MEOW: - Automatic Evolutionary Multi-Objective Concealed Weapon Detection

Anonymous Author(s)
Affiliation
Address
email

Abstract

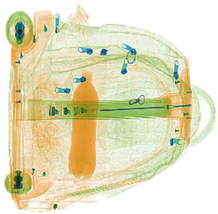
X-ray screening is crucial in ensuring the safety and security of publicly crowded areas. Since x-ray operators can be overwhelmed by the sheer amount of items they have to inspect, new computer vision-aided systems aim to reduce the operator's workload. In this work, we focus on one of the problems with developing such systems: naive application of already existing state-of-the-art methods for visual datasets does not necessarily yield satisfactory results. We use neural architecture search(NAS) and compare it to the current state-of-the-art approaches for SIXray (a famous and colossal X-ray dataset). We develop a heuristic technique to speed up the otherwise time-consuming process and make its use for industrial datasets feasible. We also devise an ensemble approach capable of utilising multiple discovered architectures simultaneously. Based on our results, AutoML shows excellent potential for such use cases, and thanks to our advancements, we manage to beat the state-of-the-art while keeping the NAS overhead to a minimum.

1 Introduction

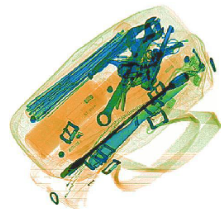
Concealed weapon detection through different screening procedures is a crucial part of security. (17; 35). As the need for such systems has significantly increased (32) many systems have emerged focusing on different techniques (31; 2). Attempts to address this problem date back more than two decades ago, connoting its complexity and arduousness (5). Most state-of-art algorithms in this domain focus on identifying if there is a threat or not (24; 27) which is a sub-optimal approach since different threats require different security protocols. Usually, millimetre wave detectors and metal detectors, etc.(30) are used to carry out concealed weapon detection. There are, however, specific use cases where computer vision algorithms are also employed to analyse the raw signals coming from these various media. One medium of interest in this work is the one which comes from X-Ray scans. These generated images significantly differ from the visual RGB images generated from CCTV or other digital cameras. Hence, they require specific preprocessing before they can be used as input to conventional computer vision algorithms (21). Moreover, the application of already established computer vision algorithms often yields sub-optimal results, which has led researchers to recognise the need for domain-specific architectures and procedures (23; 22).

Developing new architectures is a complex process, requiring additional tuning to make them fit a specific problem involving many computational and human resources. Researchers have been trying to develop techniques invariant to changes in data, distributions, and other external settings (10; 13). The idea of such approaches is to harness the tremendous computational power available nowadays and put it to work instead of a team of people with different domain expertise needing to tackle the problem manually. Automated machine learning (AutoML) allows all of this to be possible, and recently, the progress in the field has led to some remarkable breakthroughs like (29; 16; 37) . One of the significant caveats of AutoML, however, is the extreme computational requirements

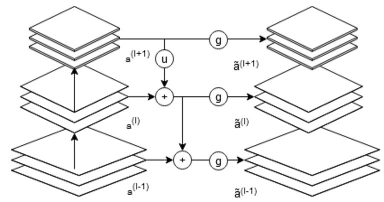
Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a) SIXray benign sample



(b) SIXray sample with multiple overlapping threats



(c) Heirarchical refinement process from (23). a^l signifies the activation a at layer l, g is the CHR denoise function. The $+$ denotes concatenation and $\tilde{a}$ denotes the filtered activation after g is applied.

Figure 1: SIXray (23) samples (1a, 1b) and CHR methodology (1c)

(16; 29) A novel method to deal with the high computational demand of these algorithms is the use of optimisation heuristics in the form of proxy scores which promise a significant reduction in resource requirements (19; 1). In this paper, we explore the potential application of AutoML approaches for concealed weapon detection using two x-ray datasets which feature different types of treats and scenarios. The contributions of the paper are as follows:

1. Novel application of AutoML state-of-the-art methods to concealed weapon detection.
2. Multi-objective optimisation of multiple heuristic proxy scores simultaneously and a 200x speed up of a multi-objective AutoML algorithm.
3. A novel ensemble approach that utilises the predictions of multiple models discovered by a multi-objective NAS to make a final prediction.

2 Related work

There are numerous ways to detect concealed weapons detection such as millimetre wave technology (8; 14) , radar (36) and most often X-ray machines (23; 3; 25) . Recently, computer vision algorithms have been tasked to automate the vast majority of the x-ray inspections and aid the machine operators as much as possible by automatically flagging potential threats or enhancing the images to make a threat more visible (23; 3). This has led to an influx of x-ray datasets featuring concealed threats in recent years (22; 23; 12). These datasets are designed to foster the upcoming automation of security systems, which would allow for security personnel to more efficiently and effectively prevent potential disasters (20) . One such dataset is SIXray (23). It contains multiple threats that can appear simultaneously and exhibit unique properties representing real-world data with over a million images. One of the most prominent of these properties is the occlusion (sample in Figure 1b) (34).

Some methods attempt to use novel attention mechanisms designed to negate these properties' adverse effects on the performance of models. Such techniques are the hierarchical refinement in (23) (Figure 1c) and Selective Dense Attention Network in (34). Alternatively, some additional preprocessing steps are presented in (22). Although we do not employ any of these techniques in our work, they are compatible with our findings and can be utilised to improve our results even further.

In these scenarios, the standard is to use convolutional neural networks(CNNs), and often, already well-performing architectures with visual images are naively applied to these x-ray problems (3; 9) . A caveat with this transfer learning approach is that, as mentioned in (20) insights and techniques used with visual datasets may be suboptimal when used with other media types such as X-rays.

Creating entirely new models and techniques specifically for the domain is highly time-consuming, and detaching it from the advancements in the visual domain would hinder the joint progress of the fields. A workaround is to use transferable automated machine learning(AutoML) methods (29). In particular, neural architecture search(NAS) is not bound to the specific data or the architectural paradigms and tricks used with visual datasets (7).

A common problem with a large portion of the state-of-the-art NAS algorithms is that they are only viable for small dimensional datasets (such as MNIST and CIFAR 10) (29; 16; 28). This is first because they require a tremendous amount of computational resources. Also, in some of them, the data loading and processing techniques are intertwined with the approach, so they cannot effortlessly scale to high-dimensional x-ray data in a "plug-and-play" fashion (10; 29; 15; 34). As the interest in the field has grown, more and more research has attempted to fill this research gap and provide alternative solutions to deal with the problem, such as (6; 37; 4).

In particular, RAMOSS (7) authors have designed their approach modularly, allowing easy integration with datasets such as SIXRay. They claim the approach provides the best balance between computational time and performance and supports multi-objective optimisation, which we intend to use in our work. However, most of these algorithms' computational time is spent on training each constructed architecture for a certain amount of epochs to evaluate its performance on a held-out validation set and compare it to the rest of the produced architectures, which is sub-optimal (1). Thankfully, some recent works have provided shortcuts that integrate performance approximation proxy scores. Even though they do not display a perfect correlation with the trained performance, they are believed to be a reliable approximation (1). Such scores are the NASWOT score (19) and the SYNFLOW score (33) . The latter was originally designed to be a score used for pruning architectures. However, a recent study (1) showcased that it might be an effective performance proxy score when used with NAS approaches.

3 Methodology

In this paper, we leverage the RAMOSS algorithm (7) with the newly surfaced proxy scores to establish the feasibility of using NAS approaches in concealed weapon detection and test how well such approaches scale to real-world use cases. Towards this goal, we benchmark a constrained search for architectures of RAMOSS to some of the most popular architectures in the domain.

The main issue with using NAS approaches like RAMOSS is the computational time. Given that SIXray has over a million data points, we recognised the need to find a heuristic evaluation method. Motivated by the promising results of *"Zero-cost NAS"*(1), come up with a rapid multi-objective neuroevolution approach capable of finding optimal architectures in less than an hour of computational time. More specifically, we use two of the proxy scores presented in (1) - NASWOT (19) and SYNFLOW (33). In contrast to *"Zero-cost NAS"*, we do not use a cumulative score to aggregate over the selected proxies. Instead, we use them as separate objectives and try to optimise for both simultaneously.

To convert the NASWOT and SYNFLOW scores from a pruning metric to an architecture performance estimation proxy, we reformulate them similarly to (1). The NASWOT score in our study is calculated using: $\ln\left(|\det(K)|\right)$ where K contains all per-neuron pruning scores, which in turn are calculated as:

$$\sum_{i=0}^{l}\left(\left(g_i(X)*g_i(X)^T\right)+\left((1-g_i(X))*\left(1-g_i(X)^T\right)\right)\right) \tag{1}$$

where l is the number of layers, $g_i(X)=1_{|f_i(X)>1|}$ is the normalised output $f_i(X)$ for layer i given input X and X is the selected batch of data fed to the model.

The SYNFLOW score in our work is computed similarly. The original per-neuron score is obtained by: $S_p\theta=\frac{\partial\ell}{\partial\theta}\odot\theta$ where ℓ is the loss function used and θ denotes the parameters of the network. Using the scores, the authors of (1) reformulate them to generate a score for each neuron in the network by using the following operation: $S_n=\sum_{N}^{i=0}S_p(\theta)_i$ where N is the total number of parameters. We take this one step further by aggregating these scores into an optimisation objective by applying the following operation where n is all the neurons in each layer i of all layers l:

$$ln(\frac{\sum_{i=0}^{l}\frac{\sum_{j=0}^{n}ln(S_j+1)}{n}}{l}) \tag{2}$$

As part of our work, we conduct an ablation study exploring different backbone architectures to establish a baseline model to compare the newly produced RAMOSS models. With this ablation study, we aim to establish which architectures work well for our given problem, if any architectures significantly outperform others and to explore why. We believe this can help search for the "ideal" architecture to solve the present problem.

Table 1: SIXRAY results

	Gun	Knife	Wrench	Pliers	Scissors	Mean
ResNet34	89.71	85.46	62.48	83.50	52.99	74.83
ResNet50	90.64	87.17	64.31	85.78	61.58	77.87
ResNet101	87.65	84.26	69.33	85.29	60.39	77.38
Inception-v3	90.05	83.80	68.11	84.45	58.66	77.01
DenseNet	87.36	87.71	64.15	**87.63**	59.95	77.36
RAMOSS-s1	94.93	89.47	67.48	86.13	89.29	85.46
RAMOSS-ens	**95.51**	**94.04**	**77.34**	76.12	**96.34**	**87.87**

To conduct our study, we use an Adam optimiser with a learning rate of 0.001, beta 1 of 0.9, beta 2 of 0.99 and epsilon of 0.0000001. We discard the top layers and add the same five dense layers for classifying the produced features to ensure a fair comparison. The loss function used is conventional binary cross-entropy. We train each architecture for 50 epochs and evaluate it using the test set. We do not use augmentation during our study to keep as many control variables as possible. We also shuffle and fetch the dataset using the same random seeds.

After setting up the corresponding data loaders and computing the benchmarks, the general process we follow can be described by the following steps: **1.** We specify any hyperparameters. **2.** We run the RAMOSS algorithm (7) for g generations with a population size p. The number of maximum convolutional layers used in all experiments in this work is set to 35 **3.** We take the Pareto front of produced solutions and train the architectures for the same 50 epochs with the other state-of-the-art methods. **4** Using the predictions of the top-performing discovered models as inputs, we construct an ensemble model. **5** We evaluate all architectures and the ensemble model on the held-out test set. We report the best performing architecture score as well as the score from the ensemble model ("RAMOSS-s1" and "RAMOSS-ens" in Table 1 respectively).

4 Results and Discussion

We report results based on average precision. We use precision for SIXray not only to match the original SIXray paper (23) but also because of the massive imbalance in the dataset. From Table 1, it becomes evident that from the state-of-the-art architectures, the best ones are ResNet50 and ResNet101, which are mostly on par with DenseNet. Interestingly, most of the architecture displays consistent performance throughout all threats, but it is worth mentioning that the number of images with the class "Scissors" is the most undersampled class.

Interestingly, DenseNet is the best architecture for detecting Pliers, which is, in fact, the class with the most samples, excluding the benign ones. However, the best-recognised classes seem to be the Gun and the Knife throughout most architectures. We notice a general increase in performance going from ResNet34 to ResNet50. At the same time, from ResNet50 to ResNet101, the performance drops except for a single class. This phenomenon can be explained by the depth of ResNet101, which may prevent it from converging for the same 50 epochs as ResNet50. Also, it is possible that the model is overparameterised for this particular problem in line with the findings of (26) and (18).

Strikingly, the models produced from RAMOSS captured the imbalance better than the state-of-the-art benchmark architectures judging by the undersampled classes and the overall precision. This connotes that the multi-objective optimisation within RAMOSS is searching for optimal architectures and implicitly enforces this discriminative ability. We attribute this implicit objective to the way the proxy scores work. As discussed above, both proxy scores are positively influenced by the distance between samples' activations. Hence, the proxy scores implicitly reward architectures capable of "seeing" distinct or simply unlike features, which in our case, helps the algorithm produce architectures capable of capturing the difference between the different classes. The best RAMOSS model significantly outperforms the state-of-the-art models used for SIXray, even with the class-balanced hierarchical refinement (CHR) technique discussed in (23). The large gap between the automatically discovered architecture and the state-of-the-art calls for future work to focus on the importance of designing specific architectures for data from different domains. Moreover, the RAMOSS-s1 architecture uses just above 11 million parameters, less than half of the 23.5 million parameters used by ResNet50. One possible explanation of how the RAMOSS model outperforms the rest of the architectures becomes apparent when taking a closer look at the architecture itself (See Figure 2). One of the key steps in the CHR (23) is the hierarchical refinement which utilises low and high-level features by concatenating intermediate activations and then filtering out noisy information based on the signals of the activations of the next layer. Each selected layer for feature extraction, thus becomes a separate stream of $\tilde{a}^l$ that is fed into an auxiliary classifier $f^l(\tilde{a}^l_n; \xi^l) = y^l_n$ where ξ^l is the hashing vectoriser of the selected layer and finally all y^l_n are averaged to obtain the final output y. Looking back at the architecture discovered by RAMOSS in Figure 2, some large skip connections can be seen. After the first convolutional layer, a skip connection is made to much lower dimensional representations. Then, the signals are added inside an auxiliary feature extraction arm, which is effectively similar to what the CHR accomplishes with the separated streams. Moreover, the architecture seems to be composed of ResNet-like blocks with various degrees of skip connection depth, which, as we can see from Table 1, seems to be working better than InceptionNet or DenseNet-like blocks.

Interestingly, the models produced from RAMOSS excelled at recognising different classes, which we attribute to the multi-objective optimisation underneath the algorithm. As we wanted to utilise a better portion of the generated front of solutions rather than just one(in the form of RAMOSSs1), we decided to select four architectures with the highest contributing hypervolume (as in (7)). Then, we designed a classifier which uses the outputs of these architectures $y^1, y^2...y^n$ in order to make a final prediction y_{ens}, which we can then evaluate using the actual labels $y*$. The architecture of the ensemble model is custom designed, and it composes of a custom layer, the aim of which is to do a weighted average of the predicted probabilities of $y^1, y^2...y^n$.

It naturally achieves the best overall scores; however, it is beaten by DenseNet only in terms of the most out-of-distribution upsampled class in the testing set - the Pliers. Thanks to our ensemble approach, we beat the state-of-the-art by more than 10%, which is more than five times better improvement than the one coming from the incorporation of CHR.

Since we used only 20 generations with a population size of 20, it is fair to assume that the algorithm did not successfully explore the enormous search space. Thus we recognise that this caveat needs to be addressed with an ablation study over these two hyperparameters in future work.

It is worth mentioning that using the advancements listed in the previous section (by using the proxy score heuristic), we managed to reduce the time needed to run RAMOSS to under 1 GPU hour for a population size of 20 for 20 generations. This is a drastic improvement over the original time required listed in the original paper (7), making it one of the fastest NAS runs. Moreover, since we do not even train the architectures during the discovery phase, we have essentially made the algorithm compatible with machines without GPUs because the CPU can handle inference on most machines. This makes the research field immensely more accessible for new practitioners and feasible to use in many new domains and industry settings, as demonstrated by our experiments. Our code base can be found on *URL withheld to comply with double-blind submission*.

5 Conclusion and Future work

In this work, we demonstrate that neural architecture search methods, which have limited use in industry, can be utilised to address real-world problems. We pick a specific modular AutoML approach, and its use with new datasets is straightforward, which is not usually the case. Although our ensemble strategy seems to generate some promising results, future work can explore using the collective knowledge of the generated networks to conduct knowledge distillation (11). We manage to beat the state-of-the-art, and our results undeniably avow that there is a need to make AutoML more practical by designing the systems so that they are disentangled from the dataset used for proof of concept and scalable to real-world scenarios. We believe that making such algorithms run faster by utilising heuristic performance estimation to substitute the regular and highly resource-greedy evaluation presents a unique opportunity, which we explore. We also modify a well-performing multi-objective approach to use multiple proxy scores to speed up the architecture search and showcase how these proxies can be used in conjunction with multi-objective optimisation to beat state-of-the-art architectures. What is more, we design an ensemble approach that successfully utilises multiple of the optimally produced set of architectures. In summary, our results and the exorbitantly low search time with a NAS method(1 hour) without the need to use a GPU are a testament to the untapped potential of the use of AutoML in industrial applications.

A Appendix 1: RAMOSS-s1 architectures

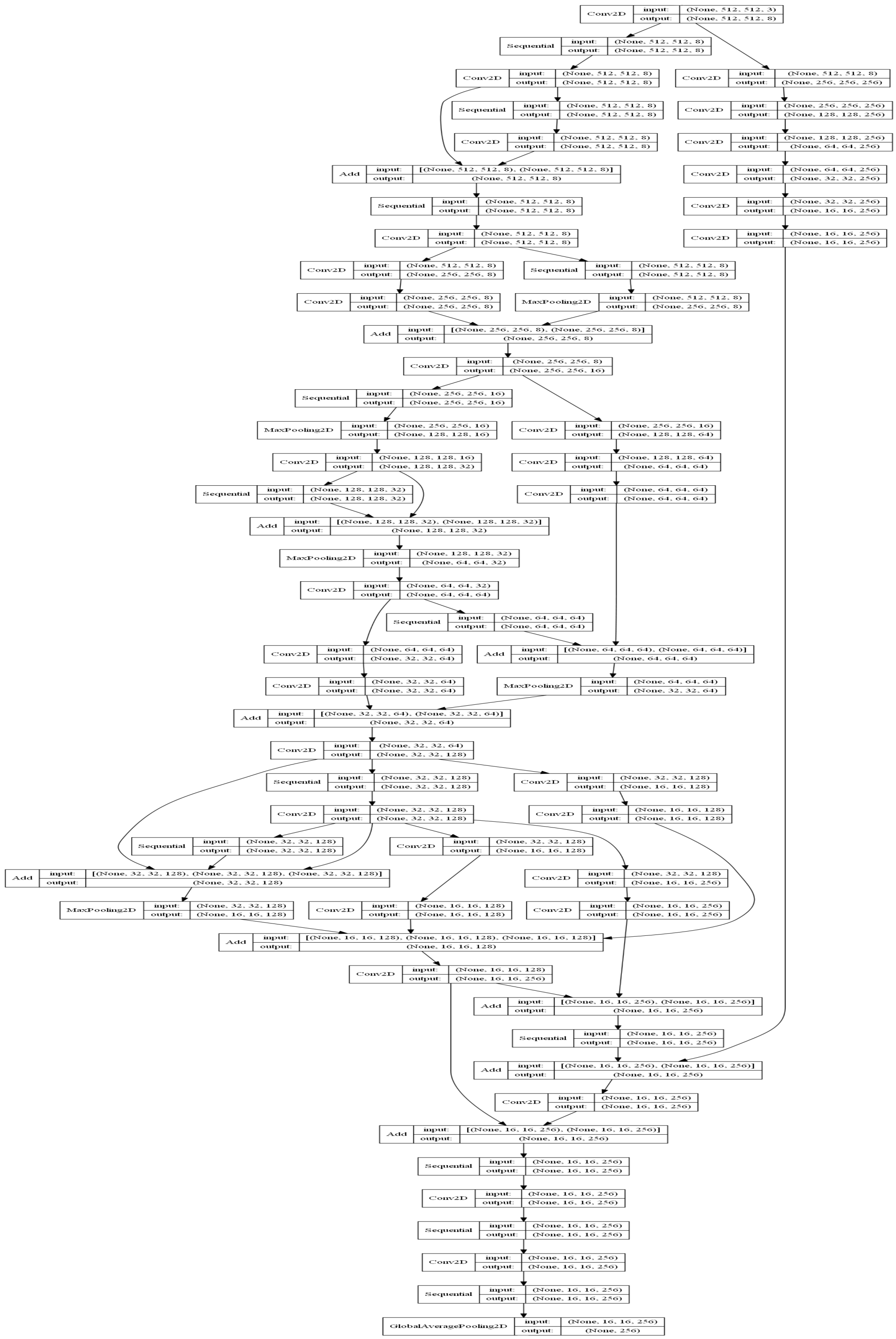


Figure 2: RAMOSS-s1 architecture for SIXray (truncated after the last convolution for legibility)

References

[1] Abdelfattah, M.S., Mehrotra, A., Dudziak, Ł., Lane, N.D.: Zero-cost proxies for lightweight nas. arXiv preprint arXiv:2101.08134 (2021)

[2] Agarwal, S., Kumar, B., Singh, D.: Non-invasive concealed weapon detection and identification using v band millimeter wave imaging radar system. In: 2015 National Conference on Recent Advances in Electronics & Computer Engineering (RAECE). pp. 258–262. IEEE (2015)

[3] Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semi-supervised anomaly detection via adversarial training. In: Asian Conference on Computer Vision. pp. 622–637. Springer (2018)

[4] Assunção, F., Lourenço, N., Machado, P., Ribeiro, B.: Fast denser: Efficient deep neuroevolution. In: European Conference on Genetic Programming. pp. 197–212. Springer (2019)

[5] Bartley, W.A., Cohen, M.A.: The effect of concealed weapons laws: An extreme bound analysis. Economic Inquiry 36(2), 258–265 (1998)

[6] Dimanov, D., Balaguer-Ballester, E., Singleton, C., Rostami, S.: Moncae: Multi-objective neuroevolution of convolutional autoencoders. arXiv preprint arXiv:2106.11914 (2021)

[7] Dimanov, D., Singleton, C., Rostami, S., Balaguer-Ballester, E.: Ramoss-resource aware multi-objective semantic segmentation through neuroevolution. In: UK Workshop on Computational Intelligence. Springer (2022)

[8] Goenka, A., Sitara, K.: Weapon detection from surveillance images using deep learning. In: 2022 3rd International Conference for Emerging Technology (INCET). pp. 1–6. IEEE (2022)

[9] Hassan, T., Shafay, M., Akçay, S., Khan, S., Bennamoun, M., Damiani, E., Werghi, N.: Meta-transfer learning driven tensor-shot detector for the autonomous localization and recognition of concealed baggage threats. Sensors 20(22), 6450 (2020)

[10] He, X., Zhao, K., Chu, X.: Automl: A survey of the state-of-the-art. Knowledge-Based Systems 212, 106622 (2021)

[11] Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)

[12] Isaac-Medina, B.K., Willcocks, C.G., Breckon, T.P.: Multi-view object detection using epipolar constraints within cluttered x-ray security imagery. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 9889–9896. IEEE (2021)

[13] Karmaker, S.K., Hassan, M.M., Smith, M.J., Xu, L., Zhai, C., Veeramachaneni, K.: Automl to date and beyond: Challenges and opportunities. ACM Computing Surveys (CSUR) 54(8), 1–36 (2021)

[14] Li, S., Wu, S.: Low-cost millimeter wave frequency scanning based synthesis aperture imaging system for concealed weapon detection. IEEE Transactions on Microwave Theory and Techniques (2022)

[15] Lu, Z., Whalen, I., Boddeti, V., Dhebar, Y., Deb, K., Goodman, E., Banzhaf, W.: Nsga-net: a multi-objective genetic algorithm for neural architecture search. arXiv preprint arXiv:1810.03522 (2018)

[16] Lu, Z., Whalen, I., Boddeti, V., Dhebar, Y., Deb, K., Goodman, E., Banzhaf, W.: Nsga-net: neural architecture search using multi-objective genetic algorithm. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 419–427 (2019)

[17] Mahajan, R., Padha, D.: Detection of concealed weapons using image processing techniques: A review. In: 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC). pp. 375–378. IEEE (2018)

[18] Malach, E., Shalev-Shwartz, S.: Is deeper better only when shallow is good? Advances in Neural Information Processing Systems 32 (2019)

[19] Mellor, J., Turner, J., Storkey, A., Crowley, E.J.: Neural architecture search without training. In: International Conference on Machine Learning. pp. 7588–7598. PMLR (2021)

[20] Mery, D.: Computer vision for x-ray testing. Switzerland: Springer International Publishing.– 2015 10, 978–3 (2015)

[21] Mery, D., Arteta, C.: Automatic defect recognition in x-ray testing using computer vision. Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017 pp. 1026–1035 (2017)

[22] Mery, D., Riffo, V., Zscherpel, U., Mondragón, G., Lillo, I., Zuccar, I., Lobel, H., Carrasco, M.: Gdxray: The database of x-ray images for nondestructive testing. Journal of Nondestructive Evaluation 34(4), 1–12 (2015)

[23] Miao, C., Xie, L., Wan, F., Su, C., Liu, H., Jiao, J., Ye, Q.: Sixray: A large-scale security inspection x-ray benchmark for prohibited item discovery in overlapping images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2119–2128 (2019)

[24] Morris, T., Chien, T., Goodman, E.: Convolutional neural networks for automatic threat detection in security x-ray images. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 285–292. IEEE (2018)

[25] Nguyen, H.D., Cai, R., Zhao, H., Kot, A.C., Wen, B.: Towards more efficient security inspection via deep learning: A task-driven x-ray image cropping scheme. Micromachines 13(4), 565 (2022)

[26] Pasupa, K., Sunhem, W.: A comparison between shallow and deep architecture classifiers on small dataset. In: 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE). pp. 1–6. IEEE (2016)

[27] Petrozziello, A., Jordanov, I.: Automated deep learning for threat detection in luggage from x-ray images. In: International Symposium on Experimental Algorithms. pp. 505–512. Springer (2019)

[28] Qin, X., Wang, Z.: Nasnet: A neuron attention stage-by-stage net for single image deraining. arXiv preprint arXiv:1912.03151 (2019)

[29] Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y.L., Tan, J., Le, Q.V., Kurakin, A.: Large-scale evolution of image classifiers. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 2902–2911. JMLR. org (2017)

[30] Rostami, S.: Preference focussed many-objective evolutionary computation. Ph.D. thesis, Manchester Metropolitan University (2014)

[31] Rostami, S., O'Reilly, D., Shenfield, A., Bowring, N.: A novel preference articulation operator for the evolutionary multi-objective optimisation of classifiers in concealed weapons detection. Information Sciences 295, 494–520 (2015)

[32] Sheen, D.M., McMakin, D.L., Hall, T.E.: Three-dimensional millimeter-wave imaging for concealed weapon detection. IEEE Transactions on microwave theory and techniques 49(9), 1581–1592 (2001)

[33] Tanaka, H., Kunin, D., Yamins, D.L., Ganguli, S.: Pruning neural networks without any data by iteratively conserving synaptic flow. arXiv preprint arXiv:2006.05467 (2020)

[34] Wang, B., Zhang, L., Wen, L., Liu, X., Wu, Y.: Towards real-world prohibited item detection: A large-scale x-ray benchmark. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5412–5421 (2021)

[35] Xue, Z., Blum, R.S.: Concealed weapon detection using color image fusion. In: Proceedings of the 6th international conference on information fusion. vol. 1, pp. 622–627 (2003)

[36] Zhang, J.: A novel intelligent radar detection network. In: Journal of Physics: Conference Series. vol. 2181, p. 012056. IOP Publishing (2022)

[37] Zhou, D., Zhou, X., Zhang, W., Loy, C.C., Yi, S., Zhang, X., Ouyang, W.: Econas: Finding proxies for economical neural architecture search. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 11396–11404 (2020)

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See our Methodology and Results
   (b) Did you describe the limitations of your work? [Yes] The discussion of the limitations is scattered across our Related works, Results, Discussion and Conclusion sections
   (c) Did you discuss any potential negative societal impacts of your work? [No] We have not recognised negative societal impacts arising directly from our work
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] All needed scripts to rerun the experiments are prepared in a zip file which can be shared with reviewers. Unfortunately, OpenReview does not support uploading zip files as supplementary materials, so these will be available on request, and a link to our repository will be disclosed upon acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All hyperparameters are specified in our methodology and in even more detail in the code.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] This is one of the limitations of our current study, and we are working on improving it.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We have used an NVidia 3090 for our experiments, and the RAMOSS run took less than 1 GPU hour, as mentioned in the methodology. The retraining of the architectures, on average, takes around 3 hours to retrain for 50 epochs.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A] Non of the assets require licensing.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The URL to our codebase (which would be made public upon acceptance). We plan to submit an anonymised version of the code as supplementary materials.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]