Vision in Adverse Weather: Augmentation Using CycleGANs with Various Object Detectors for Robust Perception in Autonomous Racing

Izzeddin Teeti [1 2]   **Valentina Muşat** [3]   **Salman Khan** [2]   **Alexander Rast** [1]   **Fabio Cuzzolin** [2]   **Andrew Bradley** [1]

Abstract

In an autonomous driving system, perception - identification of features and objects from the environment - is crucial. Autonomous racing, in particular, features high speeds and small margins that demand rapid and accurate perception systems. During the race, the weather can change abruptly, causing significant degradation in perception, resulting in ineffective manoeuvres. In order to improve detection in adverse weather, deep-learning-based models typically require extensive datasets captured in such conditions - the collection of which is a protracted and costly process. However, recent developments in Cycle-GAN architectures allow the synthesis of highly realistic scenes in multiple weather conditions. To this end, we introduce an approach of using synthesised adverse condition datasets in autonomous racing (generated using CycleGAN) to improve the performance of four out of five state-of-the-art detectors by an average of 42.7 and 4.4 mean average precision (mAP) percentage points in the presence of night-time conditions and droplets, respectively. Furthermore, we present a comparative analysis of five object detectors - identifying the optimal pairing of detector and training data for use during autonomous racing in challenging conditions.

1. Introduction

Autonomous driving is a current focus of many governments, startups, already-established corporations, and research labs (CBI Insights, 2020). Many studies aim to create an SAE level 4 or level 5 autonomous vehicle (AV) (International Society of Automotive Engineers (SAE), 2018), able to operate in unstructured environments and adverse conditions. From the AV's perspective, an adverse condition is *any* condition with which the detection model has never experienced during training. Of particular interest are weather and lighting effects that change the appearance of objects, such as droplets (which can occlude objects and blur images), or night driving.

Research in AVs has stimulated the development of autonomous racing, where cars designed for racing compete fiercely on racetracks, potentially under varied weather conditions which may be subject to rapid changes. The close, competitive nature of racing amplifies the effects of any performance gains (or weaknesses) in the AV control system in a safe, controlled test environment (Culley et al., 2020). Autonomous racing is therefore an ideal application to test the extreme operating limits of AVs, accelerating the development process of AV algorithms - including object detection in adverse conditions.

Two main approaches exist to AV control systems. The first is an 'end-to-end' AI-based approach that takes visual inputs and outputs control commands (Bansal et al., 2018). The end-to-end nature of such systems, however, makes them black boxes: they lack explainability and are difficult to debug. Furthermore, they require an extensive amount of labelled training data; however, gathering such data is time-consuming and expensive. The second approach breaks down the AV control system into a pipeline of interacting subsystems including perception, decision making, and control (Badue et al., 2021) - enhancing explainability, debugging, and a higher confidence in the ability of the autonomous vehicle to handle a variety of road scenarios. Both approaches, however, are vulnerable to adverse conditions, as the latter distorts the visual input upon which they depend.

AVs see (perceive) the world through visual inputs, acquired by sensors such as (optical) cameras and LiDAR - both of which are susceptible to adverse conditions including snow, rain, fog, and night (Mohammed et al., 2020). Rain droplets can accumulate on camera lenses, occluding objects and agents, or distorting their shapes. If a LiDAR beam intersects a rain droplet at a short distance from the transmitter,

---

[1]Autonomous Driving Group, Oxford Brookes University [2]Visual Artificial Intelligence Laboratory, Oxford Brookes University [3]Oxford Robotics Institute, University of Oxford. Correspondence to: Andrew Bradley <abradley@brookes.ac.uk>.

the raindrop can reflect enough of the beam back to the receiver to cause the raindrop to be detected as an object (Espineira et al., 2021). Night driving is particularly challenging for optical sensors; the level of reflected light from target objects or agents to the camera sensor is low, causing dim or barely recognisable objects.

Perception tasks, including object detection, semantic segmentation, action recognition, and trajectory prediction are hampered by poor visual inputs caused by adverse weather, resulting in missed and incorrect detections (Mohammed et al., 2020) - of critical importance for AVs safely to 'see' other road users. Errors in detections will propagate through downstream tasks including path planning and control, causing inaccurate and unreliable manoeuvres and compromising the safety of other road users. In the particular application of AV racing, a small error in perception may result in a considerable deviation from the optimal path (Katrakazas et al., 2015) and a loss of time or even penalties or accidents. In both on-road and racing applications, the AV industry has a critical and urgent need to mitigate or deal with the significant degradation of perception caused by adverse weather conditions (Shah et al., 2017; Fursa et al., 2021).

Since both end-to-end and independent-subsystems approaches typically use machine learning models to understand the visual inputs, one way to improve them in adverse weather is to retrain those models on more data that contains adverse weather conditions. However, obtaining this data is time-consuming, expensive, and at the mercy of the weather. It also introduces a very costly and laborious process of (manually) labelling the collected data for different tasks like detection or segmentation. To overcome this challenge, researchers have suggested synthesising adverse weather datasets using simulators (Zadok et al., 2019) or simple augmentation algorithms (Fursa et al., 2021). However, the applicability of these approaches for enhancement of object detection in a real-world setting remains uncertain, due to the lack of visual realism in the imagery.

A recent promising approach is harnessing CycleGAN to generate more realistic weather augmentations using style transfer. A small number of studies have explored the use of CycleGAN to improve perception performance in adverse weather in the context of on-road autonomous driving tasks (Porav et al., 2020; Uřičář et al., 2019; Mușat et al., 2021), but as yet, no studies have attempted to exploit this potential in the context of autonomous racing.

Furthermore, various object detection models have been proposed for real-time use in AV applications (Jocher, 2021; Jocher et al., 2020; Ren et al., 2015; Tan et al., 2020), however few studies have compared different detectors for use in autonomous driving applications (Gupta et al., 2021) - let alone with a detailed comparison of the detection performance and latencies introduced.

This study represents the first attempt to harness CycleGAN-based weather appearance transfer to generate realistic training data for multiple adverse weather conditions in autonomous racing, and explores the performance of five leading object detectors for real-time object detection in adverse conditions. Thus, the contributions of this study are:

1. An approach to using individual *synthesised* adverse weather conditions to improve detection performance under multiple *real* weather conditions in autonomous racing.
2. Performance improvements for four state-of-the-art object detectors under multiple real adverse weather conditions by training them on synthesised adverse weather datasets.
3. Benchmark results for five different object detectors, a comparative analysis of their performance in terms of accuracy and speed in both sunny and adverse conditions, and the identification of the optimal detector for the specific task of object detection for autonomous racing cars.

2. Related work

2.1. Object Detection in Autonomous Racing

Traffic cones are used to regulate traffic and to mark racing circuit layout, therefore detecting them is crucial for autonomous driving and autonomous racing. Towards cone detection in a racing context, papers such as (Qie et al., 2020) employed a colour-based detector, whereas (TIAN et al., 2018) used a Histogram of Oriented Gradients (HOG) representation. Many systems, however, including (Dhall et al., 2019; Strobel et al., 2020; Culley et al., 2020), use different versions of the well-known YOLO detector (Redmon et al., 2016) because of its speed, accuracy, and reliability. A variety of object detectors have been used in AV racing, however, there is no clear study that makes a comparative analysis between these models, therefore, an investigation into the optimal detector, considering detection accuracy and speed, is desirable.

Despite being the most accurate and commonly-used class of detectors, Convolutional Neural Network-based models exhibit significantly degraded performance in adverse weather conditions (Fursa et al., 2021). While this can be mitigated by training the model on datasets that contain adverse weather, as previously noted, generating such datasets is neither easy nor quick. Various alternative solutions to generate equivalent data have been proposed using simulators, physics models, or style transfer techniques like CycleGAN. However, augmentation affects the performance of different detectors in different ways. To aid selection, this paper makes a comparative analysis between different object detectors in terms of accuracy, speed, and behaviour

against augmentation.

2.2. Synthetic Weather Generation

Numerous approaches have been proposed to generate synthetic adverse weather images. Simple approaches have utilised open-source frameworks like OpenGL (Praveen et al., 2017), and image augmentation libraries including (Fursa et al., 2021), however, they lack visual realism. More sophisticated methods were proposed to generate more realistic adverse weather images. They harness physics models to model the dynamics of raindrops (Creus & Patow, 2013), as well as, the behaviour of the light going through them (Bernard et al., 2013). Despite their improved realism, such models demand in-depth knowledge of many controlling parameters, which, in turn, are empirically determined.

Due to the difficulty in generating synthetic data using simple augmentations and physics engines (low realism, high number of free parameters), the autonomous driving research has seen an increase in the use of Generative adversarial networks (GANs) and its variants to address the issue of data scarcity in adverse weather conditions.

The first to explore this avenue for autonomous driving was the work of (Porav et al., 2018), which generates new images with snow and various light conditions for the purpose of improving feature matching. Other applications focus on improving vehicle detection during nighttime (Lin et al., 2020), object detection and semantic segmentation (Ostankovich et al., 2020).

Other articles have used CycleGAN in the context of autonomous driving tasks (Porav et al., 2020; Uřičář et al., 2019). Musat, et al. (Mușat et al., 2021) resembles our work, using CycleGAN to create fake adverse conditions for road-going AVs in general rather than racing AVs. While the augmentations demonstrated an improvement in perception performance in adverse weather, only one detector was used, and no consideration was given to how different detectors may perform when augmented training data is used, or their suitability for real-time use on-board a vehicle.

2.3. Adverse Weather in Autonomous Racing

Adverse weather affects both the vehicle dynamics (due to, e.g., ice, water puddles or soil on the road surface) and the perception stack of the autonomous vehicle (due to both detection occlusions and changes in the road surface). Different methods have been proposed to address the problem of adverse weather from a perception point of view. (Zadok et al., 2019) who improved an end-to-end model by tackling weather appearance with the use of the existing Airsim simulator (Shah et al., 2017) to generate various time-of-day appearances. The method is scalable, however, it generates data only in daytime, and the generated images are not re-
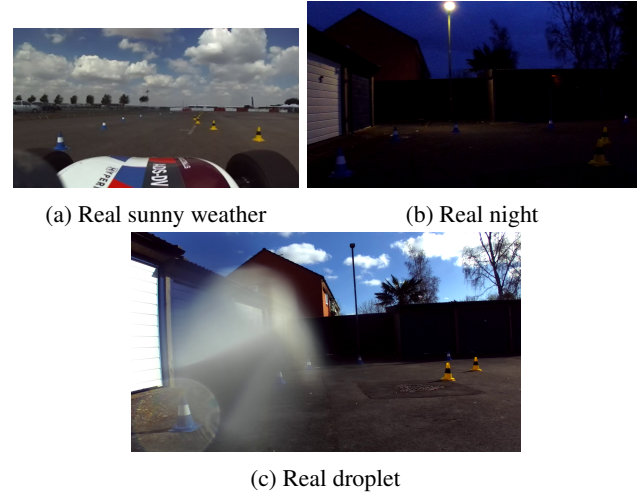

(a) Real sunny weather   (b) Real night

(c) Real droplet

Figure 1. Real datasets

alistic, thus the data may still be susceptible to differences between simulation and real world. Although such models might perform well on simulation images, they may fail on images coming from real-world unconstrained environments. Fursa, et al. (Fursa et al., 2021) employed existing simple open-source augmentation libraries to generate adverse weather datasets, but likewise potentially suffers from similar issues. Other methods including (Porav et al., 2020; Uricar et al., 2019) used GANs to improve tasks downstream from perception in urban and rural autonomous driving. To the best of our knowledge, this study represents the first attempt to harness CycleGAN-based weather appearance transfer to generate realistic datasets for multiple adverse weather conditions, in order to improve object detection in real adverse conditions in the context of autonomous racing.

3. Data

3.1. Real-world Dataset

For the object detection task, we used three real datasets of various conditions (shown in Figure 1) that were captured and annotated from an autonomous racing vehicle's vision sensor. The first dataset consists of more than $10,125$ cones captured in sunny conditions; the second ($1,548$ cones) is captured under low light conditions after sunset; the final dataset ($1,197$ cones) was captured in conditions where adherent droplets on lens completely occlude and distort some of the cones.

We used the first dataset to train and test the five object detectors (70% training, 30% testing), and kept the last two datasets to test the detectors against out-of-sample adverse illumination and rainy conditions. We exclusively focus on two object classes: blue and yellow cones.

3.2. Synthetic Dataset

Since obtaining data with ground truth in adverse conditions is a time- and effort-intensive task, we employ a CycleGAN architecture (Zhu et al., 2017) (Figure 2) to generate synthetic night and rainy (droplet) conditions with the datasets that we already have. A CycleGAN is an image-to-image translation model that learns to map images from domain A to domain B while preserving the structure of the image. It is a variant of the original GAN model, one in which two generators are trained simultaneously (one to learn the mapping from $A \rightarrow B$ and the other from $B \rightarrow A$), while enforcing a cycle-consistency (Zhu et al., 2017).

Cycle-consistency is enforced by training each GAN on the output of the other: the second generator receives as input the output of the first generator (image in domain B) and is trained to reconstruct the original image (image in domain A), by applying a reconstruction loss between the real image in domain A and its reconstruction $G_{BA}(G_{AB}(A)) = G_{BA}(\hat{B}) = \tilde{A} \approx A$. Similarly, the first generator receives as input the output of the second generator (image in domain A), and is trained to reconstruct the original image (image in domain B), by applying a reconstruction loss between the real image in domain B and its reconstruction $G_{AB}(G_{BA}(B)) = G_{AB}(\hat{A}) = \tilde{B} \approx B$.

In order to generate synthesised images that contain night or droplets conditions, we trained off-the-shelf CycleGAN implementation (Zhu et al., 2017) on two (non-racing) datasets from a mixture of resources. The first CycleGAN was trained on frames from a YouTube video captured at night, in London's streets (Stevenson, 2019) while the second CycleGAN was trained on in-house acquired images that contain droplets, both were trained at a resolution of 512×512. Then, we used them to transfer the night or droplet styles to the sunny images, ending up with 911 'fake' night images and 800 'fake' droplet images, both having the same ground truth labels as the original sunny images. Each of these cases are visualised in Figure 3. Since these fake conditions are meant for training the detectors, we split them

(a) Real sunny (original image)
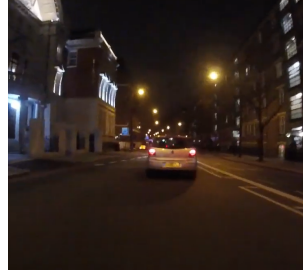
(b) Sample used to train Cycle-GAN to generate night

(c) Sample used to train Cycle-GAN to generate droplets

(d) Generated night

(e) Generated droplets

Figure 3. **a)** An original image in sunny weather, which is the input of the CycleGAN. **b, c)** Examples of images used to train the CycleGAN to generate night and droplet conditions. **d, e)** the generated images based on the input image (a) with the style of the training images (b) and (c), respectively.

into 70% training set (637 fake night and 560 fake droplet cone images) and 30% validation set (274 fake night and 240 fake droplet cone images). We used the training set to train the detectors and the validation set to measure their performance while training. Training the detectors with fake adverse conditions then testing them on real adverse conditions allows us to verify that fake generated images are a good proxy for real adverse conditions images.

3.3. Qualitative Analysis of the Generated Datasets

FID measure is the 2-Wasserstein distance (Olkin & Pukelsheim, 1982) between two multidimensional Gaussian distributions in the feature space: $\mathcal{N}(\mu_s, \Sigma_s)$ the distribution of some neural network features of the synthesized images generated by the GAN and $\mathcal{N}(\mu_r, \Sigma_r)$ the distribution of the same neural network features from the real images used to train the GAN. Mathematically, FID $= ||\mu_r - \mu_s||_2^2 + \text{trace}(\Sigma_r + \Sigma_s - 2(\Sigma_r \Sigma_s)^{1/2})$ (Olkin & Pukelsheim, 1982).
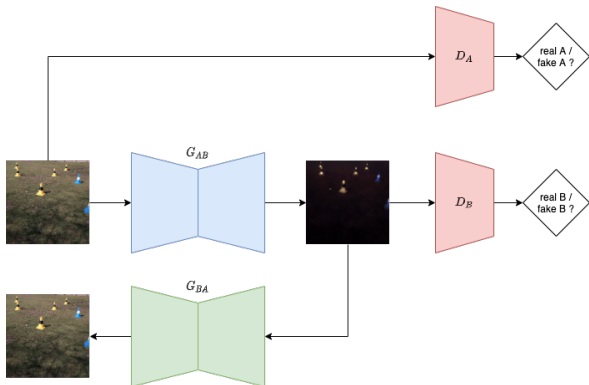

Figure 2. CycleGAN architecture

FID measure is 83.56 and 138.39 for night and droplet conditions, respectively. The lower the value of FID measure, the closer the distribution of the generated dataset to the that for the real dataset, thus, the more similar is the generated dataset to the real dataset. Therefore, the results suggest that our artificially generated night is more similar to a real night than our generated droplet images are to real images with droplets (Figure 3).

4. Object Detectors

We decided to test our concept on five state-of-the-art object detection models, not only to validate our concept, but also to run a comparative analysis between these detectors and identify the optimal one, which is crucial information for autonomous racing research. The following five object detectors were chosen due to their high speed, high accuracy, or because they can be efficiently deployed for autonomous driving applications,

YOLOv3-tiny has fewer convolution layers than YOLOv3 (Redmon & Farhadi, 2018), and as a result is claimed to be the fastest detection model (Yi et al., 2019), making it ideal for high-speed applications like autonomous racing. Counterintuitively, this high speed does not significantly compromise accuracy for our problem which only requires two classes (blue and yellow cones).

Scaled YOLOv4 (Wang et al., 2021) scales the resolution, width, and depth (number of layers) of the original YOLOv4 (Bochkovskiy et al., 2020), making it more efficient and robust. Furthermore, their authors claim it delivers the highest accuracy on the Microsoft COCO (Common Objects In Context) dataset (Lin et al., 2015). A modified, effective and real-time version of YOLOv4 was implemented in (Cai et al., 2021) for object detection in autonomous driving.

YOLOv5s (Jocher et al., 2020) is the small version of YOLOv5. It differs from YOLOv4 by being computationally more efficient in terms of storage space and training time. In terms of accuracy and inference speed, however, there is no quantitative evidence on which is better. Consequently, YOLOv5s was chosen to be compared with other detectors in our autonomous racing domain. (Gu et al., 2021) adopted YOLOv5 medium for real-time detection in autonomous driving, their system achieved a balance between accuracy and latency as it scored the second place at the Streaming Perception Challenge (Workshop on Autonomous Driving at CVPR 2021), while running in real-time. Additionally, (Benjumea et al., 2021) implemented a modified version of YOLOv5 called YOLO-Z to detect small objects in the context of autonomous racing.

Faster-RCNN (Ren et al., 2015) is a well-known Region Proposal Network (RPN)-based object detection paradigm. The model was initially proposed with a VGG (Simonyan & Zisserman, 2014) backbone; here we used a Feature Pyramid Network (FPN) (Lin et al., 2016) backbone because of its superior performance (Wu et al., 2019). Faster-RCNN was extended to stereo inputs in (Li et al., 2019) to build a 3D object detector for autonomous cars that achieved state-of-the-art results.

EfficientDet (Tan et al., 2020) applies weighted bi-directional FPN and compound scaling to the EfficientNet classifier (Tan & Le, 2019), resulting in a highly computationally efficient detector. Given the principal goal of detecting cones in real-time, we selected EfficientDet-D0 which is the lightest and most efficient among the 9 EfficientDet models.

5. Experimental Results

5.1. Experimental Setup

The general approach of the experiments is to train the five detectors on sunny weather to then test them on real adverse conditions. After that, we retrain them on fake adverse conditions then test them again on real adverse conditions. The goal is to record the difference in performance between models trained on (real) sunny condition data versus models trained on (fake) adverse conditions. If the latter proved to be better, then we would have demonstrated that our (synthesised) datasets are an effective proxy for (real) adverse condition data in the domain of autonomous racing. Furthermore, experiments aim to analysis the response of detectors to augmentation and identify the detector with best response. Additionally, the inference time of detectors is recorded to identify their suitability to real-time operation.

Our experiments were divided into four stages. Initially, all five detectors were trained on **sunny weather** data (Table 1, first row). This set the baseline against which subsequent models were compared. Next, the detectors were trained on **Sunny and Fake Night** (Table 1, second row). Third, they were trained on **Sunny and Fake Droplet** data (third row). Finally, they were trained on **Sunny, Fake Night and Fake Droplet** data (fourth row). In each stage the trained detectors were tested on **Sunny**, **Real Night**, and **Real Droplet** data (Section 3.1). The results are reported in Table 1, while a performance comparison across stages is shown in Figure 5.

In all our experiments the training was conducted on a GPU machine equipped with 4 Nvidia GTX 1080 GPUs with 12GB VRAM each. To perform fair comparisons between all five object detectors we fixed the value of several hyperparameters, including number of epochs (to 300), batch size (to 64), and input image size (to 512×512 pixels). For testing, we used an in-car PC mounted on a racing car equipped with a single Nvidia GTX 1050 TI GPU, with a batch size of 1 and an image size of 512×512 pixels.

Performance was measured using the standard mean Average Precision (mAP), with an Intersection over Union (IoU) threshold of 0.5 (Table 1). In addition, we measured computational efficiency for each detector by recording its floating point operations per second (FLOPS) and inference time in frames per second (fps) (last two rows of Table 1).

5.2. Impact of Synthetic Adverse Conditions

Detector performance (mAP) over the 60 experiments discussed in Section 5.1 is reported in Table 1. Figure 5 shows the difference in object detection performance of the models

(a) Before training on fake night and fake droplet (mAP=0.22)

(b) After training on fake night and fake droplet (mAP=0.74)

Figure 4. Impact of augmentation on night detections using YOLOv5

trained on synthetic adverse conditions dataset against that of their baseline versions (trained only on Sunny weather). Figure 5a compares the models trained upon Fake Night to their corresponding baseline (1st and 2nd rows in Table 1). It is clear that, after training the detectors on Fake Night, their performance on Real Night improves by an average of 32 percentage points (pp). Perhaps surprisingly, training on Fake Night not only improved the detectors' performance on real night, but also on Real Droplet by an average of 4.5 pp. A possible reason for this is that by introducing Fake Night in the training dataset, the models were trained to infer more diverse images that are out-of-sample of the original Sunny Weather training data.

Figure 5b compares the models trained upon Fake Droplet to their respective baseline (1st and 3rd rows in Table 1). When trained on Fake Droplet, YOLOv3, YOLOV5 and Faster-RCNN show improvements on Real Droplet, and just like the previous stage, they also show improvements on the other real adverse condition (Real Night). Counterintuitively, training on Fake Droplets slightly *reduced* performance in Real Droplet conditions and *improved* detections in Real Night using YOLOv4 and EfficientDet.

Finally, Figure 5c shows the difference between The 'monolithic' versions of the models (those trained on Sunny, Fake Night and Fake Droplet) and their related baseline (1st and 4th rows in Table 1). The 'monolithic' versions exhibit slightly better performance on real adverse conditions than those trained on a single fake adverse condition. When tested on Real Night, the monolithic models (excluding EfficientDet) show a mean performance improvement of 42.7

| Data splits | | Detection model mAP | | | | |
Training set	Test set	YOLOv3	SYOLOv4	YOLOv5	FRCNN	EfficientDet
Sunny	Sunny	0.89	0.93	0.95	**0.97**	0.45
	Real Night	**0.43**	0.21	0.22	0.23	0.24
	Real Droplet	0.53	0.54	0.53	**0.57**	0.43
Sunny + Fake Night	Sunny	0.89	0.89	**0.97**	0.96	0.47
	Real Night	0.62	0.39	**0.73**	0.67	0.55
	Real Droplet	0.57	0.55	0.58	**0.60**	0.53
Sunny + Fake Droplet	Sunny	0.89	0.88	**0.96**	0.95	0.36
	Real Night	**0.57**	0.33	0.25	0.30	0.26
	Real Droplet	0.54	0.51	0.55	**0.62**	0.36
Sunny + Fake Night + Fake Droplet	Sunny	0.89	0.86	**0.97**	**0.97**	0.37
	Real Night	0.68	0.66	**0.74**	0.71	0.25
	Real Droplet	0.58	0.55	**0.61**	0.60	0.37
Speed (fps) ↑		**74.63**	8.77	13.68	3.01	26.53
FLOPS (B) ↓		12.9	109	16.4	180	**2.5**

Table 1. Detailed comparative analysis of mAP for all five object detectors (tested on Sunny, Real Night, and Real Droplet) when trained on four different training sets. The aim of that is to show the difference of detectors' performance on real adverse conditions when trained on fake adverse condition against that when trained on sunny condition only. mAP is measured at a standard IoU threshold of 0.5. The plus (+) shows the concatenation of real and augmented data for training. The best model of each training-testing stage (each row) is bolded. Frame rate (in fps) and FLOPS (in billions) comparison of all five object detectors, tested using an AV's on-board PC with GTX 1050 TI.

pp, while the models trained only on Fake Night show a mean increase of 32.7 pp. In Real Droplet conditions, the monolithic models show a mean increase of 4.4 pp, while the models trained only on Fake Droplet show a slight degradation of 1.2 pp. Figure 4 show the qualitative results of YOLOv5 night detections before and after being trained on synthetic night.

The average improvements on all conditions for night is higher than that for droplets. This is due to the baseline models having lower performance on Fake Night than on Fake Droplets, so there was more room for improvement in the former. Furthermore, this agrees with the fact that the synthetic Fake Night data is more visually similar to the Real Night data than the Fake Droplet data is to Real Droplet data (as the Fake Night dataset has a lower FID than the Fake Droplet data (Section 3.3)). This intuitively might be due to the fact that night images, by nature, are captured with lower reflective light than images taken during daytime, resulting in images that have lower number of colours and details than images captured during the day. This, in turn, makes it easier for generative models like CycleGAN to generate them because, again, they have fewer colours and details than day-time images.

5.3. Detector Behaviour Using Synthetic Adverse Conditions

When models are trained on augmented (Fake) datasets, their performance difference on Real Sunny varies. the performance of YOLOv3 and YOLOV5 experienced marginal (negligible) improvement or degradation. However, ScaledYOLOv4 and EfficientDet performance was slightly degraded.

The different detection models use a varied architecture for the neck structure (which is placed between the head and backbone to aggregate as much information extracted by the backbone as possible before it is fed to the head) - YOLOv3 and YOLOV5 use a Path Aggregation Network (PANet) (Liu et al., 2018) as a neck and they exhibit a slight performance enhancement, while ScaledYOLOv4, FRCNN, and EfficientDet use FPN (Lin et al., 2016) and they generally show a degradation of performance. The results suggest that detectors with an FPN neck are more susceptible to augmentations degrading good weather performance than those using PANet.

Among the five detectors, YOLOv5 is the most robust in all weather conditions: trained on augmented data, it reported an improvement of 1.3, 52.1, and 7.6 pp on real sunny, real night, and real droplet respectively. On the other hand, EfficientDet performed the worst after augmentations, exhibiting a performance degradation of -8.4 and -5.5 pp for Sunny and Real Droplet and an improvement of +0.5 on Real Night.

5.4. Object Detectors' Performance

Looking at Table 1, it is observed that, in the absence of adverse weather training data (i.e., when only the sunny weather training dataset is available), Faster-RCNN is the most accurate model for sunny and rainy weather. However, YOLOv3-tiny is arguably the optimal model for this application, as object detection performance is only marginally reduced from that of Faster-RCNN in sunny and rainy conditions, but exhibits double the mAP in night-time tests - while simultaneously being 25 times faster and 18 times more computationally efficient than Faster-RCNN.

In contrast, when synthetic adverse condition training data is available, YOLOv5 and FRCNN appeared to be the best choices in terms of accuracy and robustness in all weather conditions. After training on synthetic adverse conditions (fake night and fake droplet), they scored the highest mAP across the board when tested on sunny weather, real night, and real droplet. YOLOv5 is also capable of achieving these results while running at a high frame rate of almost 14 fps.

In terms of raw computational efficiency alone (last row of Table 1), EfficientDet is the best model among the five detectors. It has the lowest number of floating point operations per second, which makes it ideal for low computational capacity hardware. Although EfficientDet has (5 times) lower FLOPS than YOLOv3, the latter is (2.8 times) faster than the former. This is because the term FLOPS describes the number of operations **per second** not the total number of operations. In other words, EfficientDet might have more operations in total than YOLOv3, but it divides and executes them over a longer time span (thus slower), ending up with a small number of operations per second. Having a lower FLOPs means the model is hardware-friendly and can be installed on relatively light processors including the popular NVIDIA Jetson. However, EfficientDet performance is decreased in almost all weather conditions with the use of augmented training data - suggesting it is a poor choice when using augmented training data in adverse conditions.

Furthermore, Table 1 in terms of inference speed, YOLOv3 is the fastest model by a clear margin - triple the speed of EfficientDet, and an order of magnitude faster than SYOLOv4 and Faster-RCNN. Interestingly, the number of frames it can process per second is more than the summation of frames processed by all the other models combined. While running at this high speed, YOLOv3 achieved a similar performance to that of YOLOv5. Therefore, in case a lower powered processor hardware is to be used or a high frame rate is critical, YOLOv3 is the optimal choice. It is worth mentioning that the frame rates in Table 1 were measured while running the detectors on the AV racing PC mentioned in section 5.1 which (at the time of writing) is 4 years old. State-of-the-art on-board hardware will further improve the performance of all detectors.
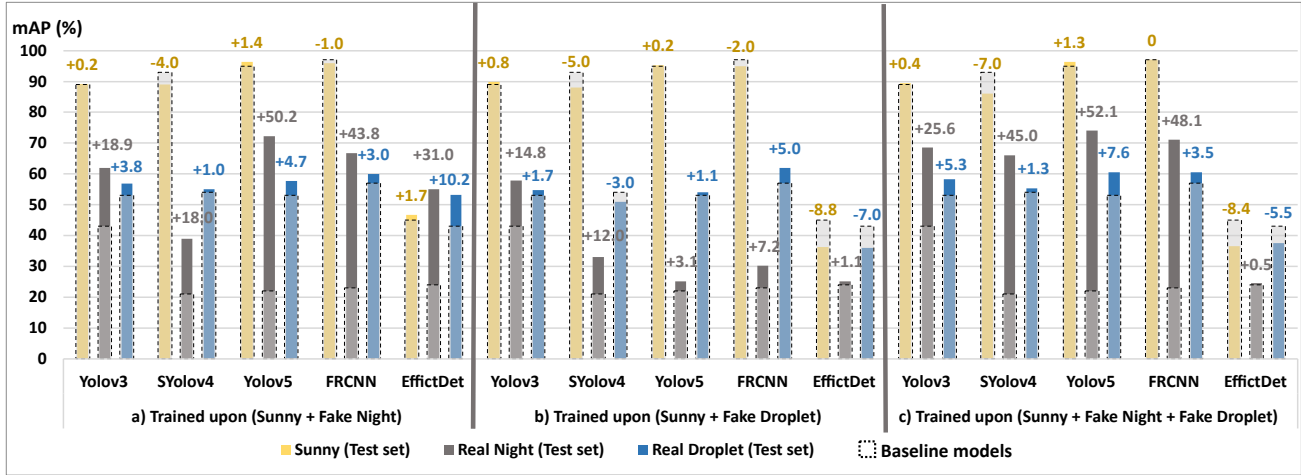
Figure 5. Dashed bars show the baseline performance (mAP) of 5 models trained on **Sunny** and tested on Sunny, Real Night, and Real Droplet (1st row of Table 1). The coloured continuous bars and coloured numbers show the improvement (+ve value) or degradation (-ve value) of performance caused by retraining the models on **a) Sunny+Fake Night, b) Sunny+Fake Droplet, c) Sunny+Fake Night+Fake Droplet**, (2nd, 3rd, and 4th rows of Table 1).

6. Conclusions and Future Work

CycleGAN-based style transfer can be efficiently used to generate synthetic *(fake)* adverse condition data that can, in turn, be used to improve the performance of object detection models for autonomous racing in *real* adverse conditions. In this paper, training on synthetic data has improved the performance of four out of five state-of-the-art detection models by 42.7 and 4.4 percentage points when tested on *real night* and *real droplet* frames, respectively. Behaviour of detectors against augmentations varies; out of the five detectors, EfficientDet was the worst while YOLOv5 has emerged as arguably the optimal one as it managed to achieve high mAP of 97%, 74% and 61% for sunny, night and droplet respectively - while being very efficient, making it ideal for autonomous racing applications. It is noted that the models with PANet neck had positive response to augmentations than those with FPN neck. In applications where speed is of the essence, YOLOv3 is recommended as it stands as the fastest detection model.

An interesting area for further exploration is the influence of the various neck structures and feature extraction backbones employed in different object detectors, in order to truly understand and optimise the performance of object detection models for applications requiring the use of augmented training data.

Results suggest that a carefully selected combination of synthetic weather training data, and an appropriate object detector, can effect a significant improvement in object detection accuracy in adverse weather. Although further work in this area is required, it is believed that these findings will be applicable to on-road autonomous driving scenarios.

All the analysed detectors are CNN-based. However, in future, Transformer-based detectors might also be analysed and compared.

References

Badue, C., Guidolini, R., Carneiro, R. V., Azevedo, P., Cardoso, V. B., Forechi, A., Jesus, L., Berriel, R., Paixão, T. M., Mutz, F., de Paula Veronese, L., Oliveira-Santos, T., and De Souza, A. F. Self-driving cars: A survey. *Expert Systems with Applications*, 165:113816, 2021. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2020.11 3816. URL https://www.sciencedirect.com/ science/article/pii/S095741742030628X.

Bansal, M., Krizhevsky, A., and Ogale, A. S. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *CoRR*, abs/1812.03079, 2018. URL http: //arxiv.org/abs/1812.03079.

Benjumea, A., Teeti, I., Cuzzolin, F., and Bradley, A. Yoloz: Improving small object detection in yolov5 for autonomous vehicles, 2021.

Bernard, E., Riviere, N., Renaudat, M., Guiset, P., Pealat, M., and Zenou, E. Experiments and models of active and thermal imaging under bad weather conditions. In Kamerman, G. W., Steinvall, O. K., Bishop, G. J., and Gonglewski, J. D. (eds.), *Electro-Optical Remote Sensing, Photonic Technologies, and Applications VII; and Military Applications in Hyperspectral Imaging and High Spatial Resolution Sensing*, volume 8897, pp. 14 – 25. International Society for Optics and Photonics, SPIE, 2013.

URL https://doi.org/10.1117/12.202897 8.

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection, 2020.

Cai, Y., Luan, T., Gao, H., Wang, H., Chen, L., Li, Y., Sotelo, M. A., and Li, Z. Yolov4-5d: An effective and efficient object detector for autonomous driving. *IEEE Transactions on Instrumentation and Measurement*, 70: 1–13, 2021. doi: 10.1109/TIM.2021.3065438.

CBI Insights. Autonomous Vehicles & Car Companies l CB Insights, 2020. URL https://www.cbinsigh ts.com/research/autonomous-driverles s-vehicles-corporations-list/.

Creus, C. and Patow, G. A. R4: Realistic rain rendering in realtime. *Computers & Graphics*, 37(1):33–40, 2013. ISSN 0097-8493. doi: https://doi.org/10.1016/j.cag.20 12.12.002. URL https://www.sciencedirect. com/science/article/pii/S00978493120 01781.

Culley, J., Garlick, S., Gil Esteller, E., Georgiev, P., Fursa, I., Vander Sluis, I., Ball, P., and Bradley, A. System design for a driverless autonomous racing vehicle. In *2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, pp. 1–6, 2020. doi: 10.1109/CSNDSP49049.2020.9249626.

Dhall, A., Dai, D., and Van Gool, L. Real-time 3D Traffic Cone Detection for Autonomous Driving. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2019-June:494–501, feb 2019. URL https://arxiv.org/abs/ 1902.02394v2.

Espineira, J. P., Robinson, J., Groenewald, J., Chan, P. H., and Donzella, V. Realistic lidar with noise model for real-time testing of automated vehicles in a virtual environment. *IEEE Sensors Journal*, 21(8):9919–9926, 2021. doi: 10.1109/JSEN.2021.3059310.

Fursa, I., Fandi, E., Musat, V., Culley, J., Esteller, E. G., Teeti, I., Bilous, L., Sluis, I. V., Rast, A., and Bradley, A. Worsening perception: Real-time degradation of autonomous vehicle perception performance for simulation of adverse weather conditions. *CoRR*, abs/2103.02760, 2021. URL https://arxiv.org/abs/2103.0 2760.

Gu, Y., Wang, Q., and Qin, X. Real-time streaming perception system for autonomous driving, 2021.

Gupta, A., Anpalagan, A., Guan, L., and Khwaja, A. S. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array*, 10:100057, 2021. ISSN 2590-0056. doi: https:

//doi.org/10.1016/j.array.2021.100057. URL https: //www.sciencedirect.com/science/arti cle/pii/S2590005621000059.

International Society of Automotive Engineers (SAE). J3016B: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International, jun 2018. URL https: //www.sae.org/standards/content/j301 6_201806/.

Jocher, G. Yolov3. *Code repository https://github. com/ultralytics/yolov3*, 2021.

Jocher, G., Nishimura, K., Mineeva, T., and Vilariño, R. Yolov5. *Code repository https://github. com/ultralytics/yolov5*, 2020.

Katrakazas, C., Quddus, M., Chen, W.-H., and Deka, L. Real-time motion planning methods for autonomous onroad driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60:416–442, 2015. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2015.09.011. URL https://www.sciencedirect.com/scienc e/article/pii/S0968090X15003447.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, P., Chen, X., and Shen, S. Stereo r-cnn based 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Lin, C.-T., Huang, S.-W., Wu, Y.-Y., and Lai, S.-H. Ganbased day-to-night image style transfer for nighttime vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2020. doi: 10.1109/TITS.2 019.2961679.

Lin, T., Dollár, P., Girshick, R. B., He, K., Hariharan, B., and Belongie, S. J. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016. URL http: //arxiv.org/abs/1612.03144.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. Microsoft coco: Common objects in context, 2015.

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. Path aggregation network for instance segmentation, 2018.

Mohammed, A. S., Amamou, A., Ayevide, F. K., Kelouwani, S., Agbossou, K., and Zioui, N. The perception system of

intelligent ground vehicles in all weather conditions: A systematic literature review. *Sensors (Basel, Switzerland)*, 20, 2020.

Muṣat, V., Fursa, I., Newman, P., Cuzzolin, F., and Bradley, A. Multi-weather city: Adverse weather stacking for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 2906–2915, October 2021.

Olkin, I. and Pukelsheim, F. The distance between two random vectors with given dispersion matrices. *Linear Algebra and its Applications*, 48:257–263, 1982. ISSN 0024-3795. doi: https://doi.org/10.1016/0024-3795(82)90112-4. URL https://www.sciencedirect.com/science/article/pii/0024379582901124.

Ostankovich, V., Yagfarov, R., Rassabin, M., and Gafurov, S. Application of cyclegan-based augmentation for autonomous driving at night. *2020 International Conference Nonlinearity, Information and Robotics (NIR)*, pp. 1–5, 2020.

Porav, H., Maddern, W., and Newman, P. Adversarial training for adverse conditions: Robust metric localisation using appearance transfer. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1011–1018, 2018. doi: 10.1109/ICRA.2018.8462894.

Porav, H., Musat, V.-N., Bruls, T., and Newman, P. Rainy screens: Collecting rainy datasets, indoors. *ArXiv*, abs/2003.04742, 2020.

Praveen, K., Mukherjee, J., and Madumbu, V. Adverse weather simulation for training neural networks. In *2017 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pp. 107–110, 2017. doi: 10.1109/ICCE-ASIA.2017.8309327.

Qie, L., Gong, J., Zhou, H., Wang, S., Zhou, S., and Chetan, N. B. Cone detection and location for formula student driverless race. *2019 6th International Conference on Dependable Systems and Their Applications (DSA)*, pp. 440–444, 2020.

Redmon, J. and Farhadi, A. Yolov3: An incremental improvement, 2018.

Redmon, J., Divvala, S., Girshick, R. B., and Farhadi, A. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

Shah, S., Dey, D., Lovett, C., and Kapoor, A. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *FSR*, 2017.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Stevenson, J. A chilly winter night ride through london. https://www.youtube.com/watch?v=8FG4ueprxqY, January 2019.

Strobel, K. L., Zhu, S., Chang, R., and Koppula, S. Accurate, low-latency visual perception for autonomous racing: Challenges, mechanisms, and practical solutions. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1969–1975, 2020.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105–6114. PMLR, 2019.

Tan, M., Pang, R., and Le, Q. V. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781–10790, 2020.

TIAN, H., NI, J., and HU, J. Autonomous driving system design for formula student driverless racecar. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–6, 2018. doi: 10.1109/IVS.2018.8500471.

Uricar, M., Ulicny, J., Sistu, G., Rashed, H., Krizek, P., Hurych, D., Vobecky, A., and Yogamani, S. Desoiling dataset: Restoring soiled areas on automotive fisheye cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

Uřičář, M., Křížek, P., Sistu, G., and Yogamani, S. Soilingnet: Soiling detection on automotive surround-view cameras. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 67–72. IEEE Press, 2019. doi: 10.1109/ITSC.2019.8917178. URL https://doi.org/10.1109/ITSC.2019.8917178.

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. Scaled-YOLOv4: Scaling cross stage partial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13029–13038, June 2021.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

Yi, Z., Yongliang, S., and Jun, Z. An improved tiny-yolov3 pedestrian detection algorithm. *Optik*, 183:17–23, 2019. ISSN 0030-4026. doi: https://doi.org/10.1016/j.ijleo.2019.02.038. URL https://www.sciencedirect.com/science/article/pii/S003040261930155X.

Zadok, D., Hirshberg, T., Biran, A., Radinsky, K., and Kapoor, A. Explorations and lessons learned in building an autonomous formula sae car from simulations. *ArXiv*, abs/1905.05940, 2019.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251, 2017. doi: 10.1109/ICCV.2017.244.