Explaining and verifying the robustness of Visual Object Trackers to noise

Iason Karakostas
Department of Informatics,
Aristotle University of Thessaloniki,
Thessaloniki, 54124, Greece
Email: iasonekv@csd.auth.gr

Vasileios Mygdalis
Department of Informatics,
Aristotle University of Thessaloniki,
Thessaloniki, 54124, Greece
Email: mygdalisv@csd.auth.gr

Ioannis Pitas
Department of Informatics,
Aristotle University of Thessaloniki,
Thessaloniki, 54124, Greece
Email: pitas@csd.auth.gr

***Abstract*—2D tracking is an important computer vision task with important applications in autonomous embedded systems such as Unmanned Aerial Vehicles and autonomous cars that particularly attracted scientists in the past few years. Many new methods have aroused that significantly pushed the state-of-the-art performance in terms of tracking precision, success rate and execution speed, in well-designed and established existing publicly available benchmarks. Despite the fact that these benchmark datasets include as many application scenarios as possible, another commonly neglected yet important aspect is the robustness of tracking methods, notably to noise related with image acquisition, capturing storing and transmission. This paper presents a robustness evaluation toolkit for 2D Visual Object Tracking, that can exploit existing datasets in order to evaluate the robustness of 2D visual tracking methods to realistic image distortion scenarios, mostly encountered in embedded systems. The source code of this toolkit will be made publicly available upon paper acceptance.**

***Keywords*—2D visual target tracking, toolkit, robustness evaluation**

I. Introduction

There are numerous applications in products that we use every day based on computer vision methods, from applications regarding road safety (e.g., pedestrian detection), autonomous systems, robotics, media production, visual surveillance, human-computer interfaces and augmented reality. Perhaps one of the most important computer vision tasks found in numerous devices is the 2D Visual Object Tracking, which can be found in Unmanned Aerial Vehicles (UAVs) assisting the pilot to follow a selected target or even further, the UAV to follow the target autonomously exploiting the visual information. The tracking methods expect the single target to be tracked to be selected by hand (e.g. from the user using a GUI) or from the output of an object detector [1], [2] and then the aim is to determine the position of target's bounding box in the following frames of a video stream.

Each tracking methodology exploits a different approach in order to solve the target tracking problem. One big family of such methods are the Correlation Filter (CF) based trackers [3], [4], [5], which learns to regress the target appearance to a distribution, using a correlation filter. This category became popular with [6] that allowed efficient adaptive online training of such filters. These methods manage to achieve high framerate/speed performance by performing most of the computations in the frequency domain. The main philosophy of such methods is that given an input target, image features are calculated and deployed in order to train a correlation filter that tries to distinguish the features of the target from the features of the background. This filter is exploited during prediction on the next frame. With the emerge of deep AI, a tracking method category that became popular is methods using Siamese Networks [7], [8], [9]. In these methods, the tracking network has two inputs, the search area and the template of the target and a single output, the desired bounding box. The networks, trained utilizing pair of images with the same target, measure the similarity in these two image patches and aim to detect similar regions between them and may still employ a CF [10]. Other popular methods also exists in the literature, that cannot be strictly put into one of the aforementioned categories [11] and a more recent approach to the tracking problem is to implement attention mechanisms [12], [13].

Many dataset benchmarks have been developed in the past years in order to measure the performance of tracking methods, containing general videos with challenging attributes such as object occlusions, illumination variations, motion blur etc. [14], [15], datasets containing huge amount of data suitable for training deep neural network based trackers [16], [17], [18], or more application specific datasets such as high framerate video stream [19] or UAV usage [20]. Despite the fact that 2D object tracking is widely used in real-life scenarios where it is common to experience hardware issues with image capturing sensors or the video stream transmission, minimal effort has been put towards investigating the robustness of tracking methods against noise from such sources. In [21], multiple methods are evaluated against various levels of additive white Gaussian noise, but the survey does not include other type of noises that can occur in real-life scenarios. An image can be noisy due to various reasons such as poor focus, distortions caused by the presence of magnetic field generated by electronic circuits etc. Even weather conditions can cause trouble to image acquisition and even more to wireless data transfer, for example in a scenario where a UAV flies autonomously or semi-autonomously and streams the video to a powerful land-based processing unit in order to perform a vision based task [22].

Ideally, a 2D tracking application must be able to handle various types of noise. Motivated by the current trend towards trustworthy and robust AI, we developed a tool for studying the robustness of tracking methods to noise. This paper particularly focuses on studying and analysing the effects of image

(a) Original frame (b) Distorted frame

Fig. 1: Gaussian Noise (GN). In a) the original frame is shown and in b) the distorted version of the same frame.

(a) Original frame (b) Distorted frame

Fig. 2: Low Quality (LQ) distortion. In a) the original frame is depicted and in b) its highly compressed distorted version.

(a) Original frame (b) Distorted frame

Fig. 3: Key-Frame Loss distortion can have a huge impact on a visual based method.

acquisition and transmission noise in state-of-the-art tracking methods. Nevertheless, the developed framework can easily be expanded to include more types of noise, adversarial attacks evaluation or even evaluation on other type of vision based methods such as Object Detection, Region Segmentation etc.

II. Robustness Evaluation Toolkit

This section presents Visual Object Tracking - Robustness Toolkit (VOT-RT), that examines the robustness of 2D visual object trackers in various conditions. The toolkit consists of three modules: a) the input module, b) the distortion module and , c) the 2D Visual Object Tracker Evaluation module. The video stream can be a single image, a video file, a folder containing image sequences or whole datasets. The distortion module creates the noisy examples and currently supports a) Image Acquisition noise types and b) Transmission noise types, that may occur in real life applications. Finally, the distorted stream is given for evaluation to the desired 2D object tracking method. The distortion module can easily be expanded and the evaluation step provides an easy way to implement more tracking methods for evaluation or attach a different evaluation module for other computer vision based tasks. The parameters of each type of noise described below are user adjustable. The user can also select to employ more than one type of noise for robustness evaluation of a method or training purposes.

A. Image Acquisition Noise

1) Gaussian noise (AG): The toolkit will add Gaussian noise to each frame of a video sequence or whole datasets. The term Gaussian noise refers to additive statistical noise having a probability density function equal to that of a normal distribution. In digital images this type of noise can occur during image acquisition due to poor illumination, high image sensor temperature or electronic circuit noise [23]. There are techniques for reducing the effects of this type of noise [24], [25], however, the resulting image may still be noisy or the system may not have enough computational resources to perform such corrections. An example is illustrated in Figure 1. The user can select during the evaluation with this type of noise the level of distortion.

2) Salt and pepper (SP): This type of noise can occur during analog to digital conversion or data transfer [26]. Frames affected by this type of noise appear to have sparsely distributed black and white pixels. This type of noise can be handled with median filtering or more complex methods [27], although in autonomous systems this type of noise may not be detected affecting the overall performance of the system. The toolkit has the option to add only salt or pepper noise to the image, and the level of distortion is adjustable as well.

B. Transmission Noise

1) Low Quality: In this scenario, each image/frame is highly compressed and decompressed before given as input to the tracking method. In such manner, cases where the processed video needs to be streamed to the processing unit but the channel bandwidth may not be enough are simulated. Technologies like 4G/5G can help towards the capabilities of the wireless connectivity, although, special permission may be needed to deploy such networks or the system should use commercially available networks, which can be costly. There are still a lot of applications where the video stream is transmitted from a big distance and WiFi straggles to perform. Lack of bandwidth can force the system to compress the video in such extent that the received video may have compression artifacts, e.g. loss of edge clarity and edge fuzziness. Figure 2 depicts an example of o Low Quality distorted frame (Fig. 2b).

2) Key-Frame Loss: During video compression, a compression algorithm, e.g. MPEG [28], apart from the implementation of other compression techniques, manages to reduce the total size of a video by creating frames that do not contain information for each pixel of the frame. Instead for these frames, only some differences/changes from a previous or later frame are stored. Thus, when a video is compressed, the algorithm selects frames that are fully stored (key frames/i-frames) and achieves further compression by converting the rest of the frames to structures with the differences from previous key frames (p-frames) or both previous and later key frames (b-frames). In order to decompress the video, its trivial to understand the importance of the key frames. Albeit, during

video transmission, network congestion can result to packet loss which can cause key frame loss, creating distorted frames during decompression as depicted in Figure 3. Such a distorted frame result can be very challenging for a visual target tracking method. In the presented framework the frame loss is simulated with the assistance of FFmpeg [29].

III. Experimental results

To demonstrate the use case of the proposed framework, we have examined the performance of state-of-the art trackers that can perform real time either on high power PCs equipped with an Nvidia GPU, or even in embedded systems such as the Nvidia Jetson family. More specifically, the evaluation includes TransT [12], PrDiMP [11], DaSiamRPN [30] and SiamFC [8], a method that does not require GPU hardware and employs tracking failure mechanisms, LDES-ODDA [31], and finally KCF [3] which is a baseline tracking method that can be used on less powerful embedded systems. The same distorted data based on the OTB dataset [14] were given as input to all of the evaluated methods. In all of the experiments the parameters of each noise are the same, i.e., for every type of noise the same video stream was given as input during evaluation.

For the quantitative evaluation, the one-pass evaluation (OPE) protocol was employed. With OPE, the tracking method is initialized in the first frame with the ground truth bounding box of the object and the goal is to produce the following bounding boxes that include the target as best as possible. As evaluation metric, the widely used Overlap Score (OS) was employed, defined as $S = \frac{|r_t \cap r_0|}{|r_t \cup r_0|}$, where r_t and r_0 is the tracked and ground truth bounding boxes respectively, $\cap$ and $\cup$ denote the intersection and union operators and $|\cdot|$ denotes the number of pixels inside the specified area. OS is calculated in a per frame basis. When the value of S is larger than a certain threshold, it is assumed that the tracker, successfully tracks the desired object.

Figure 4 depicts the success plots for the OPE evaluation and the OS score when the threshold is set at 0.5. KCF and LDES-ODDA despite suffering from performance drop, handle well the LQ and SP case. KCF, mainly due to the feature extraction mechanism that it exploits (Histogram of Oriented Gradient [32]) and the fact that it reduces the resolution of the template image which can reduce the compression artifacts and get rid to some extend of SP noise. LDES-ODDA employs also a distraction detection mechanism and has target re-detection capabilities that can assist the tracking procedure under noisy environments. GN has a greater impact for these two methods mainly to due the greater interference of this type of noise in the feature extraction process.

Noisy inputs appear to have a great impact on the Siamese based methods; both SiamFC and DaSiamRPN experience significant performance drops under noisy conditions. SiamFC's performance drops by more the 7% for the LQ case and for DaSiamRPN more than 10%. These two methods, are also affected more than other methods for the GN and SP case. This can be explained by the fact that in these methods the original template of the target is not altered, and the extracted features are affected by the specific location of the noise in the template that is not present in the later frames in the same way. Also, the relatively high resolution template and search area input does not help in removing the effect of SP or GN to some extend. The issues that rise with the evaluation on noisy datasets may be confronted by containing noisy examples in the training dataset of the Siamese networks, so that the methods can be more tolerant to such distortions.

PrDiMP, manages to perform satisfactorily against the KfL scenario, taking into account the difficulty of this task. For the rest of the scenarios, the generation of predictive probability distribution mechanism of PrDiMP, appears to be increase its robustness to some extend over these type of noises. TransT appears to handle well the three of the implemented noise types: GN, SP and LQ. In fact, the performance for GN and SP is almost identical. This achievement can be attributed to the usage of attention mechanism that shifts the focus away from noisy artifacts. For the LQ case, the bigger performance drop is somehow expected since edges and key characteristics of the target are distorted in a different way in successive frames, causing the target tracker to drift and fail. TransT has significant performance loss for the most challenging KfL scenario, although it manages to outperform the rest of the trackers.

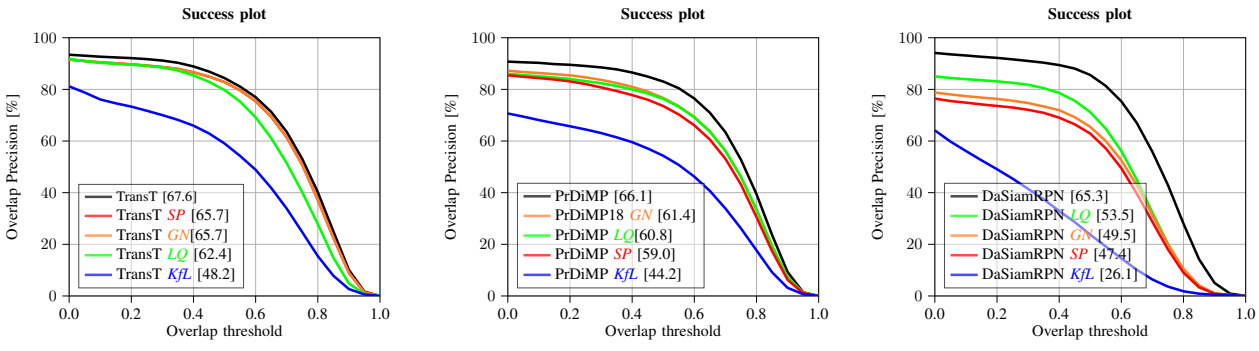
Figure 5 illustrates a qualitative evaluation for all of the types of noises studied in this paper for the PrDiMP tracker. In the original sequence the method manages to track the desired target with no issues, although in this sequence, it is noticeable that for every type of noise the method fails to perform. In the LQ scenario, due to the fact that this method does not search the target only in a portion of the frame, manages to detect the target again, but for the most part, the output of the target is incorrect. For GN and SP the tracker has a similar performance and is lost quite fast although this behaviour is not common for this type of noise for this dataset. In the KfL scenario, the tracker is not completely lost but it is noticeable that struggles to determine the correct aspect ratio or scale of the target.

IV. Conclusion

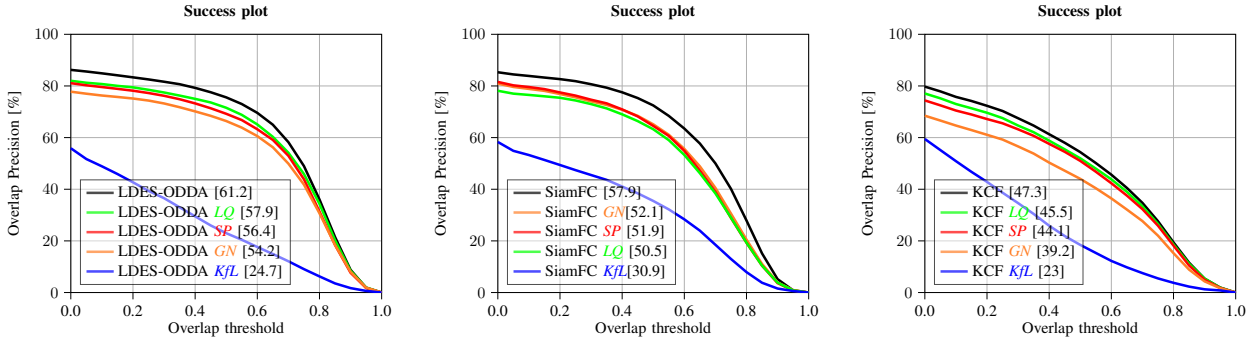
An evaluation toolkit for 2D Visual Object Tracking methods in terms of robustness has been presented. The evaluation toolkit has the ability to measure the effect of various real-world type of noises, that can appear due to image capturing issues in an autonomous system, or video transmission. In this framework, there are already implemented some of the most recent and state-of-the-art in terms of performance tracking methods and it is easy to extend the list of the supported trackers in the future. As a future work other type of noises (e.g. adversarial attacks) can be implemented to this toolkit in order to evaluate the robustness of tracking methods more thoroughly. From the evaluation results, the conclusion that is drawn, is that noise always affects the performance of the tracking methods, even if for the human eye the result of image quality may be acceptable. Severe issues arise in the case where during a video streams packets are lost causing key-frame loss. In such cases, most of the tracking methods in fact, fail to track the desired target at all.

Acknowledgment

This work has received funding from European Union's Horizon 2020 research and innovation programme under grant agreement No 871479 (AERIAL-CORE). This publication reflects only the authors' views. The European Commission is not responsible for any use that may be made of the information it contains.



(a) TransT results on OTB. (b) PrDiMP results on OTB. (c) DaSiamRPN results on OTB.



(d) LDES-ODDA results on OTB. (e) SiamFC results on OTB. (f) KCF results on OTB.

Fig. 4: Success plots for the OTB dataset and the noisy variations for each evaluated type of noise.

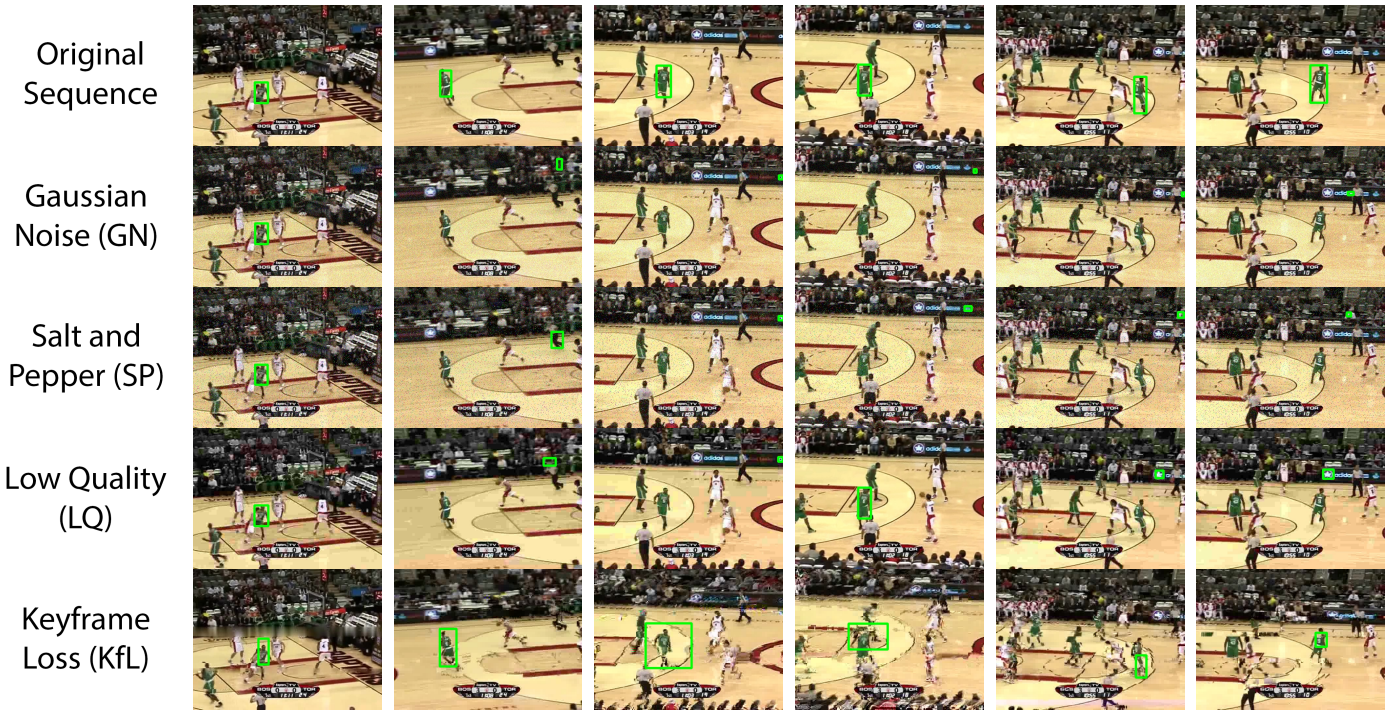


Fig. 5: Qualitative evaluation for the Gaussian Noise (GN), Salt and Pepper (SP), Low Quality (LQ) and Keyframe Loss (KfL) distortions on PrDiMP tracker. In the original sequence the tracker is able to successfully track the desired target while in GN and SP the tracker drifts from the first frames. In LQ scenario the tracker manages to track again the target but not for long. In KfL the target is not completely lost, although the scale of the produced bounding box is far from the ground truth one.

REFERENCES

[1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[3] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[4] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: Complementary learners for real-time tracking," *Computer Vision and Pattern Recognition (CVPR)*, pp. 1401–1409, 2016.

[5] H. Fan and H. Ling, "Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5486–5494.

[6] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," *Computer Vision and Pattern Recognition (CVPR)*, pp. 2544–2550, 2010.

[7] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4282–4291.

[8] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*. Springer, 2016, pp. 850–865.

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

[10] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," *Computer Vision and Pattern Recognition (CVPR)*, pp. 5000–5008, 2017.

[11] M. Danelljan, L. V. Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7183–7192.

[12] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *CVPR*, 2021.

[13] N. Wang, W. Zhou, J. Wang, and H. Li, "Transformer meets tracker: Exploiting temporal context for robust visual tracking," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[14] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[15] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, R. Pflugfelder, J.-K. Kamarainen, L. Cehovin Zajc, O. Drbohlav, A. Lukezic, A. Berg *et al.*, "The seventh visual object tracking vot2019 challenge results," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[16] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 300–317.

[17] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling, "Lasot: A high-quality benchmark for large-scale single object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5374–5383.

[18] L. Huang, X. Zhao, and K. Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[19] H. Kiani Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, "Need for speed: A benchmark for higher frame rate object tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1125–1134.

[20] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for uav tracking," *European Conference on Computer Vision (ECCV)*, pp. 445–461, 2016.

[21] M. Fiaz, A. Mahmood, and S. K. Jung, "Tracking noisy targets: A review of recent object tracking approaches," *arXiv preprint arXiv:1802.03098*, 2018.

[22] I. Mademlis, A. Torres-González, J. Capitán, R. Cunha, B. J. N. Guerreiro, A. Messina, F. Negro, C. Le Barz, T. Gonçalves, A. Tefas *et al.*, "A multiple-uav software architecture for autonomous media production," in *Workshop on Signal Processing Computer vision and Deep Learning for Autonomous Systems, EUSIPCO2019*, 2019.

[23] D. P. Cattin, "Image restoration: Introduction to signal and image processing," *MIAC, University of Basel. Retrieved*, vol. 11, p. 93, 2013.

[24] A. Makandar and B. Halalli, "Image enhancement techniques using highpass and lowpass filters," *International Journal of Computer Applications*, vol. 109, no. 14, pp. 12–15, 2015.

[25] Y. Qi, Z. Yang, W. Sun, M. Lou, J. Lian, W. Zhao, X. Deng, and Y. Ma, "A comprehensive overview of image enhancement techniques," *Archives of Computational Methods in Engineering*, pp. 1–25, 2021.

[26] J. Azzeh, B. Zahran, and Z. Alqadi, "Salt and pepper noise: Effects and removal," *JOIV: International Journal on Informatics Visualization*, vol. 2, no. 4, pp. 252–256, 2018.

[27] R. H. Chan, C.-W. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Transactions on image processing*, vol. 14, no. 10, pp. 1479–1485, 2005.

[28] D. Le Gall, "Mpeg: A video compression standard for multimedia applications," *Communications of the ACM*, vol. 34, no. 4, pp. 46–58, 1991.

[29] S. Tomar, "Converting video formats with ffmpeg," *Linux Journal*, vol. 2006, no. 146, p. 10, 2006.

[30] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 101–117.

[31] I. Karakostas, V. Mygdalis, A. Tefas, and I. Pitas, "Occlusion detection and drift-avoidance framework for 2d visual object tracking," *Signal Processing: Image Communication*, vol. 90, p. 116011, 2020.

[32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. Ieee, 2005, pp. 886–893.