Physical and Digital Adversarial Attacks on Grasp Quality Networks

Naif Wasel Alharthi and Martim Brandão

Abstract— **Grasp Quality Networks are important components of grasping-capable autonomous robots, as they allow them to evaluate grasp candidates and select the one with highest chance of success. The widespread use of pick-and-place robots and Grasp Quality Networks raises the question of whether such systems are vulnerable to adversarial attacks, as that could lead to large economic damage. In this paper we propose two kinds of attacks on Grasp Quality Networks, one assuming physical access to the workspace (to place or attach a new object) and another assuming digital access to the camera software (to inject a pixel-intensity change on a single pixel). We then use evolutionary optimization to obtain attacks that simultaneously minimize the noticeability of the attacks and the chance that selected grasps are successful. Our experiments show that both kinds of attack lead to drastic drops in algorithm performance, thus making them important attacks to consider in the cybersecurity of grasping robots. Source code can be found at `https://github.com/Naif-W-Alharthi/Physical-and-Digital-Attacks-on-Grasping-Networks`**

I. Introduction

Robots are now widespread across manufacturing, logistics and other industries, where they pick-and-place large amounts of objects everyday. Cybersecurity issues in such systems could thus lead to large economic impact. Grasp Quality Networks [1], [2], [3], [4], [5], [6], in particular, are important components of grasping-capable robots, as they allow them to evaluate multiple grasp candidates and select the one with highest chance of success. They are popular algorithms since they can leverage recent progress in computer vision and neural networks, but at the same time they might be vulnerable to similar adversarial attacks as those recently shown on deep neural networks [7].

In this paper we investigate the vulnerability of current Grasp Quality Networks to adversarial attacks. Concretely, we propose two kinds of attacks: physical and digital attacks. Physical attacks are those where an attacker has physical access to the robot's workspace, and places a new (potentially inconspicuous) object in the scene so as to trick the network to pick a new grasp with a lower probability of success. Digital attacks, on the other hand, assume that the attacker has access to the camera *software* and is able to inject single-pixel changes to the images before they reach the grasp network. This leads the network to believe the workspace is different from what it actually is, thus selecting a grasp that is more unlikely to succeed. Both kinds of attack could lead robots to damage property, thus provoking economic and reputational harm. Such attacks are not implausible, since competing businesses often adopt adversarial tactics for economic gain, and recent events have shown that users often attack robots for various reasons [8].

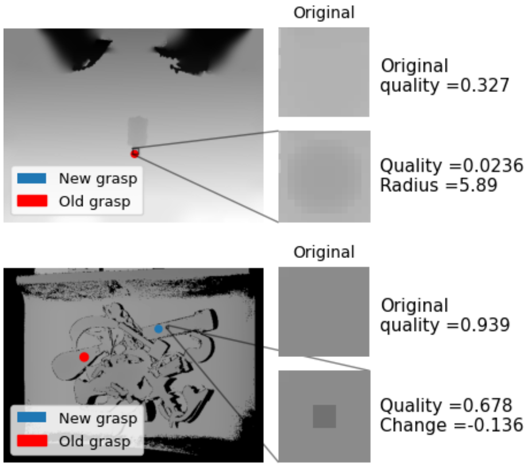

Fig. 1. Example physical (top) and digital one-pixel (bottom) attacks on Grasp Quality Networks. The physical attack corresponds to physically placing a (barely visible) spherical object on the workspace, while the digital attack corresponds to changing the intensity of a single pixel in the image (without any physical changes to the scene). Both attacks lead the network to change its preferred grasp, which is of considerably lower quality (i.e. success probability).

Our contributions are thus the following:

1) We propose and characterize the threat model of two attacks on Grasp Quality Networks (physical and digital), which we implement through evolutionary optimization.
2) We evaluate the attacks on various openly-available models, showing they are able to drastically reduce the quality of grasps—thus making this kind of attack an important one to consider in robot cybersecurity.

II. Related work

Various algorithms have been proposed to estimate the probability of a robot grasp's success. One class of algorithms uses human predictions of grasp success to train a grasp scoring network [1]. Another class of algorithms is trained directly on robots [2], and yet another uses physics simulation in synthetic datasets [3], [4], [5], [6]. Those trained directly on robots are expensive since they require many hours of training (or many robots). For example Levine et al. [2] train neural networks using many robots, in order to predict grasp success probability from an image and task-space motion command. Simulation-based networks [3], [4], [5], [6] have been proven to provide a good balance between training resources (since they do not require real experiments) and reliability, with recent Grasp Quality Networks of this kind showing high reliability even on real-world robot grasping

Both authors are with King's College London, UK.

experiments [5], [6]. Some of these networks are also capable of selecting a grasp and a gripper, in multi-gripper (multi-arm) robot scenarios [5]. In this paper we focus our attacks on this kind of simulation-based algorithm due to their popularity, reliability, and open availability. Both of our attacks are inspired by one-pixel attacks [7], which were initially proposed for image classification algorithms such as to fool them into classifying one object as a totally different one. They are also inspired by physical attacks such as camera sticker attacks [9], where a small sticker is put on the camera itself in order to lower algorithms' performance. This attack is arguably easily identifiable as an attack, and thus unlikely to be used, however. Our paper proposes one digital (one-pixel) and one physical attack, and we consider more inconspicuous physical attacks where a small object is placed on the robot's workplace.

Few attacks have been proposed in the robotics domain, despite the importance of cybersecurity aspects of robotics [10]. Most of the attacks have focused on network [11], controller [12] and signal-interference [13] attacks to robot systems, while our focus is on physical and image-based attacks. One of the exceptions is the work on characterizing reinforcement learning attacks [14], where different attacks (on the motors, sensors, and environment) are considered. In this type of classification, our attacks are environment and sensor based, though they are targeted at specific grasping algorithms whose vulnerability has not yet been investigated. Another exception is the work on physical attacks on Autonomous Vehicles and arm robots, where objects are placed in the physical environment in order to trick semantic segmentation [15], traffic sign classification [16], or motion planning [17] algorithms.

Perhaps the closest work to ours in motivation is that of Wang et al. [18], where they computationally generate objects that will be hard to grasp in terms of formal grasp quality metrics (number of antipodal points). Their algorithm adds random perturbations and texture changes, which are reasonable changes when the attacker is making a completely new object, though this could be potentially unrealistic in a pick-and-place logistics scenario, where the attacker would have to create a new product. Our physical attack, on the other hand, only requires placing or attaching a small object to an existing product. Another difference is that our work is targeted at machine learning-based grasp metrics, which are now more popular, and we characterize both a physical and a digital attack.

III. Method

Let $Q(I, g)$ be a function that estimates the quality (or equivalently the "score" or "success probability") of a grasp g given an image I. An example of such a function is a grasp quality network [3], and the grasp g is a general grasp command (e.g. a pixel or a robot motor command). As is typically the case [3], [4], [19], we assume the image is a depth image, where the value at each pixel is related to the metric distance from the camera to the nearest object in that location.

A *grasp selection* algorithm is a function $G(I)$ which internally generates N candidate grasps g_i, $i = 1, ..., N$, and outputs the best-quality candidate $g^* = G(I) = \text{argmax}_{g_i \in \{g_1,...,g_N\}} Q(I, g_i)$. Both of our proposed attacks work by generating a new image $I' \approx I$ which minimizes the probability that the resulting selected grasp succeeds.

We propose two kinds of adversarial methods that do this assuming different attacker capabilities. The first method is a physical-environment attack (i.e. where the image change is the result of physically placing a small object on top of an existing object). The second method is a pure image attack (i.e. where the image is changed by software, for example due to an injection attack on compromised computer or network). We now describe each of them in sequence.

A. Physical attack

1) Idea: The main idea of this attack is to simulate the placement of a physical object on the scene, so as to make the new grasp obtained by grasp-selection have as low chance of success as possible (i.e. as low quality Q as possible). Because the attacker might not want to get caught, the attack should make a small change to the environment, which in this paper we assume to mean the added object is of the smallest possible size.

2) Threat model: We assume the attacker has physical access to the robot's workspace, or access to objects before they reach the robot—e.g. they are able to glue a small part to an object before it reaches the robot. The objective of the attack is to make the grasp selection algorithm select a low-quality grasp, thus significantly increasing the chance of the product being dropped and damaged. Potential incentives for such attacks can be economic (i.e. to damage property and thus raise costs, for example by a competitor company) or reputational.

3) Method: We formulate the attack as a multi-objective optimization problem:

$$\begin{aligned} \underset{s=(x,y,p)}{\text{minimize}} \quad & \left(\; Q(f(I,s), G(f(I,s))) \;,\; \sqrt{p^\intercal W p} \; \right) \qquad (1) \\ \text{s.t.} \quad & (x,y) \in \text{mask}(I) \\ & p \in P \end{aligned}$$

where (x, y) is the pixel location of the center of the object in the depth image I, constrained to lie on top of existing objects (i.e. $\text{mask}(I)$ is the set of pixels of I where any object is present), $p \in P$ are the parameters of the added object, and f is a function that returns a new image $I' = f(I, s)$ which simulates the addition of the new object to the scene. W is a diagonal weight matrix, set so that $p^\intercal W p$ is a (weighted) norm of object parameters, set as to make the object as small as possible. In particular in this paper we implement physical attacks as spheres for simplicity. Thus, $p = r$ is the radius of the sphere in pixels, $W = [1]$ so as to minimize r, and f places a sphere centered at (x, y, d) where $d = I(x, y)$ is the original depth (i.e. distance) at that pixel.

To solve this multi-objective optimization problem we use an evolutionary Pareto-curve estimation algorithm, which computes the optimal trade-off curve between the two objectives. In particular we chose SPEA2 [20] for its open implementation and efficiency over other optimization approaches [21]. Basically, SPEA2 keeps and mutates a set of "individuals", where each individual holds a value of $s = (x, y, r)$ and is evaluated by two fitness functions—i.e. the two objectives in (1). The first objective corresponds to running the new image through the grasp-quality-and-selection algorithms to obtain the quality of the best grasp, while the second objective is equivalent to $|r|$.

B. Digital (one-pixel) attack

1) Idea: The main idea of this attack is to inject a change to a single pixel in the image (e.g. through access to the camera's software or ROS interface [22]), so as to obtain a grasp that has low chance of success on the real scene—which was not physically changed. So as to make the attack inconspicuous, the amount of intensity change to the image should be as small as possible.

2) Threat model: In this attack we assume the attacker has digital access to the camera software (e.g. camera driver, ROS interface [22]) and is able to inject a change to the images returned by the camera before they reach the grasp-quality algorithm. The objective of the attack is that the computed grasp, when executed on the real scene (which does not have any physical change), has a low chance of success, thus significantly increasing the chance of the product being dropped and damaged. Potential incentives are the same as those of the physical attack (i.e. economic or reputational).

3) Method: We formulate the attack as a multi-objective optimization problem:

$$\begin{aligned} \underset{s=(x,y,p)}{\text{minimize}} \quad & (\ Q(I, G(f(I,s)))\ \ ,\ \ \|p\|\ ) \\ \text{s.t.} \quad & x \in [0, \text{width}(I)],\ \ y \in [0, \text{height}(I)] \\ & p \in [-1, 1] \\ & I(x,y) + p \in [0,1] \end{aligned} \tag{2}$$

where (x, y) could lie anywhere on the image, p is the pixel intensity change, and the last constraint makes sure that the new image's pixel intensity is within limits. $f(I, s)$ is a function that returns a new image I', such that $I'(x, y) = I(x, y) + p$ and $I'(i, j) = I(i, j)$ for all $(i, j) \neq (x, y)$. Note that, in contrast to (1), this optimization problem is minimizing the quality of the new grasp on the *original image* I—this is because the physical world has actually *not* changed as a reasult of the image attack. Similarly to the physical attack, we also use SPEA2 to solve this optimization problem.

IV. Results

In this section we will evaluate the effectiveness of each kind of attack on several Grasp Quality Networks: DexNet 2.0 [3] (which tackles single-object scenes); Dexnet 2.1 [4] and FCGQCNN [6] (multi-object); and Dexnet 4.0 [5] (multi-object and multi-gripper). The networks are responsible for generating both Q and G.

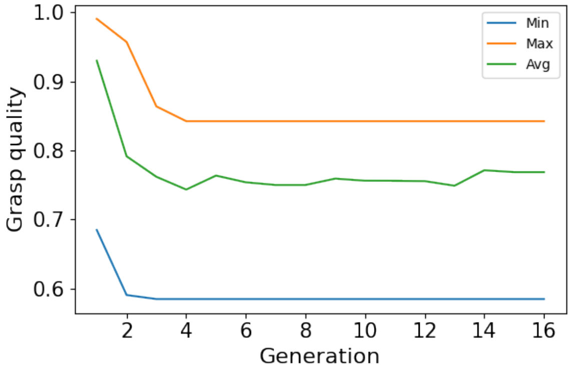

Fig. 2. Evolution of physical attacks' grasp quality with the number of algorithm generations, on an example image of the Dexnet 4.0 network. Minimum, maximum and average over the population of individuals evolved is shown.

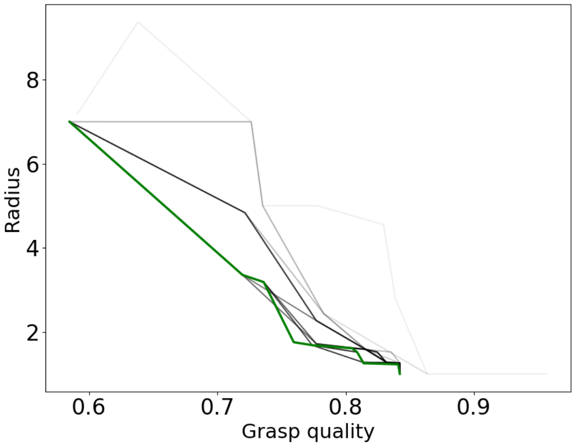

Fig. 3. Pareto front of the two objectives (grasp quality and size of the attack) on a physical attack to an example image of the Dexnet 4.0 network. The green curve is the Pareto front at the final generation of the evolutionary algorithm, while gray lines show the evolution of the front from first to last generation (light to dark).

We use the networks' original implementations and pre-trained models[1]. Since each network was trained on specific cameras, object clutter conditions, and distances to the workspace; and to make sure we are fair in our evaluation; we apply our attacks to the example images provided by the authors for each model (5-10 per model).

To solve the optimization problems (1) and (2) we used SPEA2 [20], which is an evolutionary Pareto optimizer specifically geared at black-box multi-objective optimization. In particular we used the implementation available in the DEAP [21] Python library, with the following parameters: population size 10, number of children per generation 100, and crossover and mutation probabilities 0.6 and 0.3 respectively (which are default recommended values). We ran the algorithm for 15 generations on all images of all networks, as this was sufficient for convergence. In the physical attack we set a radius limit of $r \leq 30$.

[1] Available at `https://github.com/BerkeleyAutomation/gqcnn`.

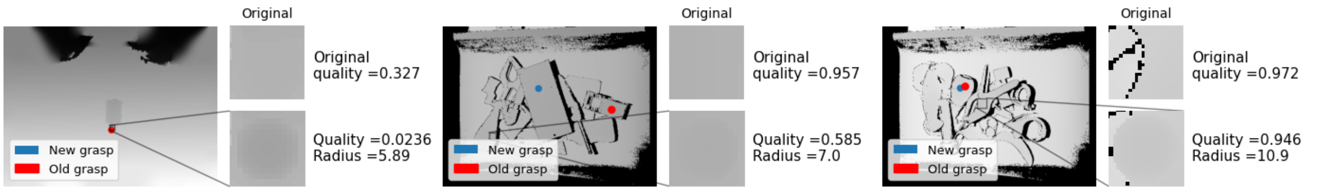

Fig. 4. Three example physical attacks obtained by our method, on three different Grasp Quality Networks. Left: Dexnet 2.0. Center: Dexnet 4.0. Right: FCGQCNN (with suction gripper).

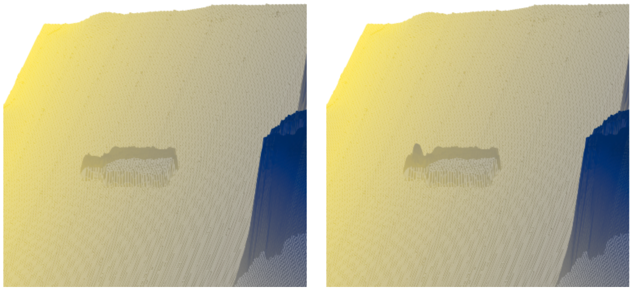

Fig. 5. 3D view of an example physical attack on Dexnet 2.0 obtained by our method. Radius of the attached sphere is 5.89, and leads grasp quality to go from 0.327 to 0.0236. Left: original scene. Right: attacked scene.

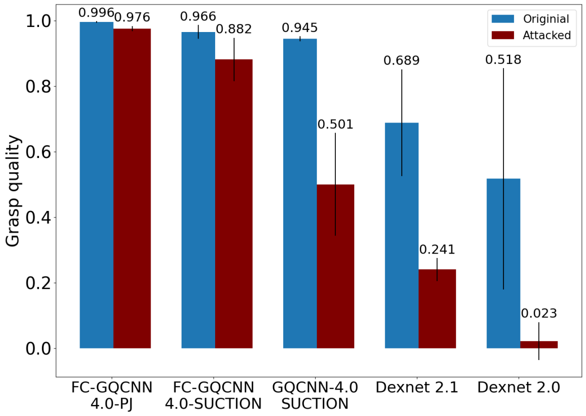

Fig. 6. Average and standard deviation of grasp quality (over the images associated with each network), obtained after physical attacks computed by our method. Results on the original images (before attack) also shown for comparison.

A. Physical attack

Fig. 2 shows the evolution of the optimization of (1) with the number of generations of the evolutionary algorithm, on an example image on Dexnet 4.0. The figure has three curves which plot the maximum, minimum, and average values over the selected population at each generation. These curves show that the quality of the grasps have converged by around the 5th generation, and that the algorithm is able to obtain attacks of varied effectiveness—with grasp quality between 0.6 and 0.85, while the quality on the original (not attacked) image was approximately 1 (see max curve, generation 1).

Fig. 3 shows the corresponding Pareto front of the two objectives (grasp quality and object radius). The figure shows that there is a trade-off between the objectives: an object of 1-pixel radius is able to reduce grasp quality to about 0.85, but increasing the radius towards 7 pixels approximately linearly decreases grasp quality down to 0.6. It is also interesting to note that increasing the radius to more than 7 pixels does not lead to a decrease in grasp quality, indicating that the worst-possible performance of the network is achieved by this object radius.

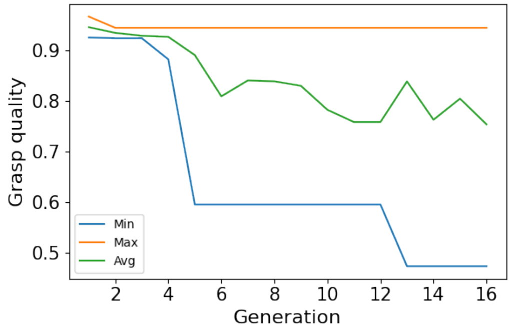

Fig. 7. Evolution of digital (one-pixel) attacks' grasp quality with the number of algorithm generations, on an example image of the Dexnet 4.0 network. Minimum, maximum and average over the population of individuals evolved is shown.

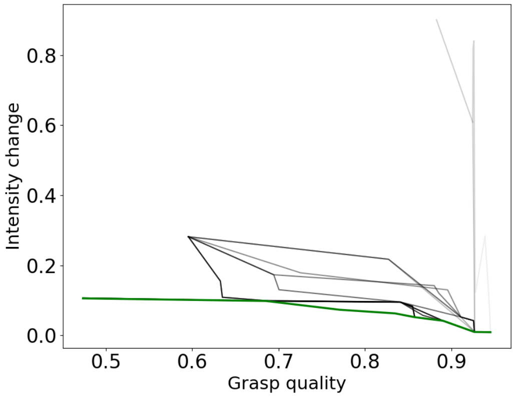

Fig. 8. Pareto front of the two objectives (grasp quality and pixel intensity change) on a digital attack to an example image of the Dexnet 4.0 network. The green curve is the Pareto front at the final generation of the evolutionary algorithm, while gray lines show the evolution of the front from first to last generation (light to dark).

Fig. 4 shows three concrete examples of physical attacks, taken from the final population of the algorithm, on three different models: Dexnet 2.0, Dexnet 4.0 and FCGQCNN. The figure shows that the attacks lead the grasp networks to

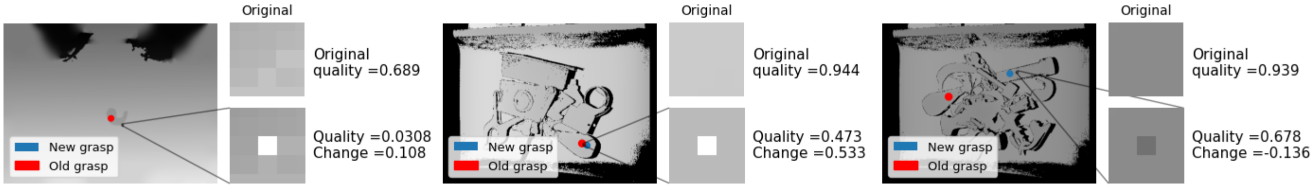

Fig. 9. Three example digital (one-pixel) attacks obtained by our method, on three different Grasp Quality Networks. Left: Dexnet 2.0. Center: Dexnet 4.0. Right: FCGQCNN (with suction gripper). Note that zoomed-in images are normalized, and therefore even though the attacked pixels looks white/black, their actual intensity change is shown on the side as "Change=".

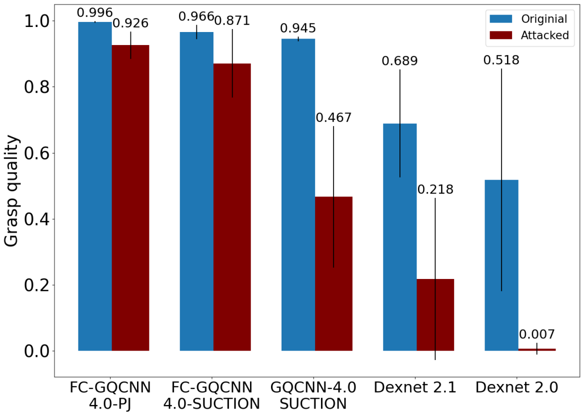

Fig. 10. Average and standard deviation of grasp quality (over the images associated with each network), obtained after digital attacks computed by our method. Results on the original images (before attack) also shown for comparison.

generate a different grasp location, and that this grasp can be of considerably lower quality—indicating a high probability of failure. The first example is of a 6-pixel radius sphere, which is barely noticeable from the depth image even when zoomed in, though it leads to a considerable drop in quality of the new grasp (from 0.327 to 0.0236). The middle and right images show examples where the attack is in a location far away from the original grasp, indicating that attacks do not have to be placed on the target object, thus making it easy for an attacker to disturb a robot by just placing an extra object on the workspace—whether or not that is the grasped object. The image on the right shows that the FCGQCNN's new grasp is still of high quality, only 3% lower. However, a 3% decrease may still represent considerable efficiency decreases or economic damage if the network is being used on a robot processing hundreds of thousands of objects per day (e.g. in automated warehouse applications).

To better visualize the physical attacks, Fig. 5 shows a 3D view of the point cloud of a physical attack, which corresponds to the example Dexnet 2.0 example on Fig. 4 (left). The attack is barely visible, noticeable only as a small bump on the shade of the object.

Finally, we computed the average grasp quality, over all images, obtained by the best-performing attack in each of the networks. Fig. 6 shows the results, together with the original grasp quality before the attack. There is a clear difference of behaviour between the networks: attacks on Dexnet 2.0, 2.1 and 4 all have similarly high performance drops of around 0.4 in value. On the other hand, the fully-convolutional networks' drop in quality is much lower. This suggests that these (newer) networks are more robust to physical attacks, which could be related to their fully-convolutional architecture and higher reliability as reported in [6]. The drop was not as high as in the other networks but still corresponds to a 2-9% decrease depending on gripper type (parallel jaw or suction).

B. Digital attack

Fig. 7 shows the evolution of the optimization of (2) with the number of generations of the evolutionary algorithm, on an example image on Dexnet 4.0. The quality of the grasps have converged by around the 13th generation, and the algorithm is able to obtain attacks of quality down to 0.5—with a single pixel change.

Fig. 8 is the Pareto front at all generations—and the last one is shown in green. The curve shows that, similarly to the physical attack, there is a trade-off between the amount of pixel intensity change and the achieved grasp quality. Interestingly, the intensity change does not have to be high for the attack to be effective (maximum 0.1 change, while depth images can take values between 0 and 1).

Fig. 9 shows three examples of digital attacks obtained by our method. The examples illustrate that one-pixel changes to the image lead to drastic performance drops in grasp quality, from 0.689 to 0.0308 for Dexnet 2.0, 0.944 to 0.473 for Dexnet 4.0, and 0.939 to 0.678 for FCGQCNN (with a suction gripper). The figure thus illustrates how FCGQCNN is less robust to digital attacks compared to physical. The grasp locations also change as a result of the pixel attacks, as shown in the figure.

Finally, we computed the grasp quality obtained on the attacked images, averaged over all images associated with each network. Fig. 10 shows the results, together with the original grasp quality before the attack. Similarly to the physical attack, there are large performance drops of around 0.4 for Dexnet 2.0, 2.1 and 4.0. The figure shows that FCGQCNN is not as robust to digital attacks as it was to physical attacks. Performance drops are more pronounced in the digital attack, at 7-10% depending on gripper type (parallel jaw or suction). The reason for this is that in digital attacks, the network does not have access to the image of the real-world (i.e. the image before the pixel change) and our method can exploit this to minimize the resulting grasp quality.

V. Conclusions

In this paper we proposed two kinds of adversarial attacks on Grasp Quality Networks, which are algorithms used to rank grasps to be used by a robot based on image input. One attack was physical, involving the optimization of the size and location of a physical object to add to the robot's workspace, so as to make the robot's new grasp be as likely to fail as possible. The other attack was digital, assuming access to the camera's software or interface, and involving the optimization of a single pixel's location and intensity change so as to make the new grasp fail. Both attacks could lead to economic and reputational damage, thus making them serious attacks to consider in the deployment of robots.

Our experiments showed that our attack methods can drastically reduce quality of grasps (and hence probability of success) by approximately 0.4 in quality units for both physical and digital attacks (or equivalently up to 22 times lower quality for physical and 74 times lower for digital attacks). The newer grasp quality network FCGQCNN was relatively more robust to physical attacks, being only 2-9% less confident on new grasps depending on gripper type, but it was less robust to digital attacks—which led to a decrease of 7-10%. Our results thus showed an advantage of the digital attack, which is that it can exploit the network's lack of access to an image of the real world.

Future directions of research include the development and evaluation of such attacks on real robots, the implementation of multi-pixel attacks, and physical attacks with various shapes. Importantly, this paper also shows there is a need for further research on protections against adversarial attacks to image-based grasping algorithms. Future work should focus on the development of protections, for example through attack detection methods, attack prevention through robust algorithms, physical damage mitigation methods in case of a successful attack, and last but not least work on increasing awareness of cybersecurity aspects of robotics within the community.

References

[1] D. Kappler, J. Bohg, and S. Schaal, "Leveraging big data for grasp planning," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 4304–4311.

[2] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[3] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," 2017.

[4] J. Mahler and K. Goldberg, "Learning deep policies for robot bin picking by simulating robust grasping sequences," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 515–524.

[5] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, p. eaau4984, 2019.

[6] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robotics and Automation Letters*, 2019.

[7] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[8] J. A. Oravec, "Robo-rage against the machine: Abuse, sabotage, and bullying of robots and autonomous vehicles," in *Good Robot, Bad Robot: Dark and Creepy Sides of Robotics, Autonomous Vehicles, and AI*. Springer, 2022, pp. 205–244.

[9] J. Li, F. R. Schmidt, and J. Z. Kolter, "Adversarial camera stickers: A physical camera-based attack on deep learning systems," *CoRR*, vol. abs/1904.00759, 2019. [Online]. Available: http://arxiv.org/abs/1904.00759

[10] J.-P. A. Yaacoub, H. N. Noura, O. Salman, and A. Chehab, "Robotics cyber security: Vulnerabilities, attacks, countermeasures, and recommendations," *International Journal of Information Security*, pp. 1–44, 2022.

[11] S. Gil, S. Kumar, M. Mazumder, D. Katabi, and D. Rus, "Guaranteeing spoof-resilient multi-robot networks," *Autonomous Robots*, vol. 41, pp. 1383–1400, 2017.

[12] D. Quarta, M. Pogliani, M. Polino, F. Maggi, A. M. Zanchettin, and S. Zanero, "An experimental security analysis of an industrial robot controller," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 268–286.

[13] Y. Li, C. Wen, F. Juefei-Xu, and C. Feng, "Fooling lidar perception via adversarial trajectory perturbation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7898–7907.

[14] X. Pan, C. Xiao, W. He, S. Yang, J. Peng, M. Sun, J. Yi, Z. Yang, M. Liu, B. Li, and D. Song, "Characterizing attacks on deep reinforcement learning," 2022.

[15] Y. Zhu, C. Miao, F. Hajiaghajani, M. Huai, L. Su, and C. Qiao, "Adversarial attacks against lidar semantic segmentation in autonomous driving," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 329–342.

[16] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on machine learning models," *CoRR*, vol. abs/1707.08945, 2017. [Online]. Available: http://arxiv.org/abs/1707.08945

[17] W. Wu, F. Pierazzi, Y. Du, and M. Brandao, "Characterizing physical adversarial attacks on robot motion planners," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, May 2024.

[18] D. Wang, D. Tseng, P. Li, Y. Jiang, M. Guo, M. Danielczuk, J. Mahler, J. Ichnowski, and K. Goldberg, "Adversarial grasp objects," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 241–248.

[19] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust robot suction grasp targets in point clouds using a new analytic model and deep learning," *arXiv preprint arXiv:1709.06670*, 2017.

[20] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK report*, vol. 103, 2001.

[21] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagn'e, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, jul 2012.

[22] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.