TU WIEN Informatics

Auswirkungen des zeitlichen Kontexts auf die Robustheit in der UAV-gestützten Bildverarbeitung

Optionaler Untertitel der Arbeit

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Medieninformatik und Visual Computing

eingereicht von

Moritz Anton Zideck
Matrikelnummer 12217036

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Senior Lecturer Dipl.-Ing. Dr.techn. Sebastian Zambanini
Mitwirkung: Dipl. Inf Marvin Burges

Wien, 1. Jänner 2001

Moritz Anton Zideck	Sebastian Zambanini

TU WIEN Informatics

Impact of Temporal Context on Robustness in UAV-based Imagery

Optional Subtitle of the Thesis

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Media Informatics and Visual Computing

by

Moritz Anton Zideck
Registration Number 12217036

to the Faculty of Informatics

at the TU Wien

Advisor: Senior Lecturer Dipl.-Ing. Dr.techn. Sebastian Zambanini
Assistance: Dipl. Inf Marvin Burges

Vienna, January 1, 2001

Moritz Anton Zideck	Sebastian Zambanini

Erklärung zur Verfassung der Arbeit

Moritz Anton Zideck

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, habe ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT-Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 1. Jänner 2001

Moritz Anton Zideck

Danksagung

Ihr Text hier.

Acknowledgements

Enter your text here.

Kurzfassung

Ihr Text hier.

Abstract

Detector performance degradation under image corruptions is an important problem in computer vision and has been extensively studied for single-frame object detection. In contrast, the robustness of video object detection (VOD) under systematic perturbations remains less explored, particularly in UAV-based scenarios where noise, blur, and compression artifacts frequently occur. This thesis investigates the impact of controlled perturbations on video object detection and analyzes how temporal context influences robustness.

Two state-of-the-art video detectors, TransVOD++ and YOLOV++ are evaluated on the VisDrone2019-VID dataset. A dedicated corruption benchmark is constructed comprising noise, blur, photometric distortions, compression artifacts, and pixelation at multiple severity levels. In addition to fully corrupted sequences, a partial-perturbation setting is introduced in which only 10–20% of frames are degraded, simulating intermittent disturbances in UAV footage. Robustness is quantified using corruption-averaged precision and recall as well as relative mAP and mAR degradation.

On clean data, temporal models achieve consistent improvements of 4–5% relative mAP over single-frame baselines. When all frames are corrupted, temporal aggregation maintains higher absolute accuracy but does not systematically improve relative robustness as the number of reference frames increases. In contrast, under partial perturbation, temporal context substantially reduces performance degradation, lowering relative mAP drop by up to 40%. Across all experiments, TransVOD++ demonstrates stronger robustness than YOLOV++, particularly under blur and pixelation.

These findings show that temporal modeling enhances robustness primarily when clean contextual information remains available and emphasize the importance of sampling strategies and architectural design for robust UAV video object detection.

Contents

CHAPTER 1

Introduction

Video object detection is a relatively new field in computer vision that aims to leverage temporal context of videos in order to improve detection performance compared to image-based object detection. Temporal context refers to information that can be extracted from the time domain of a video, such as object motion, object trajectories, and changes in object appearance over time. By exploiting this additional information, video object detection models can achieve higher accuracy as well as provide more stable detections for tasks like Tracking. This is especially important in scenarios where objects may be occluded, blurred, or undergo significant appearance changes over time. A prominent example is UAV-based imagery, where the camera is constantly moving, objects are often small, and visual conditions can change rapidly, making reliable object detection particularly challenging.

To date, most evaluations of object detection models—both image-based and video-based—focus primarily on accuracy measured on clean and unperturbed data. However, in real-world deployments visual data is frequently affected by a wide range of perturbations, including sensor noise, compression artifacts, motion blur, illumination changes, and variations in viewpoint and scale. These perturbations can substantially degrade detection performance. Although several works have investigated the robustness of image-based object detectors to such corruptions, only limited attention has been given to the robustness of video object detection models, and virtually no systematic studies exist for UAV-based video data in particular.

In this thesis, the focus is therefore placed on the robustness of video object detection models under common perturbations in UAV-based imagery. Robustness is defined as the ability of a model to maintain its detection performance when exposed to adverse conditions such as lighting changes, weather effects, motion blur, occlusions, and variations in object appearance and scale. A model may achieve high accuracy on clean data, yet still be unreliable in practice if its performance deteriorates significantly under realistic

perturbations. For safety-critical and autonomous UAV applications, such robustness is essential.

Therefore, the main research question of this thesis is formulated as follows:

RQ1: To what extent does temporal context influence the robustness of video object detection models under varying environmental and perturbation conditions on the VisDrone dataset?

- RQ1.1: How does the amount of temporal context (i.e., number of reference frames) used by video object detection models affect their robustness to different types of perturbations compared to single-frame baselines?
- RQ1.2: How do different types of perturbations (e.g., motion blur, noise, brightness changes) impact the detection performance of video object detection models with varying amounts of temporal context?
- RQ1.3: Can video object detection models use temporal context to improve detection performance on single frames that are heavily perturbed, by leveraging information from adjacent unperturbed frames?

By answering these research questions, this thesis aims to provide a deeper insight into video detection models' capabilities and limitations, to highlight the importance of temporal context for robustness in UAV-based imagery, and to identify potential avenues for future research to enhance the reliability of video object detection systems in real-world applications.

1.1 Related Work

This chapter provides an overview of existing work related to the topic of video object detection and robustness in computer vision. First i will give a brief overview of the state of the art in object detection, that serves as a foundation for video object detection. After that i will highlight the most relevant works in video object detection, like FGFA[ZWD+17] and MEGA[CCHW20], as well as the two models used in this thesis, TransVOD++[ZLH+22] and YOLOV++[SZG24]. Finally i will discuss existing works on robustness evaluation in computer vision, including benchmarks like ImageNet-C[HD19] and studies on the robustness of object detection models in UAV-based imagery[LFH+24].

1.1.1 Object Detection

In the research field of computer vision, object detection is defined as the task of **identifying and localizing objects of interest within an image or video frame**. This problem was traditionally approached using hand-crafted features and classical machine learning techniques, such as Haar cascades[VJ01] and HOG + SVM[DT05]. However, with the advent of deep learning, particularly convolutional neural networks

(CNNs), object detection has seen significant advancements. Since then multiple differnt architectures have emerged, each with its own strengths and weaknesses. In regards to this thesis, two main categories of object detection are relevant: transformer-based detectors and one-stage detectors.

Transformers, like **DETR**[CMS+20] and **Deformable DETR**[ZSL+21], use self-attention meachnisms, which makes them able to capture long-range dependencies in the input data. This feature is particularly useful for object detection, as it allows the model to consider the entire image context when making predictions. Therefore transformer-based detectors have shown strong performance on various object detection benchmarks, often outperforming traditional CNN-based methods, however they tend to be computationally more expensive and require larger amounts of training data.

On the other hand one-stage detectors, like **YOLO**[RDGF16] and **SSD**[LAE+16], are designed for real-time applications, as they can process images quickly while still achieving competitive accuracy. One-stage means that they predict bounding boxes and class probabilities directly from the input image in a single forward pass, skipping the need for region proposal generation before classification. With this design they are able to achieve high inference speeds, making them suitable for applications like autonomous driving and surveillance, where real-time performance is crucial.

1.1.2 Video Object Detection

Compared to image-based object detection, video object detection resarch has received less attention, even though many real-world applications involve video data. In most cases the video is treated as a sequence of individual frames, and image-based object detection models are applied independently to each frame. The reason for this is the increased complexity of video data, which requires models to not only consider spatial information within each frame but also temporal information across frames. Combining spatial and temporal information effectively is a challenging task, as it requires models to handle issues like motion blur, occlusions, and changes in object appearance over time. However, if done successfully, leveraging temporal context can lead to significant improvements in detection accuracy and stability. Since 2017 several works have proposed methods have been proposed to address these challenges.

Most notable works in video object detection include **FGFA**[ZWD+17], which uses optical flow to aggregate features from multiple frames, and **MEGA**[CCHW20], which introduces a memory module to store and retrieve temporal information. Optical flow is a technique that estimates the motion of objects between consecutive frames, allowing the model to align features from different frames before aggregation. Two also well known models are **TransVOD++**[ZLH+23] based on **TransVOD**[ZLH+22] and **YOLOV++**[SZG24] based on **YOLOV**[SWG22], which represent state of the art approaches to video object detection, leveraging transformer architectures and temporal feature fusion techniques, respectively. A more in depth description of these two models is given in Chapter 2.

1.1.3 Robustness in Computer Vision

Robustness is as defined by Hendrycks and Dietterich[HD19] the ability of a model to maintain predictive performance under distribution shifts caused by common, naturally occurring image corruptions and perturbations. Hendrycks and Dietterich were the first to investigate the robustness of image classification models under common corruptions and perturbations, introducing the benchmark suite **ImageNet-C**[HD19]. They applied 15 different types of corruptions, such as Gaussian noise, motion blur, and brightness changes, to the ImageNet dataset and evaluated the performance degradation of various image classification models. A more recent work by Liu et al.[LFH+24] specifically focuses on the robustness of object detection models in UAV-based imagery. They proposed the benchmark suite **UAV-C**, which consists of common corruptions and perturbations that frequently occur in UAV-based imagery, such as weather effects, motion blur, and illumination changes.

1.2 Overview

The thesis is structured as follows:

In Chapter 2, the video object detection models used in this thesis, TransVOD and YOLOV, are described in detail, including their architectures and mechanisms for leveraging temporal context. After that the dataset used for evaluation on UAV-based imagery, Visdrone, is introduced in Chapter 3, along with the definition and implementation of common perturbations applied to the data for robustness evaluation. To support reproducibility as well as describe the experimental setup, Chapter 4 outlines the training and evaluation procedures, including hyperparameter settings and hardware specifications. Chapter 5 presents the results of the robustness evaluation, analyzing the impact of temporal context on detection performance under various perturbations. Finally, Chapter 6 concludes the thesis with a summary of findings, discusses limitations, and suggests directions for future research in robust video object detection for UAV applications.

CHAPTER 2

Method

2.1 Model Overview

For this thesis two video detector models are chosen, which represent different approaches to leverage temporal context in different ways. On the one hand there is TransVOD++[ZLH+23], which is a transformer based model that uses attention mechanisms to aggregate temporal information from multiple frames. On the other hand YOLOV++[SZG24], which is a one-stage detector that extends the popular YOLO architecture to video data by incorporating temporal feature fusion techniques. For both models the Swin base backbone[LLC+21] has been chosen, to ensure a fair compairson as well as petrain weights being available for both models.

Together these models provide a good basis for evaluation, as they represent different design philosophies and represent pro and cons in terms of temporal context utilization, computational efficiency and detection accuracy.

2.2 Transvod++

Transvod[ZLH+22] is a end to end video object detection model base on DETR[CMS+20]. End to end mean that no hand crafted features as well as no post processing is needed, everthing is learned by the model itself. The models fist version was proposed in 2021 as one of the first transformer based video object detection models to streamline the detection pipeline and remove the need for hand crafted features. By encoding not only spatial but also temporal information in their attention mechanism, the model shows strong performance on various video object detection benchmarks. In the most well known video object detection benchmark, ImageNet VID[RDS+15] outperforms its single frame baseline by 3.6 mAP (%) achieving 80.7 mAP (%) on the validation set. One year later an improved version of TransVOD was proposed, called TransVOD++[ZLH+23], which

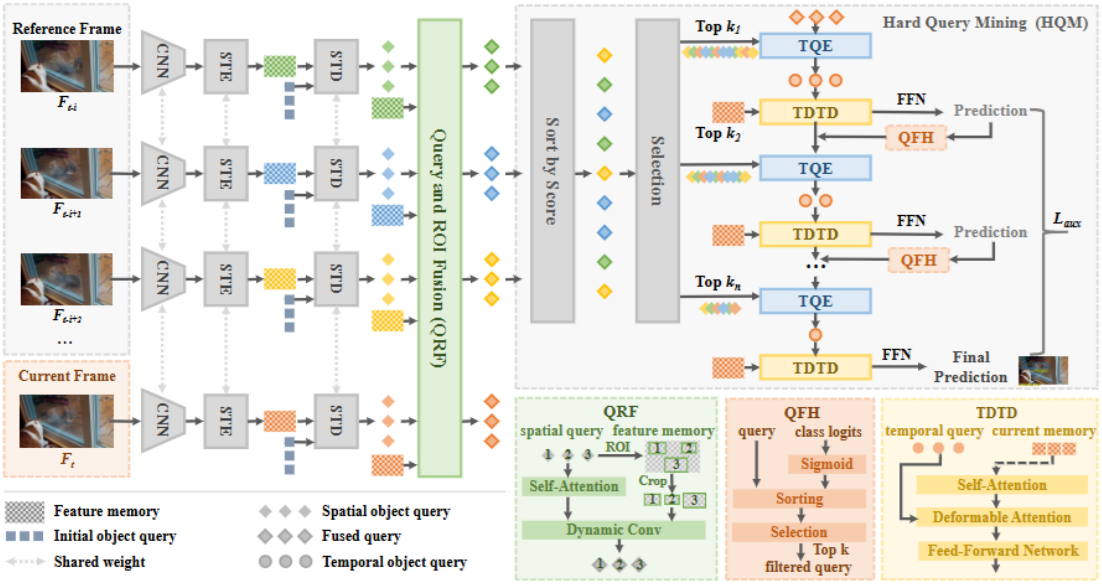


Figure 2.1: TransVOD++ architecture overview (from[ZLH+23])

builds upon the original TransVOD architecture and introduces several enhancements to further improve detection performance. The main goal of TransVOD++ is to address the heavy computation costs as well as increase the detection accuracy of its predecessor. Next to architectural improvements, which i will describe in the following, a new backbone namely Swin-Base[LLC+21] instead of ResNet-101[HZRS16] was used to further boost performance. With these improvements TransVOD++ was the first model to achieve over 90 mAP (%) on the ImageNet VID validation set, reaching 90.0 mAP (%).

Model design

TransVOD++ builds upon the deforamble DETR[ZSL+21] architecture, which itself improves the DETR[CMS+20] model by introducing deformable attention modules to better handle multi-scale features and improve convergence speed. For the backbone of the model the Swin-Base[LLC+21] architecture is used, which is a hierarchical vision transformer that utilizes shifted windows for efficient self-attention computation. To leverage temporal context of multible frames, TransVOD++ introduces two key components: Query and ROI fusion (QRF) and Hard Query Mining (HQM).

Query and ROI fusion (QRF) The goal of this module, as described in[ZLH+23], is to reduce computational cost while still effectively leveraging temporal context. QRF enables temporal encoding over object-level, RoI-refined feature embeddings instead of dense spatial feature maps. This is done by extracting RoI-aligned appearance features from predicted bounding boxes and fusing them into the corresponding transformer queries, allowing temporal aggregation to operate directly on object-centric representations.

Hard Query Mining (HQM) As the QRF module focuses on object-level features, HQM further reduces computational cost by retaining only the most informative object queries for temporal aggregation. This is achieved by evaluating object queries from the current frame and all reference frames using a lightweight classification head and selecting only those with high confidence scores. With this mechanism, redundant and low-confidence queries are discarded, allowing temporal fusion to operate on a compact and informative set of object queries.

As it can be seen in Figure 2.1, each image is first processed by the deformable DETR backbone to extract spatial features. After that the QRF module extracts RoI-aligned object features from predicted bounding boxes and fuses them into the corresponding object queries. These features are then selected by the HQM module based on their confidence scores. Finally, the selected object queries from the current frame and reference frames are aggregated using a temporal transformer, described in2.1 as Temporal Query Encoder (TQE) and Temporal Deformable Transformer Encoder to produce the final detection results.

2.3 YOLOV++

Based on the popular YOLO[RDGF16] architecture, to be more precise YOLOX[GLW+21], which are one-stage detectors known for their speed and efficiency, YOLOV[SWG22] extends this architecture to video data by incorporating temporal feature fusion techniques. The paper that was published in 2022, was able to surpass previous state of the art video object detection models on the ImageNet VID[RDS+15] benchmark with a mAP (%) of 85.5 on the validation set, while being still near real time capable with 22.7 FPS on a Nvidia TITAN RTX GPU. YOLOV achieves this by introducing a temporal feature fusion module that aggregates features from multiple frames, allowing the model to leverage temporal context effectively. Futhermore like TransVOD a improved version of YOLOV was proposed called YOLOV++[SZG24], which further enhances the temporal feature fusion mechanism and introduces additional optimizations to improve detection accuracy and efficiency. The now improved YOLOV++ was able to achieve a new record mAP (%) of 93.2 on the ImageNet VID validation set, while still being over 30 FPS on a Nvidia RTX 3090. However for this a special backbone names FocalNet-Large[YLDG22] is used. Using the same Swin-Base[LLC+21] backbone as for TransVOD++, YOLOV++ achieves a mAP (%) of 90.7 on the ImageNet VID validation set, which is still significantly higher than TransVOD++ with 90.0 mAP(%).

Model design

The YOLOX architecture, which serves as the basis for YOLOV, improves upon previous YOLO versions by introducing several enhancements, such as an anchor-free design, a decoupled head which splits the classification and regression tasks, as well as OTA (Dynamic label assignment) to improve training efficiency and detection accuracy. Based on this architecture, YOLOV++ adds the temporal context after the YOLOX processes

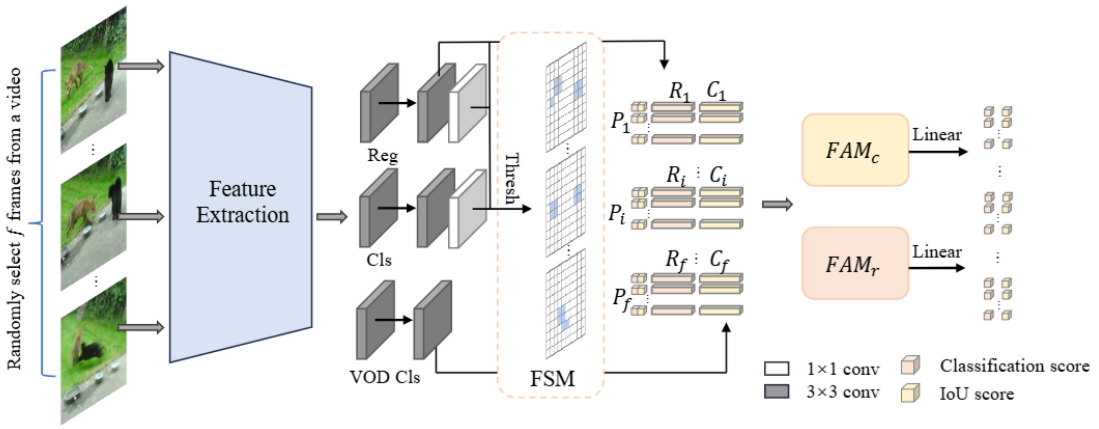


Figure 2.2: YOLOV++ architecture overview (from[SZG24])

each frame individually on the proposal-feature level. To achieve this, YOLOV++ introduces the Feature Selection Module (FSM), which as the name suggests selects the most relevant features from the all frames, as well as the Feature Aggregation Module (FAM), which aggregates features from multiple frames to enhance the feature representation for object detection.

Feature Selection Module (FSM) Since one stage detectors like YOLOX generate a large number of proposals per frame, processing all proposals from multiple frames would be computationally infisable. Therefore the FSM is introduced to select only the most relevant proposals based on there confidence for temporal aggregation, as well as adding NMS (Non Maximum Suppression) to remove redundant proposals. This way the number of proposals goes from thousands per frame to only a few hundred, making temporal aggregation feasible.

Feature Aggregation Module (FAM) The FAM is responsible for aggregating the selected features from multiple frames. This is done by to parallel modules, one for classification features and one for regression features. These modules use multi head attention to effectively combine information from different frames, however a homogenity issue is described in the paper, where standard attention tends to focus on similar features across frames, which could all be blurred, occluded or low quality. To address this, they propose their key innovation, called Affinity Manner, where the similarity between proposals is weighted by their predicted confidence, so that temporally aggregated features are drawn preferentially from high-quality, reliable frames instead of uniformly similar but degraded ones.

All together the YOLOV++ architecture can be seen in Figure 2.2, where each frame is first processed individually by the YOLOX backbone and neck to extract spatial features and generate proposals. After that the FSM selects the most relevant proposals from all frames, which are then aggregated by the FAM using the Affinity Manner to produce

enhanced features for final detection. This way YOLOV++ effectively leverages temporal context while maintaining the efficiency and speed characteristic of one-stage detectors.

CHAPTER 3

Dataset and Metrics

3.1 Visdrone Dataset

The Visdrone dataset[ZWD+21] is a large-scale dataset for different detection scenarios based on drone based imagery. The images and videos were captured by various drone platforms in different urban and suburban areas of 14 different cities across China. The objects are entitys of public street scenes, e.g., pedestrians, vehicles, bicycles, etc. All together there are 10 different object categories. It is curated by the *AISKYEYE* research group from the *Tianjin University in China*. For this thesis VISDRONE2019-VID is used, to leverage the temporal context of the videos for the object detection task. All together the dataset contains 79 sequences with 33,366 frames, which are split 56 videos with 24,198 frames for training, 7 videos with 2,846 frames for validation and 16 videos with 6,322 frames for testing. The dataset is chosen because of its large size and the challenging scenarios, e.g., different weather conditions, various altitudes and camera angles as well as high density of objects in the images. Object sizes vary significantly, ranging from very small objects with only a few pixels to large objects covering a significant portion of the image. This large difference in object sizes makes it also suitable to evaluate the performance of detection models under different perturbations and across different scales.

Since the dataset includes ignored regions, which are areas in the images where objects are not annotated due to beeing too crowded or too small, these regions are taken out of the evaluation to avoid penalizing the models for false positives in these areas. This was done by blacking out these regions in the images before feeding them into the models for training and evaluation. The method was chosen due to its simplicity and effectiveness. Other methods such as removing the annotations for these regions or masking them out during evaluation were considered, but blacking them out directly in the images was found to be the most straightforward approach. Without this step models tend to produce a high number of false positives in these ignored regions, which would skew the evaluation results.

(a) Original Image (b) Image with Ignored Regions Blacked Out

Figure 3.1: Example of an image from the Visdrone dataset with ignored regions blacked out for training and evaluation.

3.2 Perturbation

A significant part of this thesis to evaluate the robustness of video object detection models under common perturbations in UAV-based imagery. In this sentence two key concepts must be defined: what robustness means in the context of an object detection model, and which perturbations commonly occur in UAV-based imagery.

3.2.1 Definition of Robustness

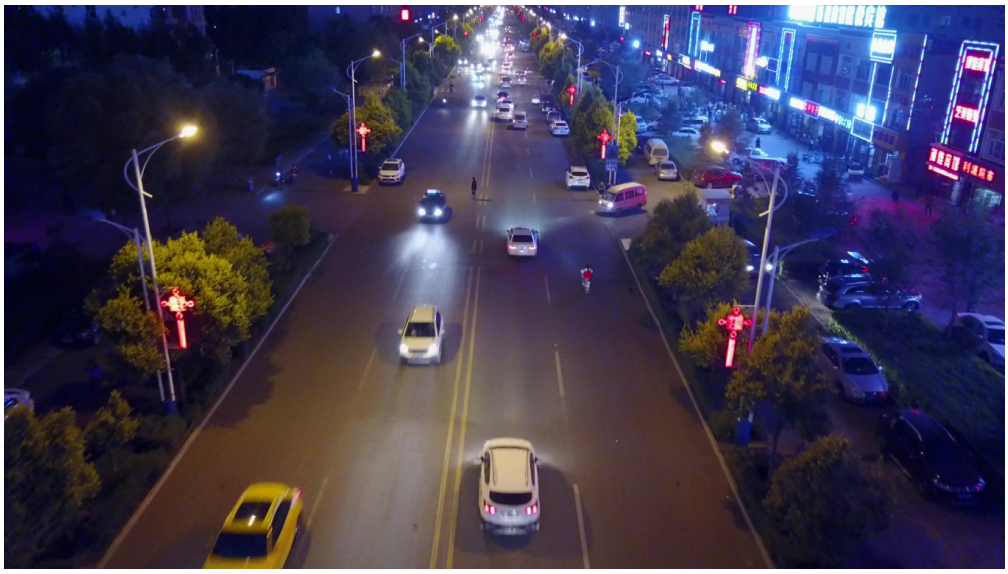
A widely used definition of robustness was proposed by Hendrycks and Dietterich [HD19], who define robustness as a model's ability to maintain predictive performance under distribution shifts caused by common, naturally occurring image corruptions and perturbations. In the context of object detection, this means that a robust model should be able to accurately detect and localize objects even when the input images are affected by various types of noise, distortions, or other adverse conditions. In the context of this thesis, the main metric to quantify robustness is the relative performance degradation of a model mean avarage precision (mAP) under different perturbations compared to its performance on clean data.

3.2.2 Common Perturbations in UAV-based Imagery

The paper by Hendrycks and Dietterich [HD19] introduced a benchmark suite called ImageNet-C, which consists of 15 different types of common image corruptions applied to the ImageNet dataset. For further insight a paper named *Benchmarking the Robustness of UAV Tracking Against Common Corruptions*[LFH+24] which was published in 2024, is used to identify perturbations that commonly occur in UAV-based imagery. Based on these works, the following perturbations are considered in this thesis:

- Gaussian Noise: Random noise following a Gaussian distribution is added to the image pixels, simulating sensor noise.

(a) Example 1



(b) Example 2

(c) Example 3

Figure 3.2: Example images from the Visdrone dataset.

- Motion Blur: Simulates the effect of camera or object motion during exposure, resulting in blurred images.
- Defocus Blur: Simulates the effect of an out-of-focus lens, resulting in blurred images.
- Brightness Changes: Adjusts the overall brightness of the image, simulating different lighting conditions.
- Contrast Changes: Adjusts the contrast of the image, affecting the distinction between light and dark areas.
- Jpeg Compression: Simulates artifacts introduced by JPEG compression at various quality levels.

The simulation of weather conditions such as fog, rain, and snow is not considered in this thesis, as these perturbations require more complex rendering techniques. For each perturbation type, multiple severity levels are defined to assess robustness across a range of adverse conditions; in this thesis, three levels are used: low, medium, and high.

3.2.3 Implementation

To apply the defined perturbations to the Visdrone dataset, a custom data augmentation pipeline is implemented into to the models data loader. Based on evaluation input parameters, the data loader applies the specified perturbation with the desired severity level to each frame before it is fed into the model for inference. For more in depth evaluation a probability parameter is added, which defines the likeliness each perturbation being applied to a frame.

Gaussian noise. We add i.i.d. Gaussian noise:

$$\tilde{I} = \mathrm{clip}(I + N,\ 0,\ 255), \qquad N_{h,w,c} \sim \mathcal{N}\Big(0,\ (\sigma \cdot 255)^2\Big), \tag{3.1}$$

where σ is the noise standard deviation.

Defocus blur. We approximate defocus blur by convolving the image with a normalized disk (pillbox) kernel:

$$\tilde{I} = I * K_{\mathrm{disk}}, \qquad K_{\mathrm{disk}}(u,v) = \frac{1}{Z}\mathbb{1}\Big(u^2 + v^2 \le r^2\Big), \tag{3.2}$$

where K_{disk} is a $k \times k$ kernel, $r = \lfloor k/2 \rfloor$, $Z = \sum_{u,v} \mathbb{1}(\cdot)$, and $*$ denotes 2D convolution.

Motion blur. We simulate linear motion blur by convolving with a sparse line kernel of size $k \times k$ oriented by an angle θ (in degrees, default $\theta = 0$). The kernel is constructed by placing ones on the discrete line

$$v = \tan(\theta)\, u, \qquad u \in [-\lfloor k/2 \rfloor,\ \lfloor k/2 \rfloor], \tag{3.3}$$

rasterized onto the kernel grid and normalized to sum to one, then $\tilde{I} = I * K_{\mathrm{motion}}$.

Brightness change. We apply a global multiplicative gain:

$$\tilde{I} = \mathrm{clip}\big(\alpha I,\ 0,\ 255\big), \tag{3.4}$$

with $\alpha > 0$.

Contrast change. We scale deviations from the per-channel mean:

$$\mu_c = \frac{1}{HW}\sum_{h,w} I_{h,w,c}, \qquad \tilde{I}_{h,w,c} = \mathrm{clip}\big((I_{h,w,c} - \mu_c)\alpha + \mu_c,\ 0,\ 255\big), \tag{3.5}$$

with contrast factor α.

Pixelation. We downsample and upsample the image using a block factor p (default $p = 8$). Specifically, we resize I to $(\lfloor W/p \rfloor, \lfloor H/p \rfloor)$ using bilinear interpolation, then resize back to (W, H) using nearest-neighbor interpolation:

$$\tilde{I} = \mathrm{NN}\big(\mathrm{BL}(I; \lfloor W/p \rfloor, \lfloor H/p \rfloor);\ W, H\big), \tag{3.6}$$

where BL denotes bilinear resize and NN denotes nearest-neighbor resize.

JPEG compression. We simulate compression artifacts by encoding and decoding the image using JPEG with quality parameter q:

$$\tilde{I} = \mathrm{JPEGdecode}\big(\mathrm{JPEGencode}(I; q)\big). \tag{3.7}$$

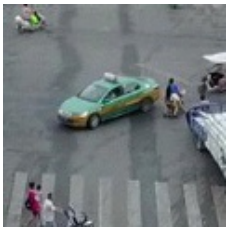
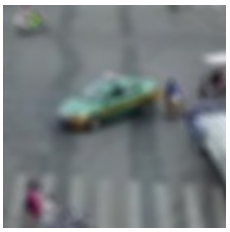
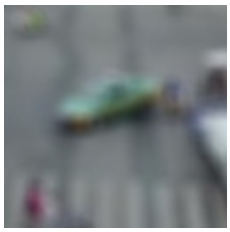
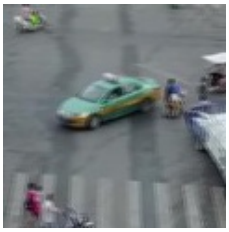
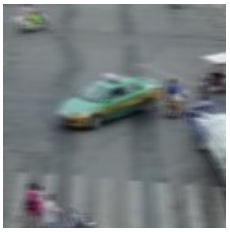
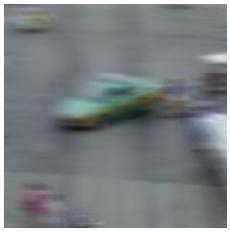
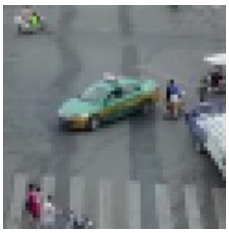
The specific severity levels are defined as follows:

Table 3.1: Perturbation presets and severity levels used in robustness evaluation.

Perturbation	**Low**	**Medium**	**High**
Gaussian noise	$\sigma = 0.01$	$\sigma = 0.05$	$\sigma = 0.10$
Defocus blur	$k = 3$	$k = 7$	$k = 11$
Motion blur	$k = 3,\ \theta = 0^\circ$	$k = 7,\ \theta = 0^\circ$	$k = 15,\ \theta = 0^\circ$
Brightness change	$\alpha = 1.10$	$\alpha = 1.25$	$\alpha = 1.45$
Contrast change	$\alpha = 1.10$	$\alpha = 1.25$	$\alpha = 1.45$
Pixelation	$p = 2$	$p = 4$	$p = 6$
JPEG compression	$q = 85$	$q = 55$	$q = 25$

With this setup, all together 18 different perturbation configurations (6 perturbation types $\times$ 3 severity levels) can be evaluated to assess the robustness of video object detection models in UAV-based imagery.

Figure 3.3: Example images illustrating perturbation severities (Low, Medium, High).

Perturbation	**Low**	**Medium**	**High**
Gaussian noise			
Defocus blur			
Motion blur			
Brightness change			
Contrast change			
Pixelation			

3.3 Evaluation Metrics

For robustness evaluation next to standard object detection metrics such as mean average precision (mAP) and mean average recall (mAR), a proper metric is needed, that quantifies the performance degradation of a model under different perturbations relative to its performance on clean data. Derived from the papers *Benchmarking the Robustness of UAV Tracking Against Common Corruptions*[LFH+24] mCE (mean corruption error) which was based on the top-1 error rate, a new metric named *mean Corruption Average Precision* (mCAP) and *relative mean Corruption Average Precision* (rmCAP) is proposed. Parallel to that *mean Corruption Average Recall* (mCAR) and *relative mean Corruption Average Recall* (rmCAR) is defined. Furthermore the *relative mAP drop under corruption* (rDrop_{AP}) as well as *relative mAR drop under corruptions* (rDrop_{AR}) is also reported to explicitly quantify performance degradation.

Mean Corruption Average Precision (mCAP) is defined as:

$$\text{mCAP} = \frac{1}{|C|} \sum_{c \in C} \frac{1}{S} \sum_{s=1}^{S} \text{mAP}_{s,c}{}^{1} \tag{3.8}$$

where C is the set of all perturbation configurations (i.e., perturbation types and severity levels), S is the number of severity levels, and $\text{mAP}_{s,c}$ is the mean average precision of the model under perturbation configuration c at severity level s.

Relative mean Corruption Average Precision (rmCAP) is defined as:

$$\text{rmCAP} = \frac{1}{|C|} \sum_{c \in C} \frac{1}{S} \sum_{s=1}^{S} \frac{\text{mAP}_{s,c}}{\text{mAP}_{\text{clean}}}{}^{1} \tag{3.9}$$

where $\text{mAP}_{\text{clean}}$ is the mean average precision of the model on clean, unperturbed data.

Relative mAP drop under corruption (rDrop_{AP}). To explicitly quantify performance degradation, we also report the relative mAP drop averaged over all corruption configurations:

$$\text{rDrop}_{\text{AP}} = \frac{1}{|C|} \sum_{c \in C} \frac{1}{S} \sum_{s=1}^{S} 1 - \frac{\text{mAP}_{s,c}}{\text{mAP}_{\text{clean}}}{}^{1} \tag{3.10}$$

By definition, $\text{rDrop}_{\text{AP}} = 1 - \text{rmCAP}$.

Mean Corruption Average Recall **(mCAR)**, relative mean Corruption Average Recall **(rmCAR)** and relative mAR drop under corruption **(rDrop_{AR})** are defined analogously, but using mean average recall (mAR) instead of mAP.

[1] Since there are no different severity levels and perturbation types in the partial corruption evaluation, the formulas are not summed over C and S.

CHAPTER 4

Experiments Setup

4.1 Hardware and Software Environment

The training as well as evaluation of the models was done on eather Nvidia RTX 3060 Ti or Nvidia RTX 3090TI GPUs. This was due due to the fact that CUDA 11.3 was required by the TransVOD implementation, which is not supported by the newer GPUs such as the RTX 40 series. Both models were implemented in PyTorch[PGM+19] and trained using the AdamW[LH19] optimizer.

4.2 Reference Frame Sampling Strategy

Since both models leverage temporal context from multiple frames, a proper reference frame sampling strategy is needed to select the frames that will be used as input to the models. As written in the original papers as well as implemented in the provided code repositories, TransVOD++ and YOLOV++ use inherently different sampling strategies.

4.2.1 Transvod Sampling

Following the sampling strategy of TransVOD++, each video is divided into 16 temporal intervals. For a given target frame, reference frames are selected by sampling one frame from each interval. When the number of reference frames $N \geq 8$, sampling is performed only on one side of the target frame; when $N \geq 8$ reference frames are sampled from both the past and future relative to the target frame. As a result, the selected reference frames are approximately evenly spaced in time around the target frame.

However, as will be discussed later, this global sampling strategy does not achieve reasonable performance on the VisDrone dataset. To address this issue, we adopt a modified, more local sampling strategy in which reference frames are selected using a fixed temporal offset relative to the target frame. Specifically, we test reference frames

sampling with a temporal offset of 1 frame and 8 frames. Overall, one single-frame baseline and twelve video-based TransVOD++ configurations were evaluated, combining three reference-frame sampling strategies with four different numbers of reference frames.

4.2.2 YOLOV Sampling

In the YOLOV++ paper, it is mentioned that test have resulted in better performance when using random sampling of reference frames in the whole video clip, called global sampling. Therefore, for YOLOV++ the original global sampling strategy is used, where reference frames are sampled from the entire video clip. To test the impact of local sampling on YOLOV++, an additional configuration with local sampling is evaluated, where reference frames are selected using a fixed temporal offset relative to the target frame with stride 10. Therefore, a total of seven YOLOV++ configurations were evaluated, consisting of one single-frame baseline and six video-based variants formed by combining two reference-frame sampling strategies with three different reference-frame counts.

4.3 Input Resolution

Since the VisDrone dataset contains a large proportion of small objects, which are particularly challenging for object detectors, the input resolution must be chosen carefully. Higher resolutions preserve fine details and improve small-object detectability, but they also increase memory consumption and thus limit the feasible batch size on the available GPUs. Consequently, an input resolution of 960x544 pixels was chosen as a compromise between detection performance and memory efficiency, as well as keeping the original aspect ratio of the images.

4.4 Training Protocol

Both models are trained under an identical training protocol to ensure a fair comparison. Differences between the two models are limited to architectural components and are detailed separately. Since both models already provided pretrained weights on the ImageNet VID[RDS+15] dataset with a Swin-Base[LLC+21] backbone, these weights were used to finetune the single frame version of the models on the Visdrone2019-VID[ZWD+21] training set.

After finetuning the single frame models, the video versions were trained by loading the finetuned single frame weights and training the temporal context modules while keeping the backbone and single frame detection head frozen. For TransVOD++ the number of reference frames was used namely 1, 3, 7 and 15 reference frames. However due to memory constraints only up to 7 reference frames could be used for YOLOV++ since the number of reference frame directly corresponds to the batch size. This means that for 15 reference frames a batch size of 16 would be required, which surpasses the available GPU memory of 24GB of the NVIDIA RTX 3090ti.

Table 4.1: Fine-tuning hyperparameters for the single-frame detectors.

Hyperparameter	**YOLOV++**	**TransVOD++**
Batch size	8	8
Precision	FP16	FP16
Learning rate	$0.01/64 \approx 1.56 \cdot 10^{-4}$	10^{-4} and 10^{-5} for backbone, with lr drops after 3 and 5 epochs
Weight decay	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$
Data augmentation	• Random horizontal flip • Multi-scale resize: original size ±64 px • Color jitter • Geometric transforms: rotation ±5°, translation 0.1, shear 2°	• Random horizontal flip • Multi-scale resize: original size ±64 px • Color jitter

For training the video models, different hyperparameters were used compared to the single frame models, which are summarized in the following table.

Table 4.2: Fine-tuning hyperparameters for the video detectors.

Hyperparameter	**YOLOV++**	**TransVOD++**
Batch size	number of reference frames	1
Precision	FP16	FP16
Learning rate	10^{-5}	$2 \cdot 10^{-6}$
Weight decay	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$
Data augmentation	• Random horizontal flip	• Random horizontal flip

All together the single frame as well as 3 different video models for YOLOV++ and 4 different video models for TransVOD++ were trained and evaluated.

For all the evaluations random seeds were fixed to 42 to ensure reproducibility of the results, except for random sample selection of reference frames during training, where different random seeds were used to increase the diversity of training samples. To see if this had an impact on the results, 5 different training runs were done for the YOLOV++ model with 7 reference frames, which showed almost no variation in mAP(%) of less than 0.3% between the runs.

For evaluation the models were first evaluated on the clean Visdrone2019-VID[ZWD+21] validation set to obtain the baseline mAP(%) and mAR(%) values. Subsequently the models were evaluated on the perturbed versions of the validation set for each perturbation type and severity level as described in the *Perturbation* chapter.

To evaluate whether temporal context enables models to maintain detection consistency when individual frames are perturbed, we conducted additional experiments using a probabilistic perturbation strategy. Specifically, we randomly selected 10% and 20% of

frames within each video sequence and applied a diverse set of perturbations at high severity levels. This approach allows us to assess whether models can leverage information from adjacent unperturbed frames to mitigate the impact of corruptions on isolated frames.

CHAPTER 5

Empirical Results

For the results alltogether 20 differnt models were evaluated, consisting of one single frame baseline and six video-based variants of YOLOV++ as well as one single frame baseline and twelve video-based variants of TransVOD++. All models were first evaluated on the clean Visdrone2019-VID[ZWD+21] validation set to obtain baseline performance metrics, followed by evaluations on perturbed versions of the validation set for each defined perturbation type and severity level. These results are then combined to compute the proposed robustness metrics mCAP, rmCAP, mCAR. As mentioned before, to further investigate the benefits of temporal context in handling sporadic perturbations, additional evaluations were performed using a probabilistic perturbation strategy where 10% and 20% of frames in each video sequence were randomly selected for high-severity perturbations.

5.1 Evaluation on Clean Data

To form a basis for the robustness evaluation, the models were first evaluated on the clean Visdrone2019-VID[ZWD+21] validation set. The results are summarized in Table 5.1.

As can it can be seen in Table 5.1, the single frame baseline of TransVOD++ outperforms the single frame baseline of YOLOV++ by a significant margin of 2.2% mAP and 6.5% mAR. When looking at the video-based configurations, both models show improvements over their respective single frame baselines.

For YOLOV++, the best performance is achieved using random sampling of reference frames, with 7 reference frames. This configuration yields the highest mAP of 18.0% and mAR of 30.0%, therefore an increase of 0.8% mAP and 1.7% mAR or 4.7% and 6.0% relative improvement compared to the single frame baseline. Alligned with the findings in the YOLOV++ paper, random sampling of reference frames outperforms local sampling in all configurations, however the performance gap is relatively small with only 0.2% mAP and 0.0-0.5% mAR difference between the two sampling strategies.

Table 5.1: Detection performance of all evaluated video object detection models on the VisDrone2019-VID validation set. (Best results in **bold** and worst results in underline per model.)

Model Configuration	$\mathbf{mAP}_{50:95}$ (%)	$\mathbf{mAR}_{50:95}$ (%)
YOLOV++		
Single-frame baseline	17.2	28.3
Sampling random, 1 reference frame	17.5	29.2
Sampling random, 3 reference frames	17.8	29.7
Sampling random, 7 reference frames	**18.0**	**30.0**
Sampling local, 1 reference frame	17.3	29.0
Sampling local, 3 reference frames	17.7	29.5
Sampling local, 7 reference frames	17.8	**30.0**
TransVOD++		
Single-frame baseline	19.4	34.8
Sampling stride 1, 1 reference frame	19.6	35.8
Sampling stride 1, 3 reference frames	20.0	36.4
Sampling stride 1, 7 reference frames	**20.4**	**36.3**
Sampling stride 1, 15 reference frames	19.8	35.2
Sampling stride 8, 1 reference frame	19.3	35.2
Sampling stride 8, 3 reference frames	19.7	35.6
Sampling stride 8, 7 reference frames	19.4	34.6
Sampling stride 8, 15 reference frames	18.2	32.8
Sampling global, 1 reference frame	19.5	35.1
Sampling global, 3 reference frames	19.3	34.8
Sampling global, 7 reference frames	18.9	33.3
Sampling global, 15 reference frames	17.2	31.0

For TransVOD++, the best performance is achieved using a local sampling strategy with a temporal stride of 1 frame and 7 reference frames. This configuration yields the highest mAP of 20.4% and mAR of 36.3%, therefore an increase of 1.0% mAP and 1.5% mAR or 5.2% and 4.3% relative improvement compared to the single frame baseline. However, it is important to note that in contrast to the orginal TransVOD++ paper, the global sampling strategy results in the worst performance on the Visdrone dataset. It even performs worse than the single frame baseline when using 15 reference frames, which indicates that this sampling strategy is not suitable for the Visdrone dataset.

To get a better insight into the impact of the sampling strategy, local sampling with a temporal stride of 8 frames was also evaluated. This configuration shows a similar trend as global sampling, where the performance degrades with an increasing number of reference frames. This further supports the hypothesis that sampling frames that are temporally distant from the target frame is not beneficial for the Visdrone dataset, likely due to the fast motion and rapid scene changes in UAV-based videos.

Since the global sampling strategy achieves the worst performance with a significant margin, it will not be considered for the robustness evaluation in the following chapter.

Therefore we we have a total of 9 instead of 13 different TransVOD++ configurations for robustness evaluation.

5.2 Evaluation on Perturbed Data

For robustness evaluation, the models were evaluated on the perturbed versions of the Visdrone2019-VID validation set for each defined perturbation type and severity level independently. Every model was evaluated on all 18 perturbation configurations, and the results were aggregated to compute the proposed robustness metrics mCAP, rmCAP, mCAR, and rmCAR. The combined model results on perturbed data are summarized in Table 5.2. For more detailed results on each perturbation type and severity level, please refer to the Appendix.

Table 5.2: Detection performance of all evaluated video object detection models on the VisDrone2019-VID validation set under perturbations. (Best results in **bold** and worst results in underline per model.)

Model Configuration	$\mathbf{mCAP}_{50:95}$ (%)	$\mathbf{mCAR}_{50:95}$ (%)	rmCAP	rmCAR
YOLOV++				
Single-frame baseline	11.3	20.1	0.66	0.71
Sampling local, 1 reference frame	11.4	21.7	0.66	0.75
Sampling local, 3 reference frames	11.5	21.8	0.65	0.74
Sampling local, 7 reference frames	11.4	21.8	0.64	0.73
Sampling random, 1 reference frame	11.7	22.1	0.67	0.76
Sampling random, 3 reference frames	11.9	22.6	0.67	0.76
Sampling random, 7 reference frames	**12.0**	**23.0**	**0.67**	**0.77**
TransVOD++				
Sampling stride 1, 1 reference frame	15.7	28.6	0.84	0.84
Sampling stride 1, 3 reference frames	15.9	29.0	0.84	0.84
Sampling stride 1, 7 reference frames	**16.1**	**29.2**	**0.85**	0.85
Sampling stride 1, 15 reference frames	15.6	28.6	0.85	**0.86**
Sampling stride 8, 1 reference frame	15.5	28.2	0.85	0.84
Sampling stride 8, 3 reference frames	15.5	28.3	0.84	0.84
Sampling stride 8, 8 reference frames	15.3	27.9	0.83	0.85
Sampling stride 8, 15 reference frames	14.3	26.2	0.83	0.84

On perturbed data, TransVOD++ significantly outperforms YOLOV++ in terms of mCAP and mCAR across all configurations, as seen in Table 5.2. This performance gap can be noticed on every model configuration, with TransVOD++ achieving mCAP values ranging from 17.0% to 19.1%, while YOLOV++ achieves mCAP values between 11.3% and 12.0%. Looking at the relative metrics rmCAP and rmCAR, TransVOD++ also demonstrates superior robustness to perturbations, on the other hand a signicant releative performance drop can be observed for YOLOV++.

The perturbation results for YOLOV++ in Table 5.2 show that the video-based configurations outperform the single-frame baseline in terms of mCAP and mCAR. This phenomenon is consistent with the clean data results 5.1, where video-based models also showed improved performance. Looking at the the realtive metrics rmCAP and rmCAR, no clear trend can be observed between model configurations. This indicates that the number of reference frames and sampling strategy have little impact on the relative robustness of YOLOV++ to perturbations. In general the YOLOV++ models show a signicant relative performance drop when evaluated on perturbed data, with rmCAP values ranging from 0.64 to 0.67 and rmCAR values between 0.71 and 0.77.

For TransVOD++, a similar trend as in the clean data results can be observed in Table 5.2, where the models gain from the first few reference frames but performance degrades when using too many reference frames. Forthermore the local sampling stagegy with a temporal stride of 1 frame outperforms the stride of 8 frames strategy, with the best performance achieved using 7 reference frames. However an important observation is that the relative robustness of all TransVOD++ configurations is shown a very strong robustness to perturbations, with rmCAP values ranging from 0.83 to 0.86 and rmCAR values between 0.84 and 0.86. As with YOLOV++ no clear trend can be observed between model configurations, indicating that the number of reference frames and sampling strategy have little impact on the relative robustness of TransVOD++ to perturbations.

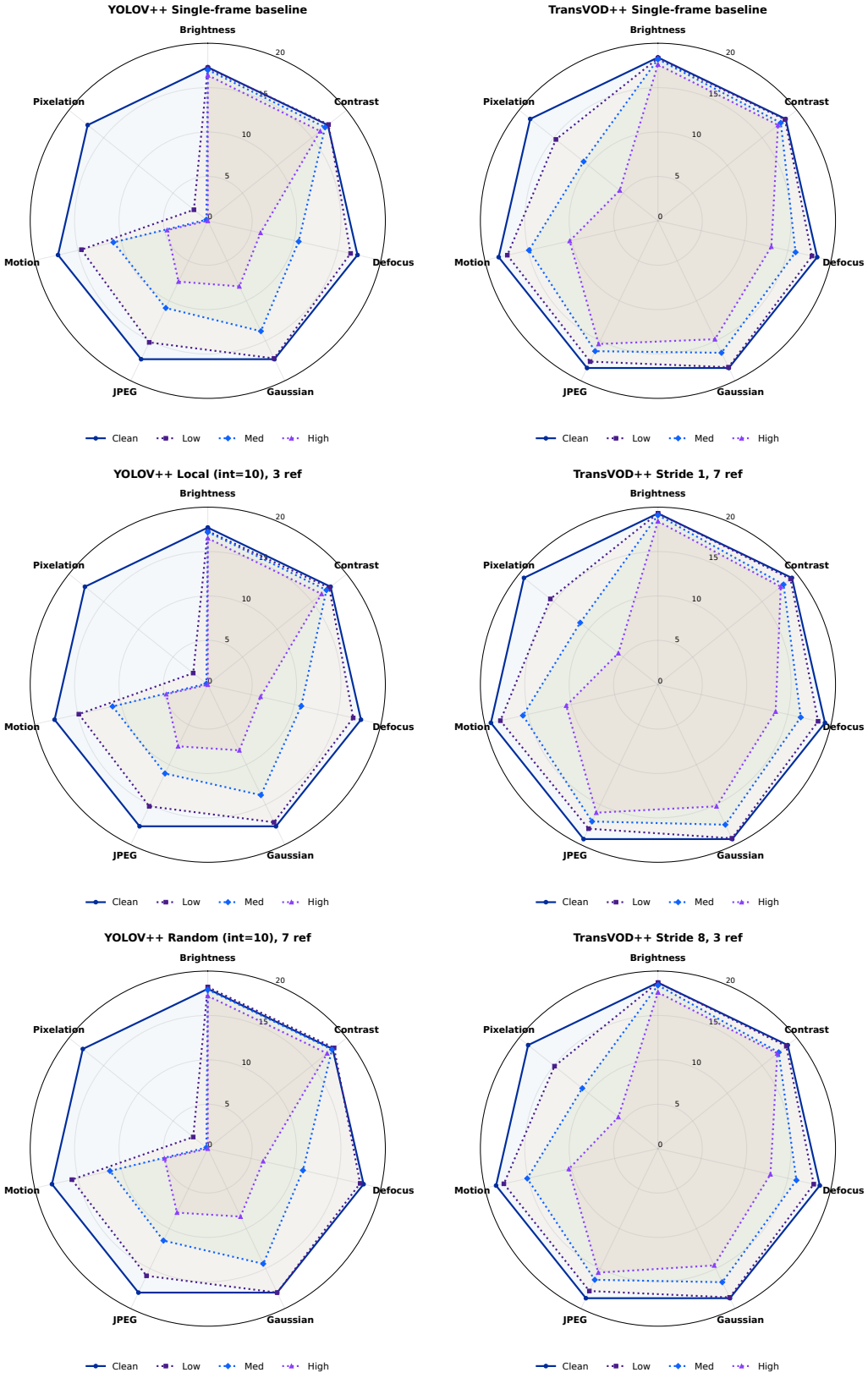
5.2.1 Perturbations and Reference Frames

Opposite of the initial hypothesis, increasing the number of reference frames does not necessarily lead to improved robustness against perturbations. This trend is can be observed in both YOLOV++ and TransVOD++ models, while video models generally outperform their single-frame baselines, relative to their clean data performance, the robustness does not improve with more reference frames. This indicates that simply adding more temporal context is not sufficient to enhance robustness against perturbations. Therefore if all frames are perturbated, the models are not able to leverage information from multibiple perturbated frames to improve detection performance.

5.3 Results by Perturbation Type

For the insight on a perturbation level, 3 model configurations were selected for each YOLOV++ and TransVOD++. These configurations represent the single-frame baseline, as well the best performing temporal window size of each sampling strategy. The key results are visualized in radar plots in Figure 5.1. Each plot illustrates the mAP performance across different perturbation types and severity levels, providing a clear comparison of how each model configuration handles various corruptions. As already observed in the aggregated results, TransVOD++ consistently outperforms YOLOV++. In general both models show a similar trend in handling different perturbation types. However, looking at **pixelation** and **motion blur**, TransVOD++ shows a significantly better robustness compared to YOLOV++, especially at higher severity levels. For

Figure 5.1: mAP performance across perturbation types and severity levels.



brightness and **contrast changes**, both models exhibit relatively stable performance where virtually no performance drop can be observed even at high severity levels. When evaluating **defocus blur**, **gaussian noise** and **JPEG Compression**, both models experience a steady performance decline as severity increases, but TransVOD++ maintains a higher mAP across all severity levels.

As seen in the overall results, increasing the number of reference frames does not consistently enhance robustness across all perturbation types. This strengthens the observation that simply adding more temporal context is not sufficient to improve robustness against perturbations, especially when all frames are affected.

5.4 Results under Partial Perturbation

Table 5.3: Detection performance under probabilistic perturbations (10% and 20% of frames perturbed).

Model Config.	Clean	mAP(10%)	mAP(20%)	rDrop(10%)	rDrop(20%)
YOLOV++					
Single-frame	17.2	16.4	15.5	4.7%	9.9%
Random, 1 ref	17.5	16.7	15.9	4.6%	9.1%
Random, 3 ref	17.8	17.1	16.4	3.9%	7.9%
Random, 7 ref	18.0	17.5	16.8	2.8%	6.7%
Local, 1 ref	17.3	16.6	15.7	4.0%	9.2%
Local, 3 ref	17.7	17.0	16.2	3.9%	8.5%
Local, 7 ref	17.8	17.3	16.7	2.8%	6.2%
TransVOD++					
Single-frame	19.4	18.8	18.2	3.1%	6.2%
Stride 1, 1 ref	19.6	19.1	18.5	2.6%	5.6%
Stride 1, 3 ref	20.0	19.5	19.0	2.5%	5.0%
Stride 1, 7 ref	20.4	20.0	19.5	2.0%	4.4%
Stride 1, 15 ref	19.8	19.4	18.9	2.0%	4.5%
Stride 8, 1 ref	19.3	18.8	18.3	2.6%	5.2%
Stride 8, 3 ref	19.7	19.3	18.8	2.0%	4.6%
Stride 8, 7 ref	19.4	19.1	18.6	1.5%	4.1%
Stride 8, 15 ref	18.2	17.9	17.4	1.7%	4.3%

In contrast to the previous evaluations where all frames in a video sequence were perturbed, additional experiments were conducted using a probabilistic perturbation strategy. This approach involved randomly selecting 10% and 20% of frames within each video sequence and applying a single perturbation at high severity level. Therefore while single frame models have no way of recovering from the perturbation, video models can leverage information from adjacent unperturbed frames to mitigate the impact of corruptions on isolated frames. This way we can evaluate wheather temporal context enables models to maintain detection consistency when individual frames are perturbed. The results of these experiments are summarized in Table 5.3. When looking at the results, all models as expected show a performance drop going from clean to

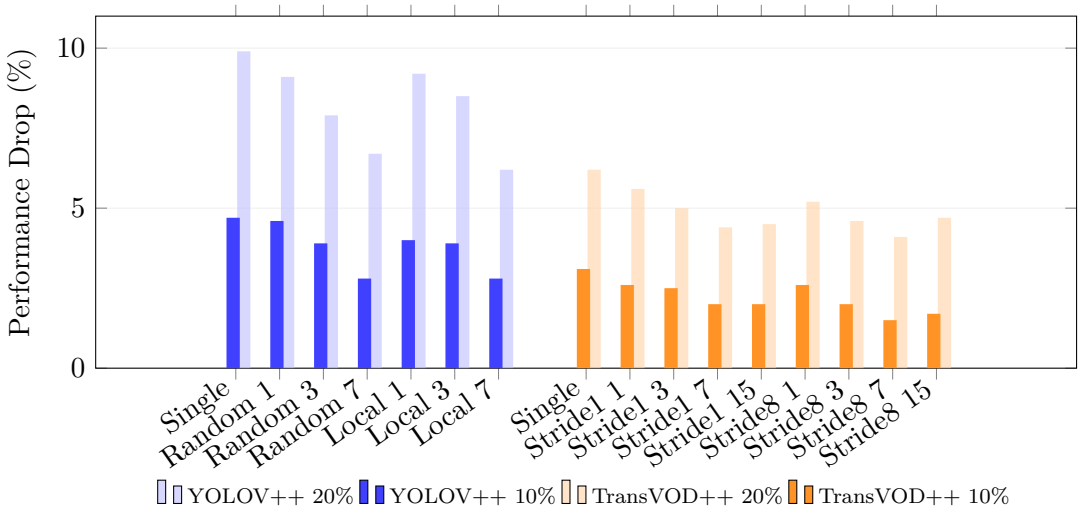


Figure 5.2: Performance drop under probabilistic perturbations. For each configuration, bars are grouped by model (YOLOV++ left, TransVOD++ right). Within each model group: 20% (back, lighter) and 10% (front, darker).

perturbated data. However in this experiment the video-based models show a smaller performance drop compared to their single-frame baselines, indicating that temporal context helps mitigate the impact of sporadic perturbations. For example for YOLOV++, the single-frame baseline experiences a performance drop of 4.7% and 9.9% for 10% and 20% perturbed frames respectively, while the video-based model with random sampling and 7 reference frames only shows a drop of 2.8% and 6.7%. This show a relative reduction in performance drop of 40.4% and 32.3% respectively. A similar trend can be observed for TransVOD++, where the single-frame baseline experiences a performance drop of 3.1% and 6.2% for 10% and 20% perturbed frames respectively, while the video-based model with stride 1 sampling and 7 reference frames only shows a drop of 2.0% and 4.4%. Also here a relative reduction in performance drop of 35.5% and 29.0% respectively can be observed. This clearly shows that temporal context enables video object detection models to better handle sporadic perturbations by leveraging information from adjacent unperturbed frames. For better visualization of the results, the performance drop values are also illustrated in the bar chart in Figure 5.2.

CHAPTER 6

Discussion

The aim of this this was to evaluate wheather video object detection models can leverage temporal context to improve robustness against perturbations in UAV-based videos. This was done by evaluating two state-of-the-art video object detection models, YOLOV++ and TransVOD++, on the Visdrone2019-VID dataset under various perturbations.

Taking a look at the clean data results, both YOLOV++ and TransVOD++ demonstrated improved detection performance compared to their single-frame baselines. However, the choice of reference frame sampling strategy and the number of reference frames had a significant impact on performance.

For YOLOV++, random sampling of reference frames consistently outperformed local sampling, with the best performance achieved using 7 reference frames. This is supported by the findings in the original YOLOV++ paper[SZG24], where random sampling with up to 32 reference frames yielded the best results. In this research, reference frame numbers were limited to a maximum of 7 due to computational constraints, but the trend indicates that more reference frames could potentially lead to further improvements.

For TransVOD++, sampling plays a crucial role for performance. The model was evaluated on three different sampling strategies: local sampling with a temporal stride of 1 frame, local sampling with a temporal stride of 8 frames, and global sampling. Contrary to the findings in the original TransVOD++ paper[ZLH+23], where global sampling yielded the best results on the ImageNet VID dataset, local sampling with a temporal stride of 1 frame outperformed both other strategies on the Visdrone dataset. Especially global sampling showed a significant performance drop, even performing worse than the single-frame baseline when using 15 reference frames. This suggests that for UAV-based videos, where objects are rather small, grouped together, often move quickly and scenes change rapidly, sampling temporally close frames is more beneficial than sampling distant frames. When using local sampling with a temporal stride of 1 frame, the best performance was achieved using 7 reference frames, indicating that leveraging more temporal context is beneficial for detection performance. In the original paper[ZLH+23], the best performance was achieved using 14 reference frames, this is not the case in this research, where 15 reference frames consistently led to a performance drop compared to using 7 reference frames.

This begs the question why YOLOV++ benefits from more random reference frames, while TransVOD++ shows a performance drop when using too many local reference frames. A possible explanation could be that since YOLOV++ is gathering information by aggregating information from random timepoints, it is able to make better predictions when more frames are available. In contrast, TransVOD++ relies on attention-based fusion of locally sampled reference frames. As the number of such frames increases, the attention space expands and becomes more susceptible to temporal noise and feature misalignment, particularly for small objects. Consequently, additional local reference frames may introduce more noise than useful information, leading to a decline in performance.

To sum up the clean data results, both YOLOV++ and TransVOD++ demonstrated improved detection performance, where the choice of sampling strategy and number of reference frames played a crucial role. With the correct configuration, YOLOV++ achieved an mAP of 18.0% and mAR of 30.0%, therefore an increase of 0.7 mAP and 1.7 mAR or 4.0% and 6.0% relative improvement compared to the single frame baseline. TransVOD++ achieved an mAP of 20.4% and mAR of 36.3%, therefore an increase of 1.0 mAP and 1.5 mAR or 5.2% and 4.3% relative improvement compared to the single frame baseline.

To answer the main research question regarding robustness against perturbations, the Visdrone dataset was perturbated multible ways, simulating real-world corruptions that can occur in UAV-based videos. For the first evaluation, all frames in a video sequence were perturbed, and the models were evaluated on their detection performance under these conditions.

Against the prehand expected outcome, using increasing the temporal context window does not show any significant improvement in robustness as seen in Table 5.2. This trend is consistent across both YOLOV++ and TransVOD++ models, where video-based configurations outperform their single-frame baselines, but the robustness does not improve with more reference frames. This indicates that simply adding more temporal context does increase detection performance but not robustness against perturbations, when all frames are affected. Furthermore this shows that the models are not able to use information from multiple perturbated frames to compensate for the precision loss caused by the perturbations. Therefore aggregating weak object features from multiple perturbated frames does not perform better than relying on a single clean frame.

However comparing the two models, TransVOD++ demonstrates significantly better robustness to perturbations compared to YOLOV++, as evidenced by the higher relative robustness metrics rmCAP and rmCAR. A good reason for this could be that the fully fetched transformer architecture of TransVOD++ is inherently more robust to perturbations due to its attention mechanism, while YOLOV++ relies on feature aggregation which may be more susceptible to noise introduced by perturbations.

Looking at each perturbation type individually 5.1, both models exhibit similar trends in handling different corruptions. Pixelation is the most challenging perturbation type of all perturbations, both model show a significant performance drop, however YOLOV++ detection pipeline completely fails. Since pixelation causes the most infomration loss, it is expected that the models struggle to detect objects under this perturbation. For brightness and contrast changes, both models show a relatively stable performance, where virtually no performance drop can be observed even at high severity levels. Since color jitter is introduced during training, the models are likely to be more robust to these types of perturbations. When evaluating defocus blur, gaussian noise and JPEG compression, both models experience a steady performance decline as severity increases, since fine image details are lost or corrupted.

Therefore the results indicate that temporal context improves detection accuracy but does not

inherently improve robustness against global perturbations affecting all frames.

Examining the results of the single-frame perturbation experiment in Table 5.3, where only 10% and 20% of frames are corrupted, further highlights the benefits of temporal context. Because the perturbations affect only a subset of frames, video-based models can utilize information from adjacent, uncorrupted frames to compensate for isolated corruptions. This behavior is reflected in the results: models with increasing temporal context exhibit a smaller performance degradation compared to their single-frame counterparts, demonstrating that temporal information helps mitigate the impact of sporadic perturbations. For both architectures, the best-performing video-based configuration achieves a relative reduction in performance drop of approximately 30% to 40% compared to the single-frame baseline. These findings clearly illustrate the advantage of temporal context, as short-lived disturbances can be smoothed out, allowing the model to maintain more consistent and robust detection performance across the video sequence.

In distinction to the previous evaluation where all frames were perturbed, this experiment shows that temporal context can indeed enhance robustness when perturbations are sporadic rather than pervasive across the entire video sequence.

CHAPTER 7

Conclusion and Future Work

This research investigated the robustness of video object detection models, specifically YOLOV++ and TransVOD++, under realistic perturbations in UAV-based videos. Using the VisDrone2019-VID dataset, both models were evaluated across different corruption types and severity levels to simulate real-world conditions. The study addressed two main questions: whether temporal context improves robustness against perturbations, and how reference frame selection influences this robustness.

The results show that temporal context can improve detection performance, but its effectiveness strongly depends on the sampling strategy and temporal window size. YOLOV++ benefits from a larger number of randomly sampled reference frames, suggesting a strong ability to aggregate diverse temporal information. In contrast, TransVOD++ performs best with a limited number of locally sampled reference frames, indicating higher sensitivity to reference frame quality and potential challenges with feature misalignment when too many frames are aggregated.

However, improved detection performance does not automatically translate into improved robustness. When all frames in a sequence are perturbed, neither model can effectively compensate for the degradation by aggregating information across multiple corrupted frames. In such cases, adding temporal context does not mitigate precision loss. In contrast, when perturbations are sporadic, video-based models can leverage clean neighboring frames to reduce the impact of isolated corruptions, demonstrating that temporal modeling enhances robustness only when reliable temporal information is available.

Overall, TransVOD++ exhibits consistently stronger robustness to perturbations than YOLOV++, likely due to its attention-based architecture, which appears more resilient to noise. These findings highlight that temporal context is not inherently robustifying; its benefit depends on both the corruption pattern and the model's ability to selectively aggregate high-quality temporal information.

Since this research focused only only on image-level perturbations, future work could explore the impact of temporal perturbations, such as frame drops or temporal jitter and dynamic motion blur, which are common in UAV videos. Additionally, since the models have not been trained on perturbed data, future research could investigate whether training with a wider variety of perturbations can enhance robustness. Finally, exploring more advanced temporal aggregation

techniques that can better handle noisy inputs may further improve the robustness of video object detection models in real-world UAV applications.

Model	Clean	Low	Mid	High
Brightness Change				
YOLOV++ Single-frame baseline	17.3	17.2	17.0	16.4
YOLOV++ Local (int=10), 3 ref	17.7	17.3	17.2	16.5
YOLOV++ Random (int=10), 7 ref	18.0	18.2	17.9	17.2
TransVOD++ Single-frame baseline	18.4	18.3	18.2	17.6
TransVOD++ Stride 1, 7 ref	19.3	19.3	19.1	18.4
TransVOD++ Stride 8, 3 ref	18.7	18.7	18.4	17.6
Contrast Change				
YOLOV++ Single-frame baseline	17.3	17.4	16.9	16.2
YOLOV++ Local (int=10), 3 ref	17.7	17.5	17.1	16.4
YOLOV++ Random (int=10), 7 ref	18.0	18.2	17.9	17.2
TransVOD++ Single-frame baseline	18.4	18.3	17.7	17.3
TransVOD++ Stride 1, 7 ref	19.3	19.1	18.1	17.7
TransVOD++ Stride 8, 3 ref	18.7	18.5	17.4	17.2
Defocus Blur				
YOLOV++ Single-frame baseline	17.3	16.5	10.5	6.1
YOLOV++ Local (int=10), 3 ref	17.7	16.8	10.8	6.1
YOLOV++ Random (int=10), 7 ref	18.0	17.6	11.0	6.4
TransVOD++ Single-frame baseline	18.4	17.8	15.9	13.1
TransVOD++ Stride 1, 7 ref	19.3	18.5	16.5	13.6
TransVOD++ Stride 8, 3 ref	18.7	18.0	16.0	13.0
Gaussian Noise				
YOLOV++ Single-frame baseline	17.3	17.2	13.8	8.2
YOLOV++ Local (int=10), 3 ref	17.7	17.2	13.8	8.2
YOLOV++ Random (int=10), 7 ref	18.0	18.0	14.4	8.5
TransVOD++ Single-frame baseline	18.4	18.3	16.5	14.8
TransVOD++ Stride 1, 7 ref	19.3	19.2	17.5	15.2
TransVOD++ Stride 8, 3 ref	18.7	18.6	16.7	14.6
Jpeg Compression				
YOLOV++ Single-frame baseline	17.3	15.2	10.9	7.6
YOLOV++ Local (int=10), 3 ref	17.7	15.2	11.1	7.7
YOLOV++ Random (int=10), 7 ref	18.0	15.9	11.5	8.0
TransVOD++ Single-frame baseline	18.4	17.6	16.3	15.4
TransVOD++ Stride 1, 7 ref	19.3	18.0	17.1	16.0
TransVOD++ Stride 8, 3 ref	18.7	17.8	16.4	15.5
Motion Blur				
YOLOV++ Single-frame baseline	17.3	14.6	10.9	4.7
YOLOV++ Local (int=10), 3 ref	17.7	14.9	11.0	4.8
YOLOV++ Random (int=10), 7 ref	18.0	15.7	11.3	5.0
TransVOD++ Single-frame baseline	18.4	17.4	14.9	10.2
TransVOD++ Stride 1, 7 ref	19.3	18.2	15.6	10.6
TransVOD++ Stride 8, 3 ref	18.7	17.8	15.1	10.3
Pixelation				
YOLOV++ Single-frame baseline	17.3	2.0	0.2	0.0
YOLOV++ Local (int=10), 3 ref	17.7	2.1	0.2	0.0
YOLOV++ Random (int=10), 7 ref	18.0	2.1	0.2	0.0
TransVOD++ Single-frame baseline	18.4	14.7	10.7	5.5
TransVOD++ Stride 1, 7 ref	19.3	15.5	11.2	5.7
TransVOD++ Stride 8, 3 ref	18.7	14.9	10.9	5.7

Table 7.1: Robustness under image corruptions (mAP$\times$100). For each perturbation type, we report Clean and Low/Mid/High severity. Rows include each model's baseline and selected best-performing temporal aggregation settings.

Overview of Generative AI Tools Used

Ihr Text hier.

Übersicht verwendeter Hilfsmittel

Enter your text here.

List of Figures

List of Tables

List of Algorithms

Bibliography

[CCHW20] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection, 2020.

[CMS+20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005.

[GLW+21] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[HD19] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[LAE+16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, pages 21–37. Springer, 2016.

[LFH+24] Xiaoqiong Liu, Yunhe Feng, Shu Hu, Xiaohui Yuan, and Heng Fan. Benchmarking the robustness of uav tracking against common corruptions. *arXiv preprint arXiv:2403.11424v1*, March 2024. [cs.CV].

[LH19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[LLC+21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith

Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, December 2015.

[SWG22] Yuheng Shi, Naiyan Wang, and Xiaojie Guo. Yolov: Making still image object detectors great at video object detection. *arXiv preprint arXiv:2208.09686*, 2022.

[SZG24] Yuheng Shi, Tong Zhang, and Xiaojie Guo. Practical video object detection via feature selection and aggregation. *arXiv preprint arXiv:2407.19650*, 2024.

[VJ01] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–511–I–518. IEEE, 2001.

[YLDG22] Jianwei Yang, Chao Li, Xiaohang Dai, and Jianfeng Gao. Focal modulation networks. In *Advances in Neural Information Processing Systems*, 2022.

[ZLH+22] Qianyu Zhou, Xiangtai Li, Lu He, Yibo Yang, Guangliang Cheng, Yunhai Tong, Lizhuang Ma, and Dacheng Tao. Transvod: End-to-end video object detection with spatial-temporal transformers. *arXiv preprint arXiv:2201.05047*, 2022.

[ZLH+23] Qianyu Zhou, Xiangtai Li, Lu He, Yibo Yang, Guangliang Cheng, Yunhai Tong, Lizhuang Ma, and Dacheng Tao. Transvod++: Improved spatial-temporal transformer models for video object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[ZSL+21] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021.

[ZWD+17] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection, 2017.

[ZWD+21] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7380–7399, 2021.