2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506112

IMPROVING FILLING LEVEL CLASSIFICATION WITH ADVERSARIAL TRAINING

Apostolos Modas[1], *Alessio Xompero*[2], *Ricardo Sanchez-Matilla*[2], *Pascal Frossard*[1], *Andrea Cavallaro*[2]

[1]LTS4, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland,
[2]Centre for Intelligent Sensing, Queen Mary University of London, UK

ABSTRACT

We investigate the problem of classifying – from a single image – the level of content in a cup or a drinking glass. This problem is made challenging by several ambiguities caused by transparencies, shape variations and partial occlusions, and by the availability of only small training datasets. In this paper, we tackle this problem with an appropriate strategy for transfer learning. Specifically, we use adversarial training in a generic source dataset and then refine the training with a task-specific dataset. We also discuss and experimentally evaluate several training strategies and their combination on a range of container types of the CORSMAL Containers Manipulation dataset. We show that transfer learning with adversarial training in the source domain consistently improves the classification accuracy on the test set and limits the overfitting of the classifier to specific features of the training data.

Index Terms— Adversarial training, Transfer learning, Classification

1. INTRODUCTION

The estimation of the amount of content (filling level) within a container is made challenging due to differences in the shape of containers, occlusions caused by the hand holding the container, and transparencies of both the container and the filling (e.g., depth estimation may be highly inaccurate for transparent objects [1]). The few approaches designed to tackle this problem use RGB [2], thermal [3], or a combination of RGB and depth data [4, 5], and usually observe the action of pouring content in a container over multiple frames [3–6]. Mottaghi *et al.* [2] showed that a Convolutional Neural Network (CNN) classifier outperforms a regression model in estimating the filling level using only one RGB image. The best performance was achieved with transfer learning [7]: self-collected data were used as task-specific dataset, the *target domain*, to fine-tune the parameters of selected layers of the CNN that was previously trained on the much larger ImageNet dataset [8], the *source domain*. Transfer learning is suitable for image recognition tasks with only small datasets available for training. Examples of these tasks include fine-grained object classification and scene classification [9, 10], and the recognition of object properties such as volume, texture, shape and material [1, 2, 11–13].

In this paper, we go beyond current approaches that use transfer learning to classify the filling level of a container [2], and evaluate transfer learning combined with adversarial training in the source domain [14, 15] on the small-scale CORSMAL Containers Manipulation (CCM) [16] dataset (Fig. 1). We thoroughly analyze the performance of standard training, adversarial training, transfer learning, and their combinations, under different setups, which include the number of fixed layers during fine-tuning, and the norm of the perturbation in adversarial training both on the source and the target domain. We show that the generalization of a ResNet-18 [17] classifier on the test set of CCM can be limited by its bias towards specific features of the CCM training data. However, adversarially training the classifier on ImageNet, followed by fine-tuning on the train set of CCM, mitigates these biases and consistently produces classifiers with better generalization performance.

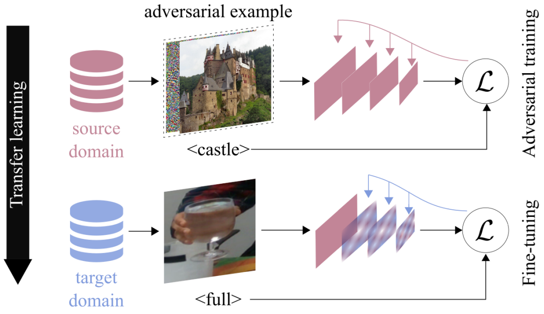

Fig. 1. An illustrative 4-layer CNN trained via transfer learning using adversarial training on the source domain (adversarial perturbations are added to the original images), followed by fine-tuning some of the layers with images from the target domain. This training strategy achieves the best accuracy in the test set of the experiments. The color of each layer corresponds to the domains used to optimize the classifier parameters using the loss $\mathcal{L}$.

2. TARGET TASK AND TRAINING STRATEGIES

We approach the problem of estimating the filling level, y, of a container captured in an image $\boldsymbol{x} \in [0,1]^{H\times W\times C}$, as a classification task, where H, W, C are the height, width, and number of channels respectively. We express the filling level as a percentage of the container's capacity: $y \in \{0\%, 50\%, 90\%, \textit{unknown}\}$, where the *unknown* class helps handling cases with opaque or translucent containers for which the filling level cannot be estimated through direct vision. Let f_θ be a CNN classifier, parameterized by a set of parameters θ, that maps an image $\boldsymbol{x}$ – drawn from a distribution $\mathcal{D}$ – to a label y, such that $f_\theta(\boldsymbol{x}) = y$. Given a train set of image-label pairs $\mathcal{T} = \{(\boldsymbol{x}^i, y^i)\}_{i=1}^N$, the goal is to find a set of parameters that minimizes a suitable loss function $\mathcal{L}(\boldsymbol{x}, y|\theta)$ such that f_θ correctly predicts y for $\boldsymbol{x} \sim \mathcal{D}$ but $\boldsymbol{x} \notin \mathcal{T}$ (generalization).

We refer to the common strategy for training a classifier on a train set, $\mathcal{T}$, as Standard Training (ST). A good generalization may be achieved if the number of image-label pairs in $\mathcal{T}$ is very large,

This work is supported by the CHIST-ERA program through the project CORSMAL, under UK EPSRC grant EP/S031715/1 and Swiss NSF grant 20CH21_180444.

e.g., $N \approx 1.2$ millions in ImageNet. However, for the target task of classifying the filling level such amount of data is not available. Transfer learning helps to overcome this limitation by using an additional training set $\mathcal{S}$, with $|\mathcal{S}| = M \gg N$, that may not be related to the target task. Transfer learning pre-trains the parameters of f_θ on $\mathcal{S}$ (source domain) and then refines them on $\mathcal{T}$ (target domain) via fine-tuning (FT). We refer to this strategy as ST→FT. With ST→FT, the parameters of some layers in the pre-trained model are fixed and FT only refines those of the remaining layers. We will denote with L the number of layers whose parameters are fixed.

Instead of using the original set of images, Adversarial Training (AT) [18–20] uses images modified with carefully crafted noise, known as adversarial perturbation. This noise is specifically designed to change the decision of a classifier [18–22]. Formally, a perturbation $\boldsymbol{\delta}$ is added to an image $\boldsymbol{x}$ in order to maximize the loss function $\mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta}, y|\theta)$ in a given ℓ_p-ball of radius ϵ around $\boldsymbol{x}$ [18,20]

$$\begin{aligned} \max_{\boldsymbol{\delta}} \quad & \mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta}, y|\theta) \\ \text{s.t.} \quad & \|\boldsymbol{\delta}\|_p \leq \epsilon \\ & \boldsymbol{x}+\boldsymbol{\delta} \in [0,1]^{H \times W \times C}, \end{aligned} \tag{1}$$

and the objective of AT is to minimize the adversarial loss $\mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}, y|\theta)$. The resulting adversarially trained models learn features that correlate better with features of the classes of interest and are thus more robust [23–26]. Hence, f_θ is expected to learn more task-relevant features with AT. While AT was originally designed to increase the robustness of deep networks to adversarial perturbations [19, 20], it has also contributed to other tasks [27]. Recently, it was shown that AT in the source domain can improve transfer learning [14, 15]: adversarially trained models from a source domain can help improving the accuracy on the target task after fine-tuning, despite performing worse, in terms of task accuracy, on the source domain.

We aim to evaluate this training strategy on the filling-level classification task and to compare it against five other strategies. As training strategies we consider ST→FT [2]; ST on the target domain; AT on the target domain; and three combinations of AT with transfer learning, namely AT on the *source* domain (AT→FT), AT on the *target* domain (ST→AFT), and AT on *both* domains (AT→AFT).

AT→FT adversarially pre-trains the parameters of f_θ on the *source* domain $\mathcal{S}$ and then fine-tunes them on the target domain $\mathcal{T}$ [14, 15]. Similarly to what was observed in [14, 15], we expect that the performance of fine-tuning on $\mathcal{T}$ will further improve if we use a model trained on $\mathcal{S}$ with AT instead of a model trained with ST, even if the classification performance of the robust model on $\mathcal{S}$ is worse than the performance of the model trained with ST. The exact reason behind this improvement is still an open question, but it is related to the differences in the learned features between standard and robust models. Also, this improvement depends on the ϵ used in Eq. 1 during AT, and the value that leads to better accuracy may differ across tasks and domains. Smaller values for ϵ generally lead to better performance [15], but its value will be selected empirically.

Finally, the last two training combinations apply AT either on the *target* domain (ST→AFT) via FT or on *both* domains (AT→AFT). Considering the effect of AT on the features learned by a classifier, we will investigate how f_θ is affected when the transferred learned features from $\mathcal{S}$ are further filtered by AT on $\mathcal{T}$. Fig. 2 summarizes the training strategies under analysis, which will be compared in the next section.

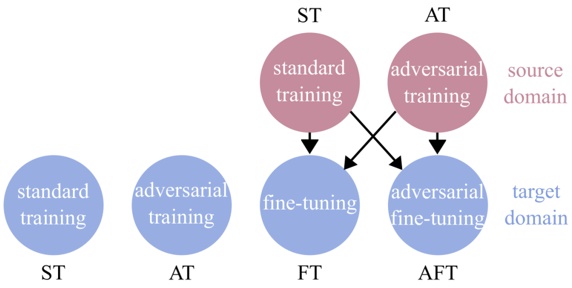

Fig. 2. The six training strategies analyzed in our experiments: independent standard training (ST) and adversarial training (AT) on the target domain, and four transfer learning strategies from source to target domain via fine-tuning (FT).

3. EXPERIMENTAL VALIDATION

3.1. Dataset

The task is to classify the filling level from a single RGB image. The CCM dataset [16] comprises of four views capturing under different backgrounds and illumination conditions cups and drinking glasses. The containers are transparent, translucent or opaque. The content is transparent (water) or opaque (pasta, rice). Each container stands upright on a surface or is being manipulated by a person. We only consider data of the public CCM repository, namely 4 cups and 4 drinking glasses.

From the CCM video data, we automatically sampled and then manually verified 10,216 frames of containers for which a pouring action was completed. To increase the variability in the sampled data, we selected frames where the container is completely visible or occluded by the person's hand, and under different backgrounds. For each frame, the final image is extracted by cropping only the region with the container using Mask R-CNN [28], followed by visual verification. Each crop is associated to an annotation of filling type and filling level (empty or filled at 50% or 90% of the capacity of the container), hand occlusion, and transparency of the container. We call this dataset Crop-CCM or C-CCM. Sample C-CCM images[1] are shown in Fig. 3.

To investigate the impact of the shape of a container on this task, we split C-CCM into train and test sets under three configurations, based on the container type. The first configuration (S_1) considers a champagne flute in the test set to further increase the shape variability of containers not previously seen in the train set. The second configuration (S_2) swaps a beer cup with a wine glass to analyze the influence of the stem of the wine glass. The last configuration (S_3) places all the containers with a stem in the train set, and the test set contains only cups without stem. Fig. 4 shows the three configurations and the number of samples for each container type.

3.2. Classifier and implementation choices

We use as classifier a ResNet-18 [17]. Note that we also conducted experiments using a ResNet-50 and a WideResNet-50 [29], and the findings are similar to the ones of ResNet-18. The discussion of the results will focus on ResNet-18 as it is the least complex network among the three. With ST we train the classifier on C-CCM, whereas with AT we train the classifier on images modified with ℓ_2

[1] Sampled images can be found at https://corsmal.eecs.qmul.ac.uk/filling.html

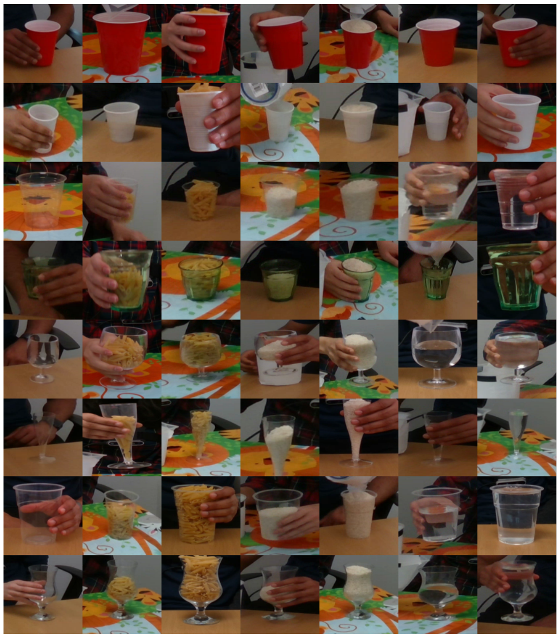

Fig. 3. Sample images (resized crops) from the CORSMAL Containers Manipulation dataset [16]. Each column shows different filling types and levels, and each row shows different backgrounds and hand occlusions.

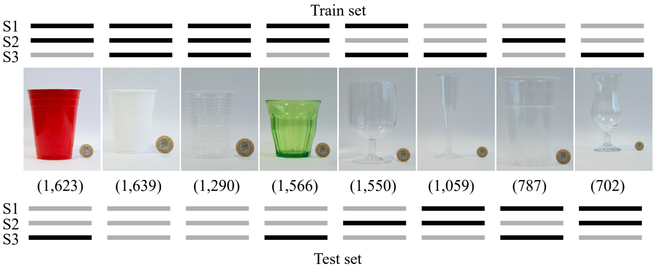

Fig. 4. Comparison of three train and test splits (S1, S2, S3) of the public containers from CCM for the shape analysis in the experiments. Black lines mean that the set of images belonging to that container are part of the train (test) set in the data split. The number of images for each container are shown in parentheses. Note the diversity in shape, color, texture, and transparency, as well as the size compared to the 1-pound coin (GBP) used as reference size.

adversarial perturbations ($p = 2$ in Eq. (1)) crafted with the 10-iteration PGD [20]. With the transfer learning strategies we fine-tune the available pre-trained models on C-CCM: for ST→FT and ST→AFT we use the pre-trained model provided by PyTorch [30], whereas for AT→FT and AT→AFT we use the robust models provided by [14].

For each strategy, we train or fine-tune the classifier for 30 epochs, using a cross-entropy loss [31] and stochastic gradient descent. The learning rate for updating the weights is set to 0.1 when training directly on C-CCM, and 0.005 when performing transfer learning. The learning rate decays linearly during training. Note that the models we evaluate are the ones obtained at the end of the training epochs (no early-stopping), while for dealing with class imbalances, the training images in a batch are randomly sampled with probabilities that are inversely proportional to the number of images of each class.

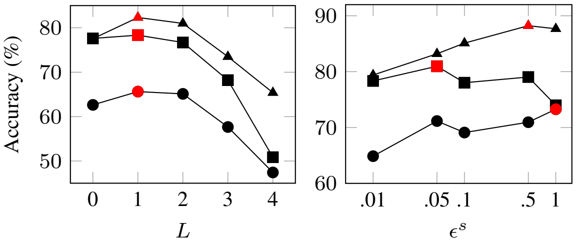

Fig. 5. Sensitivity analysis for the number of fixed layers L with ST→FT (left) and for the maximum amount of perturbation bound, ϵ^s, with AT→FT on test set of the three dataset splits: first split S_1 (■), second split S_2 (●), third split S_3 (▲). Red indicates the highest achieved accuracy. Note the different scale of the y-axis, and the logarithmic scale for the x-axis (right).

3.3. Sensitivity analysis

We perform a sensitivity analysis on the number of fixed layers (L) in fine-tuning with ST→FT, ST→AFT, AT→FT and AT→AFT; and to select the size of the bound for crafting the adversarial perturbation for AT, ST→AFT, AT→FT and AT→AFT. Note that we differentiate the bound ϵ for the source, ϵ^s, and target, ϵ^t, domain. Specifically, we perform the sensitivity analysis only for ϵ^s with AT→FT, and for each data split configuration we select the ϵ^s for which AT→FT achieves the highest accuracy. Then, based on these values of ϵ^s, for each data configuration we set $\epsilon^t = \epsilon^s$: since we use 10-iteration ℓ_2-PGD [20], performing a sensitivity analysis or a grid search on ϵ^t is computationally inefficient, as it is analogous to increasing almost $10\times$ the training epochs.

We first analyze the classification accuracy on the test sets of the three dataset splits when varying the number of fixed layers for ST→FT as $L = \{0, 1, 2, 3, 4\}$. Note that for a ResNet-18 classifier, a layer is a ResNet block of convolutions and batch normalization (see the original ResNet paper [17]). Since the target dataset is small, it is reasonable to fix the first layer ($L = 1$) in order to prevent the classifier from a possible overfitting [32]. Indeed, Fig. 5 (left) shows that the accuracy on the test set of all configurations (S1, S2, S3) is consistently higher for $L = 1$ (78.34%, 65.63%, 82.32%), while it gradually decays as L grows. This is also expected [32], since we allow fewer layers to be fine-tuned on the target datasets, and the classifiers then mostly use fixed features from ImageNet. Therefore, we set $L = 1$ for ST→FT as well as for ST→AFT, AT→FT, and AT→AFT.

By fixing $L = 1$, we analyze the classification accuracy of AT→FT when varying the size of the adversarial perturbation on the source domain, ϵ^s. Fig. 5 (right) shows that the highest achieved accuracy is different for each dataset configuration: 80.97% for S_1 with $\epsilon^s = 0.05$, 73.27% for S_2 with $\epsilon^s = 1$, and 88.23% for S_3 with $\epsilon^s = 0.5$. As mentioned in Sec. 3.3, we use these values ϵ^s also for ϵ^t when performing AT, ST→AFT, and AT→AFT. However, we observed that the model trained with ST→AFT is unable to converge (train accuracy around 45%) on S_2 for $\epsilon^t = 1$ and on S_3 for $\epsilon^t = 0.5$, while it successfully converges on S_1 for the smaller $\epsilon^t = 0.05$. We believe that this might be caused by the fact that AT with larger ϵ^t values eliminates many non-robust, yet useful, features transferred from ImageNet, and prevents the model from fitting the remaining features. Hence, we set $\epsilon^t = 0.05$ for ST→AFT across all dataset configurations for the rest of the experiments.

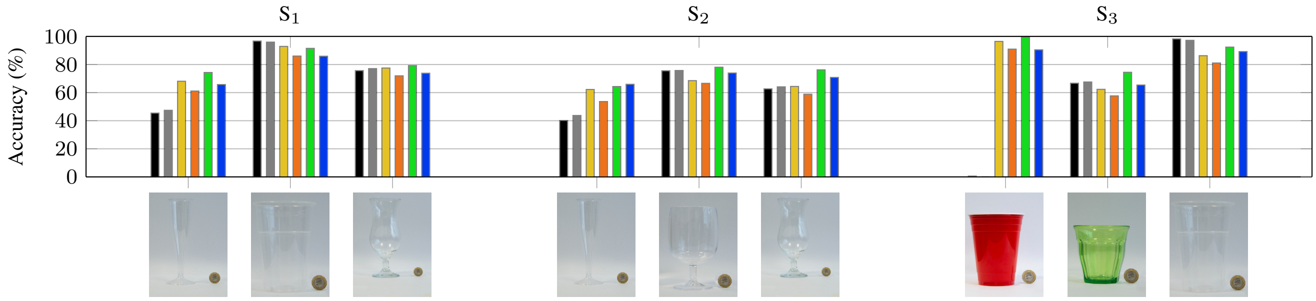

Fig. 6. Comparison of the per-container filling level classification accuracy (%) for the six training strategies. Note the different containers in the test set for each dataset split (see Fig. 4 for the train set of each split). Legend: ▬ ST, ▬ AT, ▬ ST→FT, ▬ ST→AFT, ▬ AT→FT, ▬ AT→AFT.

3.4. Results

Fig. 6 shows the filling level classification performance on the three configurations, S$_1$, S$_2$ and S$_3$, for all the training strategies. Constrained by the amount, and hence by the diversity, of training images, the differently trained classifiers could potentially develop biases or overfit to some features, such as the shape of a container. AT→FT achieved superior performance most of the times. With transfer learning, the features introduced from ImageNet (source domain) appear to decrease such biases, and enable the classifiers to identify features in the train set that are more generalizable. When combining transfer learning with AT at the source domain, the biases are modulated with the transferred features that are also filtered by AT, and the generalization of the classifier further increases. These results confirm that adversarial training improves transfer learning, even in the context of the challenging filling level classification task.

Overall, whenever the performance of ST is low, all transfer learning strategies lead to a significant improvement. On the contrary, whenever ST performs well, the contribution of transfer learning is insignificant, and sometimes it even decreases the final performance. Furthermore, applying AT on the *target* domain, either alone or combined with transfer learning, may even be harmful for the classifier.

For S$_1$, the accuracy of ST on the beer cup (middle) is already very high, and the other training strategies do not further improve it. This might be explained by the similar shape of the small transparent cup in the training set. On the other hand, the accuracy on the cocktail glass (right) is similar for all strategies, but lower than the one of the beer cup, with AT→FT performing slightly better than the rest of the training strategies. Although there is another container with a stem in the training set (wine glass), these accuracy levels might be due to the different shape above the stem that the cocktail glass has, compared to the wine glass. As for the champagne flute (left), the performance of ST and AT is quite low ($\sim$46%), which might be caused by the unique shape of the flute (narrowing towards the bottom) with respect to the shapes in the training set. However, the accuracy significantly improves with transfer learning. Especially AT→FT outperforms all the other strategies by $\sim$30 percentage points (pp).

For S$_2$, the accuracy of all strategies on the champagne flute (left) is similar to the one achieved on S$_1$. The accuracy on the cocktail glass (right) is much lower for most strategies ($\sim$10pp less compared to the performance on S$_1$), except AT→FT, which drops only by 3pp and again outperforms the rest of the strategies. The drop of the other strategies could be caused by the lack of a container with a stem in the training set. Finally, the performance on the wine glass (middle) is similar for most strategies, with AT→FT being again slightly better than the rest. Compared to the cocktail glass, the higher accuracy of all strategies on the wine glass could be caused by the similarity of its shape above the stem with the other transparent cups in the training set, despite the fact that no container with a stem is presented in the training set.

For S$_3$, the accuracy of ST on the beer cup (right) is high and the other training strategies do not improve it. Instead, the accuracy of ST on the green glass (middle), which has a different shape, is lower and reaches an accuracy of 66%. However, although ST→FT does not improve the accuracy, AT→FT significantly increases it (almost 10pp). The red cup (left) obtains the most interesting improvement compared to the 0.005% accuracy of ST: all transfer learning techniques achieve an accuracy above 90%, with AT→FT achieving 99.5% classification accuracy. By inspecting the predictions of ST and AT, the classifier assigned the label full (filling level: 90%) almost 99% of the times. In fact, predicting the *unknown* class is conceptually different from estimating the filling level, and it is more related to classifying non-transparent containers. In this sense, the features learned for transparent objects that are full with rice or pasta might be correlated with the features of the red cup.

4. CONCLUSION

We investigated how different training strategies impact the classification of the filling level of a container. Using adversarial training on the source dataset, ImageNet, followed by transfer learning on the target dataset, selected from the CORSMAL Containers Manipulations dataset, permits to consistently improve generalization to unseen containers. Our analysis demonstrates the possibilities of exploiting adversarial training for tasks that extend beyond classical image classification settings. As future work, we will explore other sources of biases that might be related to transparencies, occlusions, or the content, we will investigate alternative ways to avoid overfitting to features of the training data, and we will extend our analysis to other datasets and settings.

5. REFERENCES

[1] S. S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "ClearGrasp: 3D shape estimation of transparent objects for manipulation," in *Proc. IEEE Int. Conf. Robotics Autom.*, June 2020.

[2] R. Mottaghi, C. Schenck, D. Fox, and A. Farhadi, "See the glass half full: Reasoning about liquid containers, their volume and content," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct 2017.

[3] C. Schenck and D. Fox, "Visual closed-loop control for pouring liquids," in *Proc. IEEE Int. Conf. Robotics Autom.*, May 2017.

[4] C. Do, T. Schubert, and W. Burgard, "A probabilistic approach to liquid level detection in cups using an RGB-D camera," in *Proc. IEEE Int. Conf. Intell. Robot Syst.*, Oct. 2016.

[5] C. Do and W. Burgard, "Accurate pouring with an autonomous robot using an RGB-D camera," in *Int. Conf. Intell. Auton. Syst.*, July 2018.

[6] C. Schenck and D. Fox, "Reasoning about liquids via closed-loop simulation," in *Proc. Robotics: Science and Syst.*, July 2017.

[7] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Int. Conf. on Artificial Neural Networks*. 2018, vol. 11141 of *Lecture Notes in Computer Science*, pp. 270–279, Springer.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, L. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2009.

[9] M. Huh, P. Agrawal, and A. A. Efros, "What makes ImageNet good for transfer learning?," in *Neural Inf. Process. Syst. Workshop on Large Scale Computer Vision Systems*, Dec. 2016.

[10] S. Kornblith, J. Shlens, and Q. V. Le, "Do better ImageNet models transfer better?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2019.

[11] J. Xue, H. Zhang, and K. Dana, "Deep texture manifold for ground terrain recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018.

[12] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2009.

[13] A. Vedaldi, S. Mahendran, S. Tsogkas, S. Maji, R. Girshick, J. Kannala, E. Rahtu, I. Kokkinos, M. B. Blaschko, D. Weiss, B. Taskar, K. Simonyan, N. Saphra, and S. Mohamed, "Understanding objects in detail with fine-grained attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2014.

[14] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry, "Do adversarially robust ImageNet models transfer better?," in *Adv. Neural Inf. Process. Syst.*, Dec. 2020.

[15] F. Utrera, E. Kravitz, N. B. Erichson, R. Khanna, and M. W. Mahoney, "Adversarially-trained deep nets transfer better," in *Proc. Int. Conf. Learning Represent.*, May 2021.

[16] A. Xompero, R. Sanchez-Matilla, R. Mazzon, and A. Cavallaro, "CORSMAL Containers Manipulation," 2020, (1.0) [Dataset]. Queen Mary University of London. https://doi.org/10.17636/101CORSMAL1.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2016.

[18] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learning Represent.*, May 2015.

[19] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2016.

[20] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learning Represent.*, Apr. 2018.

[21] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learning Represent.*, Apr. 2014.

[22] A. Modas, S.-M. Moosavi-Dezfooli, and P Frossard, "SparseFool: A few pixels make a big difference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2019.

[23] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *Proc. Int. Conf. Learning Represent.*, May 2019.

[24] Z. Allen-Zhu and Y. Li, "Feature purification: How adversarial training performs robust deep learning," *arXiv:2005.10190*, May 2020.

[25] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, B. Tran, and A. Madry, "Adversarial robustness as a prior for learned representations," *arXiv:1906.00945*, June 2019.

[26] S. Santurkar, A. Ilyas, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Image synthesis with a single (robust) classifier," in *Adv. Neural Inf. Process. Syst.*, Dec. 2019.

[27] G. Ortiz-Jimenez, A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard, "Optimism in the face of adversity: Understanding and improving deep learning through adversarial robustness," *Proc. IEEE*, vol. 109, no. 5, pp. 635–659, May 2021.

[28] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017.

[29] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, Sept. 2016.

[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inf. Process. Syst.*, 2019.

[31] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014.

[32] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Adv. Neural Inf. Process. Syst.*, Dec. 2014.