCT to handle high saturation images in the Convolutional Neural Networks (CNN's) input. Extracting CNN features involves utilizing the Inverse Discrete Cosine Transform (IDCT).

Subsequently, the approach harnesses multichannel correlation maps generated by CNNs to precisely determine the target's position. This is achieved through the amalgamation of convolutional features. Newton's method plays a pivotal role in this process, bolstering the system's long-term memory regarding the target's appearance and aiding in recovery from tracking failures.

Further, the updating parameter for the correlation filters is determined by selecting the highest value among the output maps derived from correlation filters using convolutional features extracted from the HOG features of the image template. The conclusive results unequivocally establish the superiority of the proposed method, surpassing the performance of most recent tracking techniques.

Keywords**:** Convolutional neural network, Discrete Cosine Transform (DCT), Correlation filter, Visual tracking, Newton’s method.

Résumé

Cette thèse introduit une approche novatrice de suivi visuel d'objets (VOT) conçue pour résoudre un défi majeur rencontré dans les méthodes existantes : l'altération considérable de l'apparence des objets, principalement causée par une occlusion étendue et des conditions d'éclairage changeantes.

La méthode proposée intègre plusieurs éléments clés, incluant les Réseaux de Neurones Convolutifs Profonds (DCNN), la Transformation en Cosinus Discret (DCT), les Histogrammes de Gradients Orientés (HOG), ainsi qu'une condition énergétique basée sur l'espace colorimétrique HSV. Initialement, une condition énergétique basée sur le HSV enrichit le processus d'apprentissage en fusionnant les bases de couleurs RVB et HSV, améliorant ainsi l'adaptabilité du modèle. Plutôt que de se baser sur un modèle d'image, cette technique utilise les coefficients issus de la DCT de l'image pour gérer les images à forte saturation dans l'entrée des Réseaux de Neurones Convolutifs (CNN). L'extraction des caractéristiques CNN implique l'utilisation de la Transformation Inverse en Cosinus Discret (IDCT).

Ensuite, l'approche exploite les cartes de corrélation multi-canaux générées par les CNN pour déterminer précisément la position de la cible. Ceci est accompli en fusionnant les caractéristiques convolutives. La méthode de Newton joue un rôle central dans ce processus, améliorant la rétention à long terme des caractéristiques de la cible par le système.

De plus, le paramètre de mise à jour des filtres de corrélation est déterminé en sélectionnant la valeur la plus élevée parmi les cartes de sortie dérivées des filtres de corrélation, en utilisant les caractéristiques convolutives extraites des caractéristiques HOG du modèle d'image. Les résultats concluants établissent sans équivoque la supériorité de la méthode proposée, dépassant les performances des techniques de suivi les plus récentes.

***Mots clés* :** Réseaux de neurones à convolution profonde, Transformation Cosinus Discrète (DCT), Filtre de Corrélation, Suivi Visuel, Méthode de Newton.

ملخص

تقدم هذه الأطروحة نهجًا مبتكرًا لتتبع الكائنات المرئية (VOT) مصممًا لمواجهة التحدي السائد في الأساليب الحالية: التغيير الكبير في مظهر الكائن، والذي ينبع في المقام الأول من الانسداد الواسع النطاق وظروف الإضاءة المتغيرة.

تدمج الطريقة المقترحة العديد من المكونات الرئيسية، بما في ذلك الشبكات العصبية التلافيفية العميقة(DCNN) ، وتحويل جيب التمام المنفصل(DCT) ، وميزات الرسوم البيانية للتدرجات الموجهة(HOG) ، وحالة الطاقة المستندة إلى .HSV في البداية، تعمل حالة الطاقة المستندة إلى HSV على إثراء عملية التعلم من خلال دمج قواعد الألوان RGB وHSV، مما يعزز قدرة النموذج على التكيف. بدلاً من الاعتماد على قالب الصورة، تستخدم هذه التقنية المعاملات المشتقة من DCT للصورة للتعامل مع الصور عالية التشبع في مدخلات الشبكات العصبية التلافيفية (CNN) يتضمن استخراج ميزات CNNاستخدام تحويل جيب التمام المنفصل العكسي(IDCT) .

بعد ذلك، يستخدم هذا النهج خرائط الارتباط متعددة القنوات التي تم إنشاؤها بواسطة شبكات CNN لتحديد موقع الهدف بدقة. يتم تحقيق ذلك من خلال دمج الميزات التلافيفية. تلعب طريقة نيوتن دورًا محوريًا في هذه العملية، حيث تعمل على تعزيز ذاكرة النظام طويلة المدى فيما يتعلق بمظهر الهدف والمساعدة في التعافي من فشل التتبع.

علاوة على ذلك، يتم تحديد معلمة التحديث لمرشحات الارتباط عن طريق تحديد أعلى قيمة بين خرائط الإخراج المستمدة من مرشحات الارتباط باستخدام الميزات التلافيفية المستخرجة من ميزات HOG لقالب الصورة. وتثبت النتائج الحاسمة بشكل لا لبس فيه تفوق الطريقة المقترحة، متجاوزة أداء أحدث تقنيات التتبع.

***الكلمات المفتاحية*:**الشبكة العصبية التلافيفية، تحويل جيب التمام المنفصل (DCT)، مرشح الارتباط، التتبع البصري، طريقة نيوتن.

Dedication

First and foremost, I thank Allah Almighty for the strength and patience he has given me to accomplish this work.

I dedicate this work to my mother "Allah Yarhmha", and to my father

Myhusband and brother and sisters for their precious assistant and encouragement.

All my family.

All my friends and colleagues.

All my teachers.

And for you ...

Acknowledgements

*First and foremost, I extend my heartfelt gratitude to **ALLAH** for bestowing upon me the strength and determination required to accomplish this modest undertaking. I am deeply appreciative of the unswerving support and guidance provided by Professor **TERKI Nadjiba** and **Mohammed BOURENNANE** throughout this endeavor.*

My thanks also go to the jury members for having agreed to review and evaluate this work:

Pr. Salim sbaa	*professor at the University of Biskra and the jury president.*
Pr. Yahia kourd	*professor at the University of Souk Ahras.*
Pr. Kheireddine chafaa	*professor at the University of Batna*

.

*I also extend my sincere thanks and appreciation to **abida toumi** and **Medouakh Saadia***

Finally, I express my gratitude to my family and friends for their patience, encouragement, and invaluable support that greatly aided me throughout my work.

Table of Contents

Chapter I
Introduction

Chapter II
State-Of-The-Art

Chapter III
Methods and tools for object tracking

Chapter IV
Proposed methods for visual object tracking

Chapter V
Experminent results

List of Figures

List of Figures

List ofTables

List of Tables

List ofAcronyms

List of Acronyms

VOT	:	Visual Object Tracking
CNN	:	Convolutional Neural Networks
DCNN	:	Deep convolutional neural networks
LWT	:	Lifting Wavelet Transforms
ILWT	:	Integer Lifting Wavelet Transforms
FFT	:	Fast Fourier Transform
IFFT	:	Inverse Fast Fourier Transform
DCT	:	Discrete Cosine Transform
HOG	:	Histograms of Oriented Gradients
SOT	:	Single Object Tracking
MOT	:	Multiple Object Tracking
PCA	:	Principal Component Analysis
MIL	:	Multiple Instance Learning
KLT	:	Kanade-Lucas-Tomasi
MOSSE	:	Minimum Output Sum of Squared Error
KCF	:	Kernelized Correlation Filters
CSR-DCF:		Channel and Spatial Reliability of Discriminative Correlation Filter
CIE	:	Commission Internationale de l'Eclairage
HSV	:	Hue Saturation Value
RGB	:	Red, Green and Blue color

ReLU	:	Rectified Linear Unit
ILSVRC	:	ImageNet Large Scale Visual Recognition Challenge
SVHN	:	Street View House Number
KCF	:	Kernels correlation filter
DSST	:	Discriminative Scale Space Tracker
SRM	:	Spatial Reliability Maps
DFT	:	Discrete Fourier transform
OTB	:	Object Tracking Benchmark dataset
AUC	:	Area Under the Curve
DP	:	Distance Precision
OS	:	Overlap Success
OPE	:	One-Pass Evaluation
OS	:	Overlap Success
TC128	:	Temple Color 128 dataset
UAV	:	Unmanned Aerial Vehicles

Chapter I

Introduction

Summary

Introduction

I.1. Context

Visual tracking poses as one of the most complex problems in computer vision, finding applications in diverse fields like human-computer interaction, video surveillance, and unmanned driving. The main objective in generic visual tracking algorithm is to anticipate the path of a target through a series of images, starting from its initial position. Nonetheless, creating a fast and dependable tracker is a formidable undertaking due to numerous hurdles, such as occlusion, swift motion, and distortion. Moreover, the scarcity of training samples further complicates the development of an efficient and robust tracking system. To address these challenges, various pioneering trackers have been proposed, leading to significant advancements in tracking performance and robustness. Notably, discriminative-filter-based trackers [1] have garnered considerable attention owing to their competitive performance. Typically, visual tracking methods can be divided into two main types: generative approaches and discriminative methods [2]. Discriminative approaches have witnessed significant progress based on correlation filters, and Examples of successful tasks include object identification, image segmentation, and image classification, which have been effectively achieved [3]. In recent times, DCNN have gained significant popularity as a prominent method in visual tracking [4]. The utilization of CNN for human tracking was introduced in [5]. The VGG-Net-19 model was utilized to train three adaptive correlation filters in [6]. The effectiveness of this approach was evaluated using contemporary methods. However, despite its advantages, the sustainability of long-term tracking was found to be limited[7].

Huang et al [8] employed reinforcement learning to train an early decision policy, resulting in improved speed for object tracking using CNN. Similarly, Wang et al [9] introduced an approach that involves utilizing an automatic denoising encoder stack to learn generic features for visual tracking. Furthermore, He et al. [10] developed a two-part Siamese network consisting of a semantic branch and an appearance branch, aiming to enhance the discrimination capabilities of SiameFC in tracking. Nevertheless, despite the advantages offered by these techniques, the challenge of sustaining long-term tracking effectiveness persists. In an attempt to overcome this challenge, several researchers have endeavored to improve tracking performance by integrating feature representations from various CNN layers with correlation filters [11, 12]. In [13], the authors have presented an effective hybrid image fusion method that combines the Integer Lifting Wavelet Transforms (ILWT) and the DCT. This method is capable of generating fused images with superior visual quality, making it a potential solution to mitigate certain visual tracking issues.

The utilization of the DCT in visual tracking has received limited attention in the existing literature [14], despite its effectiveness in diverse visual applications like image retrieval [15], face recognition [16], and video object segmentation [17]. In [18], authors introduced a particle filter framework that integrated a sparse appearance model based on structural local DCT, which included occlusion detection for visual tracking.

In recent times, HOG features have emerged as a valuable tool for addressing various challenges in detection and classification. Notably, the successful identification of faces [19] has been accomplished by leveraging the magnitudes and orientations of image derivatives. Y. Wei et al. [20] have introduced a Haar-HOG-based technique, which has shown promise by delivering remarkable speed and efficiency compared to algorithms relying solely on Haar-like features or isolated HOG descriptors. Additionally, this proposed method demonstrated a lower false positive rate and a higher detection rate when compared to techniques that solely rely on the HOG descriptor.

I.2. Contributions

The key contributions outlined in this thesis are as follows:

- By applying the HSV energy condition, we tackle the issue of light variation in individual color frames. Our approach allows to opt for either RGB or HSV color bases. Additionally, the DCT has the capability to capture pertinent spatial frequency information. Notably, in the top left corner of the corresponding 2-D DCT matrix, a concentrated cluster of low-frequency coefficients is observed.

- Given the significance of integrating feature representations from multiple CNN layers, as exemplified in [21], a HCF model has been devised.

- HOG features derives from the 2D-DCT coefficients, utilizing them as a basis instead of the original image. This introduction of 2D-DCT coefficients aims to enhance the performance of HOG features. Moreover, a technique has been formulated to counteract model drift, aiding in the identification of alterations in appearance and demonstrated superior empirical results for both object detection and real-time tracking [22]. This involves the utilization of Newton's method to compute the maximum value within the maps generated through correlation filters. Unlike [23], we compute the convolutional feature products derived from the HOG features extracted from an existing image template. Subsequently, this computed value is employed as a parameter for updating the correlation filters.

- The proposed approach evaluates using a comprehensive benchmark dataset known as OTB50, which consists of 50 challenging image sequences.

I.3.Thesis Organization

This thesis work constitutes of five chapters.

Chapter II : State-Of-The-Art

In this chapter we present the state-of-the-art of visual tracking. We give a detailed introduction to visual tracking, and we also give the challenges they encounter and two different types of tracking algorithms.

Chapter III : Methods and tools for object tracking

We give in this chapter a detailed of image colors spaces, and imports methods of Image Features extraction, Convolutional neural networks and Histograms of Oriented Gradients .

Chapter IV : Proposed methods for visual object tracking

This chapter presents in detail the main steps of our methods.

Chapter V : Experminent results

In this chapter, first, we present Benchmark Datasets, followed by an assessment of visual object tracking performance. Second, a comprehensive examination of results and discussions pertaining to each database is provided.

Chapter II

State-Of-The-Art

Sommary

State-Of-The-Art

II.1. Introduction

Object tracking is the process of spatially and temporally pinpointing a moving entity within a video sequence. In every method employed for tracking moving objects, an object detection system is a necessity, either in each frame or at the moment when the object initially emerges in the video.

The task of tracking objects across a sequence of video frames involves the extraction and analysis of data from intricate images. This challenge becomes even more formidable when real-time constraints are imposed. Within the literature, a wide array of tracking methods is documented. Most of the standard methods cannot effectively address demanding scenarios, such as substantial occlusion, deformation, rapid motion, changes in scale, and variations in illumination, all of which have a direct impact on tracking performance.

In this section, we will delve into the core principles and real-world uses of tracking algorithms, as well as the challenges they encounter. Furthermore, we will scrutinize the two different types of tracking algorithms. Lastly, we will introduce state-of-the-art approaches for object tracking, classified into several categories such as Tracking by detection, Tracking by correspondence, Tracking by correlation filter, and more, as well as the advancement in Visual Object Tracking Technology.

II.2. Object Tacking Applications

Over the past decade, there has been a noticeable upsurge in interest regarding object tracking within video sequences, largely owing to its extensive array of potential applications. Here are several significant domains where this technology finds application:

- Video surveillance involves the detection, tracking, and recognition of people's activities.
- Human-Computer Interactions, such as gesture recognition and augmented.

- reality, frequently leverage object tracking to enhance interactions between humans and computers.
- Robotics employs this technology to assist autonomous robots in navigating complex environments by maneuvering through obstacles during avoidance phases.
- Vehicle monitoring, traffic management, and analytic systems benefit from theutilization of object tracking.
- Military applications utilize it to guide missiles and track targets, enhancing accuracy and precision in combat operations.
- In healthcare, medical imaging utilizes object tracking to monitor and assess movement or changes, aiding in diagnosis and treatment planning.

II.3. The Challenges in Object Tracking

Visual object tracking plays a pivotal role in the realm of Computer Vision by providing critical trajectory data for behavior analysis through the prediction of an object's status in a video. Despite the extensive research and notable progress made in recent years [21][22][23][24], object tracking remains a highly intricate challenge [25]. There isn't a universally applicable tracking solution capable of efficiently handling all scenarios. Multiple factors influencing the effectiveness of object tracking algorithms encompass [25][26]:

▪**Deformation :**This scenario occurs when all elements of the tracked object move and rotate in conjunction with one another, leading the tracker to perceive them as a unified and rigid entity.

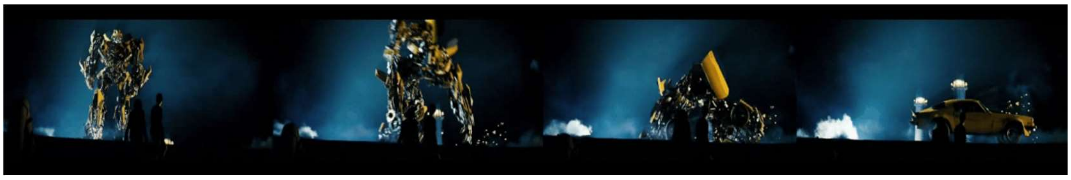


Fig. II.1. Example of Deformations

▪ **Occlusion :** This can be a challenging scenario, as it can arise in consecutive frames when the target, or parts of it, become obscured by either the background or other objects. In such instances, updating the appearance model becomes crucial as it directly impacts the accuracy of the target's position estimation in subsequent input frames.

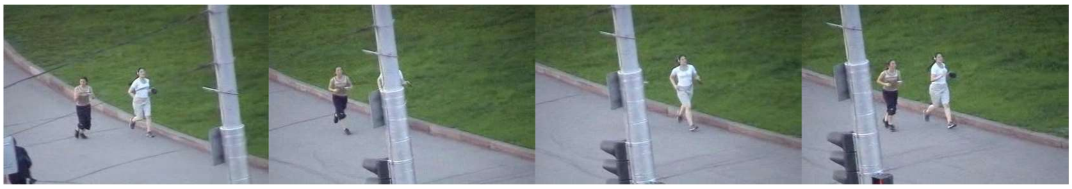


Fig. II.2. Example of Occlusions

- **Background clutters :** This situation may arise when the object's features closely resemble those of its background, making it so that even a slight alteration in the object's appearance could lead the tracker to perceive the background as more akin to the target than the actual target itself.

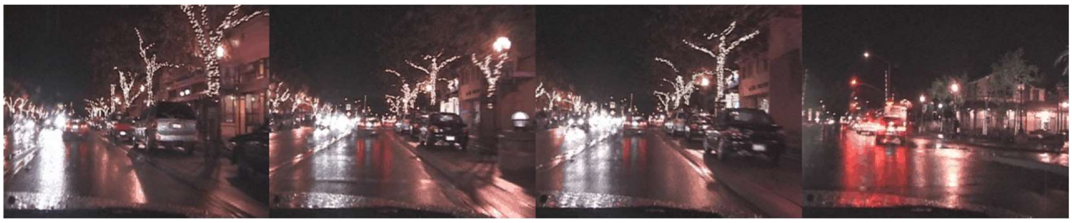


Fig. II.3. Example of Background Clutter

- **Scale variation :** This situation frequently arises as a result of the camera's proximity or distance from the target. To mitigate this issue, resizing the input frame is a fundamental technique employed in visual tracking.

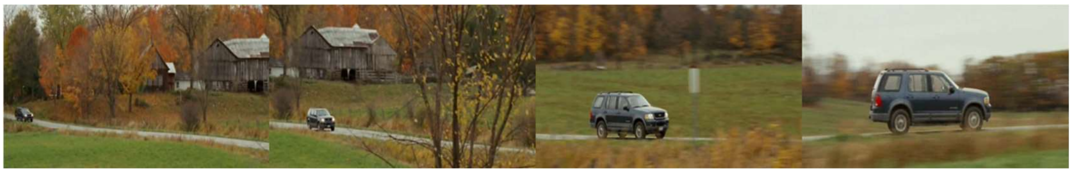


Fig. II.4. Example of Scale Variations

- **Fast Motion :** This represents another crucial scenario in visual object tracking, involving rapid or substantial movements of the object and/or the camera between two successive input images. Such motion can result in tracker drift.

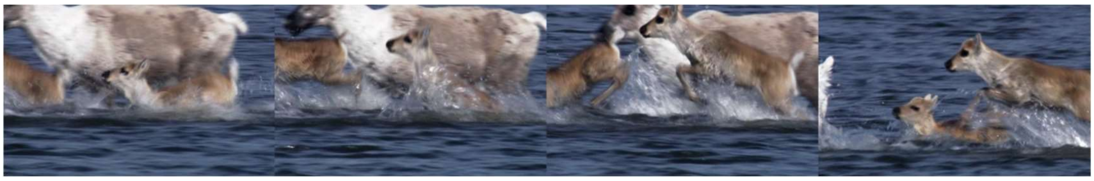


Fig. II.5. Example of Fast Motion

- **Blur Motion :** It is when the rapid motion of an object results in a disruption or distortion in its appearance.

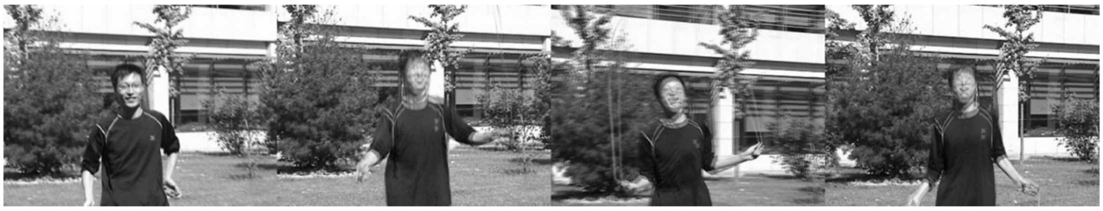


Fig. II.6. Example of Motion Blur

- **Illumination variation :** The tracked object can experience direct or indirect influences from changes in the lighting conditions, whether caused by environmental factors or material properties. This issue arises particularly when there are light fluctuations or movingprojectors, which can indeed impact the effectiveness of feature extraction from images. Various approaches tackle this challenge by utilizing invariant features to account for variations in lighting conditions [27,28].

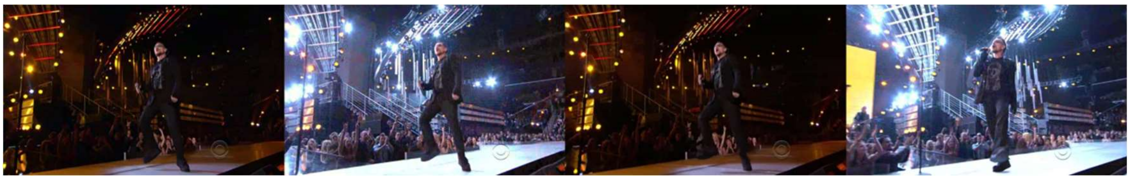


Fig. II.7. Example of Illumination Variations

- **In-Plane Rotation :** Videos have the capability to visually convey rotations that occur within a two-dimensional image plane. One familiar illustration of this phenomenon is observing the image of a motorcycle rider who has turned backward, revealing the side view.

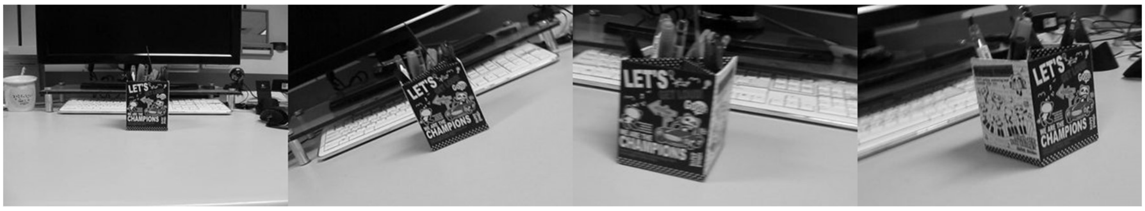


Fig. II.8. Example of In-plane Rotations

- **Out-of-Plane Rotation :** This type of rotation is distinct from rotations that may occur out of the image plane, as it has the potential to result in the vanishing or concealment of specific sections of the target.

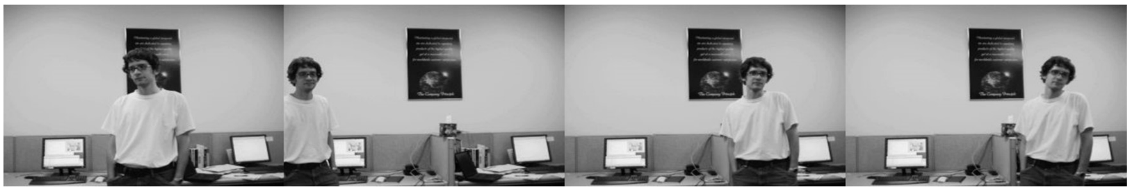


Fig. II.9. Example of Out-of-Plane Rotations

- **Out of view:**Numerous tracking techniques struggle to reacquire the target when it traverses the image's periphery. In such situations, the target may vanish from view or exhibit movement within the video.

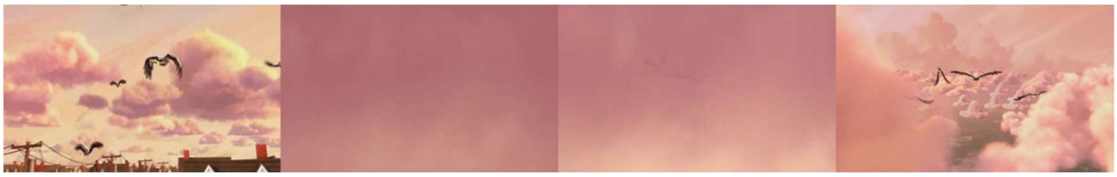


Fig. II.10. Example of Out-of-View

- **Low resolution:**Another significant factor is low resolution, which can be described as a decrease in the amount of information available from the target's patch. This reduction in information diminishes the accuracy of location estimation. Typically, this arises from using a low-resolution camera or the substantial physical distance between the camera and the object.

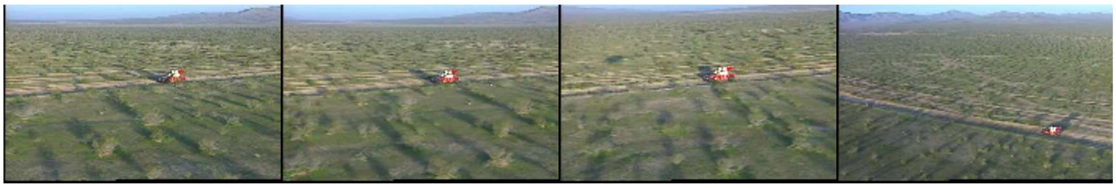


Fig. II.11. Example of Low Resolution

II.4.Types of Object Tracking Algorithms

The categorization of object tracking tasks can be determined by the number of objectstracked within a sequence, as outlined in [29].

II.4.1.SingleObjectTracking (SOT)

SOT algorithm is designed to monitor the movement of a solitary object within a video sequence. Its success lies in its ability to track a single object, even when the surrounding environment contains multiple objects. The SOT process involves the initial selection of a region of interest in the first frame of the video and subsequently tracking the object's position (i.e., its coordinates) in the subsequent frames of the video. In this study, we will explore a selection of algorithms utilized for tracking single objects.

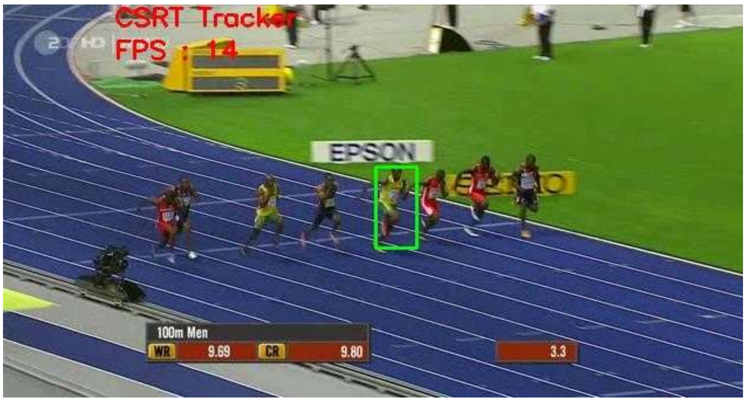


Fig. II.12. SingleObjectTrackingExample

II.4.2.Multiple Object Tracking (MOT)

MOT involves the challenge of monitoring the movement of more than one object within a video. In this scenario, the algorithm attributes a distinct variable to each of the objects detected in the video frame. It then proceeds to recognize and track all these multiple objects across successive frames of the video.

Given that a video may contain a substantial number of objects, or the video quality might be suboptimal, leading to ambiguity in the direction of an object's motion, MOT becomes a demanding task, often relying on single-frame object detection. The process of MOT is illustrated in figure II.13.

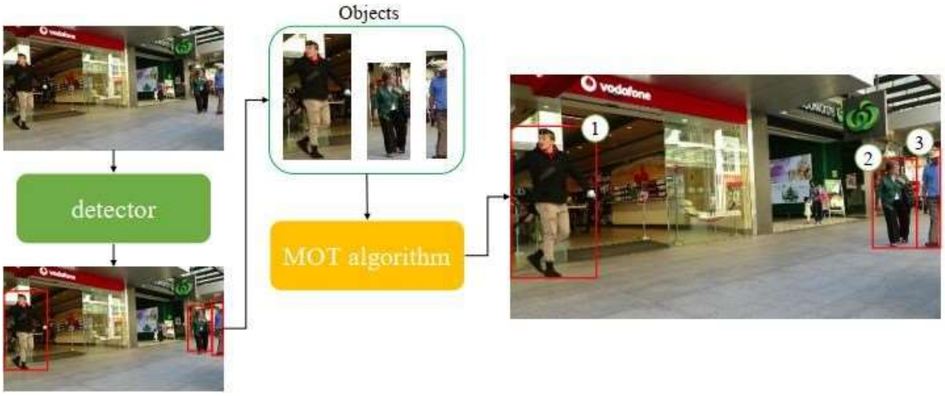


Fig. II.13. MultipleObjectTrackingProcess

II.5.Object tracking methods

Numerous object tracking methodologies have been put forth, and the distinctions between these approaches primarily arise from the selection of object representation, shape, image features, and the nature of motion estimation. This choice is contingent on the specific application and the characteristics of the processed video. Various categorizations of tracking algorithms exist in the literature. Yilmaz et al. [21] have proposed one classification based on the object representation employed, distinguishing between point tracking, kernel tracking, and silhouette tracking. More recently, another categorization has emerged, based on the appearance model utilized. In [30] and [31], the authors categorize tracking methods into two groups: generative and discriminative approaches. Generative methods are centered on modeling the object's appearance, which may vary between frames. Discriminative methods, on the other hand, separate the object from the background, transforming the tracking problem into a binary classification task [32]. Li *et al*. [31] offer a comprehensive overview of the existing appearance models in tracking, delving into their visual representation and statistical modeling of appearance. In this section, we will utilize a classification of track

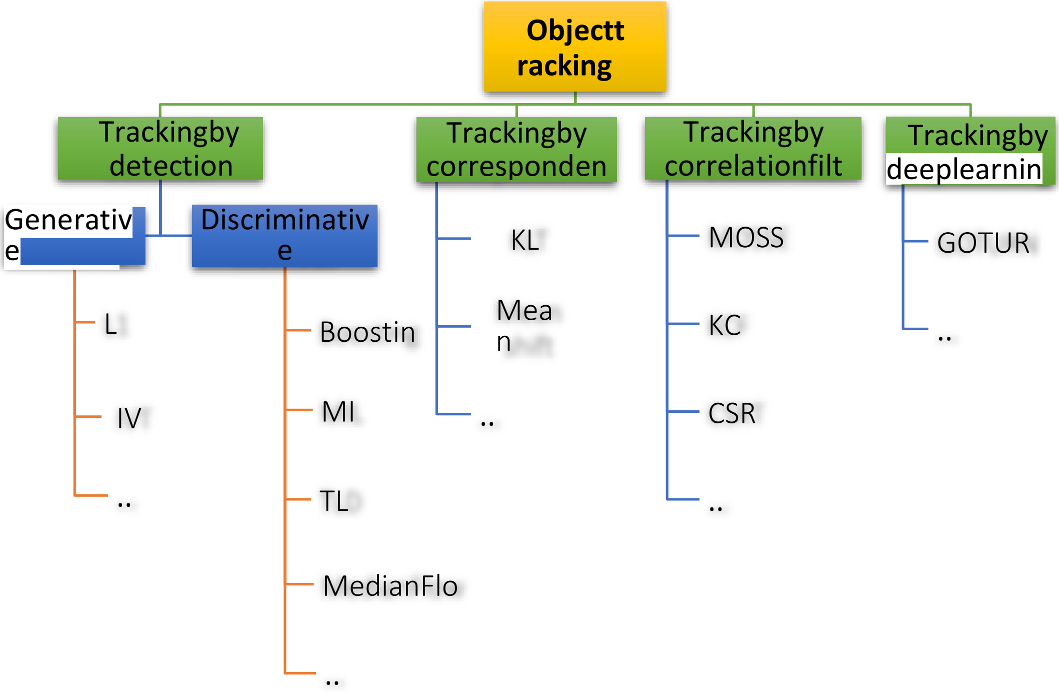


Fig. II.14. Taxonomyofobjecttracking methods.

II.5.1.Tracking by detection

Over the recent years, tracking by detection has emerged as a highly successful approach in the field of visual object tracking, reaching the pinnacle of performance. The tracking-by-detection paradigm primarily comprises two crucial elements: visual representation and statistical modeling. Within this section, our attention will be directed towards the most noteworthy publications pertaining to statistical modeling, which can be classified into two distinct categories: generative and discriminative models [33].

II.5.1.1. Generative Method

Generative modeling-based tracking primarily revolves around the concept of acquiring a model that accurately represents the target object and subsequently employs this model to identify the most analogous region in subsequent frames [33]. As a rule, methods falling into this category do not necessitate an extensive training dataset [34].

- **Tracker L1:** Mei and Ling [25] introduced a resilient tracking technique that perceives object tracking as an approximation challenge through rigorous handling of noiseand occlusion. In the course of tracking, potential targets are portrayed as a sparse linear combination of model sets, which encompass both target models derived from previous frames and simple models. The L1 tracker demands substantial computational resources because of the numerous L1 minimization computations involved.

- **Tracker IVT:** Ross and colleagues [26] introduced a tracking algorithm that employs an incremental subspace model for characterizing the target object, allowing it to accommodate changes in appearance. This algorithm facilitates the gradual learning of an object representation subspace, specifically using Principal Component Analysis (PCA), and adjusts the model by incorporating the updated appearance of the object while considering a forgetting factor for past appearances. However, this approach exhibits limited robustness, particularly in cases where the object's location is imprecise.

II.5.1.2. Discriminative Method

While generative modeling has achieved some success, it often encounters challenges when trying to depict the target object while disregarding background information. This is especially evident when the appearance of the target object undergoes significant changes or when the background is cluttered. In contrast, discriminative modeling approachesdiscriminative modeling approaches the problem of object tracking as a classification task, aiming to differentiate the target object from the background. Consequently, it tends to exhibit greater robustness in complex scenarios by explicitly modeling the background as negative training samples. Tracking methods based on discriminative modeling have advanced swiftly and have taken the lead in most datasets in recent years [33].

- **Boosting Tracker:**

Grabner *et al*. [35] employed a comparable online boosting framework for real-time object tracking in [33]. This approach is grounded in the online iteration of the AdaBoost algorithm, where the algorithm elevates the weights of inaccurately classified objects, enabling a weak classifier to concentrate on their identification. As the classifier is trained in an "online" manner, the user designates the frame containing the tracked object. Initially, this object is treated as a positive detection result, with objects in its proximity considered as background. Upon receiving a new image frame, the classifier assigns scores to the surrounding detection pixels from the previous frame, and the new position of the object is determined in the area where the score attains the highest value [35].

- **MIL Tracker:**

Babenko *et al*. [36] suggested the adoption of Multiple Instance Learning (MIL) in the context of visual object tracking. This approach empowers the classifier to choose from a variety of potential positive samples based on its current state. In the MIL tracking scheme, training samples are viewed as "bags." A bag is designated as positive if it comprises at least one positive instance; otherwise, it is categorized as negative [33].

- **TLD:**

Kalal *et al*. [37] introduced a resilient visual tracking algorithm. This algorithm breaks down the long-term tracking task into three sub-tasks: tracking, learning, and detection. The corresponding elements of these sub-tasks are integrated to create a comprehensive tracker known as TLD. The tracking component is responsible for estimating object motion and maintaining continuous object tracking to generate smooth trajectories. However, it also accumulates errors over time and may lose track if the target becomes invisible. The detection component is tasked with localizing the object in all its previously observed appearances and reinitializing the tracker when it encounters failure. The learning process assesses the quality of the results and updates them with only the most reliable outcomes [33].

- **MEDIANFLOW Tracker:**

This algorithm utilizes the Lucas-Kanade method as its foundation. It employs a bidirectional time-based approach to monitor object movement and calculates the error in these trajectories. This capability enables the tracker to make real-time predictions about the future position of the object.

II.5.2.Tracking by correspondence

Utilizing the correspondence between a target object's representation in two successive frames is an intuitive method for inferring its motion and maintaining tracking continuity. This approach was prevalent in the early stages of tracking due to its reasonably effective performance, uncomplicated structure, and minimal computational demands [33].

KLT: As a more efficient template-matching technique, Lucas and Kanade [38] introduced theKanade-Lucas-Tomasi (KLT) tracker. The KLT tracker identifies affine correspondences that have undergone transformation between two consecutive frames by utilizing spatio-temporal derivatives. The tracker determines the new position of the target by matching its location in the previous frame to its position in the current frame through the estimated affine transformation.

Mean shift: Several strategies have emerged to address the challenge of tracking non-rigid objects effectively, leveraging the mean shift algorithm introduced by Fukunaga *et al*. in 1975. Originally designed for data clustering, Comaniciu *et al*. [39] adapted mean-shift trackers to perform matching with color histograms, which remains consistent even when target shapes change. In each new frame, the mean shift algorithm is employed to pinpoint the target's location by maximizing a similarity metric. Nevertheless, this tracker can face difficulties in regions with similar color distributions, primarily due to the absence of spatial information.

II.5.3. Tracking by Correlation Filter

In recent times, there has been a substantial surge of interest in correlation filter-based tracking methods, primarily because of their straightforward design, outstanding performance, and computational efficiency. Correlation filters, a fundamental tool in digital image processing, are employed to identify regions within an image that resemble a predefined template. Ideally, a correlation filter yields high responses when matched with a predefined template, while yielding low responses for background elements [33].

- **MOSSE Tracker:** Bolme *et al*. [40] introduced the Minimum Output Sum of Squared Error (MOSSE) filter for grayscale image-based visual tracking. This method relies on adaptive correlation calculations in Fourier space. The filter minimizes the sum of squared errors between the actual correlation output and the predicted correlation output. MOSSE filter-based tracking is known for its computational efficiency, achieving speeds of several hundred frames per second, and its robustness in handling variations in illumination, scale, pose, and non-rigid deformations.

- **KCF Tracker**: Henriques *et al.* [41] introduced the Kernelized Correlation Filters (KCF) tracker. This tracking method extends the concepts from the previous two trackers, BOOSTING and MIL. KCF leverages the observation that the multiple positive samples employed in the MIL tracker exhibit significant overlap. This overlap within the data gives rise to certain advantageous mathematical properties that the KCF tracker harnesses to enhance both tracking speed and accuracy simultaneously [24].

- **CSRT Tracker:** CSRT stands for the Channel and Spatial Reliability of Discriminative Correlation Filter (CSR-DCF) implementation. This algorithm employs spatial reliability maps to adapt the filter support to a specific region selected from the frame for tracking. This capability allows for an expanded search area and the tracking of non-rectangular objects. The reliability indices provide insights into the filter quality acrossdifferent channels and are utilized as weighting factors for localization. Consequently, by utilizing feature sets such as HoGs and Colornames, this algorithm demonstrates commendable performance [42] [43].

II.5.4.Tracking by Deep Learning

In recent years, Deep Learning approaches, particularly those employing CNNs, have exhibited significant empirical achievements and have emerged as dominant solutions for numerous computer vision challenges. However, the application of deep learning to the realm of visual object tracking presents unique challenges, primarily stemming from the scarcity of appropriate training data and the dynamic nature of target objects throughout video sequences [33].

GOTURN Tracker: TheGOTURN tracker, an acronym for "Generic Object Tracking Using Regression Networks," is an advanced tracking algorithm rooted in Deep Learning, specifically relying on CNN [44].

According to informationfrom the OpenCV documentation, it demonstrates robustness in the face of alterations in viewpoint, lighting conditions, and object deformations. However, it may not perform as effectively in scenarios involving object occlusions [45].from the

OpenCV documentation, it demonstrates robustness in the face of alterations in viewpoint, lighting conditions, and object deformations. However, it may not perform as effectively in scenarios involving object occlusions [45].

II.6.Evolution of Visual Object Tracking Technology

The evolution of visual object tracking algorithms exhibits a progression from traditional tracking approaches [46],[47],[48] to those rooted in deep learning [49],[50], and from generative methods [51],[52],[53] to discriminative techniques [54],[55],[56].

Between 2005 and 2010, the focal point of visual object tracking centered on the Bayesian framework, Particle filter, and Kalman filter, all falling under the generative methods category. During this time frame, object tracking predominantly involved the challenge of template matching. Manual design features were employed to create an appearance model for comparison, and a Gaussian distribution was utilized as the motion model to generate potential object candidates. The ultimate object state was determined by identifying the candidate with the highest similarity. Abdel-Hadi *et al.* [57] and Han *et al.* [58] introduced visual object tracking techniques based on the Kalman filter and particle filter, respectively, while Yang *et al.* [59] employed superpixel features in constructing the appearance model.

From 2010 to 2014, researchers extensively explored correlation filtering-based trackers using kernel methods, which are part of the discriminative methods category [60],[61],[62].

These trackers aimed to train a correlation filter to position the object center at the peak value in the response map after correlation filtering. Bolme *et al.* [61] applied correlation filters to locate eyes and developed the ASEF filter. Subsequently, Bolme *et al.* [63] improved the ASEF filter and applied it to visual object tracking, introducing the MOSSE tracker, the pioneer among correlation filter-based trackers. Henriques *et al.* [64] proposed the CSK tracker, which addressed the correlation filter using a linear classifier. In 2014, Henriques *et al.* [60] treated tracking as a ridge regression problem and employed a circulant matrix to collect positive and negative samples around the object for training the correlation filter. To tackle scale variation in the KCF tracker, Danelljan *et al.* [62] introduced two correlation filters: one for translation and one for scale estimation.

Between 2015 and 2017, the potent representation capabilities of deep features were incorporated into correlation filtering-based trackers [65],[66],[67]. These trackers leveraged well-pre-trained networks as feature extractors. Ma *et al.* [65] used a pretrained deep network to extract object deep features and amalgamated multi-features from different layers of the deep network to create the HCF tracker. Moreover, the characteristics of feature maps from different deep network layers were explored [67]. Hong *et al*. [66] introduced a learnable saliency map based on CNN and fused it with an SVM-based classifier to establish the appearance model. Danelljan *et al*. [67] introduced continuous convolution operators to amalgamate multiple resolution feature maps and achieve precise sub-pixel location.

From 2018 to 2020, deep learning-based tracking methods, particularly the Siamese network, made substantial progress in visual object tracking [68],[69],[70]. Bertinetto *et al*. [69] merged the Siamese network with the correlation filter to propose the SiameseFC tracker. Li *et al*. [68] incorporated the region proposal network into the Siamese network to provide object candidates, which effectively served as a motion model in visual object tracking. Wang *et al*. [70] unified visual object tracking with instance segmentation, enhancing tracking accuracy and instance segmentation speed.

Lately, to address online updates and few-shot learning challenges in deep learning-based trackers, Siamese network-based trackers have integrated online update strategies and meta-learning techniques to enhance tracking robustness [71],[70]. Zhang *et al*. [71] treated the update model as a function of ground truth from the first frame, the template from the last frame, and the current frame's appearance model. This function was expressed as a deep network and introduced the UpdateNet for model updates. Huang *et al*. [73] and Wang *et al*. [72] introduced meta-learning into the Siamese network-based tracking method to improve tracking performance robustness through network initialization.

II.7. Conclusion

A significant domain within the realm of computer vision pertains to real-time object tracking, a technology that enjoys widespread usage today. Within this chapter, we provided a comprehensive overview of object tracking and its associated methodologies. Initially, we delved into the foundational principles of object tracking, exploring its practical applications and addressing the challenges that can impact the performance of tracking algorithms. Moreover, we examined the two primary categories of tracking algorithms. In conclusion, we presented the cutting-edge methods for object tracking, which can be categorized into several groups, including Tracking by detection, Tracking by correspondence, Tracking by correlation filter, and more, as well as delving into the evolution of Visual Object Tracking Technology.

Chapter III

Methods and tools for object tracking

Sommary

Methods and tools for object tracking

III.1. Introduction

Color, a fundamental image attribute, serves as a cornerstone within diverse domains of computer vision, particularly in object tracking, owing to its effectiveness and efficiency. Within this domain, a multitude of feature choices are widely embraced, ranging from color labels to Histogram of Oriented Gradients (HOG), alongside features derived from deep CNN. The feature selection adopted for this thesis encompasses an array of elements extracted from both RGB and HSV color spaces, complemented by features derived from CNN and HOG methodologies.

III.2 Image Color Spaces

III.2.1RGB Color Space

The RGB color space is fundamental in computer technology, extensively used in image and video processing. This model relies on the additive combination of three primary colors—R (red), G (green), and B (blue)—due to its strong resemblance to human visual perception. Represented as a Maxwel cube following the Cartesian coordinate system, it was initially introduced by the Commission Internationale de l'Eclairage(CIE) in 1931 and continues as a primary standard, despite its limitations. One limitation involves the significant correlation among its channels, which merges luminance and chrominance data into each channel [186]. This intrinsic correlation presents a challenge for the RGB color space, affecting its efficacy in color analysis and recognition algorithms relying on color-based identification. Colors are expressed as (R, G, B) triples, for instance, (255, 0, 0) represents red, while (255, 255, 255) signifies white. By adjusting contributions from each primary color, any desired color can be generated. Conversely, specific colors can be deconstructed into their red, blue, and green constituents using equations III.1 to III.3 [74].

$$r = \frac{R}{R+G+B} \tag{III.1}$$

$$g = \frac{G}{R+G+B} \tag{III.2}$$

$$b = \frac{B}{R+G+B} \tag{III.3}$$

Another significant concern relates to the spectral composition, where a negative section within the spectra obstructs the reproduction of certain colors through the mere combination of three spectra. This limitation obstructs the precise replication of particular colors within the RGB color space, posing a hindrance to its comprehensive use for color representation and analysis.

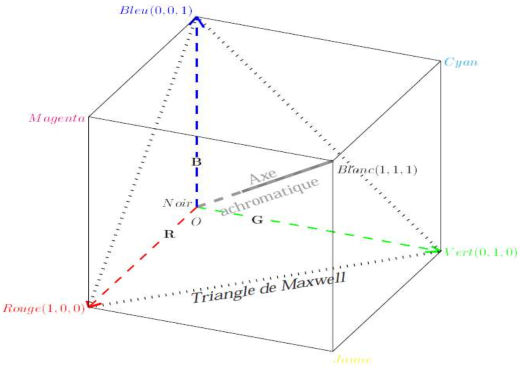


Fig III.1. Space Color Cube RGB.

III.2.2. Hue Saturation Value (HSV) Color

The HSV color space, an acronym for hue, saturation, value, emerges as a color model inspired by the human visual system. Its inception aimed to provide a more intuitive methodology for manipulating colors, closely mirroring color perception and interpretation. Widely employed within the realm of computer graphics, HSV embodies a nonlinear transformation that shifts from a cartesian coordinate representation (RGB) to a cylindrical coordinate framework (as illustrated in figure III.2). Tailored for a more instinctive color portrayal, HSV significantly streamlines the quantification of perceived colors. In this color space, color representation is encapsulated in a triplet: hue (*H*), saturation (*S*), and brightness (*V*). The transformations inherent in HSV delineate the following aspects:

$$\begin{aligned} M &= \min(R,G,B) \\ V &= \max(R,G,B) \end{aligned} \tag{III.4}$$

$$H = \begin{cases} 0, & if\ R = G = B \\ 60 \times \dfrac{G-B}{V-M}, & if\ V = R \\ 60 \times \dfrac{B-R}{V-M} + 120, & if\ V = G \\ 60 \times \dfrac{B-R}{V-M} + 240, & if\ V = B \end{cases} \tag{III.5}$$

$$S = \begin{cases} 0, & if\ V = 0 \\ \dfrac{V-M}{V}, & Otherwise \end{cases} \tag{III.6}$$

Hue H traverses from 0 to 1.0, exhibiting colors ranging from red, transitioning through yellow, green, cyan, blue, magenta, back to red. Saturation S spans from 0 to 1.0, depicting colors varying from unsaturated (comprising shades of gray) to fully saturated hues (devoid of any white component). Value V or brightness progresses from 0 to 1.0, leading to an escalating brightness within the corresponding colors. The hue component within HSV spans from 0° to 360° in angular measurement.

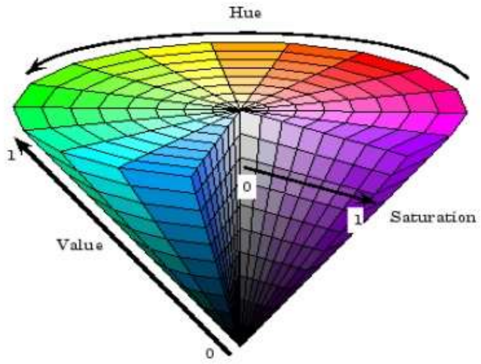


Fig III.2. Space Color Cube HSV Color

III.2.3.YCbCr Color Space

Originally conceptualized to ensure compatibility between color and monochrome televisions, this system aimed to separate luminance and chrominance components. The YCrCb color space, introduced by the International Radio Consultative Committee(IRCC), sought to optimize storage and transmission efficiency by harnessing perceptually relevantinformation

[190]. Tailored for digitally encoding television images, it conforms to the ITU.BT-601 standard and holds significant prominence within the JPEG2000 compression standard. Within this color space, RGB undergoes a linear transformation into luminance (Y) and chrominance (Cb, Cr) components (as demonstrated in figure 3.8). The specifics of this transformation differ contingent on television standards like National Television System Committee(NTSC), PAL (phase alternating line), or Sequential color memory(SECAM).

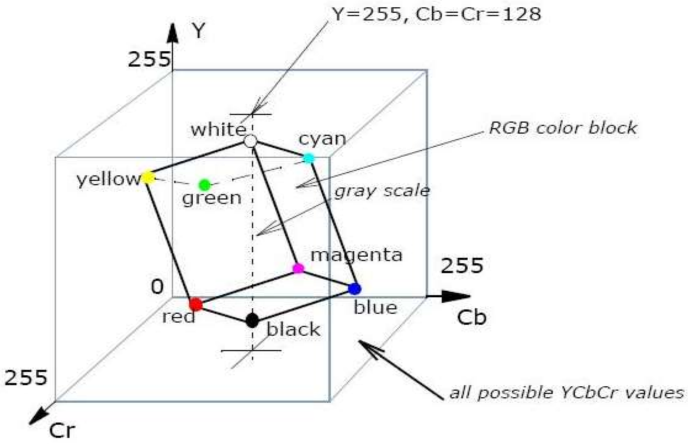


Fig III.3. Space Color Cube YCrCb.

As mentioned earlier, diverse YCrCb-type systems exist. Among them, YIQ and YUV serve as standard color spaces utilized in analog television transmission. The YIQ system, aligning with the NTSC standard, strategically capitalizes on human eye color response characteristics to optimize fixed transmission bandwidth [190]. In contrast, the YUV system corresponds to the PAL standard. These color spaces, resembling YCrCb, stem from the RGB space, where Y denotes the luminance component, while U, V, I, and Q represent the chrominance components.

In this arrangement, luminance information is singularly stored as a component (Y), while chrominance information is preserved as two color-difference components (Cb and Cr). Cb denotes the variance between theblue component and a reference value, whereas Cr signifies the difference between the red component and another reference value.

The principal transformations from the RGB space into the YCrCb spaces are expressed through the following equations.

$$Y = 0.299R + 0.287G + 0.11B \tag{III.7}$$

$$Cr = R - Y \tag{III.8}$$

$$Cb = B - Y \tag{III.9}$$

III.3. Image Features extraction

III.3.1. Convolutional neural networks

In the fast-changing field of computer vision and image processing, a consistent challenge prevails: the effective recognition and classification of images. Over recent years, CNNs have demonstrated impressive outcomes across various computer vision research domains. These networks have highlighted the significance of CNN features in capturing both semantic and intricate details of target objects, surpassing the utility of alternative features in a broad spectrum of visual recognition tasks. Essentially, a CNN comprises a sequence of pre-defined operations, where the operation types are predetermined, and their parameters are learned from extensive datasets. The configuration of these operation types is commonly referred to as the network architecture, organized into layers, each specifically dedicated to a particular type of operation.

III.3.2.Architecture

CNNs, represent a specialized type of neural network architecture tailored for processing data with a grid-like topology. This design makes them particularly adept at handling spatial and temporal data, such as images and videos, where there is a notable correlation between adjacent elements. While CNNs share similarities with other neural networks, they introduce an added layer of complexity by incorporating convolutional layers.

A typical CNN comprises an input layer, multiple hidden layers, and an output layer. The hidden layers typically include convolutional layers, ReLU layers, pooling layers, and fully-connected layers. In contrast to traditional neural networks, where each hidden layer consists of neurons fully connected to the previous layer, CNNs take advantage of the three-dimensional arrangement of layers—height, width, and depth.

Neural networks process input through hidden layers, each composed of neurons that are fully connected within the layer but operate independently. The last fully-connected layer serves as the "output layer," particularly crucial in classification scenarios as it produces class scores.

The key distinction between CNNs and standard neural network architecture lies in the three-dimensional organization of layers in CNNs—height, width, and depth. Here, "depth" refers to the third dimension of each layer rather than the overall architecture depth.

Figure III.4. illustrates that a CNN structures its neurons in three-dimensional layers, transforming a 3D input volume into a 3D output volume of neurons.

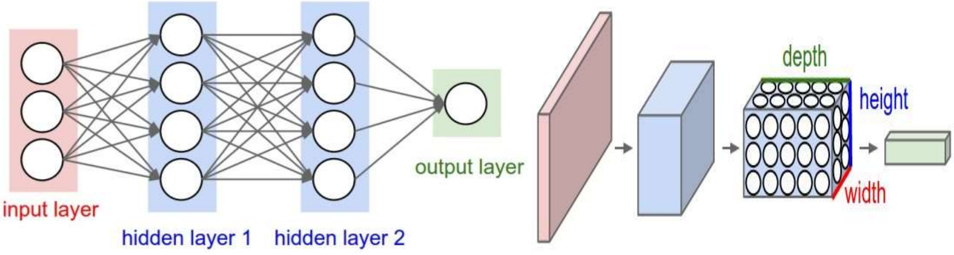


Fig III.4. Left: a network of neurons with 3 layers. Right: A convolutional neural network organizes its neurons in three dimensions (width, height, depth), visualized for each layer. The red input layer presents the input image. The width and height correspond to the dimensions of the image and the three-channel red, green and blue [82].

The primary layers in CNNs, widely recognized, include convolutional layers, ReLU layers, pooling layers, and fully-connected layers [82].

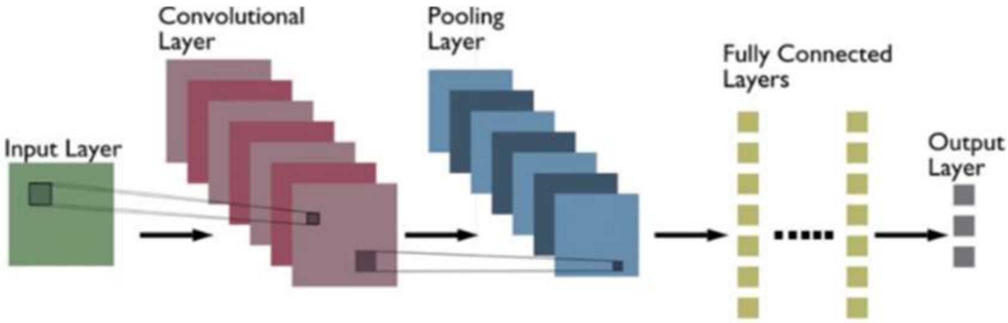


Fig III.5.A Standard CNN Structure

*III.3.2.1.**Convolutional Layer***

The convolutional layer serves as the fundamental building block of a CNN, comprising a collection of filters. These layers function by sliding a set of 'filters' or 'kernels' across the input data. Each filter is designed to identify specific features or patterns, such as edges, corners, or more intricate shapes in deeper layers. As these filters traverse the image, they generate a map indicating the locations where these features were detected.

Conceptually, a filter can be viewed as a smaller window that convolves (slides) across the input image, computing dot products between the filter values and the pixel values of the input image. The result is a 2-dimensional activation map, illustrating the responses of the filter at each position. Within each convolutional layer, a set of filters is applied, and each filter produces an independent 2-dimensional map. These individual activation maps are then stacked to form the output of the convolutional layer.

a) RELU layer:

Following the convolutional layers, Rectified Linear Unit (ReLU) layers are commonly applied, employing the activation function

$$f\left(x\right)=\max\left(0,x\right) \tag{III.10}$$

On the input *x*. This introduces increased non-linearity to the network while eliminating negative values from the activation maps. Traditionally, alternative functions like $f\left(x\right)=\tanh\left(x\right)$ or the sigmoid function $f\left(x\right)=\left(1+e^{-1}\right)^{-1}$ were utilized, but the ReLU function has demonstrated faster performance and is generally preferred [83],[84].

b) Pooling Layer:

Following the convolutional layers, pooling layers are employed to decrease the spatial dimension of the input, facilitating easier processing and demanding less memory. In the context of images, "spatial dimensions" pertain to the width and height of the image. An image can be envisioned as a grid composed of pixels, akin to rows and columns of tiny squares. By diminishing the spatial dimensions, pooling layers contribute to the reduction of parameters or weights in the network. This is instrumental in mitigating overfitting and expediting the model training process. Max pooling contributes to the reduction of computational complexity by decreasing the size of the feature map, rendering the model invariant to minor transitions. Without max pooling, the network would lack the capability to discern features irrespective of slight shifts or rotations, potentially compromising accuracy.

There are two primary types of pooling: max pooling and average pooling. Max pooling selects the maximum value from each feature map within a specified window, such as 2×2. It effectively captures the most prominent feature or characteristic in that region. On the other hand, average pooling computes the average of all values within the pooling window, offering a smoother, averaged feature representation.

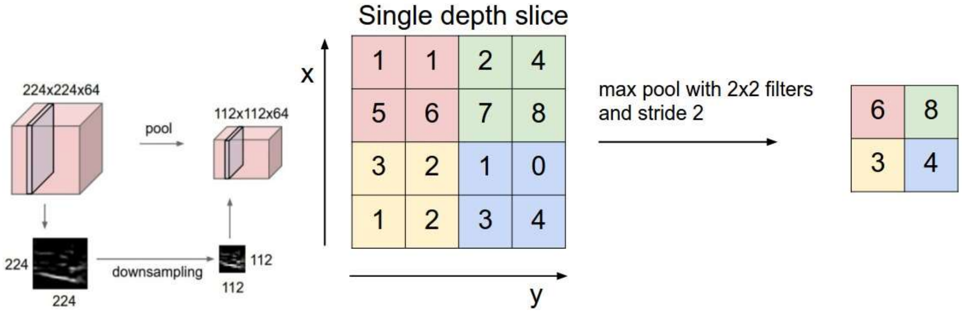


Fig III.6. Pooling layer

c) Fully-connected layers

Fully-connected layers constitute a fundamental element in the architecture of a (CNN). As the name implies, every neuron within a fully-connected layer establishes connections with all other neurons in the preceding layer. Typically positioned towards the conclusion of a CNN, fully-connected layers play a crucial role in leveraging the features acquired by convolutional and max pooling layers for predictive tasks, such as classifying input into specific labels. In the context of image classification, the final fully-connected layer might utilize the learned features to categorize an image as containing a dog, cat, bird, etc.

These fully connected layers take the high-dimensional output from preceding convolutional and pooling layers and flatten it into a one-dimensional vector. This flattening process enables the network to integrate all extracted features across the entire image, transcending localized considerations and grasping the global context of the image. The responsibility of fully connected layers extends to mapping the integrated features to the desired output, such as class labels in classification tasks. Serving as the final decision-making component of the network, these layers ascertain the significance of the extracted features within the context of the specific problem, such as recognizing a cat or a dog.

The amalgamation of a convolutional layer followed by a max-pooling layer and similar subsequent sets establishes a hierarchy of features in a CNN. The initial layer detects rudimentary patterns, and subsequent layers progressively build upon these to discern more intricate patterns.

CNNs find extensive application in image recognition and classification tasks. They excel in identifying objects within images, classifying images (e.g., distinguishing between a cat and a dog), and undertaking more complex endeavors such as generating image descriptions or pinpointing points of interest. While CNNs are predominantly associated with image data, they can also be adapted for time-series data like audio or text. However, for the latter scenarios, other network architectures such as Recurrent Neural Networks (RNNs) or transformers are often favored. CNNs stand as a potent tool in the realm of deep learning, consistently achieving state-of-the-art results across diverse applications.

III.3.2.2 Various CNN Architectures

Within the realm of Convolutional Networks, numerous architectures bear distinct names. The most prevalent include:

LeNet. stands as the pioneering CNN architecture, developed in 1998 by Yann LeCun, Corinna Cortes, and Christopher Burges specifically for addressing handwritten digit recognition challenges. Regarded as one of the initial triumphs in the field of CNNs, LeNet is often considered the "Hello World" equivalent in the realm of deep learning. It represents one of the earliest and extensively employed CNN architectures, showcasing notable success in tasks such as handwritten digit recognition.

The LeNet architecture comprises multiple convolutional and pooling layers, culminating in a fully-connected layer. With a structure encompassing five convolution layers and two fully connected layers, LeNet marked the advent of CNNs in tackling computer vision problems. Despite its pioneering role, LeNet faced challenges related to the vanishing gradients problem, impeding its training efficacy. To address this issue, a max-pooling shortcut connection layer was introduced between convolutional layers, reducing the spatial size of images. This implementation aids in preventing overfitting, facilitating more effective training for CNNs. The diagram below illustrates the LeNet-5 architecture.

The LeNet CNN, despite its simplicity, remains a potent model that has found applications in diverse tasks, including handwritten digit recognition, traffic sign recognition, and face detection. Even though LeNet was conceived over two decades ago, its architecture retains relevance in contemporary contexts and remains actively utilized.

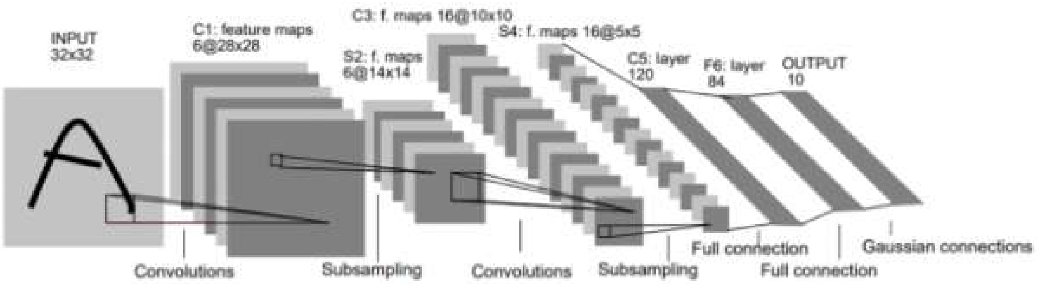


Fig III.7. LeNet Architecture .

AlexNet: AlexNet, a pivotal deep learning architecture that propelled CNN into popularity, was developed by Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. While sharing a resemblance to the LeNet architecture, AlexNet distinguished itself by its increased depth, size, and the stacking of Convolutional Layers. This network marked a breakthrough as the first large-scale CNN and secured victory in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. Tailored for deployment with expansive image datasets, the AlexNet architecture yielded state-of-the-art results upon its introduction. Comprising 5 convolutional layers interspersed with max-pooling layers, 3 fully connected layers, and 2 dropout layers, AlexNet incorporates the ReLU activation function across all layers, with Softmax serving as the activation function in the output layer.

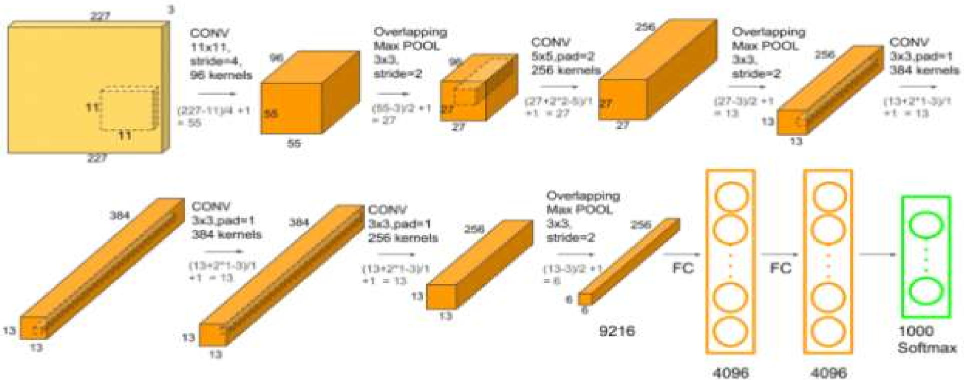


Fig III.8. AlexNet Architecture .

ZFNet : is a CNN architecture that integrates both fully-connected layers and CNNs, devised by Matthew Zeiler and Rob Fergus. Emerging as the ILSVRC 2013 champion, ZFNet boasts relatively fewer parameters compared to AlexNet while surpassing it in the ILSVRC 2012classification task, achieving top accuracy with only 1000 images per class. This architecture represents an enhancement over AlexNet, achieved through fine-tuning the hyperparameters, particularly by augmenting the size of the middle convolutional layers and reducing the stride and filter size in the first layer. Based on the Zeiler and Fergus model trained on the ImageNet dataset, the ZFNet CNN architecture encompasses seven layers: a Convolutional layer, a max-pooling layer for downscaling, a concatenation layer, a convolutional layer with a linear activation function and stride one, and dropout for regularization applied prior to the fully connected output. ZFNet enhances computational efficiency relative to AlexNet by introducing an approximate inference stage through deconvolutional layers in the middle of CNNs.

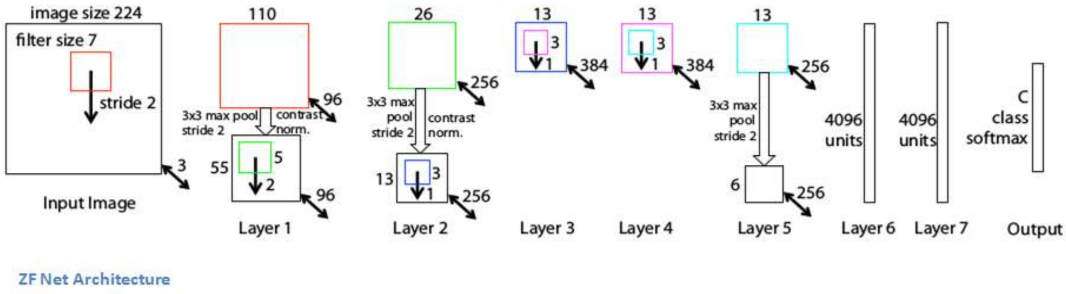


Fig III.9.ZF Net Architecture

<u>GoogLeNet :</u> the CNN architecture employed by Google to secure victory in the ILSVRC 2014 classification task, was developed by Jeff Dean, Christian Szegedy, Alexandro Szegedy, and others. Its key innovation lies in the introduction of an Inception Module, significantly reducing the network's parameter count (4M, compared to AlexNet's 60M). This architecture achieves greater depth through various techniques, including 1×1 convolution and global average pooling. Although computationally demanding, GoogLeNet utilizes heavy unpooling layers atop CNNs to mitigate spatial redundancy during training and incorporates shortcut connections between the first two convolutional layers before introducing new filters in subsequent CNN layers. Real-world applications of the GoogLeNet CNN architecture include tasks such as Street View House Number (SVHN) digit recognition, often utilized as a proxy for roadside object detection.

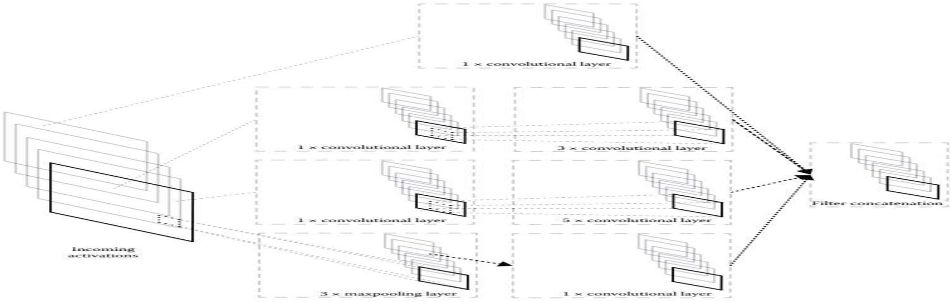


Fig III.10. GoogLeNet’s inception module

VGGNet : crafted by Karen Simonyan, Andrew Zisserman, and others at Oxford University, stands as a 16-layer CNN boasting up to 95 million parameters, trained on a dataset comprising over one billion images categorized into 1000 classes. Designed to accommodate large input images sized at 224 x 224 pixels, VGGNet yields 4096 convolutional features. However, CNNs with such extensive filters incur high training costs and demand abundant data. This characteristic explains why CNN architectures like GoogLeNet (AlexNet architecture) often outperform VGGNet in image classification tasks involving input images ranging from 100 x 100 pixels to 350 x 350 pixels.

Real-world applications of the VGGNet CNN architecture encompass the ILSVRC 2014 classification task, coinciding with the victory of the GoogLeNet CNN architecture.Acknowledged for its computational efficiency, VGGNet serves as a robust baseline for various computer vision applications, particularly in tasks like object detection. Its deep feature representations find application across multiple neural network architectures such as YOLO, SSD, and others.

VGG16: In the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" by K. Simonyan and A. Zisserman (referenced in[69]), the VGG16 network achieved an impressive 92.7% accuracy on the ImageNet test. Its training spanned weeks, employing a dataset of 16 million images, and utilized the computational prowess of an NVIDIA Titan Black graphics card. Notably, this network comprises 16 hidden layers and demonstrates the capability to accurately classify images into 1000 distinct classes.

The diagram below illustrates the standard VGG16 network architecture.

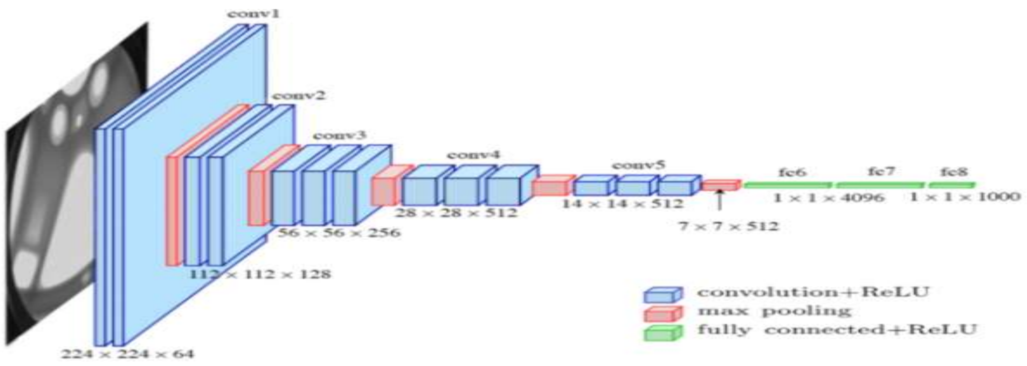


Fig III.11.VGG16 network architecture

VGG19: In accordance with [69], this network mirrors the architecture of VGG16 but incorporates an additional three convolution layers, resulting in a total of 16 convolution layers alongside 3 fully connected layers, summing up to 19 layers in total (illustrated in figure II.22).

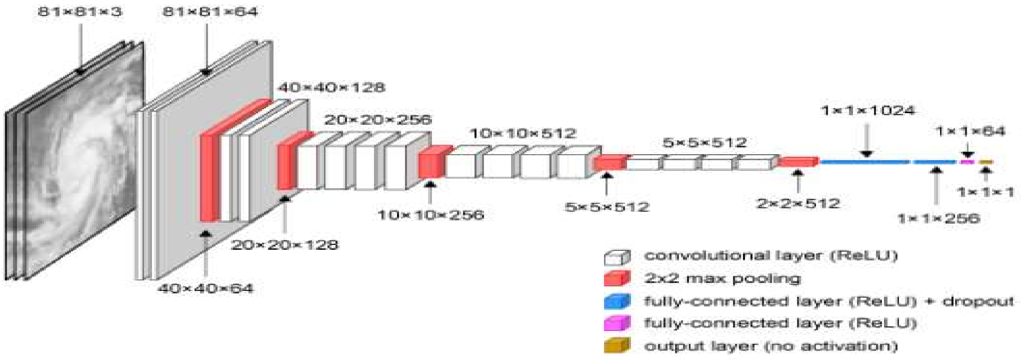


Fig III.12.VGG16 network architecture

ResNet:devised by Kaiming He and his colleagues, emerged as the CNN architecture that secured victory in the ILSVRC 2015 classification task, achieving a top-five error rate of merely 15.43%. This network, characterized by its 152 layers and over one million parameters, qualifies as deep even among CNNs. The training process for ResNet on the ILSVRC 2015 dataset would have exceeded 40 days when conducted on 32 GPUs. While CNNs are typically associated with image classification tasks featuring 1000 classes, ResNet demonstrates the versatility of CNNs by successfully addressing natural language processing challenges, such as sentence completion and machine comprehension. Notably, it was employed by the Microsoft Research Asia team in 2016 and 2017 for these purposes.

Real-world applications of the ResNet CNN architecture extend to Microsoft's machine comprehension system, where CNNs are utilized to generate answers for over 100,000 questions across more than 20 categories. ResNet is recognized for its computational efficiency and scalability, enabling adjustment to match the computational power of GPUs, whether scaling up or down.

MobileNets: represent CNNs designed to operate on mobile devices, facilitating image classification or object detection with minimal latency.Developed by Andrew G. Trillion and colleagues, MobileNets typically feature compact CNN architectures, rendering them suitable for real-time execution on embedded devices like smartphones and drones. Despite their reduced size, the architecture remains flexible and has been tested with CNNs comprising 100-300 layers, consistently outperforming other architectures such as VGGNet.

In practical terms, MobileNets find application in CNNs integrated into Android phones, powering Google's Mobile Vision API. This API enables the automatic identification of labels for popular objects in images, exemplifying the real-world utility of MobileNets in mobile device scenarios.

III.3.3.Histograms of Oriented Gradients

In the realm of computer vision, numerous algorithms aim to extract spatial features for object identification by leveraging information related to image gradients. One such algorithm is HOG, short for Histogram of Oriented Gradients. A histogram serves as an approximate representation of the distribution of numerical data, resembling a bar graph where each bar corresponds to a data group within a specific value range, known as bins. The term "orientation" refers to the direction of an image gradient. HOG generates a histogram depicting the directions of gradients present in an image.

To capture shape features, we applied the **HOG**technique proposed byN. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,”. This technique stores information about the shapes within the image, represented by histograms of object edge slopes. Each bin in the histogram signifies the count of edges with orientations falling within a particular angular range. The concatenation of computed histograms from all four sub-bands produces the HOG descriptor, housing both shape and texture information. This descriptor proves valuable for content-based image retrieval. Byemploying DBC and Haar wavelet transforms to enhance edges and other high-frequency local features, the use of HOG yields more comprehensive shape information compared to an unprocessed image.

Below is a summary of the process for extracting HOG features:

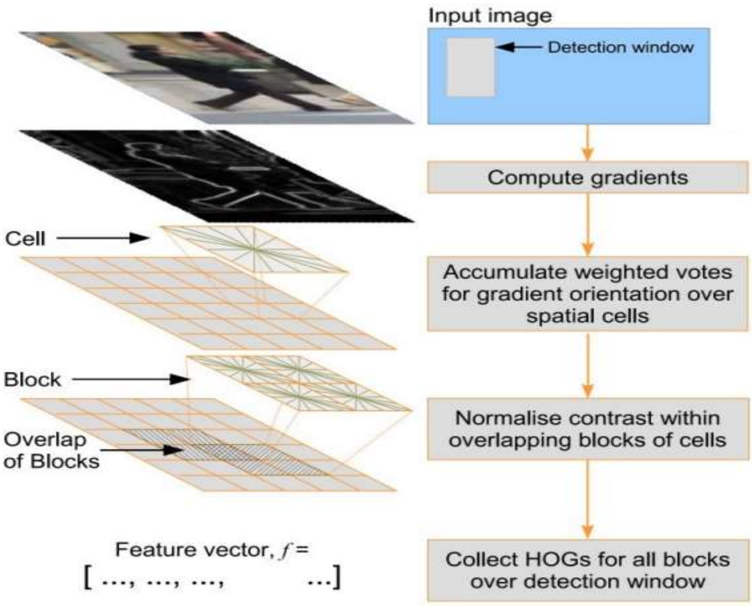

Input image
Detection window
Compute gradients
Accumulate weighted votes for gradient orientation over spatial cells
Normalise contrast within overlapping blocks of cells
Collect HOGs for all blocks over detection window
Cell
Block
Overlap of Blocks
Feature vector, *f* =
[..., ..., ..., ...]

Fig III.13. An overview of HOG feature extraction

- Gradient calculation : This stage involves the application of a 1-D discrete derivative mask in both the horizontal and vertical directions to each centered point. The mask is defined as follows: $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$

The magnitude and orientation at each pixel $I(x,y)$ are computed using the following formulas:

$$\begin{aligned} Magnitude &= \sqrt{\left(Gx(x,y)\right)^2 + \left(Gx(x,y)\right)^2} \\ Orientation &= \operatorname{atan}2\left(Gx(x,y), Gx(x,y) + \frac{\pi}{2} \right) \end{aligned} \tag{III.11}$$

Here,Gx(x,y) and Gy(x,y) represent the gradient values in the horizontal and vertical directions at each pixel. In the case of color images, the channel with the highest magnitude is chosen to determine the dominant magnitude and orientation of the pixel. It's important tonote that the addition of π/2 is necessary because the arctan operator yields a range between −π/2 and π/2. However, for an unsigned orientation scheme that enhances performance, the range is adjusted to be between 0 and π.

- Orientation Binning: In the second step of creating cell histograms, each pixel contributes a weighted vote to an orientation-based histogram channel, determined by the values obtained in the gradient computation. The cells, which are rectangular in shape, distribute the histogram channels evenly over a range of 0 to 180 degrees or 0 to 360 degrees, depending on whether the gradient is unsigned or signed. In their experiments, N. Dalal and B. Triggs discovered that unsigned gradients paired with 9 histogram channels yielded optimal performance.

- **Bloc normalization:**There are three distinct methods for normalizing blocks. Let v represent the non-standard feature vector that compiles all the cell histograms within a given block. The k-norm of $\|v\|_k$ is used, where $k = 1, 2$, and λ is a constant. The normalization schemes take the following forms:

$$\hat{v} = \frac{v}{\sqrt{\|v\|_2^2 + \lambda^2}} \tag{III.12}$$

$$\hat{v} = \frac{v}{\|v\|_1 + \lambda} \tag{III.13}$$

$$\hat{v} = \frac{\sqrt{v}}{\|v\|_1 + \lambda} \tag{III.14}$$

Additionally, an L2-norm is applied, followed by clipping (constraining maximum values of v to 0.2), and then normalization through Laplacian normalization [85]. All these normalization approaches demonstrate superior performance compared to non-standard cases. The ultimate HOG feature descriptor comprises a vector containing the elements of normalized cell histograms from all block regions.

III.4. Conclusion

In this chapter, the primary focus on visual tracking centers around the image as its main component. We provided a comprehensive discussion on critical aspects related to images, encompassing color, feature extraction, and delving into the significant role in processing the images that will be utilized in the subsequent section.

Chapter IV

Proposed methods for visual object tracking

Sommary

Proposed methods for visual object tracking

IV.1. Introduction

This chapter presents our method for tracking visual objects, our method is presented, focusing on effectively handling diverse challenging appearance changes of the target, such as substantial occlusion, illumination variations, and scale variations. Figure IV.1 showcases the different stages of the tracking algorithm. The fundamental algorithm can be succinctly described in three essential steps.

Initially, following a similar methodology as presented in [21], we utilize CNN features to train four two-dimensional correlation filters to estimate the target's location.

Next, we introduce a novel approach that involves integrating RGB and HSV color transformations with DCT decomposition. This innovative technique allows us to enhance the tracking process further.

Lastly, we calculate the maximum value from the resultant maps utilizing the correlation filters, Newton's method, and the convolutional features extracted from the HOG feature-based image template. This computed value plays a crucial role as a parameter in the correlation filters' update process.

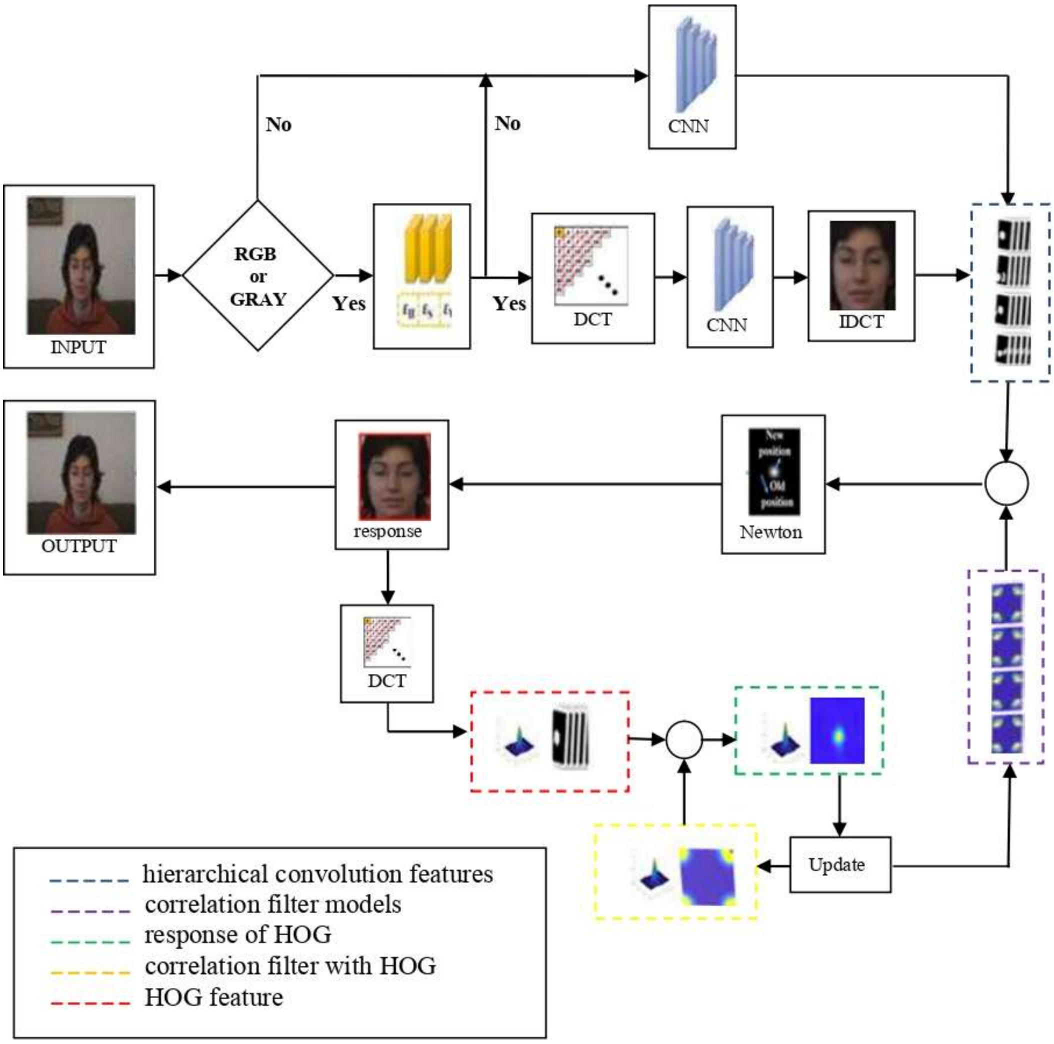


Fig. IV. 1. Key Phases in the Proposed Algorithm Implementation.

IV.2. Conditionbasedon HSV-Energy

In this section, we present an innovative strategy to tackle the issue of managing fluctuations in illumination, which proves to be a formidable obstacle for numerous benchmark trackers. The technique revolves around harnessing the energy constituents within the HSV color space. The notion of energy finds broad utility across various domains, encompassing wireless sensor networks [28], image reconstruction [29], and beyond.

For every input RGB frame, we utilize the energy utilization of individual components within the HSV color space to categorize the frame into two groups: low light and high light. The initial category encompasses frames with low energy consumption and minor alterations in lighting, while the subsequent category encompasses frames with elevated energy consumption and notable fluctuations in lighting.

Figure IV.2 the depiction showcases how the HSV color space establishes the fundamental framework for computing the energy consumption of every input frame.

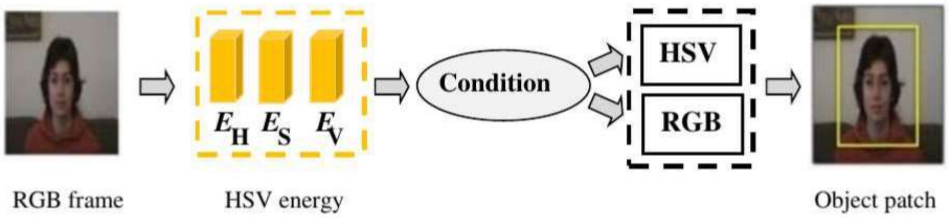


Fig. IV.2. Condition based on HSV-Energy.

The energy associated with the k^{th} component is represented by E_k. The proportion of energy consumption attributed to each individual HSV component is defined in the subsequent manner:

$$E_k = 100 \times \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \left(S_{ij}\right)^2}{E_T} \tag{IV.1}$$

$$E_T = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(H_{ij}\right)^2 + \sum_{i=1}^{m} \sum_{j=1}^{n} \left(S_{ij}\right)^2 + \sum_{i=1}^{m} \sum_{j=1}^{n} \left(V_{ij}\right)^2 \tag{IV.2}$$

If E_k is greater than $23\times100\%$, it indicates that the illumination is very weak. In such situations, the coefficients of the image's DCT are utilized as input for the CNN. Conversely, in the opposite scenario, the input image is decomposed into its RGB components. This step follows the approach proposed in this section.

IV.3. Discrite cosine transform

The DCT is a mathematical technique that converts a signal from its spatial representation to the frequency domain. By utilizing the DCT, important spatial frequency information in a 2-D signal can be efficiently captured using a small set of low-frequency coefficients, which typically group together in the upper left corner of the corresponding 2-D DCT matrix. This exceptional energy compaction characteristic has led to widespread adoption of the DCT in various applications, including data compression and image quality evaluation. The 2-D DCT of an $M \times N$ image matrix f can be defined as follows:

$$F\left(u,v\right)=\alpha_u\alpha_v\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}f\left(i,j\right)\times\cos\left(\frac{\left(2i+1\right)u.\pi}{2M}\right)\times\cos\left(\frac{\left(2j+1\right)v.\pi}{2N}\right) \tag{IV.3}$$

where the 2-D IDCT transform is defined as follows:

$$f\left(i,j\right)=\alpha_u\alpha_v\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}F\left(u,v\right)\times\cos\left(\frac{\left(2i+1\right)u.\pi}{2M}\right)\times\cos\left(\frac{\left(2j+1\right)v.\pi}{2N}\right)$$

In this transformation, the indices $u$and v are constrained within the range of 0 to $\left(M-1\right)$ and 0 to $\left(N-1\right)$, respectively. The pixel intensity at coordinates $\left(i,j\right)$ in the original signal is denoted by $f\left(i,j\right)$, while the corresponding transform coefficient located at row u and column $v$in the DCT matrix is represented by $F\left(u,v\right)$. To ensure appropriate normalization during the DCT calculation, vital scalar values α_u and α_v are defined as normalization coefficients. These coefficients play a crucial role in the normalization process of the DCT transformation.

$$\alpha_u=\begin{cases}1/\sqrt{M}, & u=0\\ \sqrt{2/M}, & 1\le u\le M-1\end{cases} \tag{IV.4}$$

$$\alpha_v=\begin{cases}1/\sqrt{N}, & v=0\\ \sqrt{2/N}, & 1\le v\le N-1\end{cases} \tag{IV.5}$$

The DCT coefficient $F\left(0,0\right)$located at the top left corner of the matrix is referred to as the DC term. As for the other DCT coefficients, they represent AC terms and correspond to high spatial frequency coefficients arranged in increasing order.

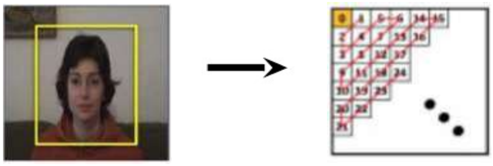


Fig. IV.3. Displays image patches on the left and the DCT coefficient matrix on the right. The yellow color highlights the dc term, while the remaining terms represent the selected ac terms.

The suggested methodology introduces a robust method to address illumination variation, leveraging a novel conceptual framework. Central to this innovative concept is the utilization of DCT coefficients in instances where the image's saturation reaches high levels. This strategic approach bypasses the direct utilization of the image components RGB within the network, offering an alternative and potentially more effective means of managing extreme saturation scenarios.

The amalgamation of the DCT and CNN techniques has exhibited considerable resilience, thereby enabling effective mitigation of illumination variation issues. This integrated method showcases robustness, showcasing its capacity to effectively address and alleviate challenges stemming from fluctuations in illumination within images.

FigureIV.4 portrays the precise positioning of the target across six selected frames within the Singer2 sequence. The blue frame denotes the initial position of the tracked object, while the red frame represents the tracking achieved using the suggested approach, a fusion of DCT coefficient and CNN methodologies. Evidently, the proposed technique enables robust tracking of moving objects amidst illumination fluctuations while maintaining a sustained memory of the target's appearance. This ensures a notably high accuracy in locating the target across the majority of frames in the Singer2 sequence.

Moreover, to validate the proposed approach, focusing on the saturation condition, two tests were conducted to calculate tracking errors. These tests involved comparing scenarios where the saturation condition was considered (depicted in red) versus situations where the saturation condition was disregarded (depicted in blue). Notably, figure IV.4 (center) highlights a substantial reduction in tracking error when employing the proposed approach, emphasizing its effectiveness in minimizing errors compared to the standard case that neglects variations in saturation conditions.

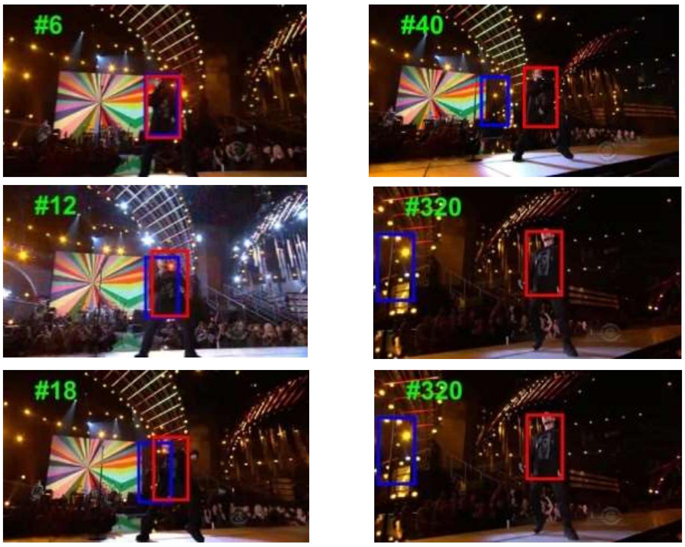


Fig. IV.4.A frame-by-frame display of the results of the Singer2 sequence tracking, with and without the saturation condition (in pixel).

IV.4. Correlation Filters

Within this segment of the thesis, the primary objective centers on enhancing the learning phase embedded within correlation filter models. The intention is to delve into the fundamentals of the correlation filter tracking approach and trace the evolutionary path of its algorithms across time. This exploration will encompass a comprehensive overview of the development journey, highlighting the key advancements and refinements. Subsequently, the focal point will narrow down to our specific algorithm within the domain of correlation filter tracking.

IV.5. Evolution of Correlation Filters

Within the domain of digital image processing, the correlation filter stands as a fundamental tool employed to identify specific locations within an image that closely align with a predefined template. The ideal functionality of a correlation filter involves producing heightened responses when encountering the predetermined template, while concurrently eliciting minimal responses when confronted with background elements [21].

The advent of the MOSSE [31] filter marked a significant leap, integrating correlation filters into tracking using grayscale images. Despite its swift object tracking capabilities, this filter lacked the reliability to accurately track objects when their appearances underwent changes. Subsequently, Henriques *et al*. introduced circulant structure tracking with kernels Kernels correlation filter (KCF), in 2012, propelling advancements in tracking methodologies [43]. Further progressions surfaced in 2014 with Danwelljan *et al*., presenting the KCF, which fine-tuned channel characteristics to accommodate multi-channel features and introduced CN features to enhance tracking capabilities [32]. Despite these advancements, challenges persisted in augmenting the filter's adaptability to handle rotation, objects moving out of view, and rapid motion.

The innovation continuum persisted as Danelljan *et al*. proposed the Discriminative Scale Space Tracker (DSST) in 2014, employing feature pyramids to address issues arising from scale variations [44]. Additionally, the improved fDSST algorithm emerged as a subsequent iteration. With the meteoric rise of deep learning, the C-COT algorithm emerged in 2016, effectively amalgamating spatial position information through shallow CNN features, bridging correlation filtering and CNN methodologies. This pioneering algorithm clinched victory in the VOT2016 competition. Similarly, the CSR-DCF algorithm harnessed CNN features, fortifying algorithmic resilience and robustness through their integration with correlation filtering techniques [35].

IV.5.1. KCF algorithm

KCF, short for Kernelized Correlation Filter, amalgamates techniques from two tracking algorithms, BOOSTING and MIL tracker. Its core function involves translating the object's position using a circular offset within the bounding box. In essence, the KCF tracker's primary focus lies in discerning changes within an image, whether in movement, extension, or orientation, thereby striving to probabilistically ascertain the object's position being tracked [47].

The KCF [32] is a stalwart among traditional discriminant methods, known for its ability to learn filters derived from a series of training samples. The KCF sample generation method employs the cyclic shift technique. Considering one-dimensional data represented as $\boldsymbol{x} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}]$, the cyclic shift of x is denoted as $\boldsymbol{Px} = [\boldsymbol{x_n}, \boldsymbol{x_1}, \dots, \boldsymbol{x_{n-1}}]$. The entirety of cyclic shift samples formulates a cyclic matrix.

$$X = C(x) = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_n & x_{n-1} & \cdots & x_1 \end{bmatrix} \tag{IV.6}$$

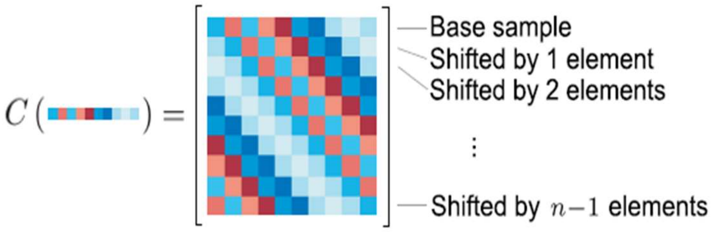


Fig. IV.5. Circulant matrix.

In other words, it employs an *(M* × *N)*image block x to train a filter $\boldsymbol{f(x)} = \langle \boldsymbol{\omega}, \boldsymbol{\emptyset_x} \rangle$,, creating a training sample through a cyclic shift operation on x. These training samples encompass all possible cyclic shifts *Pi*, where *i* ∈ *{0, …, M* − *1}* × *{0, …, N* − *1}*. Each Pi yields a respective score y_i (y_i∈[0, 1]*)*, generated by a Gaussian function contingent on the offset distance. Through the minimization of the regression error, the classifier undergoes training as follows:

$$w = \arg\min_{w} \sum_{i} \left(\langle w, \phi(x) \rangle - y_i \right)^2 + \lambda \|w\|^2 \tag{IV.7}$$

Among these elements, $\phi(x)$ represents the mapping within Fourier space. The parameter $\lambda \geq 0$ serves as the regularization parameter, indicating the model's level of simplicity. The periodic assumption facilitates efficient training and detection through the utilization of a fast Fourier transform. Leveraging the translation invariance of the kernel function, a can be efficiently obtained as $\hat{a} = \hat{y} / \left(\hat{k}^{xx} + \lambda\right)$ given the special nature of the circulant matrix. During the filtering conversion process, the assessment of a candidate image block of size $m \times n$ within the search space z is determined by the following formula:

$$f(z) = \mathscr{F}^{-1}\left(\hat{k}^{xx} \odot \hat{a}\right) \tag{IV.8}$$

The filter response $f(z)$ spans across all cyclic matrices z, with the highest response pinpointing the object within the current frame.

Through exploitation of cyclic matrix properties within the candidate window, the KCF algorithm generates a sequence of potential samples, drastically enhancing tracking speed when contrasted with traditional window sampling methods. This problem subsequently undergoes swift conversion into a frequency domain operation via Fourier transform, shifting the ridge regression issue from the time domain to a frequency domain cross-correlation problem.

Distinct from a single-channel grayscale feature, the KCF algorithm incorporates a multichannel HOG feature. Nevertheless, due to the use of cyclic shifts, the KCF encounters challenges associated with boundary effects. Furthermore, the KCF maintains a fixed search area, potentially leading to surpassing the search range during swift motion.

IV.5.2. CSRT algorithm

CSRT, the OpenCV implementation of the CSR-DCF represents a sophisticated algorithm capable of adapting to alterations such as object enlargement and non-rectangular shapes. Fundamentally, it employs HoG features in conjunction with Spatial Reliability Maps (SRM)to facilitate object localization and tracking .The spatial reliability map adjusts the filter's support to focus on the part of the object most suitable for tracking. This adaptation tackles two issues: it overcomes the limitations of circular shift, allowing for a flexible search range, and it surpasses constraints tied to assuming a rectangular shape for the object. Another innovation in CSR-DCF lies in channel reliability. This reliability is determined based on the propertiesof theconstrained least-squares solution. These reliability scores for each channel are employed to weigh the filter responses individually during localization (refer to figure IV.6) [35].

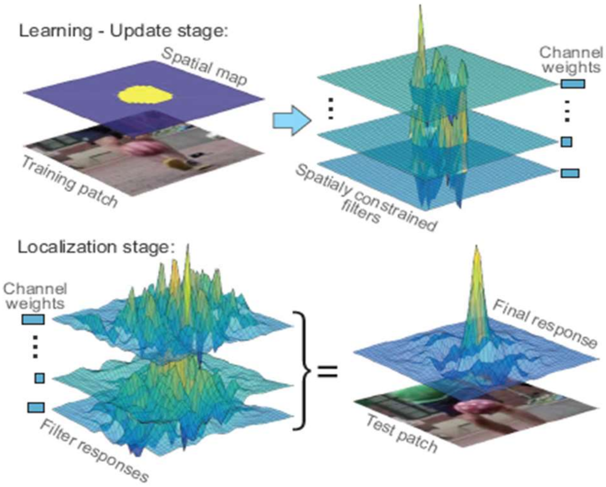


Fig. IV.6. The CSR-DCF approach

IV.5.3. MOSSE algorithm

The integration of correlation filter technology into the realm of visual tracking was catalyzed by the MOSSE algorithm [31]. This algorithm possesses the capability to adjust to challenges associated with occlusion and rotation, attaining an impressive tracking speed of 669 frames per second. Primarily trained on the initial image, the MOSSE filter showcases robust performance, effectively handling variations in lighting, scale, and posture. In instances of target occlusion, the algorithm adeptly assesses the object's tracking status, refining filter parameters based on the PSR value. Upon the object's reappearance, seamless tracking is resumed [9].

During the process of visual tracking, the selected area of interest, which can be a specified object, a point of interest, or the entire image, is referred to as a sample. The set N , denoted as x_i where $x_i = \{x_1, x_2, \cdots, x_N\}$ represents training samples, each denoting a rectangular region with a width of M_1 and a height of M_2 .

Typically, each sample x_i functions as $x_i = \{0,\cdots,M_1\}\times\{0,\cdots,M_2\}\to\mathbb{R}$. The objective of the MOSSE tracker revolves around identifying a filter that maintains the relationship $x_i \otimes h \approx y_i$, presented as a cost function in equation (IV.9).

$$h_{opt} = \arg\min_{h} \sum_{i=1}^{N} \left\| h \otimes x_i - y_i \right\|^2 \qquad \text{(IV.9)}$$

The $\{y_i\} N_i$ denotes the anticipated response, illustrating the intensities within the region of interest. Ordinarily, y_i embodies a definition derived from a sampled Gaussian function, characterized by a narrow peak strategically centered on the targeted object, as visually depicted in figure IV.2.6. The symbol $\otimes$ signifies the correlation operation, an outcome achieved through cyclically shifting the features of the original patch x_i and the previously acquired model *h*. The resultant h_{opt} represents the freshly acquired filter model. It's essential to emphasize that the correlation and convolution, delineated by equations (IV.10)and (IV.11), are invoked to resolve this linear regression quandary, particularly in the frequency domain. These operations are instrumental in addressing this specific linear regression challenge, employing a frequency domain-based approach.

$$c[n] = \sum a[m]\, b[n-m] = \mathcal{F}^{-1}\{A \odot B\} \qquad \text{(IV.10)}$$

$$c[n] = \sum a[m]\, b[n+m] = \mathcal{F}^{-1}\{A^* \odot B\} \qquad \text{(IV.11)}$$

In the context of discrete signals, the expressions $a[.]$and $b[.]$represent unidimensional signals, wherein $a[n-m]$signifies the circular form of $a[n]$experiencing a delay of m units to the right. The symbol * denotes the complex conjugate, while the $\mathcal{F}$ symbol signifies an elementwise multiplication. The transformation denoted by $\mathcal{F}$ refers to the Discrete Fourier transform (DFT), distinguishing all uppercase letters as signals in the frequency domain and lowercase letters as signals in the spatial domain.

Expanding the theorem to encompass the two-dimensional scenario is straightforward owing to the separable nature of the DFT operation in the two-dimensional realm. It's presupposed that any derivation applied to the one-dimensional signal can be seamlessly extended to encompass two-dimensional signals, provided that the operations can be segregated into two dimensions [38].Utilizing these principles, the correlation problem articulated in equation (IV.9) can be expressed as follows:

$$H_{opt} = \arg\min_{H} \sum_{i=1}^{N} \left\| H \otimes X_i - Y_i \right\|^2 \qquad \text{(IV.12)}$$

To streamline the optimization process for each individual element of H_{opt}, we can express:

$$\mathscr{L} = \sum_{i=1}^{N} H^* \odot H \odot X_i^* \odot X_i + Y_i^* \odot Y_i - H^* \odot X_i \odot Y_i^* - H \odot X_i^* \odot Y_i \qquad \text{(IV.13)}$$

Suming independence between H and H^*, upon deriving this function with respect to each element of H^*, we acquire:

$$\frac{\partial \mathscr{L}}{\partial H^*} = \sum_{i=1}^{N} H \odot X_i^* \odot X_i - X_i \odot Y_i^* \qquad \text{(IV.14)}$$

When $\frac{\partial \mathscr{L}}{\partial H^*}$ equals zero, the derivative of equation (IV.13)concerning H^* results in:

$$H = \frac{\sum_{i=1}^{N} X_i \odot Y_i^*}{\sum_{i=1}^{N} X_i^* \odot X_i} \qquad \text{(IV.15)}$$

The acquired model H effectively upholds the minimization of the cost function within the frequency domain. the Fourier transform serves as the optimal tool for executing convolution (elementwise multiplication) due to the computational efficiency offered by the Fast Fourier Transform (FFT) at a complexity of $\mathrm{O}\left(M\log(M)\right)$, necessitating a signal complexity of $\mathrm{O}\left(M^2\log(M)\right)$. Furthermore, this method streamlines operations involving the $M^2 \times M^2$ matrix (inversion and multiplication), providing a practical solution for the linear least-squares method.

Throughout the tracking process, estimating the object's position occurs in each input frame, facilitating the localization of the object in the subsequent frame. At the instance $t+1$, the correlation response map f_{t+1} materializes by element-wise multiplication of the acquired filter model H_t at instant t with the sample Z extracted from the object patch at the instance $t+1$, as follows:

$$f_{t+1} = \mathcal{F}^{-1}\left(H_t^* \odot Z_{t+1}\right) \tag{IV.16}$$

The significance of this response map lies in its role in pinpointing the new location of the object, reliant on the coordinates of the pixel with the highest value. Essentially, these coordinates serve as the indication of the shift in the center between consecutive frames at instances t and $t+1$.

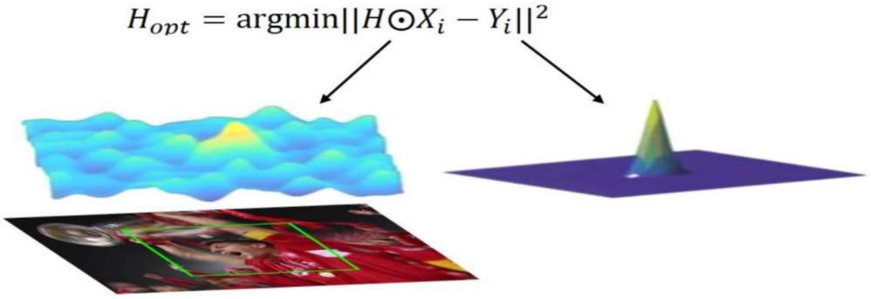


Fig. IV.7. The MOSSE filter

As the tracked object's position is determined in each frame, preserving the efficiency of the filter model derived from equation (IV.16)necessitates updating it. This pivotal process is executed employing a learning rate denoted as γ. The significance of this parameter resides in facilitating the learning process for the numerator and denominator of the model H at instant $t+1$, while conserving a portion of the previous filter model H_t, thus involving the following:

$$A_{t+1} = (1-\gamma)A_t + \gamma\left(X_{t+1} \oplus Y_{t+1}^*\right) \tag{IV.17}$$

So, we can write

$$H_{t+1} = \frac{A_{t+1}}{B_{t+1}} \tag{IV.18}$$

where A_{t+1} and B_{t+1} represent the numerator and de-numerator of H_{t+1}, while X_{t+1} and Y_{t+1} correspond to the samples and the desired response map, respectively.

IV.5.4. Correlation Filters

The correlation filters showcase a proficient encoding of the visual attributes of the target object [24]. The procedure of acquiring the correlation filter models W entails addressing the subsequent minimization challenge:

$$W^{*}=\arg\min_{W}\sum_{m,n}\left\|W.x_{m,n}-y\left(m,n\right)\right\|^{2}+\lambda\left\|W\right\|^{2} \tag{IV.19}$$

The learned correlation filter model is denoted as W^{*}.

The feature vector x is characterized by its dimensions, which are M, N and D, with M representing width, N representing height, and D representing the number of channels.

The regularization parameter λ assumes values that are non-negative.

where

$$W.x_{m,n}=\sum_{d=1}^{D}W_{m,n,d}^{T}.x_{m,n,d} \tag{IV.20}$$

with $W_{m,n}^{T}$ and d the transposed weight for each channel d at position $\left(m,n\right)$. The correlation filter model's dimensions are $M\times N$ [25]. Each shifted sample of $x_{m,n}\left(m,n\right)\in\left\{0,1,...,M-1\right\}\times\left\{0,1,...,N-1\right\}$ is associated with a Gaussian function label $y\left(m,n\right)$ through the regression process:

$$y\left(m,n\right)=e^{-\frac{\left(m-\frac{M}{2}\right)^{2}-\left(n-\frac{N}{2}\right)^{2}}{2\sigma^{2}}} \tag{IV.21}$$

where σ is the standard deviation.

The optimization problem outlined in equation (IV.19) can be separately addressed for each feature channel by utilizing FFT, akin to the vector correlationfilter training method described in [26]. In the frequency domain, the learned filter for the d^{th} channel (where d takes values from 1 to D) is defined according to equation (IV.22).

$$W^d = \frac{Y \odot \overline{X}^d}{\sum_{i=1}^{D} X^i \odot \overline{X}^i + \lambda} \tag{IV.22}$$

where y is the Fourier transformation form of $y = y(m,n) \mid (m,n) \in \{0,1,\ldots,M-1\} \times \{0,1,\ldots,N-1\}\}$ and the bar refer to the complex conjugation. The operator $\odot$ is the Hadamard product.

To compute the d^{th}correlation response map f_l, the Inverse Fast Fourier Transform (IFFT) is employed, as expressed by the equation:

$$f_l = \mathcal{F}^{-1}\left(\sum_{d=1}^{D} W^d \odot \overline{Z}^d\right), \text{where } l = 1, 2, \ldots, 3 \tag{IV.23}$$

During the tracking process, a multi-channel vector Z is employed to compute the value of f_l. The uppercase letters indicate the Fourier transform signals associated with it, the IFFT operation is represented by $\mathcal{F}^{-1}$, and the complex conjugation is indicated by the bar symbol.

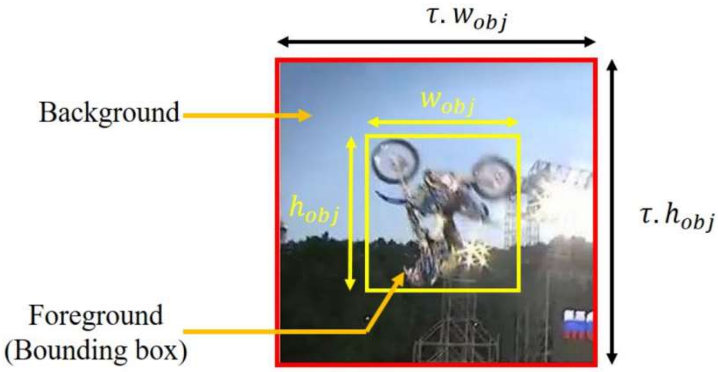


Fig. IV.8. Object patch extraction.

IV.6. Convolution Features

CNN have exhibited remarkable success across a range of computer vision tasks. In this investigation, we introduce a novel approach involving translation estimation by leveraging a CNN model to extract features and establish a translation model. Specifically, we harness four layers from the VGGNet-19 model to extract convolutional features.

In the context of visual object tracking, the precise determination of the target object's position takes precedence over its semantic category. As a result, we employ bilinear interpolation [11] to resize each input frame to dimensions of 224 × 224. Subsequently, we collect the outputs from pool 1, pool 2, pool 3, and pool 4 layers to create a multichannel feature map.

As the depth of the CNN increases, the spatial resolution of the target object gradually diminishes due to the pooling operations. To mitigate this challenge, we address each feature map's size using bilinear interpolation, as outlined in equation (IV.24), ensuring that they are resized to a consistent spatial resolution of $M/4 \times N/4$. Here, $M$and Ndenote thedimensions of the feature vector x. This approach guarantees uniform spatial resolution across the pooling layers.

$$x_i = \sum_k \alpha_{ik}.h_k \tag{IV.24}$$

In this context, x_i stands for the upsampled feature vector at the i^{th} location, and h_k represents the feature map corresponding to the k^{th} feature. Meanwhile, $\alpha_{ik}$is a weight interpolation factor that relies on the specific positions of the i^{th} and k^{th} vectors within the adjacent features.

IV.7. Estimation of Coarse-to-Fine Translation

To determine the target translation within the correlation response maps of each layer, denoted as f_l, a search is performed to locate the maximum value in the previous layer $(l-1)^{th}$. The corresponding location in the current layer l^{th} is taken as a reference point for regularization. The most suitable position of the target in the $(l-1)^{th}$ layer is subsequently identified by maximizing the weighted summation of responses from the$(l-1)^{th}$ and l^{th} layers, while adhering to certain constraints.

$$\arg\min f_{l-1}(\mathrm{m},\mathrm{n}) + \gamma f_l(\mathrm{m},\mathrm{n}), \quad |m-\widehat{m}| + |n-\widehat{n}| \leq r \tag{IV.25}$$

Within a region of size $r \times r$ centered around $(\widehat{m}-\widehat{n})$, the search is confined to neighboring areas, ensuring limitations on the search range. Progressing from the outermost to the innermost layers, each response value is subject to multiplication by a regularization factor γ and subsequently propagated back to the response map of preceding layers [11]. Ultimately, through the maximization of equation (IV.25) on the layer boasting the highest spatial resolution, the estimation of the target location is achieved.

Furthermore, by employing equations(IV.19), (IV.22), (IV.25), and Newton's method, the utmost response of the correlation filter derived from HOG can be calculated for $l=1$, and $\gamma = 1$.

Newton's method, a technique in the field of optimization, is employed to discover global extrema. By calculating both the gradient and the hessian [27], this method seeks the highest score during each iteration. The process achieves convergence with only a limited number of iterations.

IV.8. Model Update

During the tracking process, a significant change in the object's appearance between two consecutive images is evident, leading to potential tracker drifts [26]. To address this issue, it becomes crucial to update the correlation filter model obtained through equation (IV.19) by incorporating a learning rate denoted as η, as demonstrated in equation(IV.26).

$$\begin{cases} \widehat{x}^t = (1-\eta)\, x^{t-1} + \eta x^t \\ \widehat{W}^t = (1-\eta)\, W^{t-1} + \eta W^t \end{cases} \tag{IV.26}$$

The adaptive updating of correlation filters derived from both CNN and HOG features is executed with a conservative approach. This conservative learning strategy endows the filters with resilience against noisy updates, thereby enhancing their ability to accurately estimate the confidence level associated with each tracked outcome.

To discern instances of potential tracking failures, a threshold value, denoted as T_0, is established. If the maximum response of the correlation filter derived from the HOG features surpasses the T_0 threshold, indicating a notably high level of confidence in the tracked result z, we proceed with updating the correlation filters. Conversely, when the confidence score falls below the T_0 threshold, refraining from updating the filters is warranted, ensuring a cautious approach to filter modifications.

IV.9. Conclusion

Within this chapter, we introduce a novel approach for visual object tracking that encompasses a series of methodological steps, commencing with image preprocessing and progressing through feature extraction to fulfill the dual objectives of translation and scale estimations. The initial stage of preprocessing is adept at efficiently extracting patches from the target object, facilitating the retention of critical information pertaining to both foreground and background aspects.

Moving forward to the feature extraction phase, our methodology integrates the utilization of DCT, CNN and HOG features. This amalgamation is instrumental in capturing and encoding significant visual attributes essential for tracking tasks. In the pursuit of maintaining robust object tracking, correlation filters play a pivotal role in both the translation and scale estimation tasks, forming a crucial component of our proposed methodology.

Chapter V

Experminent results

Sommary

Experminent results

V.1. Introduction

The realm of visual tracking encompasses a vast array of applications, each with its unique set of prerequisites and demands. These requirements may encompass constraints related to real-time processing speed or specialized performance expectations tailored for specific video classifications. In practical terms, the selection of an optimal tracker is inherently application-specific and typically involves a trade-off among various factors.

In order to facilitate a thorough and equitable assessment, the community has developed standardized datasets and evaluation metrics for testing and comparing generic visual object trackers. Within the academic field, any published work is generally expected to subject its proposed methodology to evaluation using these collective datasets.

Within the scope of this thesis, the proposed method undergoes meticulous evaluation across three such datasets, notably the OTB-2015 dataset, which comprises 50 videos. Renowned for their diversity and complexity, these datasets encompass a wide spectrum of scenes and target scenarios, presenting significant challenges for tracking algorithms. Several experiments detailed in this thesis are conducted utilizing either the OTB-2015 dataset, serving as robust testing grounds to gauge the effectiveness and adaptability of the proposed methodology.

V.2. **Databases**

In recent years, a plethora of applications in visual tracking has surfaced, leading to assessments conducted via subjective evaluations or intricately devised metrics aimed at scrutinizing tracker performance. Acknowledging the inherent biases in subjective assessments, several studies have aimed to mitigate this concern by introducing sophisticated databases like the OTB dataset, TC128, and UAV20 datasets. These repositories stand as comprehensive collections, encompassing attributes and scenarios pivotal for the meticulous evaluation of tracking methodologies.

V.2.1. Object Tracking Benchmark

The Object Tracking Benchmark dataset (OTB) comprises three primary datasets: OTB-2013 [96], OTB100 (OTB-2015) [97] , and OTB50. Initially introduced by Y. Wu *et al.*[96] at CVPR in 2013 as OTB-013, OTB-2013 encompasses 51 video sequences with over 2900 frames annotated with artificially labeled target boxes. Notably, the Skating video sequence is divided into two separate sequences due to distinct labeling of objects. Furthermore, OTB-2013 categorizes visual object tracking into 11 challenge types, such as scale change, illumination variation, and occlusion. Each video sequence in the dataset is annotated with these tracking challenges, facilitating method analysis concerning diverse challenges. A subset of the OTB dataset's video sequences is depicted in figureV.1.

Fig.V.1. Examples of Sequences within the OTB Benchmark

It's important to note that a single video sequence may correspond to multiple tracking challenges. Subsequently, OTB-2015, an extension of OTB-2013 by Y. Wu *et al.*[97], expands the dataset to 100 video sequences, hence its alternate name, OTB100. Due to the widespread use and success of OTB-2013 and OTB-2015 datasets, numerous tracking methods have demonstrated commendable performance on these two datasets. To intensify the dataset's difficulty level, an additional 50 complex video sequences were extracted from OTB-2015, forming a distinct dataset termed OTB50.

Moreover, the OTB dataset introduces an evaluation tool that boasts compatibility not only within its own datasets but also with others like TC128, UAV123, among others. Consequently, the OTB dataset stands as the most extensively utilized dataset in the domain of visual object tracking.

V.2.2.Temple Color 128

The Temple Color 128 dataset (TC128), introduced by Liang et al. [98] from Temple University in the United States, was presented in the IEEE Transactions on Image Processing journal in 2015. Comprising 128 video sequences accompanied by manual annotations, the dataset draws from two primary sources. Firstly, it includes 50 video sequences commonly utilized in other video datasets, while the remaining 78 sequences were manually labeled. Designed to investigate the impact of color information on video target tracking algorithms, the video dataset exclusively consists of color images. A segment of the video sequences from the TC128 dataset is illustrated in figure V.2.

Fig.V.2. Examples of Sequences within the TC128 Benchmark

V.2.3.Unmanned Aerial Vehicles123

During the 2016 ECCV conference, Mueller *et al.* [99] from King Abdullah University of Science and Technology introduced the UAV123 dataset (Unmanned Aerial Vehicles, UAV). This dataset exclusively comprises video sequences captured from aerial viewpoints, predominantly sourced from unmanned aerial vehicles, while a subset of sequences is computer-generated. The UAV123 dataset caters to specific tasks and application scenarios, encompassing 123 video sequences meticulously annotated with manual labels.

Moreover, the dataset features an additional 20 ultra-long video sequences, denoted as UAV20L, intended specifically to evaluate the tracking efficacy of visual object tracking methods over extended durations. These extended sequences serve as a platform for assessing tracking capabilities within prolonged videos. A segment of the video sequences housed within the UAV123 dataset is showcased in figure V.3.

Fig.V.3. Examples of Sequences within the UAV123 Benchmark

V.3. Assessment Approach

The evaluation of the proposed tracking method encompasses its performance analysis across multiple datasets, specifically the OTB, TC-128, and UAV20L datasets. This evaluation employs the one-pass evaluation (OPE) protocol, integrating measures such as distance precision or overlap success rates. Through the utilization of these standardized evaluation metrics, the method's tracking capabilities are rigorously assessed across diverse datasets, offering insights into its robustness and effectiveness in varying scenarios.

V.3.1.Robustness evaluation

The assessment of tracker robustness conventionally involves executing trackers across a designated test sequence, initiating them from the ground truth position in the initial frame, and subsequently calculating metrics such as average precision or success rate. This evaluation approach, termed one-pass evaluation (OPE), serves as the standard benchmarking method for assessing tracker performance. By commencing tracking from the ground truth position in the first frame and observing its behavior across the entire sequence, OPE provides a comprehensive overview of a tracker's performance in terms of precision or success rates on average.

V.3.2.Precision plot

The assessment of tracking precision commonly involves employing the center location error as a key metric, calculated as the mean Euclidean distance between the center locations of tracked targets and manually labeled ground truths across frames within a sequence. However, inherent limitations arise when trackers lose the target, resulting in potentially random output locations that may skew the average error value and inaccurately reflect tracking performance [6]. In response to this challenge, the precision plot [6,95] has emerged as a more nuanced measure of overall tracking efficacy.

The precision plot provides insights by illustrating the percentage of frames where the estimated location falls within a predefined threshold distance of the ground truth. This plot effectively captures the tracker's performance in maintaining proximity to the actual target location throughout the sequence. A widely accepted representation of the precision score for each tracker utilizes a threshold distance of 20 pixels [6], offering a standardized measure for comparison among different tracking methodologies.

V.3.3.Success plot

An additional evaluation metric employed in assessing tracking performance involves bounding box overlap, which compares the tracked bounding box r_t with the ground truth bounding box r_a. This metric is quantified by the overlap score, denoted as $S = \frac{|r_t \cap r_a|}{|r_t \cup r_a|}$, where $\cap$ and $\cup$ signify the intersection and union of two regions, respectively, while $|.|$ represents the pixel count within the region. Evaluating performance across a sequence of frames entails counting the successful frames, defined by an overlap score S exceeding a predefined threshold t_0.

The success plot portrays the ratios of successful frames across a spectrum of threshold values, varying from 0 to 1. Singularly relying on a specific success rate value at a given threshold (e.g. $t_0 = 0.5$) for tracker evaluation might not offer a comprehensive or equitable assessment. To address this limitation, the Area Under the Curve (AUC) of each success plot is employed as a more comprehensive metric, allowing for the ranking of tracking algorithms based on their overall performance across various threshold values.

V.4. Experiments

The presented algorithm underwent validation and assessment using the OTB50 benchmark dataset, which comprises 50 videos. The tracking algorithm was coded in MATLAB and operated on an Intel I5-12400F 2.50 GHz CPU equipped with 16 GB RAM, with additional assistance from the MatConvNet toolbox. Feature extraction involved carrying out CNN forward propagation on a GeForce GTX1060 GPU.

CNN introduced by the Visual Geometry Group in 2012, VGG-Net-19, has been Exploited in this study. This network was made of 19 layers, featuring 16 convolutional layers, 3 fully connected layers, 5 MaxPool layers, and 1 SoftMax layer [100]. To extract features, the network underwent training on the comprehensive hierarchical image repository, ImageNet [101].

During the process of feature extraction, only the outputs from pool 1, pool 3, pool 4, and pool 5 were employed. The search window size was remained constant at 1.8 times the target size. A regularization parameter η of 10^{-4} has been chosen, and the kernel width for generating Gaussian function labels has been set at 0.1. Furthermore, the learning rate η in equation (IV.26) was established as 0.01, also, the control updating parameter was fixed to 0.3. Additionally, the value of $\gamma$was varied across different layers: 1 for conv5-4, 0.5 for conv4-4, 0.25 for conv3-4, and 0.15 for conv1-4 layers.

Method evaluation employs Distance Precision (DP) and Overlap Success (OS) metrics. A comparison is conducted against other reference methods [11], [102], [103]. The outcomes for the two performance metrics are presented via two curves within One-Pass Evaluation (OPE). The first curve depicts the distance precision rate based on the location error threshold, indicating the portion of frames where tracking results lie within a specific number of pixels from the ground truth. The second curve portrays the success rate relative to the overlap threshold, signifying the percentage of frames where tracking was successful. The location error threshold spans from 0 to 50, while the overlap threshold is adjusted across the range of 0 to 1.

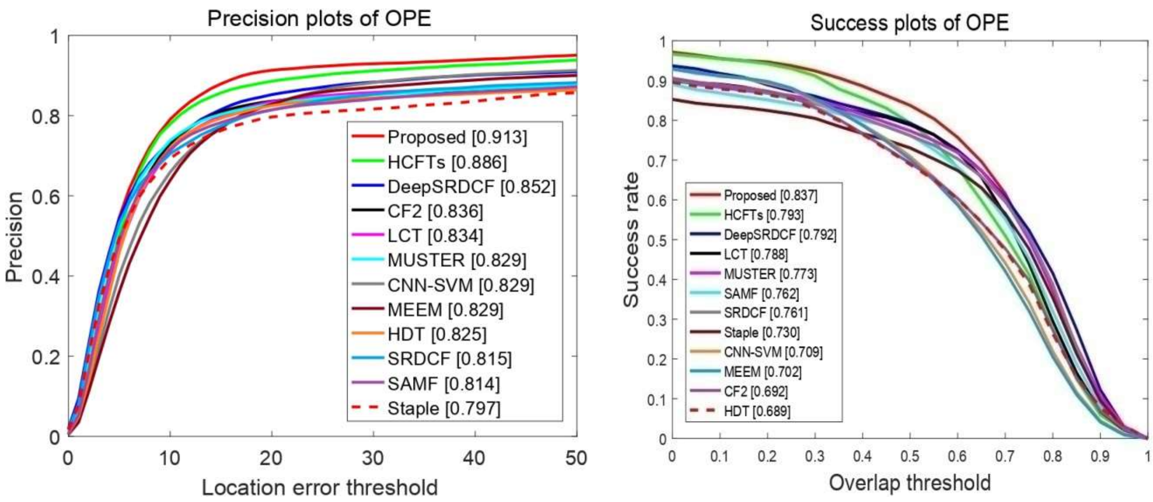
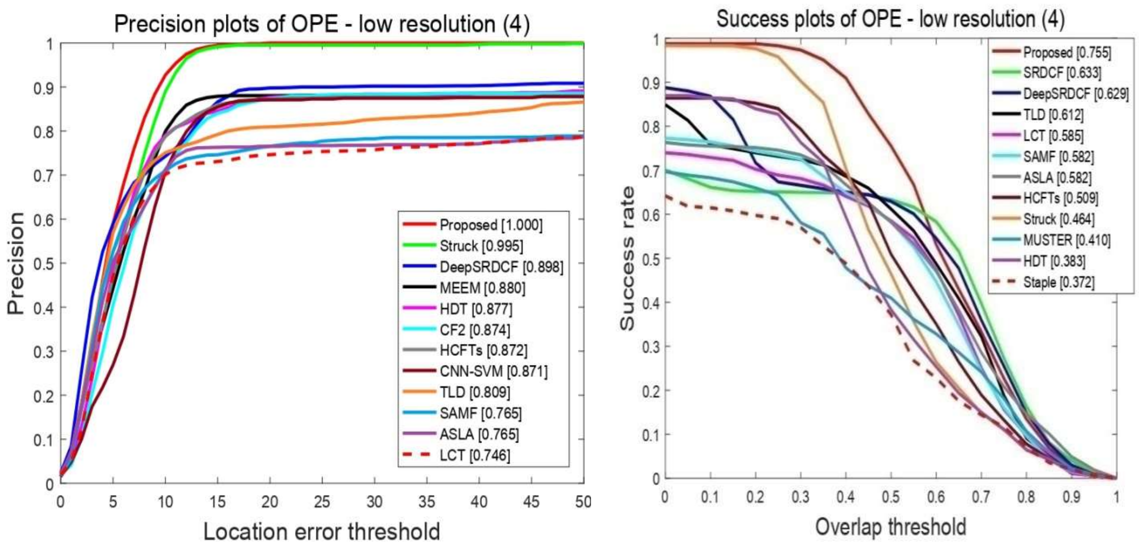
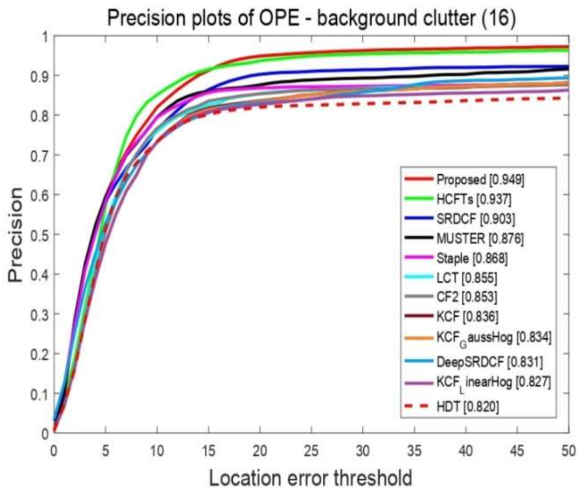
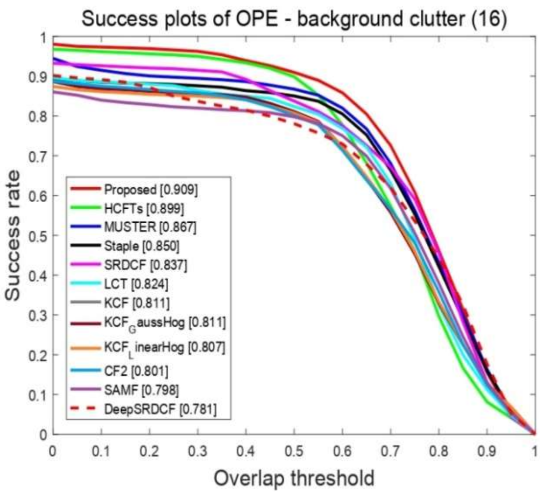


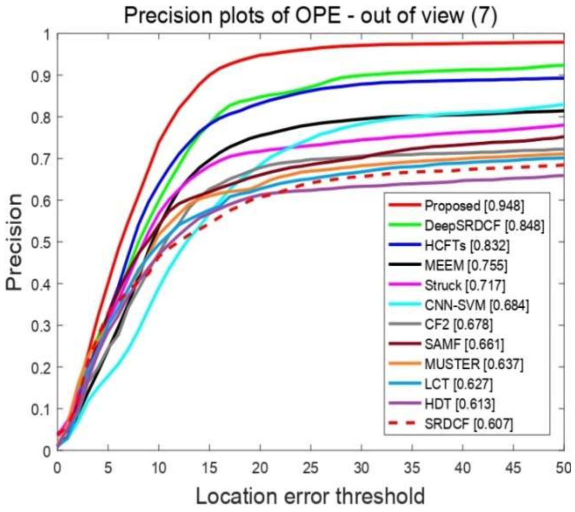
Fig.V.4. Comparison with nine reference trackers using distance precision and overlap success on the OTB-50 database

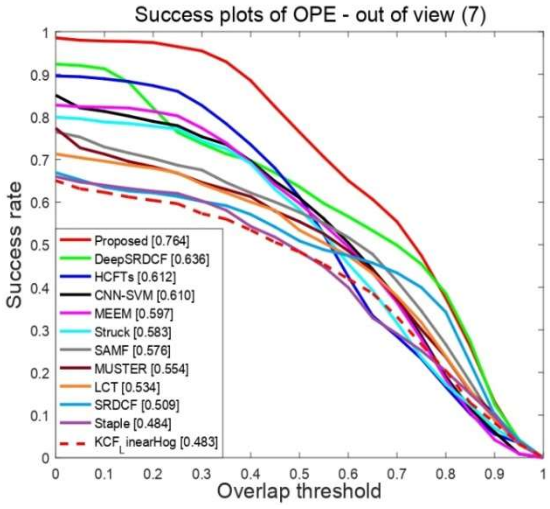
Examining the comparative results presented in figure V.4, one observes the performance metrics on the OTB-50 dataset. Notably, the HCFTs tracker emerges as a standout performer, securing the second-highest levels of efficacy. This is evidenced by its commendable distance precision and overlap success rate. Importantly, the innovative approach introduced in this study shines through as particularly effective, demonstrating notable improvements. Specifically, the proposed method achieves significant enhancements, boasting a noteworthy increase in distance precision and a substantial rise in overlap success. These compelling results underscore the efficacy of the newly suggested approach, positioning it as a promising advancement in the realm of tracking methodologies.

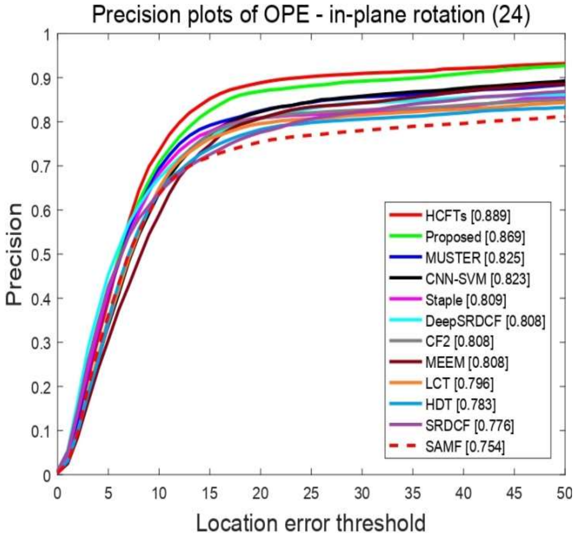


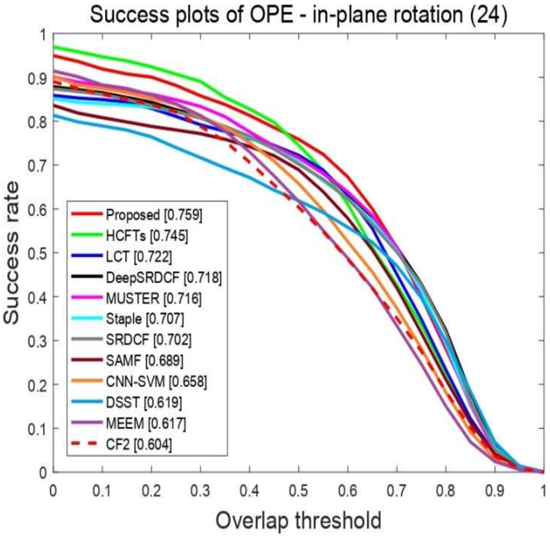

Precision plots of OPE - background clutter (16)
Precision
Location error threshold
0
5
10
15
20
25
30
35
40
45
50
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Proposed [0.949]
HCFTs [0.937]
SRDCF [0.903]
MUSTER [0.876]
Staple [0.868]
LCT [0.855]
CF2 [0.853]
KCF [0.836]
KCF_GaussHog [0.834]
DeepSRDCF [0.831]
KCF_LinearHog [0.827]
HDT [0.820]

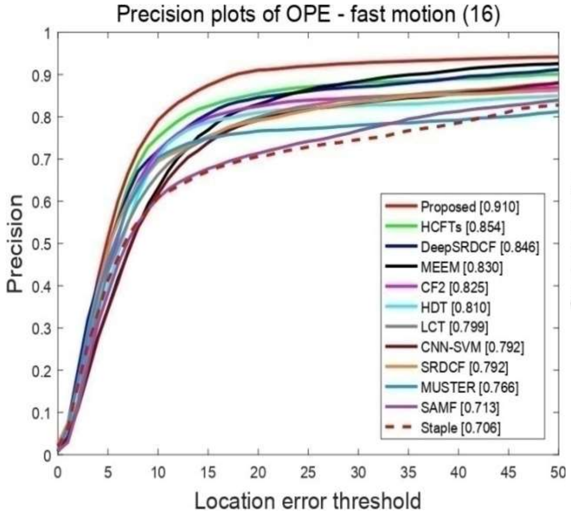

Success plots of OPE - background clutter (16)
Success rate
Overlap threshold
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Proposed [0.909]
HCFTs [0.899]
MUSTER [0.867]
Staple [0.850]
SRDCF [0.837]
LCT [0.824]
KCF [0.811]
KCF_GaussHog [0.811]
KCF_LinearHog [0.807]
CF2 [0.801]
SAMF [0.798]
DeepSRDCF [0.781]


Precision plots of OPE - out of view (7)
Precision
Location error threshold
0
5
10
15
20
25
30
35
40
45
50
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Proposed [0.948]
DeepSRDCF [0.848]
HCFTs [0.832]
MEEM [0.755]
Struck [0.717]
CNN-SVM [0.684]
CF2 [0.678]
SAMF [0.661]
MUSTER [0.637]
LCT [0.627]
HDT [0.613]
SRDCF [0.607]

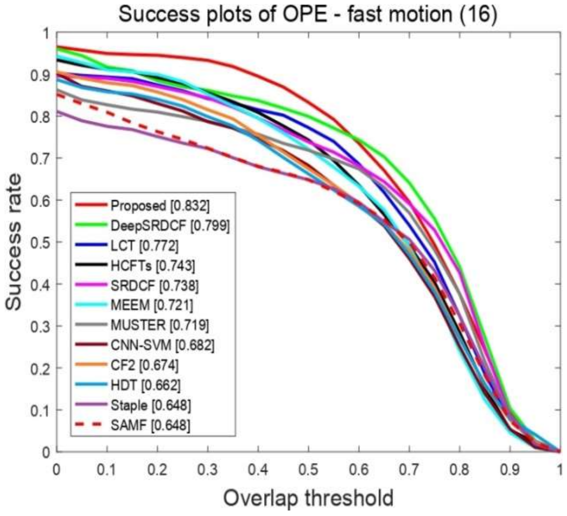

Success plots of OPE - out of view (7)
Success rate
Overlap threshold
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Proposed [0.764]
DeepSRDCF [0.636]
HCFTs [0.612]
CNN-SVM [0.610]
MEEM [0.597]
Struck [0.583]
SAMF [0.576]
MUSTER [0.554]
LCT [0.534]
SRDCF [0.509]
Staple [0.484]
KCF_LinearHog [0.483]


Precision plots of OPE - in-plane rotation (24)
Precision
Location error threshold
0
5
10
15
20
25
30
35
40
45
50
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
HCFTs [0.889]
Proposed [0.869]
MUSTER [0.825]
CNN-SVM [0.823]
Staple [0.809]
DeepSRDCF [0.808]
CF2 [0.808]
MEEM [0.808]
LCT [0.796]
HDT [0.783]
SRDCF [0.776]
SAMF [0.754]

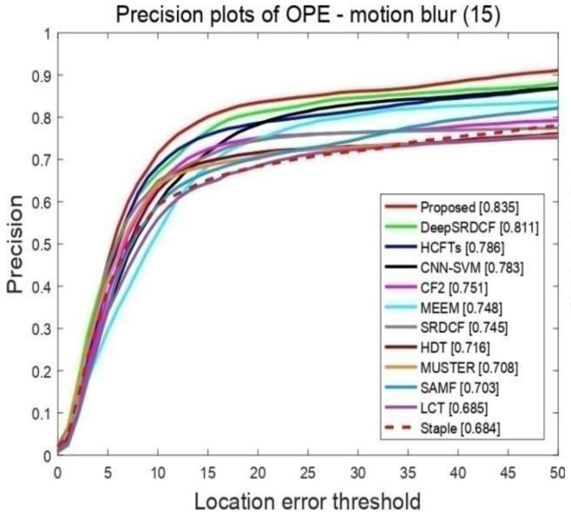

Success plots of OPE - in-plane rotation (24)
Success rate
Overlap threshold
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Proposed [0.759]
HCFTs [0.745]
LCT [0.722]
DeepSRDCF [0.718]
MUSTER [0.716]
Staple [0.707]
SRDCF [0.702]
SAMF [0.689]
CNN-SVM [0.658]
DSST [0.619]
MEEM [0.617]
CF2 [0.604]



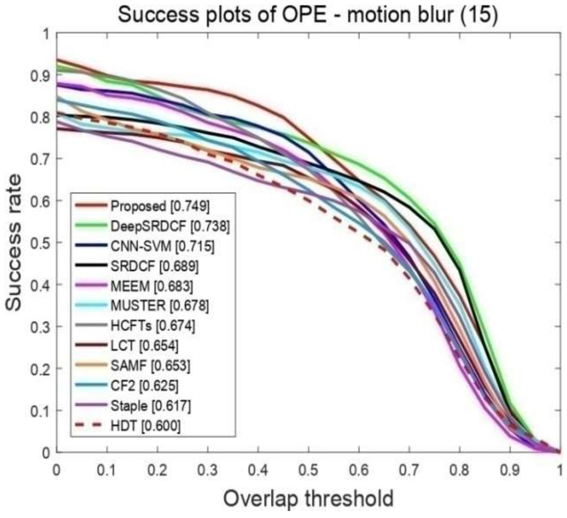
Precision plots of OPE - fast motion (16)



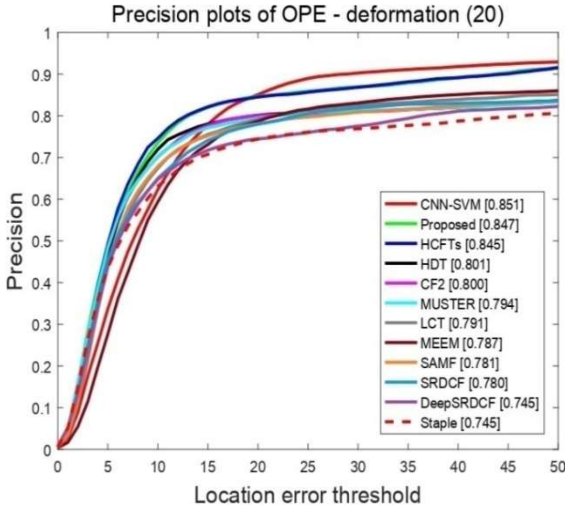
Success plots of OPE - fast motion (16)



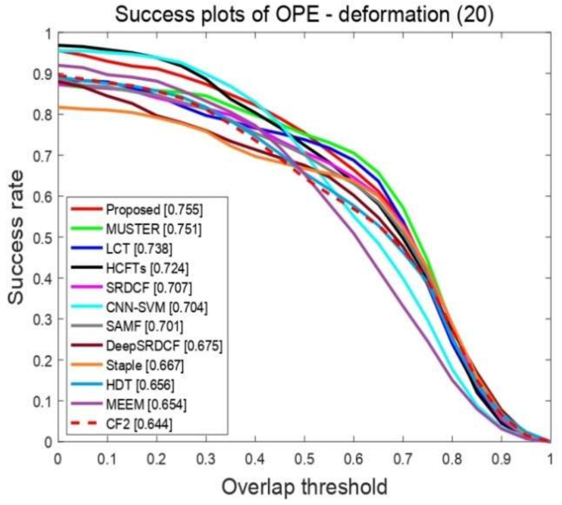
Precision plots of OPE - motion blur (15)



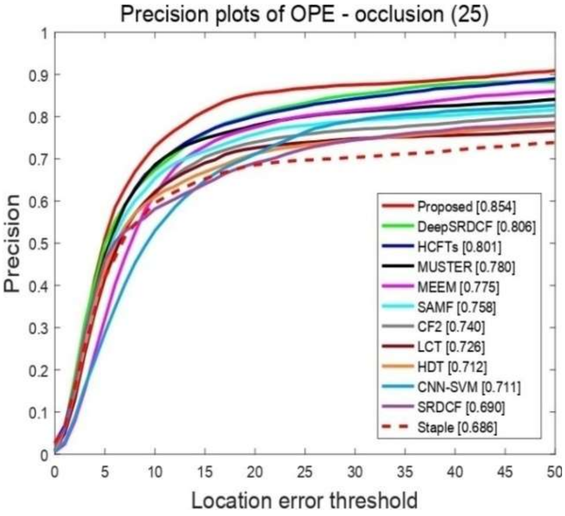
Success plots of OPE - motion blur (15)

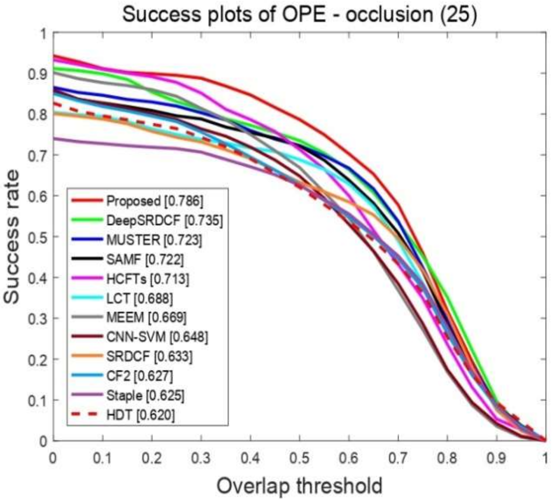


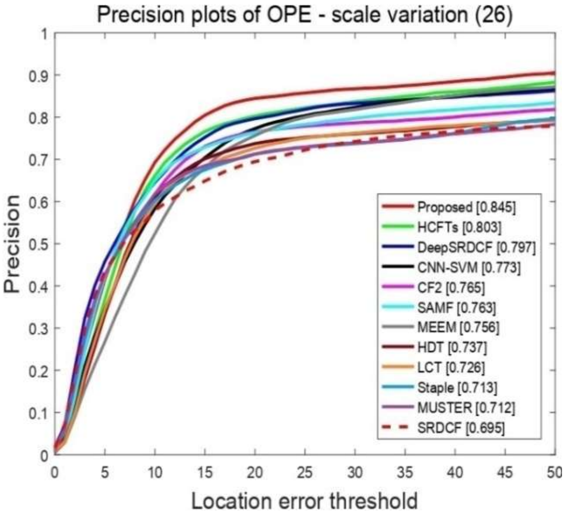
Precision plots of OPE - deformation (20)

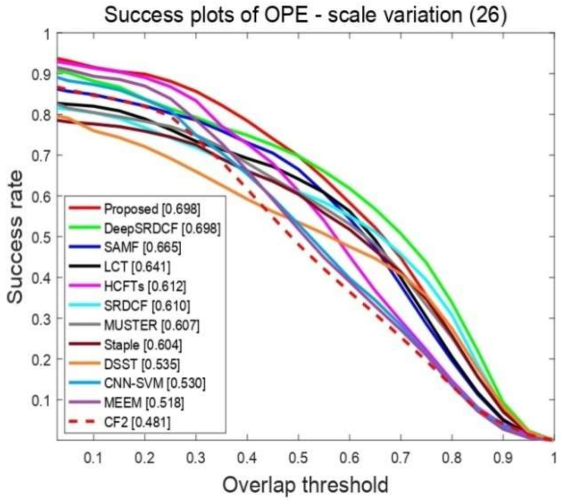


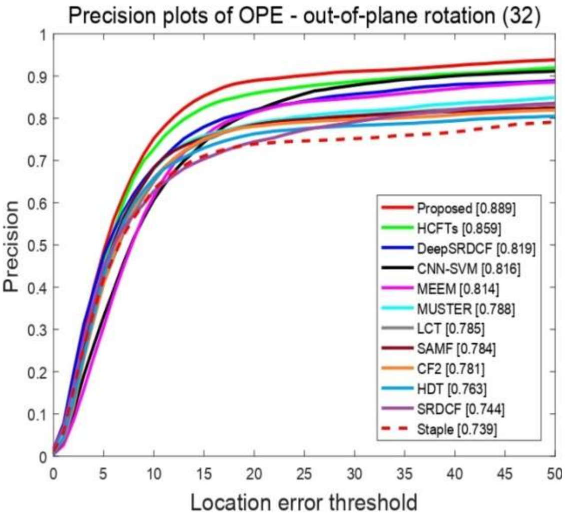
Success plots of OPE - deformation (20)

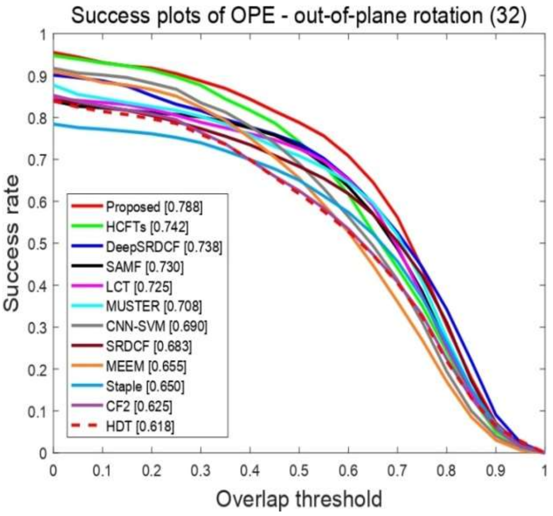

Precision plots of OPE - occlusion (25)
Precision
Location error threshold
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25 30 35 40 45 50
Proposed [0.854]
DeepSRDCF [0.806]
HCFTs [0.801]
MUSTER [0.780]
MEEM [0.775]
SAMF [0.758]
CF2 [0.740]
LCT [0.726]
HDT [0.712]
CNN-SVM [0.711]
SRDCF [0.690]
Staple [0.686]


Success plots of OPE - occlusion (25)
Success rate
Overlap threshold
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Proposed [0.786]
DeepSRDCF [0.735]
MUSTER [0.723]
SAMF [0.722]
HCFTs [0.713]
LCT [0.688]
MEEM [0.669]
CNN-SVM [0.648]
SRDCF [0.633]
CF2 [0.627]
Staple [0.625]
HDT [0.620]


Precision plots of OPE - scale variation (26)
Precision
Location error threshold
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25 30 35 40 45 50
Proposed [0.845]
HCFTs [0.803]
DeepSRDCF [0.797]
CNN-SVM [0.773]
CF2 [0.765]
SAMF [0.763]
MEEM [0.756]
HDT [0.737]
LCT [0.726]
Staple [0.713]
MUSTER [0.712]
SRDCF [0.695]


Success plots of OPE - scale variation (26)
Success rate
Overlap threshold
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Proposed [0.698]
DeepSRDCF [0.698]
SAMF [0.665]
LCT [0.641]
HCFTs [0.612]
SRDCF [0.610]
MUSTER [0.607]
Staple [0.604]
DSST [0.535]
CNN-SVM [0.530]
MEEM [0.518]
CF2 [0.481]


Precision plots of OPE - out-of-plane rotation (32)
Precision
Location error threshold
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 5 10 15 20 25 30 35 40 45 50
Proposed [0.889]
HCFTs [0.859]
DeepSRDCF [0.819]
CNN-SVM [0.816]
MEEM [0.814]
MUSTER [0.788]
LCT [0.785]
SAMF [0.784]
CF2 [0.781]
HDT [0.763]
SRDCF [0.744]
Staple [0.739]


Success plots of OPE - out-of-plane rotation (32)
Success rate
Overlap threshold
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Proposed [0.788]
HCFTs [0.742]
DeepSRDCF [0.738]
SAMF [0.730]
LCT [0.725]
MUSTER [0.708]
CNN-SVM [0.690]
SRDCF [0.683]
MEEM [0.655]
Staple [0.650]
CF2 [0.625]
HDT [0.618]

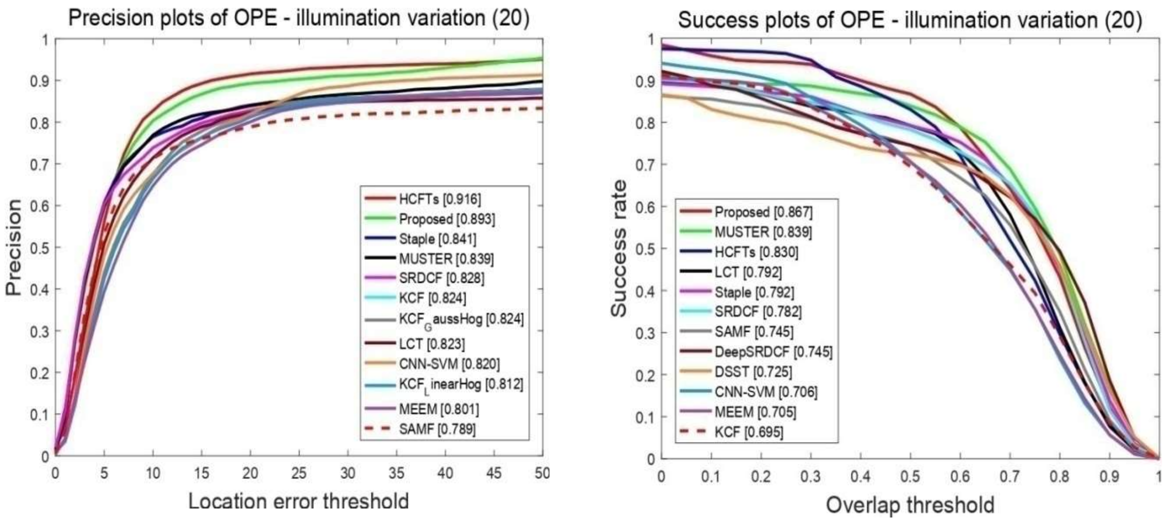


Fig.V.5. The charts depict how well tracking performs in 11 different challenging scenarios, measured in terms of both overlap success and distance precision.

Figure V.5 exhibits the plotted graphs showcasing the overlap success rate and distance precision attained through the rigorous evaluation using the OTB50 dataset. This comprehensive assessment specifically centers on 11 intricate scenarios, each presenting unique challenges encompassing scale variation, fast motion, in-plane rotation, deformation, motion blur, occlusion, illumination variation, out-of-plane rotation, background clutter, out-of-view, and low resolution. A meticulous scrutiny of all the sub-figures within figure V.5 illuminates a compelling observation: the newly proposed tracker excels remarkably when compared to existing state-of-the-art counterparts across various aspects. However, it's noteworthy to mention that in the case of deformation, the proposed tracker doesn't outperform its counterparts, standing as an exception amidst its otherwise superior performance across the spectrum of challenging scenarios.

We utilized OTB2015 for further assessment of the proposed tracker. Tables V.1 and V.2 present the DP and OS rate results across 11 diverse attributes for the proposed tracker and 10 other trackers. Notably, our tracker excels in 9 out of 11 attributes in terms of DP, encompassing scale variation, fast motion, deformation, motion blur, occlusion, illumination variation, out-of-plane rotation, background clutter, and out-of-view scenarios.

Additionally, it demonstrates superior performance in nine attributes regarding the overlap success rate. However, the proposed tracker exhibits limitations in low-resolution scenes due to the challenge of capturing adequate features in such settings. Despite this, the proposed tracker surpasses its state-of-the-art counterparts overall.

Tab.V.1 The distance precision results achieved by the proposed tracker and other 10 trackers on 11 different attributes on the OTB-2015 benchmark. The best, second best, and third best values are highlighted in red, green, and blue, respectively

	Proposed	MemDTC	DeepSRDCF	SRDCFdecon	MemTrack	DRVT	BACF	SRDCF	HCFTs	CF2	siamfc3s
ALL	**0.901**	0.845	**0.851**	0.825	0.820	0.834	0.824	0.789	**0.870**	0.837	0.771
IV	**0.900**	0.805	0.791	**0.835**	0.793	0.822	0.831	0.792	**0.888**	0.817	0.736
SV	**0.871**	0.818	0.819	0.805	0.799	**0.820**	0.774	0.745	**0.827**	0.799	0.735
OCC	**0.864**	0.797	**0.825**	0.768	0.762	0.781	0.745	0.735	**0.814**	0.767	0.722
DEF	**0.875**	0.783	0.783	0.753	0.718	**0.792**	0.778	0.734	**0.826**	0.791	0.690
MB	**0.864**	0.790	**0.823**	0.814	0.767	0.742	0.766	0.767	**0.822**	0.804	0.705
FM	**0.849**	0.814	0.814	0.775	0.797	0.776	0.808	0.769	**0.823**	**0.815**	0.743
IPR	**0.893**	0.829	0.818	0.776	0.818	0.792	0.795	0.745	**0.895**	**0.854**	0.742
OPR	**0.891**	**0.844**	0.835	0.797	0.817	0.811	0.787	0.742	**0.849**	0.807	0.756
OV	**0.825**	**0.804**	**0.781**	0.641	0.720	0.751	0.765	0.597	0.746	0.677	0.669
BC	**0.930**	0.802	0.841	**0.850**	0.794	0.789	0.830	0.775	**0.887**	0.843	0.775
LR	0.950	**0.995**	0.847	0.747	**0.998**	**1.000**	0.795	0.765	0.860	0.847	0.900

Tab.V.2. The overlap success rate results achieved by the proposed tracker and other 10 trackers on 11 different attributes on the OTB-2015 benchmark. The best, second best, and third best values are highlighted in red, green, and blue, respectively

	Proposed	MemDTC	DeepSRDCF	SRDCFdecon	MemTrack	DRVT	BACF	SRDCF	HCFTs	CF2	siamfc3s
ALL	**0.645**	**0.637**	**0.635**	0.627	0.626	0.625	0.621	0.598	0.598	0.562	0.582
IV	**0.656**	0.624	0.621	**0.646**	0.614	0.631	**0.634**	0.613	0.603	0.540	0.568
SV	0.604	**0.608**	0.605	**0.607**	0.602	**0.611**	0.576	0.561	0.525	0.485	0.552
OCC	**0.629**	**0.604**	**0.601**	0.589	0.581	0.592	0.576	0.559	0.558	0.525	0.543
DEF	**0.616**	0.568	0.566	0.553	0.539	**0.569**	**0.583**	0.544	0.560	0.530	0.506
MB	**0.662**	0.625	**0.642**	**0.639**	0.611	0.603	0.586	0.594	0.606	0.585	0.550
FM	**0.638**	**0.626**	**0.628**	0.606	0.623	0.611	0.606	0.597	0.581	0.570	0.568
IPR	**0.619**	**0.613**	0.589	0.573	**0.606**	0.588	0.584	0.544	0.603	0.559	0.557
OPR	**0.627**	**0.619**	**0.607**	0.591	0.605	0.601	0.584	0.550	0.573	0.534	0.557
OV	**0.608**	**0.590**	0.553	0.510	0.549	**0.574**	0.552	0.460	0.519	0.474	0.506
BC	**0.665**	0.610	**0.627**	**0.641**	0.599	0.595	0.625	0.583	0.620	0.585	0.523
LR	0.505	**0.665**	0.561	0.517	**0.684**	**0.635**	0.514	0.514	0.435	0.388	0.618

These findings affirm the superiority of the proposed tracker, achieving notable values of 91.3% for the DP score and 83.7% for the AUC score compared to other trackers.

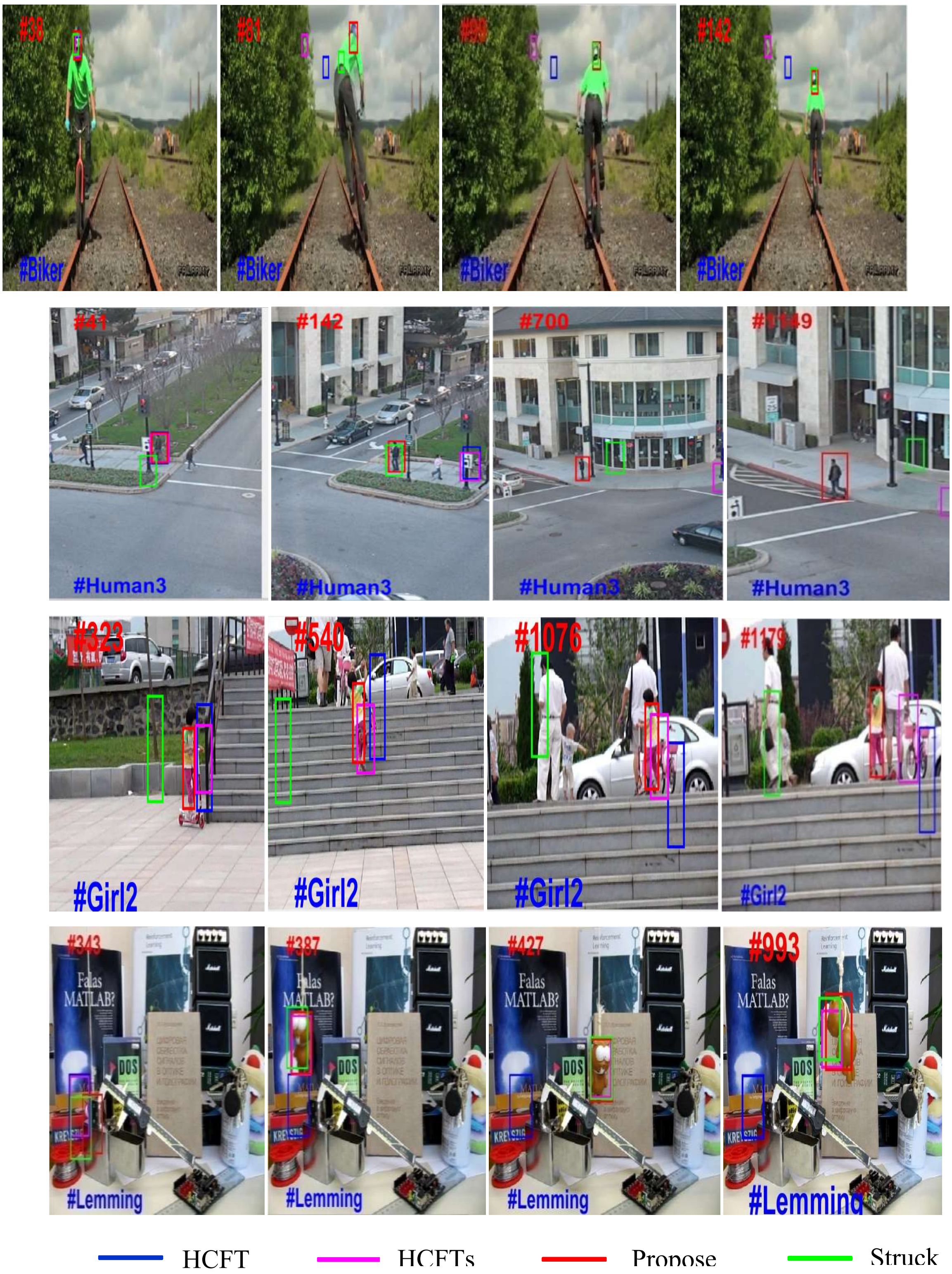


Fig.V.6. Showcases the qualitative results of our proposed method, along with HCFT [11],[103], HCFTs [102], and Struck [103], on four challenging sequences.

In figure V.6, a compilation of tracking results stemming from a selected subset of the OTB-50 benchmark sequences is showcased. The primary objective here revolves around a qualitative assessment of the trackers' performance, featuring a comparison between HCFTs [11],[102],[103], Struck[103], and the recently proposed tracker. This evaluation is specifically carried out on four intricate sequences, distinguished by blue, magenta, green, and red markers, denoting the sequences Biker, Human3, Girl2, and Lemming, respectively, arranged in a top-to-bottom sequence. Each of these sequences presents a unique set of challenges spanning scale variation, occlusion, motion blur, rapid motion, out-of-plane rotation, low resolution, deformation, background clutter, and out-of-view scenarios.

The experimental findings distinctly outline Struck's adeptness in skillfully navigating a range of challenges, encompassing scale variation, occlusion, motion blur, and background clutter. This particular strength is prominently showcased in sequences such as Biker and Lemming. However, when faced with deformation, Struck's effectiveness diminishes, notably revealing a limitation in sequences like Human3 and Girl2.

Contrastingly, HCFTs exhibit remarkable proficiency in scenarios involving scale variation, occlusion, motion blur, and background clutter, notably excelling in sequences like Lemming. However, its effectiveness falters when handling deformation challenges, evident in sequences Biker, Human3, and Girl2.

Across all mentioned sequences (Biker, Human3, Girl2, and Lemming), HCFT displays reduced effectiveness. In contrast, our proposed method consistently demonstrates precise target tracking, surpassing HCFTs, HCFT, and Struck. Particularly noteworthy is its excellence in tracking small targets due to its heightened robustness.

V.5. **Conclusion**

The suggested algorithm undergoes thorough validation utilizing the OTB50 datasets. The evaluation of the proposed trackers' performance hinges on two key metrics: AUC and the DP. Additionally, to validate the performance of the proposed tracker, a comprehensive comparison is conducted against various trackers. The simulation results unequivocally showcase the superior performance of the proposed tracker over numerous contemporary counterparts, particularly excelling in demanding scenarios characterized by background clutter, motion blur, partial occlusions, and diverse appearance alterations.

Conclusion

The work presented in this Doctoral thesis makes contributions to tracking a singular object visually problem, focusing keenly on estimating the precise position of a target object within every frame constituting a video sequence.

In this work, we are particularly interested to employ correlation filter-based tracking methodologies.

The selection of these methods was underpinned by their remarkable performance, notable computational efficiency, and their effectiveness in updating models seamlessly. The standout attributes of these correlation filter-based tracking techniques, including their superior performance metrics and computational efficacy, made them an ideal choice for addressing the complexities associated with real-time object tracking scenarios. Their efficiency in accurately estimating object positions across video frames aligned strategically with the core objectives of this thesis, emphasizing the significance of their adoption within this research endeavor.

Within this thesis, a novel approach aimed at enhancing visual object tracking algorithms was introduced, capitalizing on an effective fusion of CNN layers' features, Hog features, and coefficients derived from the image's DCT. This method leveraged hierarchical CNNs, trained extensively on a large-scale database, to derive a multifaceted strategy. Specifically, the output layers of the CNNs were harnessed to retain the semantic essence of target objects, rendering them robust against substantial appearance variations. In contrast, the input layers of the CNNs were exploited to encode finer spatial details crucial for precise localization. Combining these features with intricate details simultaneously enriched the visual object tracking process.

To deduce the target's location, a linear correlation filter was trained on each CNN layer, facilitating a coarse-to-fine estimation through hierarchical correlation maps. Simultaneously, to enhance the tracker's accuracy and combat drifting issues during the correlation filter's update, this thesis proposed a real-time approach based on training the correlation filter on

HOG features. This method served as a means to update the filters derived from CNN and HOG features. Moreover, the integration of DCT served dual purposes: it replaced RGB in computing HOG features and supplemented CNN features for images with high saturation, bolstering their performance.

Additionally, the employment of Newton's method enhanced long-term memory regarding the target's appearance and aided in recovery from tracking failures. Extensive simulations conducted within this study demonstrated the superiority of the proposed tracker over numerous contemporary counterparts. However, despite its proven effectiveness, further robustness enhancements were essential for this proposed tracker to become a versatile solution applicable across diverse scenarios, aiming to minimize deficiencies compared to existing counterparts.

In the following, we give a summary of the results obtained and perspectives on work.

In Chapter II, we presented the state-of-the-art of visual tracking. We then gave a detailed introduction to visual tracking, along with the challenges encountered and two different types of tracking algorithms.

In Chapter III, we gave a detailed overview of image color spaces and imported methods of Image Features extraction, Convolutional Neural Networks, and Histograms of Oriented Gradients.

In Chapter IV, we presented in detail the main steps of the proposed methods.

In Chapter V, first, we presented Benchmark Datasets, followed by an assessment of visual object tracking performance. Second, a comprehensive examination of results and discussions pertaining to each database was provided.

Finally, this work was concluded with a general conclusion in which the main results obtained were presented, and the perspectives to be considered as a follow-up to this work were discussed.

Perspectives

The work done throughout this thesisopens up various perspectives. Among the issues not detailed here, which could be the subject of future research, one main avenue is the possible improvement of the methods used in this work by employing alternative approaches that mimic the use of deep artificial intelligence.

Bibliography

[1] J. Yang, W. Tang and Z. Ding, "Long-Term Target Tracking of UAVs Based on Kernelized Correlation Filte," International Journal of Innovative Computing, Information and Control. 2021, vol.9, no. 23, pp. 1-18.

[2] C. Ma, J. B. Huang, X. Yang, and M. H. Yang, "Robust Visual Tracking via Hierarchical Convolutional Features," IEEE Transactions on Pattern Analysis and Machine Intelligence. 2019, vol. 41, no. 11, pp. 2709--2723.

[3] H. Grabner, C. Leistner, H. Bischof, Semi-supervised on-line boosting for robust tracking, In: Forsyth, D., Torr, P., Zisserman, A. (eds) Computer Vision ECCV 2008. ECCV 2008. Lecture Notes in Computer Science. 2008, vol. 5302, Springer, Berlin, Heidelberg.

[4] J. Zhang, J. Sun, J. Wang et al., "Visual object tracking based on residual network and cascaded correlation filters," Springer, Journal of Ambient Intelligence and Humanized Computing. 2021, vol. 12, pp. 8427--8440 .

[5] J. Fan, W. Xu, Y. Wu, Y. Gong, "Human tracking using convolutional neural networks," Transactions on Neural Networks. 2010, vol. 21, no. 10, pp. 1610--1623.

[6] L. Yang, C. Kong, X. Chang et al, "Correlation filters with adaptive convolution response fusion for object tracking, " Knowledge-Based Systems. 2021, vol. 228, 107314.

[7] B. Babenko, M. -H. Yang and S. Belongie, "Robust Object Tracking with Online Multiple Instance Learning," IEEE Transactions on Pattern Analysis and Machine Intelligence. 2011, vol. 3.

[8] C. Huang, S. Lucey and D. Ramanan, Learning policies for adaptive tracking with deep feature cascades, IEEE International Conference on Computer Vision (ICCV). Venice, Italy, 2017, pp. 105--114.

[9] N. Wang and D.-Y. Yeung, Learning a Deep Compact Image Representation for Visual Tracking, Advances in Neural Information Processing Systems 26 (NIPS 2013). 2013.

[10] A. He, C. Luo, X. Tian et al., A twofold siamese network for real-time object tracking, IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA, 2018, pp. 4834--4843.

[11] C. Ma, J. -B. Huang, X. Yang et al., Hierarchical Convolutional Features for Visual Tracking, IEEE International Conference on Computer Vision (ICCV). Santiago, Chile, 2015, pp. 3074--3082.

[12] M. Y. Abbass, K. C. Kwon, N. Kim et al., "Efficient Object Tracking Using Hierarchical Convolutional Features Model and Correlation Filters," Springer-Verlag, The Visual Computer, 2021, vol. 37, no. 04, pp. 831--842.

[13] B. Latreche, S. Saadi, M. Kious et al., "A novel hybrid image fusion method based on integer lifting wavelet and discrete cosine transformer for visual sensor networks," Springer, Multimedia Tools and Applications, 2019, vol. 78, pp. 10865--10887.

[14] H. Chen, W. Zhang, X. Zhao et al., DCT representations based appearance model for visual tracking, IEEE International Conference on Robotics and Biomimetics (ROBIO 2014). Bali, Indonesia, 2014, pp. 1614--1619.

[15] D. He, Z. Gu and N. Cercone, Efficient image retrieval in DCT domain by hypothesis

testing, 16th IEEE International Conference on Image Processing (ICIP). Cairo, Egypt, 2019, pp. 225--228.

[16] M. Uzair, A. Mahmood, A. Mian, Hyperspectral Face Recognition using 3D-DCT and Partial Least Squares, In Proceedings of the British Machine Vision Conference. BMVA, 2013, vol. 78, pp. 10pp.

[17] D. Chen, Q. Liu, M. Sun and J. Yang, "Mining Appearance Models Directly From Compressed Video," IEEE Transactions on Multimedia, 2008, vol. 10, no. 02, pp. 268--276.

[18] B. K. Shreyamsha Kumar, M. N. S. Swamy and M. Omair Ahmad, "Visual tracking using structural local DCT sparse appearance model with occlusion detection," Springer, MMultimedia Tools and Applications, 2019, vol. 78, pp. 7243--7266.

[19] W. -h. Yun, D. Kim, B. Song et al., Block comparison based face identification using HOG feature, The 18th IEEE International Symposium on Robot and Human Interactive Communication, Toyama, Japan, 2009, pp. 484--487.

[20] Y. Wei, Q. Tian and T. Guo, "An Improved Pedestrian Detection Algorithm Integrating Haar-Like Features and HOG Descriptors," Advances in Mechanical Engineering, 2013, vol. 05, pp. 484--487.

[21] Y. Li, Y. Zhang, Y. Xu et al., "Robust scale adaptive kernel correlation filter tracker with hierarchical convolutional features," IEEE Signal Processing Letters, 2016, vol. 23, no. 08, pp. 1136--1140.

[22] S.Stabinger,A.Rodríguez-Sánchez,andJ.Piater,“25 yearsofCNNS:Canwecomparetohuman abstraction capabilities?,” in Lecture Notes in Computer Science (including subseriesLectureNotesinArtificialIntelligenceandLectureNotesinBioinformatics),2016,vol.9887LNCS, pp. 380–387. doi: 10.1007/978-3-319-44781-0_45.

[23] M. Y. Abass, KC. Kwon, N. Kim, et al. "A survey on online learning for visual tracking," The Visual Computer. 2021, vol. 37, pp. 993--1014.

[24] SatyaMallick,“ObjectTrackingusingOpenCV(C++/Python),”Feb.17,2017.https://learnopencv.com/object-tracking-using-opencv-cpp-python/(accessedJun.14, 2022).

[25] X. Mei, H. Ling: "Robust visual tracking using l1 minimization ". In IEEE 12thInternationalConferenceonComputer Vision,2009, pp. 1436–1443.

[26] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang, “Incremental learning for robust visualtracking,” International Journal of Computer Vision, vol. 77, no. 1–3, pp. 125–141, May2008,doi: 10.1007/s11263-007-0075-7.

[27] N. Dalal, “Finding people in images and videos,” Ph.D. dissertation, Institut National Polytechnique de Grenoble-INPG, 2006.

[28] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1. IEEE, 2005, pp. 886–893.

[29] N.Dardagan,A.Brđanin,D.Džigal,andA.Akagic,“MultipleObjectTrackersinOpenCV:ABenchmark,”Oct. 2021,[Online].Available: http://arxiv.org/abs/2110.05102

[30] S. He, Q. Yang, R. W. H. Lau, J. Wang, and M. H. Yang, “Visual tracking via localitysensitive histograms,” in Proceedings of the IEEE Computer Society Conference on Com-puterVisionandPatternRecognition,2013,pp.2427–2434.doi:10.1109/CVPR.2013.314.

[31] X.Li,W.Hu,C.Shen,Z.Zhang,A.Dick,andA.vandenHengel,“ASurveyofAppearanceModelsin VisualObjectTracking,”Mar.2013,[Online].Available:http://ar-xiv.org/abs/1303.4803

[32] S.Medouakh.“Détectionetsuivid’objets.’’Thèsededoctorat.UniversitédeBiskra,2019.

[33] Yang. H. : Vers un suivi robuste d’objets visuels : sélection de propositions et traitementdesocclusions. 2016.

[34] D.Riahi,Suivimulti-objetsparladétection :«Applicationàlavidéo surveillance.Diss». EcolePolytechnique,Montréal (Canada),2016.

[35] H. Grabner, M. Grabner, and H. Bischof, “Real-time tracking via on-line boosting,” inBMVC 2006 - Proceedings of the British Machine Vision Conference 2006, 2006, pp. 47–56. doi: 10.5244/c.20.6.

[36] B. Babenko, M.-H. Yang, and S. Belongie, “Visual tracking with online multiple instancelearning,” in 2009 IEEE Conference on computer vision and Pattern Recognition.

IEEE,2009, pp. 983–990.

[37] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” IEEE transactionsonpattern analysis and machineintelligence, vol.34, no.7, pp. 1409–1422, 2011.

[38] B.D.Lucas,T.Kanade, etal. (1981).“Aniterativeimageregistrationtechniquewithan1pplicationto stereo vision.’’InIJCAI, volume81, pages674–679

[39] D. Comaniciu, V. Ramesh, and P. Meer, “Real-Time Tracking of Non-Rigid Objects usingMean Shift.” In Computer Vision and Pattern Recognition, 2000. Proceedings. IEEEConferenceon, volume2, pages 142–149.IEEE.

[40] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, “Visual object tracking usingadaptivecorrelationfilters,”in2010IEEEComputerSocietyConferenceonComputerVisionand Pattern Recognition.IEEE,2010, pp. 2544–2550.

[41] J.F.Henriques,R.Caseiro,P.Martins,andJ.Batista,“High-speedtrackingwithkernelizedcorrelationfilters,”IEEEtransactionsonpatternanalysisandmachineintelligence,vol.37,no. 3, pp. 583–596, 2014.

[42] broutonlap,“ACompleteReviewoftheOpenCVObjectTrackingAlgorithms[Blogpost].”https://broutonlab.com/blog/opencv-object-tracking(accessedJun. 14,2022).

[43] A.Lukezic,T.Vojir,L.CehovinZajc,J.Matas,andM.Kristan,“Discriminativecorrelationfilterwithchannelandspatialreliability,”inProceedingsoftheIEEEConferenceonCom-puterVision and PatternRecognition, 2017, pp. 6309–6318.

[44] D.Held,S.Thrun,andS.Savarese,“LearningtoTrackat100FPSwithDeepRegressionNetworks,” inEuropeanConferenceonComputerVision.Springer,2016,pp.749–765.

[45] SatyaMallick,“ObjectTrackingusingOpenCV(C++/Python),”Feb.17,2017.https://learnopencv.com/object-tracking-using-opencv-cpp-python/(accessedJun.14, 2022).

[46] X. Mei and H. Ling, “Robust visual tracking using ℓ 1 minimization,” in 2009 IEEE 12th international conference on computer vision, 2009, pp. 1436–1443.

[47] L. Matthies, T. Kanade, and R. Szeliski, “Kalman filter-based algorithms for estimating depth from image sequences,” Int. J. Comput. Vis., vol. 3, no. 3, pp. 209–238, 1989.

[48] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” Int. J. Comput. Vis., vol. 1, no. 4, pp. 321–331, 1988.

[49] H. Nam and B. Han, “Learning multi-domain convolutional neural networks for visual tracking,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4293–4302.

[50] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, “Learning dynamic siamese network for visual object tracking,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 1763–1771.

[51] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, “Incremental learning for robust visual tracking,” Int. J. Comput. Vis., vol. 77, no. 1, pp. 125–141, 2008.

[52] A. Abdel-Hadi, “Real-time object tracking using color-based Kalman particle filter,” in The 2010 International Conference on Computer Engineering & Systems, 2010, pp. 337–341.

[53] F. Yang, H. Lu, and M.-H. Yang, “Robust superpixel tracking,” IEEE Trans. Image Process., vol. 23, no. 4, pp. 1639–1651, 2014.

[54] S. Hare et al., “Struck: Structured output tracking with kernels,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 10, pp. 2096–2109, 2015.

[55] M. Godec, P. M. Roth, and H. Bischof, “Hough-based tracking of non-rigid objects,” Comput. Vis. Image Underst., vol. 117, no. 10, pp. 1245–1256, 2013.

[56] Q. Wang, F. Chen, W. Xu, and M.-H. Yang, “Object tracking via partial least squares analysis,” IEEE Trans. Image Process., vol. 21, no. 10, pp. 4454–4465, 2012.

[57] A. Abdel-Hadi, “Real-time object tracking using color-based Kalman particle filter,” in The 2010 International Conference on Computer Engineering & Systems, 2010, pp. 337–341

[58] Z. Han, T. Xu, and Z. Chen, “An improved color-based tracking by particle filter,” in Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE), 2011, pp. 2512–2515.

[59] F. Yang, H. Lu, and M.-H. Yang, “Robust superpixel tracking,” IEEE Trans. Image Process., vol. 23, no. 4, pp. 1639–1651, 2014.

[60] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-Speed Tracking with Kernelized Correlation Filters,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 3, pp. 583–596, Apr. 2014, doi: 10.1109/tpami.2014.2345390.

[61] D. S. Bolme, B. A. Draper, and J. R. Beveridge, “Average of synthetic exact filters,” in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2105–2112.

[62] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, “Accurate scale estimation for robust visual tracking,” BMVC 2014 - Proc. Br. Mach. Vis. Conf. 2014, 2014, doi: 10.5244/C.28.65.

[63] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, “Visual object tracking using adaptive correlation filters,” in 2010 IEEE computer society conference on computer vision and pattern recognition, 2010, pp. 2544–2550.

[64] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “Exploiting the circulant structure of tracking-by-detection with kernels,” Springer Berlin Heidelb., vol. Computer V, pp. 702–715, 2012, doi: 10.1007/978-3-642-33765-9_50.

[65] X. Yang, C. Ma, J.-B. Huang, and M.-H. Yang, “Hierarchical Convolutional Features for Visual Tracking,” Proc. IEEE Int. Conf. Comput. Vis., pp. 3074–3082, 2015, doi: 10.1109/ICCV.2015.352.

[66] S. Hong, T. You, S. Kwak, and B. Han, “Online tracking by learning discriminative saliency map with convolutional neural network,” in International conference on machine learning, 2015, pp. 597–606.

[67] M. Danelljan, A. Robinson, F. Shahbaz Khan, and M. Felsberg, “Beyond correlation filters: Learning continuous convolution operators for visual tracking,” in European conference on computer vision, 2016, pp. 472–488.

[68] L. Zhang, A. Gonzalez-Garcia, J. van de Weijer, M. Danelljan, and F. S. Khan, “Learning the model update for siamese trackers,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 4010–4019.

[69] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, “Fully-convolutional siamese networks for object tracking,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9914 LNCS, pp. 850–865, 2016, doi: 10.1007/978-3-319-48881-3_56.

[70] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, “Fast online object tracking and segmentation: A unifying approach,” in Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, 2019, pp. 1328–1338.

[71] L. Zhang, A. Gonzalez-Garcia, J. van de Weijer, M. Danelljan, and F. S. Khan, “Learning the model update for siamese trackers,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 4010–4019.

[72] G. Wang, C. Luo, X. Sun, Z. Xiong, and W. Zeng, “Tracking by instance detection: A meta-learning approach,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 6288–6297.

[73] L. Huang, X. Zhao, and K. Huang, “Bridging the gap between detection and tracking: A unified approach,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3999–4009.

[74] Ford, Adrian, and Alan Roberts. "Colour space conversions.: Westminster University, London (1998).

[75] Y. M. P. Prajakta M.Patil, “Robust Skin Colour Detection And Tracking Algorithm,” Int. J. Eng. Res. Technol., vol. 1, no. 8, pp. 1–6, 2012.

[76] A. Karpathy, F. Li, and J. Johnson, “Cs231n: Convolutional neural networks for visual recognition, 2016,” URL http://cs231n. github. io, 2017.

[77] Gritzman, A.D., Rubin, D.M., Pantanowitz, A.: Comparison of colour transforms used in lip segmentation algorithms, Signal Image Video Processing., vol 9, pp: 947-957 2015.

[78] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Adv. Neural Inf. Process. Syst., vol. 25, 2012.

[79] V. Romanuke, “Appropriate number and allocation of ReLUs in convolutional neural networks,” Res. Bull. Natl. Tech. Univ. Ukr. Kyiv Politech. Institute", no. 1, pp. 69–78, 2017.

[80] Karen Simonyan & Andrew Zisserman, (2015). « Very Deep Convolutional NetworksnFor Large-Scale Image Recognition », ICLR 2015.

[81] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.

[82] W. Wei, X.-L. Yang, B. Zhou et al., "Combined energy minimization for image reconstruction from few views," Hindawi, Mathematical Problems in Engineering, 2012, vol. 2012, pp. 1--15.

[83] W. Wei, Z. Sun, H. Song et al., "Energy balance-based steerable arguments coverage method in WSNs," IEEE Access. 2018, vol. 06, pp. 33766--33773.

[84]

[85] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking usingadaptivecorrelationfilters,"in2010IEEEComputerSocietyConferenceonComputerVi-sionand Pattern Recognition.IEEE,2010, pp. 2544–2550.

[86] M. Danelljan, G. Häger, S. Khan, and M. Felsberg, "Accurate Scale Estimation for RobustVisual Tracking Ours ASLA SCM Struck LSHT." In: British machine vision conference,2014, pp 1–11.DOI: 10.5244/C.28.65

[87] J.F.Henriques,R.Caseiro,P.Martins,andJ.Batista,"High-speedtrackingwithkernelizedcorrelationfilters,"IEEEtransactionsonpatternanalysisandmachineintelligence,vol.37,no. 3, pp. 583–596, 2014.

[88] Danelljan M, Robinson A, Khan F S, Felsberg M. Beyond correlation filters: Learn-ingcontinuousconvolutionoperatorsforvisualtracking. In:Europeanconferenceoncom-putervision, (2016), pp 472–488

[89] A.Lukezic,T.Vojir,L.CehovinZajc,J.Matas,andM.Kristan,"Discriminativecorrelationfilterwithchannelandspatialreliability,"inProceedingsoftheIEEEConferenceonCom-puterVision and PatternRecognition, 2017, pp. 6309–6318.

[90] Stack Overflow Documentation. (n.d.). Python® Notes for Professionals [e-book]. Re-trievedfrom https://goalkicker.com/PythonBook/

[91] S.Liu,D.Liu,G.Srivastava,D.Połap,andM.Woźniak,"Overviewandmethodsofcorre-lationfilteralgorithmsinobjecttracking,"Complex&IntelligentSystems,vol.7,no.4,pp.1895–1917,Aug. 2021, doi: 10.1007/s40747-020-00161-4.

[92] M. Danelljan, G. Hager, F. S. Khan et al., Learning Spatially Regularized Correlation Filters for Visual Tracking, IEEE International Conference on Computer Vision (ICCV). Santiago, Chile, 2015, pp. 4310--4318.

[93] C. Ma, J.-B. Huang, X. Yang et al., "Adaptive Correlation Filters with Long-Term and Short Term Memory for Object Tracking," International Journal of Computer Vision, 2017, vol. 126, pp. 771796.

[94] V. N. Boddeti, T. Kanade and B. V. K. V. Kumar, Correlation filters for object alignment, IEEE Conference on Computer Vision and Pattern Recognition. Portland, OR, USA, 2013, pp. 2291—2298.

[95] H. K. Galoogahi, T. Sim and S. Lucey, Correlation filters with limited boundaries, In CVPR2015. pp. 4630--4638 (2015).

[96] M. D. Vision, "A Computational Investigation into the Human Representation," Process. Vis. Information, Free. WH Company. San Fr., 1982.

[97] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 7, pp. 1442–1468, 2013.

[98] D. M. Chu and A. W. M. Smeulders, "Color invariant surf in discriminative object tracking," in European Conference on Computer Vision, 2010, pp. 62–75

[99] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," Comput. Vis. image Underst., vol. 110, no. 3, pp. 346–359, 2008.

[100] A. Abdel-Hadi, Real-time object tracking using color-based Kalman particle filter, The 2010 International Conference on Computer Engineering & Systems. Cairo, Egypt 2010, pp. 337--341.

[101] J. Deng, W. Dong, R. Socher et al., ImageNet: A large-scale hierarchical image database, 2009 IEEE Conference on Computer Vision and Pattern Recognition. Miami, FL, USA,

2009, pp. 248-255.

[102] C. Ma, J. -B. Huang, X. Yang et al., "Robust Visual Tracking via Hierarchical Convolutional Features," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, vol. 41, no. 11, pp. 2709--2723.

[103] S. Hare et al., "Struck: Structured Output Tracking with Kernels," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, vol. 38, no. 10, pp. 2096--2109.