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

Enter your text here.

Contents

CHAPTER 1

Introduction

Video object detection is a relatively new field in computer vision that aims to leverage the 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, improved robustness, and greater temporal consistency when detecting objects across frames. 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, and especially in UAV-based applications, 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.

The central research question of this thesis is how the incorporation of temporal context in video object detection models affects their robustness to common perturbations in UAV-based imagery. In particular, the impact of the number of reference frames used to construct the temporal context is analyzed. To address this question, a comprehensive evaluation of state-of-the-art video object detection models is conducted on a benchmark dataset designed for UAV scenarios. In addition, a novel robustness evaluation metric is proposed, which quantifies the performance degradation of a model under different perturbations relative to its performance on clean data.

CHAPTER 2

Method

2.1 Models

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 [?], which is a transformer based model that uses attention mechanisms to aggregate temporal information from multiple frames. On the other hand YOLOV [?], 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.

2.1.1 Transvod

Transvod [?] 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++ [?], which builds upon the original TransVOD architecture and introduces several enhancements to

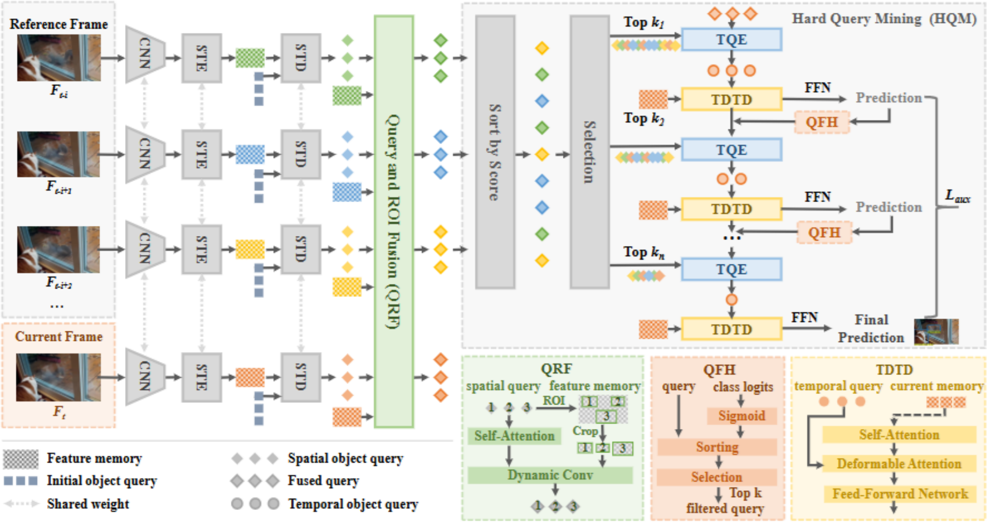


Figure 2.1: TransVOD++ architecture overview (from [?])

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(%). TransVod short summary: ...

Model design

TransVOD++ builds upon the deforamble DETR [?] architecture, which itself improves the DETR [CMS+20] model by introducing deformable attention modules to better handle multi-scale features and improve convergence speed. This baseline is used to extract spatial features from individual frames. 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 [?], 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 seend 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 in 2.1 as Temporal Query Encoder (TQE) and Temporal Deformable Transformer Encoder to produce the final detection results.

2.1.2 YOLOV

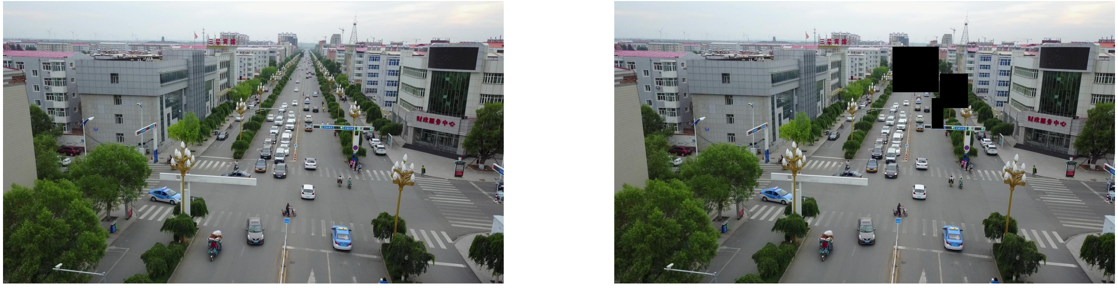
Based on the popular YOLO [RDGF16] architecture, to be more precise YOLOX [GLW^{+}21], which are one-stage detectors known for their speed and efficiency, YOLOV [**?**] extends this architecture to video data by incorporating temporal feature fusion techniques. The paper that was published in 2023, was 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

2.2 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.

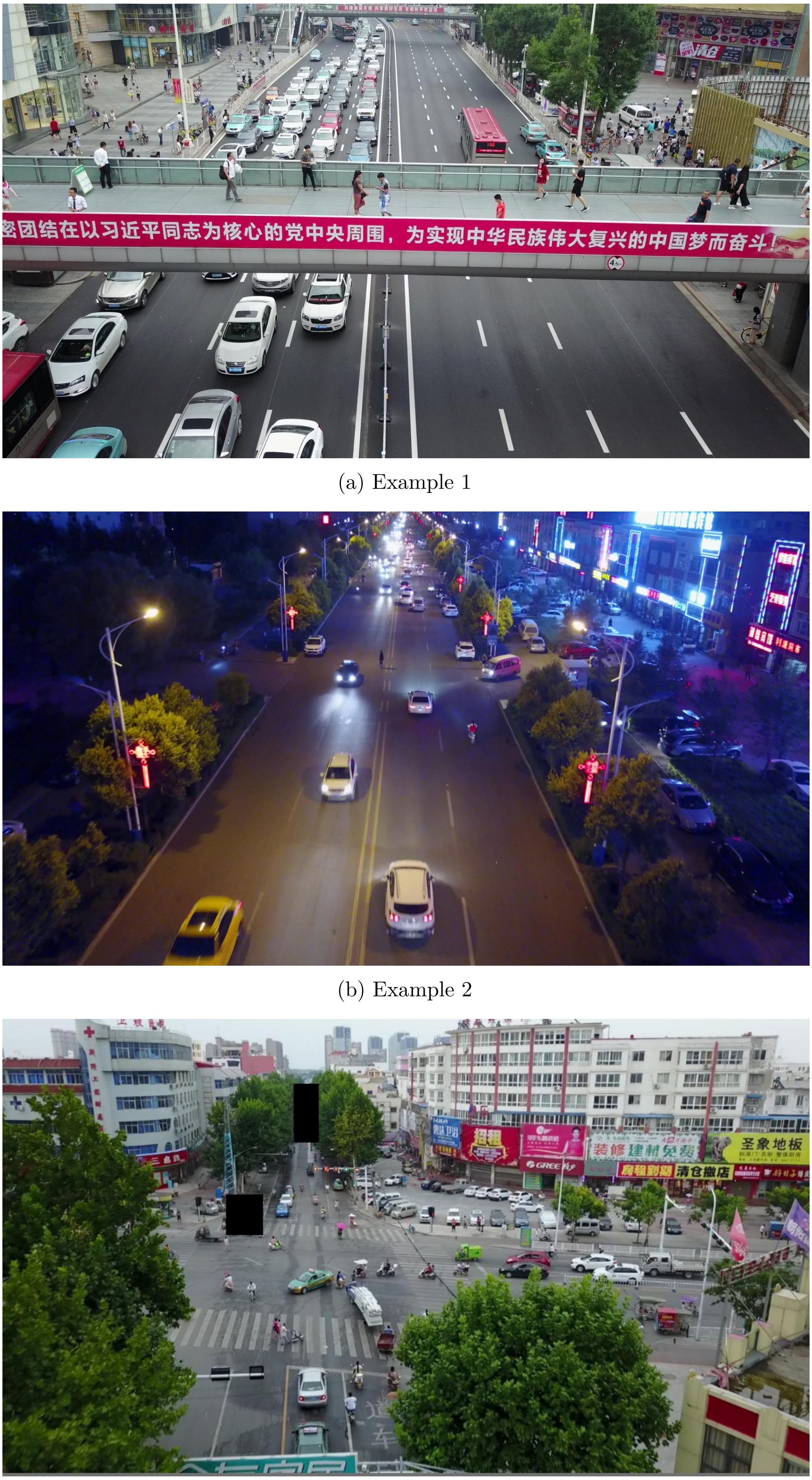
Since the dataset includes ignored regions, which are areas in the images where objects are not annotated due to beeing too crowded or too small, these regions are taken out of the evaluation to avoid penalizing the models for false positives in these areas. This was done by blacking out these regions 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 2.2: Example of an image from the Visdrone dataset with ignored regions blacked out for training and evaluation.


密团结在以习近平同志为核心的党中央周围，为实现中华民族伟大复兴的中国梦而奋斗！
圣象地板
装修建材免费
清仓撤店

(a) Example 1

(b) Example 2

(c) Example 3

Figure 2.3: Example images from the Visdrone dataset.

CHAPTER 3

Perurbation

A signficant 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.

3.0.1 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.

3.0.2 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.

- 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.

3.0.3 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:

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

where σ is the noise standard deviation.

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

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

where K_{disk} is a $k \times k$ kernel, $r = \lfloor k/2 \rfloor$, $Z = \sum_{u,v} \nVdash(\cdot)$, 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 $\tilde{I} = I * K_{\text{motion}}$.

Brightness change. We apply a global multiplicative gain:

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

with $\alpha > 0$.

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

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

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:

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

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:

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

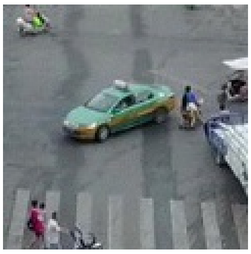
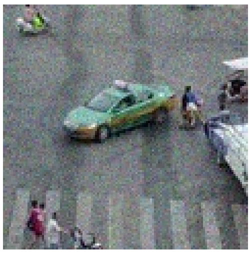
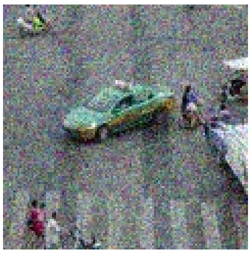
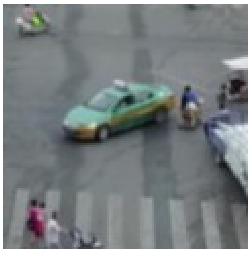
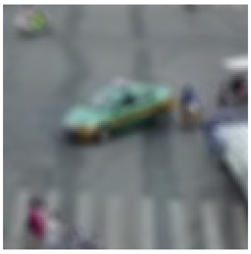
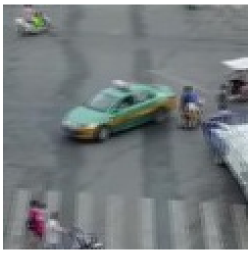
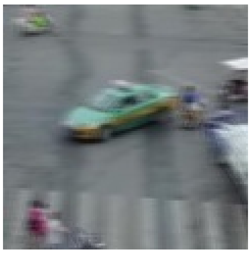
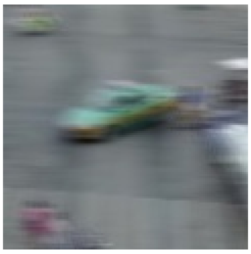
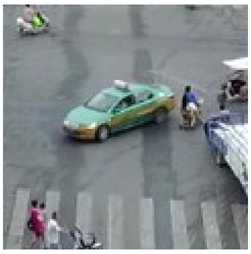
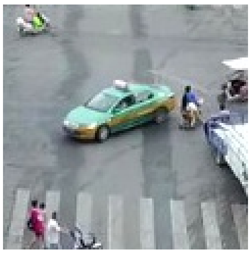
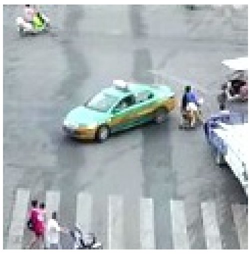
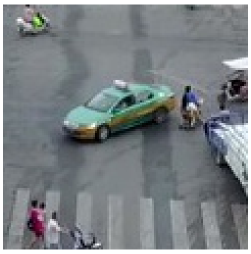
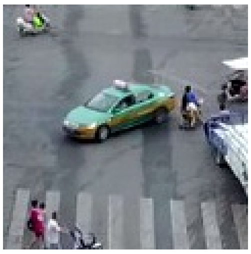
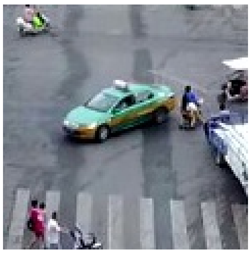
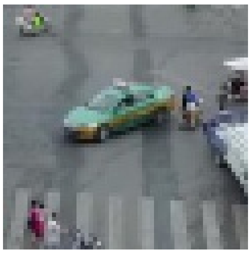
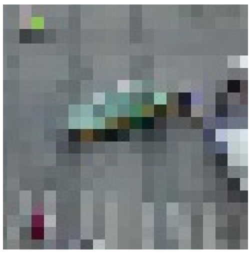
The specific severity levels are defined as follows:

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	$k = 3,\ \theta = 0°$	$k = 7,\ \theta = 0°$	$k = 15,\ \theta = 0°$
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 × 3 severity levels) can be evaluated to assess the robustness of video object detection models in UAV-based imagery. To give an impression of the applied perturbations, example images for each perturbation type and severity level are shown in Figure 3.1.

Figure 3.1: Example images illustrating perturbation severities (Low, Medium, High).

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

CHAPTER 4

Evaluation Metric

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.

Mean Corruption Average Precision (mCAP) is defined as:

$$\mathrm{mCAP} = \frac{1}{|C|} \sum_{c \in C} \frac{1}{S} \sum_{s=1}^{S} \mathrm{mAP}_{s,c}, \tag{4.1}$$

where C is the set of all perturbation configurations (i.e., perturbation types and severity levels), S is the number of severity levels, and $\mathrm{mAP}_{s,c}$ 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:

$$\mathrm{rmCAP} = \frac{1}{|C|} \sum_{c \in C} \frac{1}{S} \sum_{s=1}^{S} \frac{\mathrm{mAP}_{s,c}}{\mathrm{mAP}_{\mathrm{clean}}}, \tag{4.2}$$

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

Probability-based mean Corruption Average Precision (p-mCAP). In addition to corruption-type robustness, we evaluate robustness with respect to the probability of perturbation occurrence. Let P denote the set of perturbation probabilities applied independently per frame. We define probability-based mean Corruption Average Precision (p-mCAP) as

$$\text{p-mCAP} = \frac{1}{|P|} \sum_{p \in P} \frac{1}{S} \sum_{s=1}^{S} \text{mAP}_{p,s}, \tag{4.3}$$

where $\text{mAP}_{p,s}$ denotes the detection performance when each frame is perturbed with probability p at severity level s.

Relative probability-based mean Corruption Average Precision (rp-mCAP). Analogous to rmCAP, we define a relative probability-based robustness metric by normalizing p-mCAP with respect to a baseline detector:

$$\text{rp-mCAP} = \frac{1}{|P|} \sum_{p \in P} \frac{\sum_{s=1}^{S} \text{mAP}_{p,s}^{f}}{\sum_{s=1}^{S} \text{mAP}_{p,s}^{\text{baseline}}}. \tag{4.4}$$

CHAPTER 5

Experiments Setup

5.0.1 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 [?] and trained using the AdamW [?] optimizer.

5.0.2 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 int the provided code repositories, TransVOD++ and YOLOV++ use inherently different sampling strategies.

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.

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.

5.0.3 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.

5.0.4 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.

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

Table 5.1: 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

Table 5.2: 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 evaluttions random seeds were fixe 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. Futhermore evaluations with different perturbation probabilities were conducted to assess the models robustness when only a subset of frames are perturbed. For this two different probabilities were chosen namely 0.5 and 0.75 meaning that each frame has a 50% or 75% chance of being perturbed. The goal of this evaluation is to simulate more realistic scenarios where not all frames are affected by perturbations and conclude on how well the models can leverage unperturbed frames to maintain detection performance.

CHAPTER 6
Results

6.1 Clean Results

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 6.1.

As can it can be seen in Table 6.1, 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.

6.1.1 YOLOV++ clean data results

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.

6.1.2 TransVOD++ clean data results

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

Table 6.1: 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	**mAP**$_{50:95}$ (%)	**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

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. FIND EXPLENATION FOR DEGRADING PERFORMANCE WITH GLOBAL SAMPLING 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.

6.2 Perturbation Results

The results are summarized in Table **??**.

Table 6.2: 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	mCAP$_{50:95}$ (%)	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++				
Single-frame baseline	18.3	33.3	0.94	0.96
Sampling stride 1, 1 reference frame	18.7	34.0	0.95	0.95
Sampling stride 1, 3 reference frames	18.9	34.5	0.95	0.95
Sampling stride 1, 7 reference frames	**19.1**	**34.7**	0.94	0.96
Sampling stride 1, 15 reference frames	18.5	34.0	0.93	**0.97**
Sampling stride 8, 1 reference frame	18.5	33.5	**0.96**	0.95
Sampling stride 8, 3 reference frames	18.5	33.7	0.94	0.95
Sampling stride 8, 7 reference frames	18.2	33.2	0.94	0.96
Sampling stride 8, 15 reference frames	17.0	31.2	0.93	0.95

On perturbed data, TransVOD++ significantly outperforms YOLOV++ in terms of mCAP and mCAR across all configurations, as seen in Table 6.2. 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++.

6.2.1 YOLOV++ perturbation results

The perturbation results for YOLOV++ in Table 6.2 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 6.1, 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.

6.2.2 TransVOD++ perturbation results

For TransVOD++, a similar trend as in the clean data results can be observed in Table 6.2, 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 0.93 to 0.96 and rmCAR values between 0.95 and 0.97. 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.

6.2.3 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.

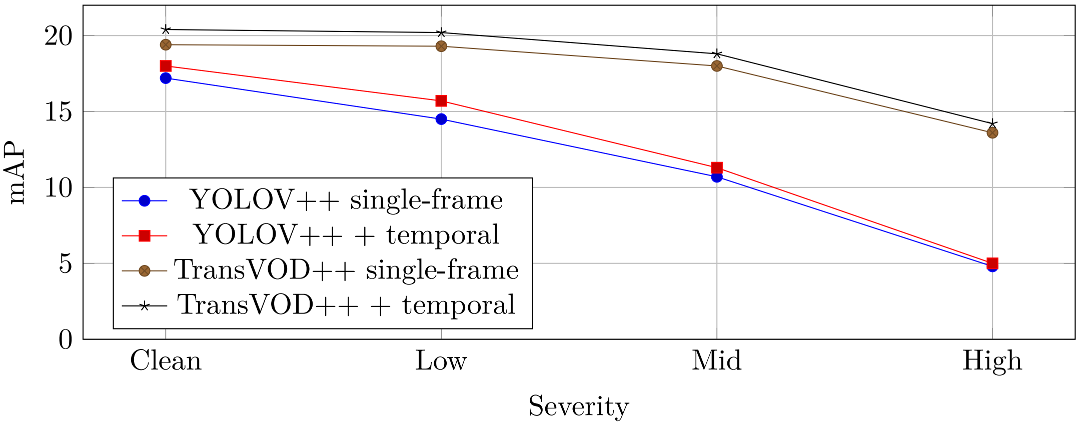
6.3 Deeper insight into Perturbations

To gain a deeper understanding of into model robustness in the following sections, the performance of the best YOLOV++ and TransVOD++ configurations on each perturbation type are analysed separately.

. . .

6.4 Perturbation Probability Results

. . .



Model	Clean	Low	Mid	High
YOLOV++ Single-frame baseline	17.2	14.5	10.7	4.8
YOLOV++ Sampling stride 8, 7 ref frames	18.0	15.7	11.3	5.0
TransVOD++ Single-frame baseline	19.4	19.3	18	13.6
TransVOD++ Sampling stride 1, 7 ref frames	20.4	20.2	18.8	14.2

Figure 6.1: Robustness under image corruptions. Top: mAP vs severity. Bottom: mAP table (clean/low/mid/high).

CHAPTER 7

Conclusion and Future Work

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

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

[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.

[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].

[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.

[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.

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

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

[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.