An Empirical Study on Cross-Data Transferability of Adversarial Attacks on Object Detectors

Alexander Michael Staff[1], Jin Zhang[1], Jingyue Li[1], Jing Xie[2], Elizabeth Ann Traiger[3], Jon Arne Glomsrud[3] and Kristian Bertheussen Karolius[3]

[1]*Computer Science Department, Norwegian University of Science and Technology, Trondheim,7050, Norway*
[2]*Tidal, Square, Oslo, 0187, Norway*
[3]*Group Technology and Research, DNV GL, Høvik, 1363, Norway*

Abstract

Object detectors are increasingly deployed in safety-critical systems, including autonomous vehicles. Recent studies have found that object detectors based on convolutional neural networks are fundamentally vulnerable to adversarial attacks. Adversarial attacks on object detectors involve adding a carefully chosen perturbation to the input, which causes the object detector to make mistakes. The potential consequences of adversarial attacks must be known to make sure these safety-critical systems are reliable. This paper investigates the influence of transfer attacks on object detectors, where the attacker does not access the target detector and its training set. Devising an attack with this assumption requires the attacker to train their model on data that resembles the target detector's training set. Using their model as a surrogate, attackers can generate adversarial attacks without accessing the target detector. Our study investigates whether one can effectively attack a black box model using publicly available data. We have performed targeted objectness gradient attacks on the state-of-the-art object detector (i.e., YOLO V3). Initial transferability between the attacking and target model is low. However, increasing attack strength from 8 to 24 strengthens transferability and reduces the target detector performance by about half. Transferability is also studied when the datasets for the attacking and the target model intersect. Attack performance is proportional to the size of the intersection. With the stronger transferability caused by intersecting datasets, attack strength can be dropped to 16 and retain the attack performance.

Keywords

Object detector, adversarial examples, transfer attacks, targeted objectness gradient(TOG) dataset intersection,

1. Introduction

Object detection and classification have seen a rapid improvement since the ImageNet competition in 2012, where the introduction of deep convolutional networks almost halved the error rates of the best competing approaches [1], causing a paradigm shift in the field of computer vision. With the increased performance, object detectors have seen wide deployment in a range of areas, including safety-critical applications like autonomous vehicles [2]. However, despite

AI-CyberSec 2021: Workshop on Artificial Intelligence and Cyber Security, December 14, 2021, Cambridge, UK

alexandermstaff@gmail.com (A. M. Staff); jin.zhang@ntnu.no (J. Zhang); jingyue.li@ntnu.no (J. Li); jxie@squareup.com (J. Xie); elizabeth.traiger@dnv.com (E. A. Traiger); jon.arne.glomsrud@dnv.com (J. A. Glomsrud); kristian.bertheussen.karolius@dnv.com (K. B. Karolius)
https://github.com/alexmstaff/ (A. M. Staff)
0000-0003-3036-3075 (J. Zhang)

the increased performance, it has been proven that object detectors can misdetect the targets if imperceptible non-random perturbations are added to the input. These inputs, first described in [3] are called adversarial samples. It turns out that all neural networks are fundamentally susceptible to making critical errors if exposed to these adversarial samples. Furthermore, it became apparent that adversarial samples that emerged to attack a particular model were even effective on models with entirely different architectures [4, 5].

Object detectors are different from classifiers because they are multi-task learners performing a more complex task than classifiers. Recently, attacks have been developed that target object detectors directly [5, 6, 7]. However, there is a significant research gap on cross-data blackbox attacks, where the attacker cannot use the same dataset as the target detector. This scenario is essential, as it is unlikely that an attacker would have access to the dataset used to train an autonomous commercial vehicle.

This study aims to find the level of exposure one can expect if an attacker tries to attack a model they have no direct access to and do not know essential model details. We investigate whether it is possible to transfer attacks across models trained on different data, i.e., models are trained to detect the same semantic class, namely boat, with various datasets. In addition, the effect on attack performance when the surrogate and target models are trained on datasets with intersections is investigated. We trained some models on datasets that share images, while others were trained on entirely disjoint datasets. The adversarial samples are generated using a source model trained to detect ships on one dataset. Then the target model trained to detect ships on a separate dataset attempts to make predictions on the sample. We employ Targeted Objectness Gradient (TOG) attack [5] to generate adversarial samples.

The results show that perturbations generated with an epsilon=8 of the TOG attack under the distance metric L_∞ have poor transference between the models. However, an epsilon in the 20-30 range results in better transference and more effective attacks. Models trained on intersecting datasets show transference which is proportional with the size of the intersection. Performing a cross-resolution attack revealed that an attacker who does not know the resolution of the target detector should choose a lower resolution like 416×416 pixels to raise the probability of the attack being upsampled rather than downsampled.

Our evaluation demonstrates the potential consequences of cross-data transfer attacks on object detectors. We provide the performance results for models trained with dataset intersections, models without dataset intersections, and several high-performance single-class object detection models.

The remainder of the paper is organized as follows: Section 2 introduces background and terminology related to adversarial attacks to object detectors and summarizes related work. Section 3 describes the research design, and section 4 reports the experiment results. Section 5 discusses some insights about transfer attacks on object detectors. Finally, section 6 concludes the study.

2. Background

Object detection is a computer vision task that takes an image as input and gives bounding boxes with class labels as output [8]. To make real-time predictions one-stage object detectors are developed, such as Single Shot MultiBox Detector (SSD) [9] and You Only Look Once (YOLO) [10]). One-stage detectors predict bounding boxes and object classes at the same time. This section provides a brief overview of the TOG attacks developed to target real-time object detectors and the transferability of attacks.

2.1. Targeted Objectness Gradient (TOG) attacks

The TOG attacks, developed by Chow *et al.* [5], are a class of adversarial attacks explicitly developed to target object detectors. The attacks use an iterative gradient method to cause object detectors to make mistakes. For example, with detection expressed as $\mathcal{O}^*(x)$ and attack loss as $\mathcal{L}^*$, TOG can be expressed as:

$$x^{'}_{t+1} = \prod_{x,\epsilon}[x^{'}_{t} - \alpha_{TOG}\Gamma(\nabla_{x^{'}_{t}}\mathcal{L}^*(x^{'}_{t}, \mathcal{O}^*(x); \theta))], \tag{1}$$

where $x^{'}_{t}$ is the adversarial sample at the t-th iteration, Γ is the sign function, and α_{TOG} is the attack learning rate. Further details can be found in [5]. Using this TOG can find adversarial perturbations targeting different functions of object detectors. The *untargeted* attack is successful when it causes the detector to misdetect, such as failing to detect misclassifying or fabricating an object. The *object-vanishing* attack seeks to suppress detections and is successful when the detector finds no objects which actually exist in the sample. The *object-fabrication* makes the detector hallucinate multiple objects, making the prediction useless. Finally, the *targeted object-mislabeling* attack can cause the detector to mislabel detected objects with the chosen target class while maintaining the correct bounding boxes [5, 11]. This attack is less relevant to this paper as most of the models being studied are single class models.

2.2. Transfer Attacks

Transfer attacks are adversarial attacks with different source and target models. In this context, different models mean that they have different model architectures or different training sets. Even generating the adversarial sample at a different resolution than the detection operates at can be seen as a transfer attack. Transfer attacks rely on the fact that even though two models have been trained on different datasets to detect the same objects, it can be assumed that the two models share commonalities. These can be exploited through adversarial attacks when the attacker lacks access to the target model. Normally, to generate adversarial examples, one needs to give the model input and read the output. If one lacks access to the model, one technique is to train a stand-in model (the source model) to make similar predictions to the target model. Then adversarial samples generated using the source model will also affect the target model [5, 7]. For instance, [12] develops a cross-architecture transferable attack, called *relevance attack on detectors* by using a YOLO model as a surrogate model to attack other object detectors. [4] develops an attack that generates universal perturbations which can transfer across tasks, models, and datasets.

3. Research design

This section explains the research strategy of data preparation, attack scenario considered, and performance analysis metrics.

3.1. Data preparation

YOLO uses a fixed internal resolution and resizes the images before training and detection. This results in changing the appearance of objects in the image. When one trains the model on a dataset with the same aspect ratio as the images used to test it, this effect disappears as it learns how objects look when they are warped. The issue arises when detecting images of an aspect ratio not seen in the training set, and in this case recall and confidence drop.

In this study, we use four data sources focused on autonomous ships. The images shared from [13] are compiled to train an object detector for the ReVolt, which is a conceptual autonomous ship that has been developed by DNV[1] since 2014[2]. Most images are 1280×720 and are captured in Høvik, Norway. These images are also largely of smaller craft and kayaks. These images were re-annotated to only buildings and boats to ensure consistent labeling. Figure 1 shows a sample of the images from [13].

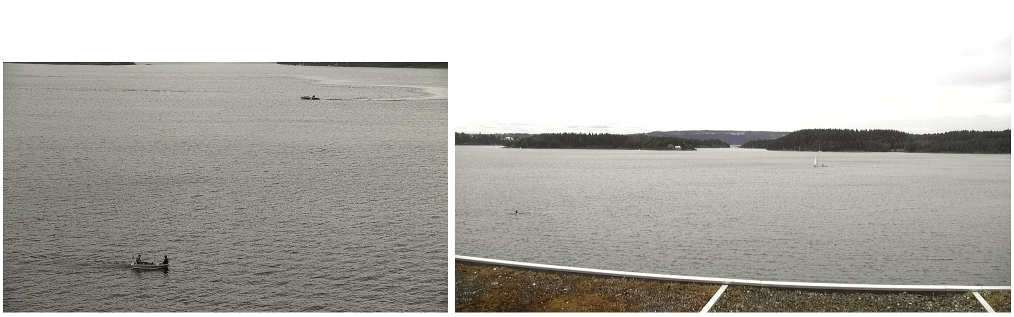


Figure 1: Sample images from the dataset shared from [13]

The Hurtigruten data is shared from DNV and contains images captured from the bow of the Hurtigruten ferry as it travels up the Norwegian coast. These images are captured at a resolution of 1920×1080. The watercrafts in the images are largely small to medium-sized pleasure craft. Figure 2 shows a sample of the images from DNV. Like the images from [13], most of the crafts will have water as the background.

The third dataset is Singapore Maritime Dataset [14], in which images were collected in the Singapore harbour. These images are dominated by large freight ships.

The fourth data source used is the Common Objects in Context (COCO) [15] dataset. The COCO dataset is a large dataset used to benchmark and rank object detectors. It consists of over 200000 labeled images divided into 80 categories. As the name implies, a focus for the COCO dataset is presenting the objects in context. For the images in the boat class, this results

[1] DNV is a world-leading classification society and the independent expert in assurance and risk management.

[2] Webpage: https://www.dnv.com/technology-innovation/revolt.

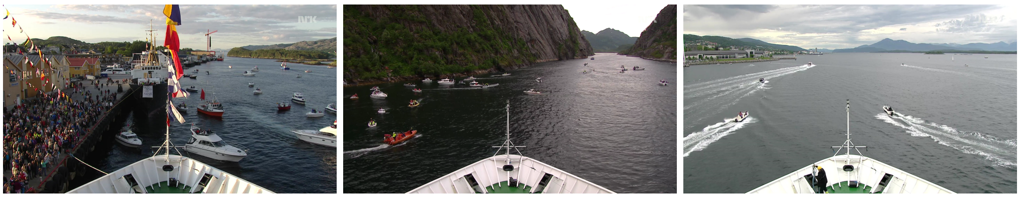


Figure 2: Sample images from the dataset shared from DNV

in more watercrafts on the beach or docked in the harbour than is found in the other datasets. In addition, the resolution of the images is not uniform, and the images are highly varied in lighting, composition, and environment. These attributes make it much harder for an object detector to achieve a high mAP on the COCO dataset.

Table 1
Data sources with their description

Data Source	Description	Size
Grini data	Images shared from [13]. Re-annotated for this paper, two classes: Buildings, boats.	622
Hurtigruten	Data shared from DNV. Collected from the show "Hurtigruten minute by minute". Initially separate classes for different watercraft, but these are are combined to a single "boat" class for this paper.	3740
SMD	The Singapore Maritime Dataset [14] Consists of a set of high definition videos taken in the Singapore harbour. Initially separate classes for different watercraft, but these are combined to a single "boat" class.	3590
COCO	The COCO [15] dataset consists of over 200K labeled images divided into 80 classes.	5123
Total		13075

3.2. Attack Scenario

The attack scenario considered in this paper is modeled to establish a worst-case scenario for an autonomous ferry under attack where the attacker does not have access to the target model or critical information about the model. The attacker can apply adversarial noise directly to the input but cannot generate adversarial samples using the target model. The attacker uses the same architecture as the target model. This effectively establishes a worst-case scenario as no cross-model attack could be stronger than the attack with the same source and target architecture. Additionally, there are not many different real-time object detection architectures suitable for autonomous vehicles. YOLO is the best performing architecture on the COCO dataset, which makes it reasonable for an attacker to guess that a YOLO model might be used.

3.3. Performance Analysis metrics

For an image classifier or object detector, we use precision to measure how accurate the predictions are. i.e., the percentage of the predictions is correct. Recall measures how well the model finds all the positives. To better view model performance, one can plot a precision-recall curve at different confidence thresholds. Calculating the area under the precision-recall curve gives average precision.

$$AP = \sum_n (R_n - R_{n-1}) P_n,$$

where R_n and P_n represent recall and precision at nth threshold respectively. Since average precision is generated for each query, one can take the mean to get a single number that represents the models performance on the whole dataset.

$$mAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q},$$

where Q is the number of queries. One of the reasons mAP became an essential metric for object detection is that object detection challenges like PASCAL VOC [16] and COCO [15] rank object detectors by mAP. In this paper, mAP is the primary metric to measure model performance.

3.4. Research questions

To demonstrate the attack transferability, we ask the following two research questions:

Research Question 1 (RQ1): Is it possible to transfer black box attacks across models and particularly models trained on different data?

Research Question 2 (RQ2): If the answer to RQ1 is yes, how does the image sharing of the dataset affect the transference of the attack?

RQ1 is based on the adversarial sample performance of the models compared to the clean sample performance. First, models are trained using YOLO - Darknet as this implementation produces real-time object detectors with leading performance [17]. One model is trained purely on private data to enable black box attacks. Other models are trained on various customized datasets based on publicly available data. Then the performance of adversarial attacks using these models is measured in mAP. Adversarial samples are generated with different combinations of source and target models. Attack performance is inferred from the drop in mAP between clean and adversarial sets.

RQ2 is based on attack performance between the various public models. As these models are trained on various compositions of public datasets, comparisons can be drawn between the intersection of the datasets and the performance of the adversarial samples. The intersection between the datasets is measured as the number of images that exist in both datasets, divided by the size of the dataset. This is similar to Intersection Over Mean (IOU), but importantly, IOU is a symmetric measure, i.e., $IOU(A, B) = IOU(B, A)$, while the measure we use reflects the difference between attacking a model trained on a subset of the model's own training set, and attacking a model trained on a superset of the model's training set.

4. Results of the empirical study

This section presents the results of RQ1 and RQ2. The composition of datasets is detailed, and the performance of black box attacks on YOLO using the TOG family of attacks is reported. The initial performance of models on clean samples forms a baseline to compare the adversarial attacks with. All mAP data is mAP .50. All the code used for generating adversarial samples can be found here: https://github.com/alexmstaff/TOG/.

4.1. Results of RQ1

To demonstrate black box attacks and answer RQ1, there should be two disjointed datasets, meaning no images appearing in both datasets. Furthermore, the performance between the models should be as similar as possible. This is due to the assumption that models with similar performance have higher attack transferability.

Model performance before attack Data sources in Table 1 were compiled throughout this study. The central goal was to obtain a dataset that would train a representative object detector for autonomous ships. Additionally, another dataset was needed to train the surrogate model, which would enable black box attacks. The assembled datasets for our study are shown in Table 2. Numbers in Table 2 include only images in the training set, images in the validation set are excluded, and models with no intersections are highlighted in green.

Table 2
Assembled datasets with their composition. Datasets with no intersections are highlighted in green.

Dataset ID	Source Datasets	No. Images	Total Size
D1	Grini data	497	3048
	Hurtigruten	2551	
D2	COCO	3025	5602
	Grini data	517	
	SMD	2060	
D3	Grini data	517	4107
	SMD	3590	
D4	Hurtigruten	3573	3573
D5	COCO	3025	10705
	Hurtigruten	3573	
	Grini data	517	
	SMD	3590	
D6	COCO	5123	5640
	Grini data	517	

We trained six YOLO models (i.e., M1-M6) by using the assembled datasets D1-D6. In Table 3, the Validation column shows the performance of M1-M6 on the validation set associated with the training set for the model. The Benchmark column shows the performance of models on the Benchmark dataset, i.e., the validation set associated with D1. M2 and M4 are highlighted, as these models are the primary models used to answer RQ1. M2 is the highest performing model trained on public data (i.e., D2), and M4 is trained on private data (i.e., D4). Thus, D2 and D4 are two disjoint datasets. M1 does not have the same performance on the validation and benchmark datasets because the images were resized to fit YOLO's internal resolution in the benchmark column. The input image must match the model dimensions to generate adversarial samples, but annotations are not corrected for this difference.

Table 3
Performance of models on clean samples. The rows highlighted in green indicate models trained with disjoint datasets.

Model ID	Dataset	Validation	Benchmark
M1	D1	47.96	45.75
M2	D2	78.28	26.32
M3	D3	87.06	10.36
M4	D4	70.8	68.29
M5	D5	85.35	74.56
M6	D6	59.84	24.74

Attack Performance In the mathematical expression of TOG attacks (see Eq. 1), epsilon represents the maximum change allowed to any point using an L_∞ distance metric. For an image, epsilon represents how much one can change a single pixel with no limit on the number of pixels changed. Eight is the default value for the TOG attacks, and using this value makes the numbers comparable to the data gathered by [5]. Since a greyscale pixel can hold the values from 0 to 255, an epsilon value of 8 means no pixel is changed more than $\frac{8}{255} = 0.031$ [18].

Table 4 (a) shows the mAP achieved on the adversarial set generated by M2. The Target Model column shows which model makes detections on the adversarial set. White-box attacks, where the source and target model are the same, are highlighted in grey. Source model refers to the model used to generate the adversarial set, also called the attacking model. The Clean column is the model performance on the unperturbed but resized set. This is to ensure that the only effect being compared is the adversarial perturbation. The Untargeted column refers to the model's performance on images perturbed by the TOG untargeted attack. Similarly, the Vanishing column refers to the model's performance on the TOG Vanishing attack. An explanation of these attacks can be found in section 2.1. It is apparent from this table that transference is very low. Although the white-box is very effective, M4 only takes a minor hit to its mAP. An important aspect of these attacks is that there is also an element of cross-model transference to the attack beyond testing attack performance with different source and target models. This is a consequence of generating samples using the TensorFlow - YOLO model and detecting with the Darknet - YOLO model. Table 4 (b) shows the mAP achieved on the adversarial set generated by M4. In Table 4 (b), we see that only the untargeted attack is successful in white-box mode. The untargeted and vanishing attacks do not transfer to M2 effectively.

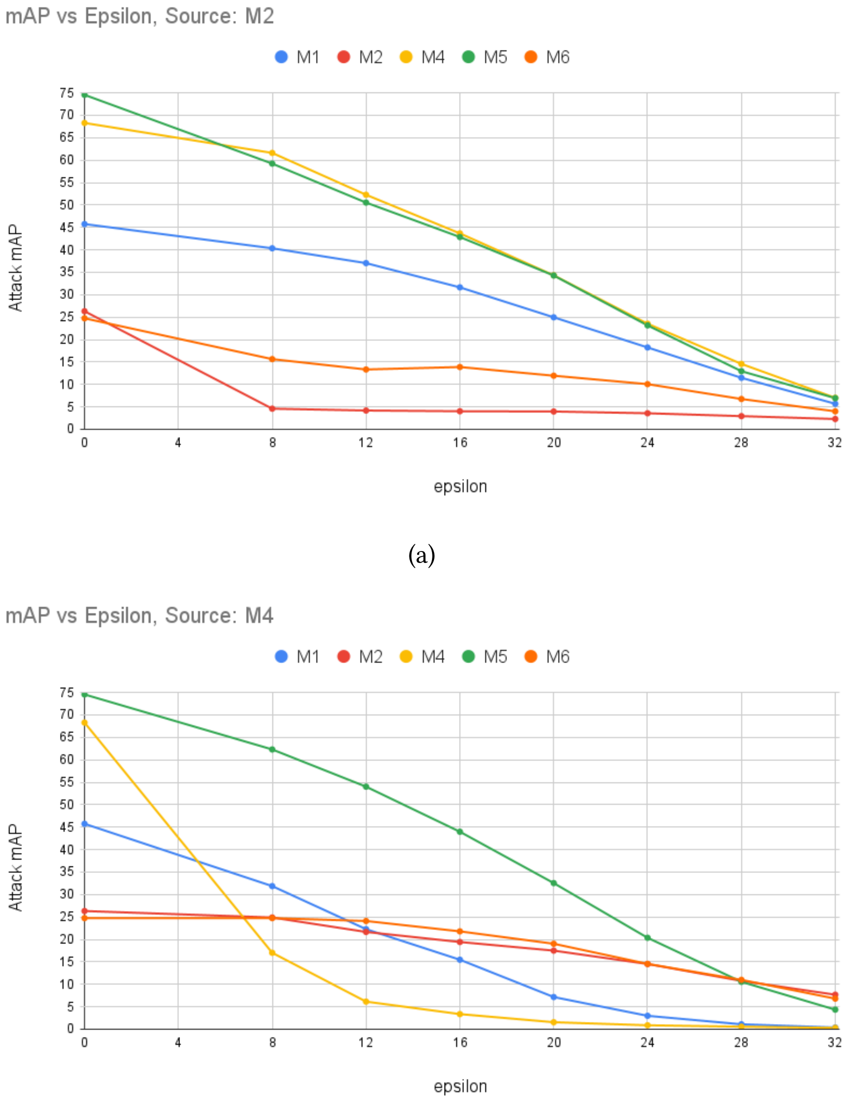
Table 4
Performance of target models on adversarial samples (a) generated by M2; (b) generated by M4

Target Model	Clean	Untargeted	Vanishing
M2	26.32	4.63	3.27
M4	68.29	61.62	63.55

(a)

Target Model	Clean	Untargeted	Vanishing
M2	26.32	24.13	25.17
M4	68.29	14.57	57.61

(b)



(a)

(b)

Figure 3: The blackbox and white-box performance plotted against epsilon. (a) The adversarial set is generated by the M2 model, (b) The adversarial set is generated by the M4 model

Attack performance compared to epsilon Figure 3 (a) shows the mAP achieved by the models on adversarial sets generated by M2 with increasing epsilon values. At the left extreme, we see the clean set performance. The most significant mAP change is from clean to 8 epsilon on M2. As this is the white-box attack, this is to be expected. An interesting observation is that though mAP is not reduced to 0, increases in epsilon do not decrease performance further. Looking at the line for M4, we see a steeper slope between 8 and 32 than between 0 and 8. This indicates that the attack transference is poor while epsilon is small, but once epsilon is large enough, the attack strength improves proportionally with epsilon. Attacking M4 effectively with M2 requires a quite high epsilon, but by epsilon=24, the mAP of M4 is reduced to less than half. On the other side, adversarial samples generated by M4 show worse transferability on M2, as presented in Figure 3 (b). Epsilon in the 20-30 range is relatively high in comparison to the white-box attacks. However, when attacking object detectors, epsilon=20 is in line with other efforts [4, 12]. The white-box attack stands out as mAP decreases much faster than for any other model.

Answer to RQ1: The black-box attacks with disjoint datasets show a low transference while epsilon is small. Increasing epsilon strengthens attack transferability.

4.2. Results of RQ2

To answer RQ2, we compare the performance of models trained with intersecting datasets to the models trained with disjoint datasets. In particular, we investigate whether there is a correlation between the relative size of the dataset intersection and the performance of the transfer attack.

Dataset Intersection Table 6 shows the intersection between the six datasets. Each row shows what percentage the intersection with the dataset on the column is of its total size. For instance, D5 has 52.33% intersection with D2. and D4 has no intersection with D2.

Table 6
The intersection between the various datasets

	D1	D2	D3	D4	D5	D6
D1	100	16.31	16.31	65.58	81.89	16.31
D2	8.87	100	46	0	100	63.23
D3	12.1	62.75	100	0	100	12.59
D4	55.95	0	0	100	100	0
D5	23.32	52.33	38.37	33.38	100	33.09
D6	8.81	62.8	9.17	0	62.8	100

Attack Performance by Intersecting Models Attack transferability varies between source models trained with different datasets. We report the attack performance of M5 and M1 as representatives and show the results in Table 7. Performance of other models is included in https://github.com/alexmstaff/TOG/. In Table 7, rows highlighted in gray indicate the performance of white box attacks. The others are the performance of black box attacks. M5 is trained on D5, which is a superset of all other five datasets. Table 7 (a) shows that M5 achieves the greatest mAP reduction on M2, i.e., of 26.32-7.31=19.01 for the untargeted attack and 16.67 mAP reduction for the vanishing attack. On average, the black box attack of M5 achieves a 16.79 mAP reduction for the untargeted attack and a 12.39 mAP reduction for the vanishing attack. M1 is the only two-class model among all six models, which shows poor attack transfer performance. In Table 7 (b), the average mAP reduction for the untargeted attack is 8.01 and 4.35 for the vanishing attack. M5 also shows a significant success on white box attack with 74.56-0.97=73.59 mAP reduction for the untargeted attack and 63.69 mAP reduction for the vanishing attack. On the contrary, the white box attack of M1 is not particularly effective (i.e., 16.77 mAP reduction for the untargeted attack and 9.45 mAP reduction for the vanishing attack).

Table 7
Performance of target models on adversarial samples (a) generated by M5; (b) generated by M1. The rows highlighted in gray indicate the performance of the white box attacks. Others are the performance of black box attacks.

Target Model	Clean	Untargeted	Vanishing
M1	45.75	35.94	39.57
M2	26.32	7.31	9.65
M3	10.36	4.63	5.22
M4	68.29	50.64	57.24
M5	74.56	0.97	10.87
M6	24.74	7.95	12.35

(a)

Target Model	Clean	Untargeted	Vanishing
M1	45.75	28.98	36.30
M2	26.32	24.79	24.59
M3	10.36	9.09	9.07
M4	68.29	43.91	59.51
M5	74.56	63.18	64.66
M6	24.74	23.25	24.71

(b)

Figure 4 plots the correlation between dataset intersection and attack performance. It is apparent that a greater intersection leads to more transferable attacks. Training on the same images should make the two object detectors make similar predictions, which make them vulnerable to the same adversarial sample.

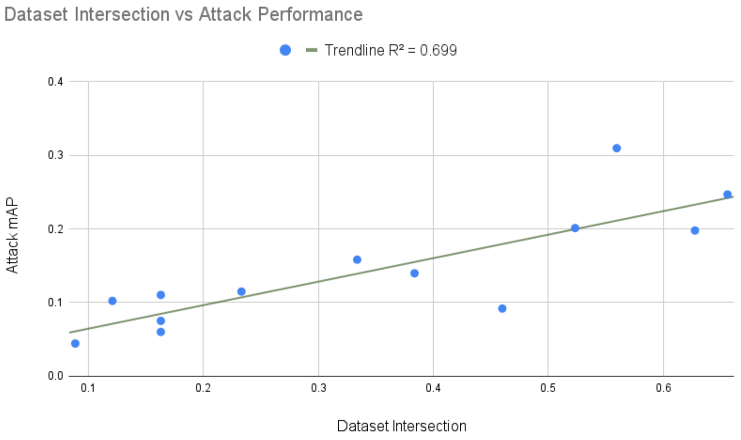


Figure 4: The correlation between dataset intersection and attack performance

Transfer attack performance compared to epsilon Figure 5 shows that although epsilon=8 is a little too low to get a strong attack transference between some models, by epsilon=20, the model performance has dropped off sharply. This is a good indication of how vital the dataset intersection is to get stronger transference. Comparing Figure 3 (b) to Figure 5, it is apparent that M5 generates adversarial samples that are more likely to fool M4 than M4 is to generate adversarial samples that fool M5. The intersection between M5 and M4 allows for an epsilon reduction of almost 10 to achieve the same mAP reduction as in Figure 3 (b).

Together with Table 6, we can explain the performance difference of targeted models on adversarial samples generated by different source models. That is the more intersection between two datasets, the higher transferability of the attack is. In Figure 5, the performance curve of

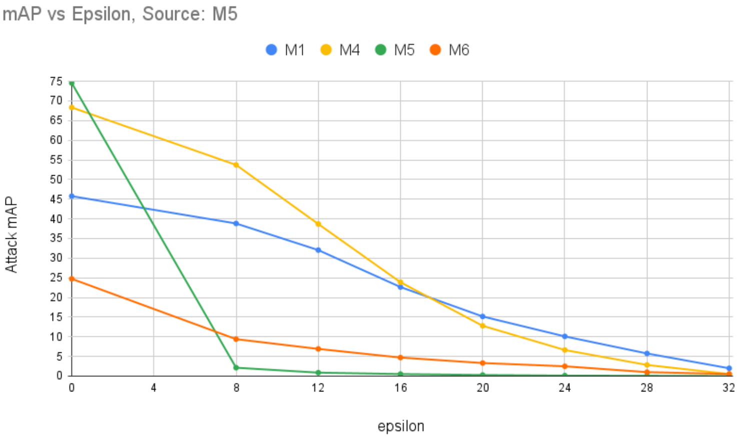


Figure 5: The blackbox and white box performance plotted against epsilon. The adversarial set is generated by the M5 model

M4 is steeper than in Figure 3 (a) due to the intersection between D4 and D5. This intersection allows M5 to attack M4 more effectively than M2.

Answer to RQ2: The more intersection between two datasets, the higher transferability of the attack is.

4.3. Extra findings regarding different source and target resolution

In our experiments to answer RQ1 and RQ2, we have datasets with images of different resolutions. The data in Table 9 shows that a mismatch in resolution between source and target model incurs a penalty in attack strength. Table 10 compares the relative mAP loss for samples generated and detected at the same resolution (see Table 7 (a)) to samples generated and detected at different resolutions (see Table 9). Table 10 shows that the average performance drop between the identical resolution and different resolution attacks is on average 22.5 percentage points. This is in line with the findings from [5]. They found that downsampling an adversarial sample to a lower resolution always produced a weaker attack than upsampling the image. This implies that an attacker who does not know the victim detector's resolution should choose a lower resolution like 416×416 to raise the probability of the attack being upsampled rather than downsampled.

5. Discussion

This section describes the comparison to related work, as well as the implications contribution of the findings presented in section 4.

Comparison to Related Work Many studies have been done on transferable adversarial attacks but few target real-time object detectors. Of the studies done on transferable samples targeting object detectors [5, 12], none operate with the same black box model as used in

Table 9
Performance on adversarial samples generated by M5 at 416x416, but detected at 832x832 demonstrating the cross-resolution performance. The row highlighted in gray indicates the performance of the white box attack. Others are the performance of black box attacks.

Target Model	Clean	Untargeted	Vanishing
M1	37.96	36.47	37.29
M2	12.52	5.71	6.19
M3	4.47	2.02	2.07
M4	50.47	43.76	44.16
M5	51.25	40.06	37.53
M6	11.14	4.76	4.96

Table 10
Relative mAP loss for samples generated and detected at the same resolution compared to samples generated and detected at different resolutions. The row highlighted in gray indicates the performance of the white box attack. Others are the performance of black box attacks.

Target Model	Untargeted 832	Untargeted 416	Difference
M1	21.44%	3.93%	17.51
M2	72.23%	54.39%	17.84
M3	55.31%	54.81%	0.5
M4	25.85%	13.30%	12.55
M5	98.70%	21.83%	76.87
M6	67.87%	57.27%	10.6
Average			22.50

this paper, but rather perform cross-resolution and cross-architecture transference, but not cross-dataset. [4] studied the cross-dataset transference, and their attack method can reduce the mAP of the target YOLO model by 30.9% at an L_∞ epsilon of 20. Our method can reduce the mAP of the target model by 49.7% at the same L_∞ epsilon of 20. Cross-dataset transference on real-time object detectors is largely unexplored and no other studies were found that examined the correlation between attack transference and relative dataset intersection. [19] found they could discover and filter adversarial dataset poisoning attacks. However, if the samples added to the dataset had no perturbation added to them but only served to create an intersection between the target detector and the attacker's surrogate detector, such techniques would be useless.

Implications to Academia This study shows expanded knowledge about cross-dataset attacks. We found that the model trained on the superset of the other datasets generated adversarial samples with better transference to the other models than vice versa. This indicates that it is not simply the size of the intersection that matters but the size of the intersection relative to the whole training set.

Implications to Industry The training set for an object detector deployed in an autonomous vehicle must be secured. The inclusion of public data must be carefully considered, as the public data increases attack transference, allowing effective attacks with a smaller epsilon. Transfer attacks with subtle perturbations may not be an immediate threat, while the transfer attacks become effective at higher epsilon values.

Threats to Validity External validity is susceptible to poor generalizability and replicability of the results. This study is conducted with a single object detector and attack framework, limiting the generalizability of the results. However, the fact that adversarial samples transfer between dataset, architectures and even tasks show that results on one set of detectors and attacks are relevant to other detectors and attacks. Additionally, the scenario described in section 3.2 severally constrained the object detectors and adversarial attacks, which were relevant to this study. Since the mAP score an object detection model achieves is highly dataset-dependent, several datasets were compiled and used to train models.

6. Conclusion and future work

This paper studied cross-dataset transference by training Darknet models on disjoint training sets. Then the TOG framework was used to generate adversarial sets. We compared the attack performance on the source model with the target model and measured how well the attack transfers. At an epsilon of 8, the attack barely transferred, leaving the target model unaffected. Raising the epsilon to the mid-20s makes the transference stronger and reduces mAP on the target model by about half. Additional models were trained on datasets with various intersection sizes to further investigate the impact of dataset intersection. The experimental results demonstrate the correlation between attack transference and dataset intersection. Besides, we identified the average penalty to attack strength associated with cross-resolution attacks.

The natural next step for this research is to add additional architectures to the analysis. For example, comparing YOLO results with SSD and testing cross-dataset transference on adversarial patches would be interesting. Adversarial patches avoid the issue of minimizing the perturbation, which could suggest that they offer robust and cross-dataset transference. In addition, studying physical patches would make the attack even more relevant to autonomous vehicles.

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (2015) 436–444.

[2] M. J. Shafiee, A. Jeddi, A. Nazemi, P. Fieguth, A. Wong, Deep neural network perception models and robust autonomous driving systems: Practical solutions for mitigation and improvement, IEEE Signal Processing Magazine 38 (2021) 22–30. doi:10.1109/MSP.2020.2982820.

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, 2014. arXiv:1312.6199.

[4] Q. Zhang, Y. Zhao, Y. Wang, T. Baker, J. Zhang, J. Hu, Towards cross-task universal perturbation against black-box object detectors in autonomous driving, Computer Networks 180 (2020) 1. URL: https://search.proquest.com/scholarly-journals/towards-cross-task-universal-perturbation-against/docview/2476554302/se-2?accountid=12870, copyright - Copyright Elsevier Sequoia S.A. Oct 24, 2020; Last updated - 2021-01-11.

[5] K.-H. Chow, L. Liu, M. E. Gursoy, S. Truex, W. Wei, Y. Wu, Understanding object detection through an adversarial lens, in: European Symposium on Research in Computer Security, Springer, 2020, pp. 460–481.

[6] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, A. Yuille, Adversarial examples for semantic segmentation and object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1369–1378.

[7] X. Wei, S. Liang, N. Chen, X. Cao, Transferable adversarial attacks for image and video object detection, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 954–960. URL: https://doi.org/10.24963/ijcai.2019/134. doi:10.24963/ijcai.2019/134.

[8] S. Grigorescu, B. Trasnea, T. Cocias, G. Macesanu, A survey of deep learning techniques for autonomous driving, Journal of Field Robotics 37 (2020) 362–386.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, Lecture Notes in Computer Science (2016) 21–37. URL: http://dx.doi.org/10.1007/978-3-319-46448-0_2. doi:10.1007/978-3-319-46448-0_2.

[10] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv (2018).

[11] K.-H. Chow, L. Liu, M. Loper, J. Bae, M. Emre Gursoy, S. Truex, W. Wei, Y. Wu, Adversarial objectness gradient attacks in real-time object detection systems, in: IEEE International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications, IEEE, 2020, pp. 263–272.

[12] S. Chen, F. He, X. Huang, K. Zhang, Relevance attack on detectors, 2021. arXiv:2008.06822.

[13] S. V. Grini, Systematic training and testing of deep learning-based detection methods for vessels in camera images, 2019.

[14] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabaly, C. Quek, Video processing from electro-optical sensors for object detection and tracking in maritime environment: A survey, 2016. arXiv:1611.05842.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.

[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International Journal of Computer Vision 88 (2010) 303–338.

[17] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection, 2020. arXiv:2004.10934.

[18] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, 2017. arXiv:1608.04644.

[19] A. Paudice, L. Muñoz-González, A. Gyorgy, E. C. Lupu, Detection of adversarial training examples in poisoning attacks through anomaly detection, 2018. arXiv:1802.03041.