JCSE Journal of Computing Science and Engineering

Regular Paper

Journal of Computing Science and Engineering,
Vol. 19, No. 1, March 2025, pp. 24-35

A Survey of Efficiency Requirements in Computer Vision

Hyeon Choe and Dongsu Kang*
Department of Computer Engineering, Korea National Defense University, Nonsan, Korea
choi0249@gmail.com, dasekang@korea.kr

Abstract

Computer vision has been automatically performing challenging tasks such as image classification, object detection and recognition, and image segmentation due to the rapid development in the industry and performing beyond human capabilities. As social expectations and interests grow in computer vision, there is a growing tendency to utilize computer vision technologies in several other fields such as defense, finance, education, and other related industries. However, many computer vision products are often underutilized for several reasons after they are developed. The important problem is that researchers are less interested in the knowing the cause of this. Even though many studies have been conducted to improve computer vision technology, studies that focus on quality of the computer vision products that people use are insufficient. Hence, this paper discusses the quality of computer vision products. Through a requirement engineering approach, we analyze the efficiency requirements that mostly affect the quality of computer vision products and identify the trends in related technologies. Through this, we present the relevant technical limitations and ways to overcome them.

Category: Computer Graphics / Image Processing

Keywords: Computer vision; Efficiency requirements; Non-functional requirements; Domain knowledge

I. INTRODUCTION

Computer vision (CV) techniques are performing well in image recognition and classification, object detection, image segmentation, and more, with advancements in machine learning (ML) that contribute to CV. However, many CV products are developed and rarely used because the customer's requirements are not properly understood in the early stages of product development [1].

Since CV is a technology that predicts outcomes based on data, it requires a different approach to quality control than traditional software (SW). However, requirements engineering (RE) studies for CV products are very scarce. As a result, even when CV products are developed, they are often underutilized.

This paper analyzes techniques related to efficiency requirements among non-functional requirements (NFR) for improving CV product quality. It draws meaningful conclusions that should be of interest to CV developers and researchers.

In Section II, we explain the research questions (RQ) that were set and what scientific process was used to conduct the survey. Section III discusses why efficiency requirements are important for CV products. In Section IV, we identify and characterize the trends in technologies that can improve efficiency. In Section V, we provide a logical explanation for the importance of domain knowledge based on the results of the technology trend analysis.

Open Access http://dx.doi.org/10.5626/JCSE.2025.19.1.24 http://jcse.kiise.org

Received 12 September 2024; **Accepted** 30 May 2025

*Corresponding Author

pISSN: 1976-4677 eISSN: 2093-8020

II. RESEARCH METHOD

For a systematic literature review, this paper follows the method suggested by Kitchenham [2]. This method is more useful when the number of papers is large, as it efficiently verifies that the papers have been thoroughly researched.

Our study has been performed in a sequence as shown in Fig. 1.

A. Research Questions

1) RQ1: Importance of Efficiency

From previous studies, we found that there is a lack of research on RE compared to the societal interest in CV products. Through various surveys, we found that the study of efficiency requirements should be prioritized to improve CV quality. RQ1 explains why we focus on analyzing efficiency requirements.

2) RQ2: Techniques for Efficiency

As a result of RQ1, we found that efficiency is a critical factor in improving CV quality. RQ2 identifies trends in technology to improve efficiency.

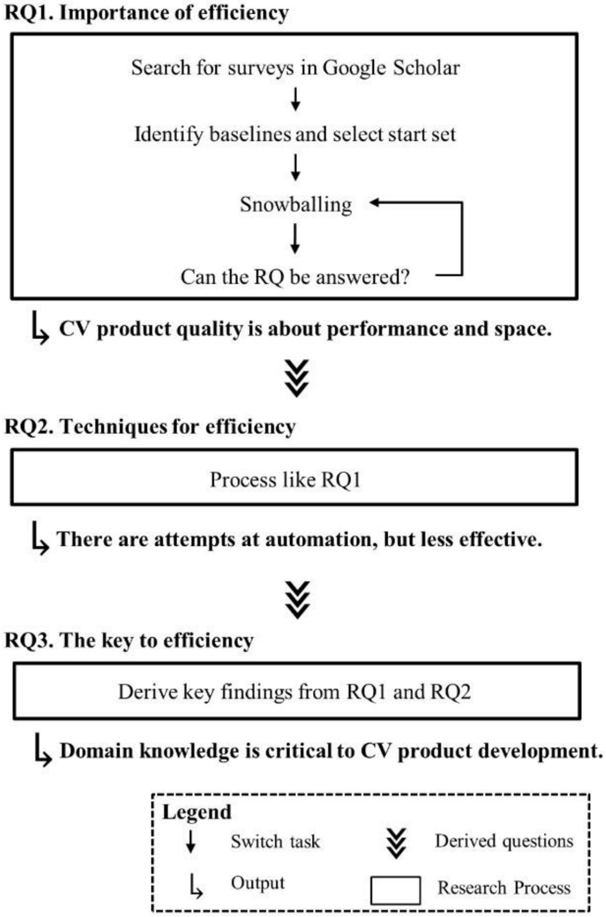
RQ1. Importance of efficiency
Search for surveys in Google Scholar
Identify baselines and select start set
Snowballing
Can the RQ be answered?
CV product quality is about performance and space.
RQ2. Techniques for efficiency
Process like RQ1
There are attempts at automation, but less effective.
RQ3. The key to efficiency
Derive key findings from RQ1 and RQ2
Domain knowledge is critical to CV product development.
Legend
Switch task
Derived questions
Output
Research Process

Fig. 1. Chain of research questions and process.

Table 1. Search keyword combination

	Keywords
1	Survey; Review; Analysis
2	Machine learning; Computer vision; Artificial intelligence
3	Requirement engineering; MLOps; Data cleaning; Data augment; Continuous learning; Light weight

3) RQ3: The Key to Efficiency

In analyzing the technology trends in RQ2, we found an important fact: the importance of domain knowledge has become even greater than that in traditional SW development. In RQ3, we explain this logically.

B. Search Strategy

We first checked the surveys and summarized the relevant information. This is because it gives us indirect access to many papers in a short time, with little effort. We primarily used Google Scholar to search for papers from a variety of sources. The search keywords are shown in Table 1 [3-18]. Each line is joined by an OR condition. For example, keywords = line1 AND line2 AND line3, where line1 = survey OR review OR analysis.

Search keywords allow for many paper results. Therefore, we set the following exclusion rules. Non-English papers were excluded. We excluded papers with duplicate content. Papers that were published in journals after the conference were removed based on the author's name. We excluded papers that were not directly related to CVs. The time period for the papers was set to 2017–2024. We prioritized highly cited articles in our search.

For each survey, we set the core content and related papers as a startset. From these startsets, we used a snowballing approach [19] to expand the scope of the study from baseline to the present. Table 2 summarizes the surveys.

III. IMPORTANCE OF EFFICIENCY

A. Previous Studies

Research on RE for ML first started in 2019 and has been going on ever since [3]. However, till now, there is no clear practice for ML development [4, 20]. Recent survey results show that the number of papers on NFRs for ML is very limited [4]. The development of RE is slow due to the lack of academic interest on the same.

The researchers are trying to find answers through interviews, case studies, and literature reviews. It is not a quantitative analysis, but more of a conceptual conclusion.

Table 2. Summary of survey on CV quality

Category	Study	Year	Method	Main contributions
Computer vision	Guo et al. [5]	2021	M	-Categorize and analyze attention by how it works: channel, spatial, branch, temporal, etc. to see the difference between each technique
	Bi et al. [6]	2022	L	-Domain knowledge needs make content extraction from CV difficult -Identify evolutionary computer vision techniques to easily extract content
Software engineering	Masuda et al. [7]	2018	L	-Identify testing practices to improve ML quality -More research is needed in areas such as fault localization and prediction
	Kumeno [8]	2019	L	-Broadly describes challenges for ML applications and attempts to map them to knowledge areas in Software Engineering Body of Knowledge (SWEBOK)
	Atoum et al. [9]	2021	L	-Analyzed 19 validation technologies and 27 tools -Define the relationship between validation technologies and SW domains
Requirement engineering	Vogelsang & Borg [3]	2019	I	-First RE for ML paper, interviews with 4 data scientists -Categorize RE for ML activities based on criteria classified in the SWEBOK
	Pei et al. [10]	2022	L	-Identifies RE for ML challenges focusing on stakeholders (business experts, required engineers, SW engineers, domain experts, and data scientists)
	Gjorgjevikj et al. [4]	2023	C	-Snowballing paper reviews and case studies to draw conclusions -No basic practices for eliciting requirements for ML
Data cleaning	Chalapathy & Chawla [11]	2019	L	-Systematically categorize anomaly detection techniques -Analyze the effectiveness of each technology
	Pang et al. [12]	2020	M	-Categorize outlier detection from a modeling perspective and check techniques for problem identification
Data Augment	Zhang et al. [13]	2020	M	-Showing how using Mixup helps with model robustness -Explain why it is robust against multiple types of adversarial attacks
	Park et al. [14]	2022	M	-It provides a theoretical analysis of mixed sample data augmentation and provides a high level of understanding of how different design choices work
Continuous learning	Guo et al. [15]	2017	M	-Evaluate the performance of different post-processing calibration methods -Temperature scaling works well
	Van de Ven & Tolias [16]	2019	C	-Categorize Continual Learning methods into three scenarios -See the effects of applying techniques in different scenarios
Light weight	Sanchez-Iborra & Skarmeta [17]	2020	E	-Investigate the TinyML framework available for integrating ML algorithms within microcontroller units
	Ray [18]	2022	L	-Identify tool sets and components to support TinyML -Present the state of the art on the TinyML framework

L: literature review, C: case study, E: experiment, M: mathematical analysis, I: interview.

Masuda et al. [7] analyzed technical keywords for ML quality. It shows that there is a wide range of technologies being developed, but it only covers specific parts of the field, not all of them.

The main stakeholders of CV development are data scientists [3, 4, 10]. Data scientists are important because securing the right data is key to ML development. Kumeno [8] emphasize that data acquisition is the most difficult part of the entire process.

To analyze ML testing methods, Zhang et al. [21] divided ML into workflows, components, and properties, and found that correct data collection is the most important. Previous studies have shown that data is the most important thing in improving the quality of CV products. So we identify efficiency requirements that are directly related to data.

B. The Challenge of CV Development

Requirements are critical to the entire SW engineering process; most SW failures are caused by incomplete and inaccurate requirements (64.7%) [9]. Currently, there is no requirement elicitation practice for CV development.

In [22], there is no standardized model for MLOps.

Therefore, each industry plans and applies its own development process. With MLOps, it is important to identify the initial business problem and identify the correct dataset. MLOps are represented as end-to-end processes, where the workflow is repetitive and automation is emphasized. If the wrong data is used, it affects the whole thing because the process is chained.

We know that every industry has its own applicable workflows. The important thing is that continuous learning is essential for CV product development in any industry, which is why it is necessary to use clean data.

C. Efficiency Requirements for Quality

Software requirements are mainly classified into functional requirements and non-functional requirements. NFR includes everything related to quality. NFRs can be classified in various ways depending on their perspectives. In this paper, we interpret efficiency requirements by referring to software engineering textbooks [23]. Efficiency requirements are separated into performance requirements and space requirements.

To improve the quality of ML products, it is important to pay attention to data. The data development process is divided into data collection, labeling, preparation, reduction, and augmentation. Of these, data cleaning and augmentation require a significant amount of engineering work [24]. Successful artificial intelligence (AI) essentially requires research in two areas: first, the ability to clean up dirty datasets into clean labels, and second, the ability to automatically generate new, high-quality data. This is the part that directly affects performance.

Meanwhile, CV has become multimodal and very large in size. The latest trend is for companies to miniaturize these large models. In addition, lightweight models can be run on mobile devices, expanding their application area. This is why space requirements are needed for recent CV products. For this reason, we considered efficiency requirements to be the key NFR. Performance and space are important to achieve efficiency in CV products.

IV. TECHNIQUES FOR EFFICIENCY

A. Performance Requirements

Improving CV performance is directly related to data. Techniques for this include how to improve data quality and quantity and retraining models with new data. Check technology trends for this.

1) Data Cleaning for Single Class

Anomaly detection is a technique for checking whether a training dataset contains anomalous data. Unnecessary training with anomalous data reduces the performance of the model. Anomaly detection can also be applied to various fields such as cyber-intrusion detection, medical anomaly detection, and video surveillance [11].

When training, anomaly detection is classified as supervised, semi-supervised, and unsupervised, depending on whether the data is labeled or not. We categorized the techniques according to the data features used and identified the features as shown in Table 3.

Supervised anomaly detection ensures that all labels are present in the given data. It is more accurate than the other methods. However, in the real world, where anomaly detection is applied, there is much less anomalous data, resulting in a data imbalance.

Semi-supervised anomaly detection uses two methods to overcome this imbalance: using only normal data and using a small number of labeled data. Since it is a real-world application, this is a relatively active area of research [25]. The density-based method is simple to implement and highly effective. For example, SPADE [26] collects

Table 3. Taxonomy of anomaly detection

Category	Data	Description	Method
Supervised			
Binary classification	O,X	Apply when labelled for normal and abnormal data	-Over/under sampling -Weight optimization -Hybrid method
Semi-supervised			
One-class classification	O	Use only normal data to detect differences between the normal and outlier	-Density-based -Classification-based -Reconstruction-based, etc.
Distribution mismatch	O,X,U	Use when domains between labeled/unlabeled data are different	-Uncertainty-aware-self distillation -Open-set-semi-supervised learning
Unsupervised	U	Detect outliers using only unlabeled data without label information	-Deep unsupervised anomaly detection -Self-supervised, Refine, Repeat

O: normal, X: abnormal, U: unlabeled

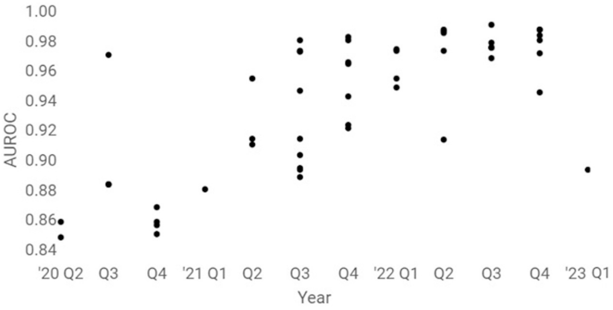

Fig. 2. Performance graph of anomaly detection using MVTec [31].

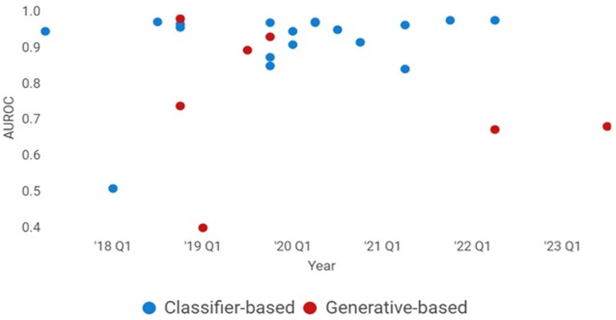

Fig. 3. Out-of-distribution performance graph [39]. IN: CIFAR10, OUT: SVHN.

the features of normal data and creates a global average pooling and uses it as an anomaly score. PaDiM [27] identifies anomaly data by calculating the Mahalanobis distance between normal and anomalous data distributions.

Semi-supervised anomaly detection studies have mainly used MNIST and CIFAR-10 datasets [28]. It is somewhat unnatural because the datasets were not originally created for anomaly detection. Since the numbers 0 and 1–9 are easily distinguishable by humans, there is a limitation that even if the dataset shows good performance, it is not practical [29].

For practical research, researchers have begun using the MVTec anomaly detection dataset introduced in CVPR2019 [30]. As shown in Fig. 2, studies using the MVTec dataset continue to improve the performance [31]. The state-of-the-art method achieved an AUROC score of 99.8%, indicating excellent detection performance [32].

Despite the sophistication of semi-supervised anomaly detection, it still suffers from low anomaly detection recall rates on real-world data [12]. Reducing false positives and increasing recall rates still require data scientists' involvement and validation.

Unsupervised anomaly detection is a label-free learning method that assumes that most of the data is normal. Autoencoder-based methods extract features of normal data through important information in the compressed data during the encoding process and use them to distinguish between normal and abnormal data [33]. Unsupervised anomaly detection does not require a labelling process, but it has the disadvantage of poor judgement accuracy and sensitivity to hyperparameters.

Semi-supervised anomaly detection is well researched, and the next challenge is to increase the performance of unsupervised anomaly detection with unlabeled data.

2) Data Cleaning for Multiple Class

CV products often require the ability to classify multiple classes, not just a single class. In the open world, there are various classes that CV has not learned. This can lead to performance degradation and sustainability degradation. Data that is not trained on multiple classes is called out-of-distribution. The methods for handling out-of-distribution data can be broadly classified into classifier-based methods and generative-based methods.

Hendrycks and Gimpel [29] is the first paper to address out-of-distribution detection and propose an experimental protocol using maximum softmax probability. The proposed metrics, AUROC curve and area under the precision-recall (AUPR) curve, have been widely used in subsequent papers.

The follow-up paper, ODIN [34], uses the previously proposed methods of temperature scaling and input preprocessing appropriately to find the out-of-distribution without learning. In [35], under the assumption that the features of the trained network follow a class-conditional Gaussian distribution, Mahalanobis distance is obtained and used as a confidence score. Outlier exposure [36] improves the performance of out-of-distribution detection by adding independent auxiliary data sets.

Generative-based methods assume that feeding out-of-distribution samples into a density estimator will result in lower likelihood values. In early studies, the likelihood values obtained from the generative model did not distinguish between in-distribution and out-of-distribution datasets well.

Choi et al. [37] observed an inconsistent phenomenon when training with CIFAR-10 and testing with traffic sign and SVHN; where the generative model had higher likelihood values for SVHN, an out-of-distribution dataset, than for CIFAR-10, an in-distribution dataset. Later, they found that the higher the complexity of the input image, the lower the likelihood was observed, and proposed a method to utilize the complexity of the input as the out-of-distribution score [38].

Fig. 3 shows the performance trends of generative-based and classifier-based methods on the CIFAR-10 (in-distribution) and SVHN (out-of-distribution) datasets [39]. The performance is similar, but the generative-based method greatly depends on the dataset. For generative-based performance, the dataset used must be as realistic as possible [36].

Recently, it has been recognized that studies on out-of-distribution detection are not suitable for the real world. Therefore, a new concept of out-of-model-scope has been defined to reflect more appropriate problem settings. For realistic out-of-model-scope use, developers need to study the domain of the product.

3) Data Augmentation

With the development of deep learning, CV has a large number of parameters compared to the amount of data. This leads to the problem of overfitting the training data. Overfitted models have significantly lower prediction performance on new data not seen in training [40].

The method of data augmentation should be chosen according to the purpose of CV. For example, classification typically produces a single correct label for an image, whereas segmentation involves assigning a class label to each pixel. To improve segmentation performance, image data augmentation techniques are commonly used. These include basic manipulations (e.g., flipping, rotation) and deep learning-based approaches. Both are increasingly being integrated into meta-learning frameworks that can be applied without requiring domain-specific knowledge.

By manipulating the images by flipping, rotating, zooming, etc., they have the effect of imposing variety on the data. It's also very simple to apply. These methods improve model performance by diversifying the data just by manipulating the original image [41].

However, flipping and rotation are inappropriate for semantic segmentation and stereo matching, which require spatial information. For example, a flipped image of the sky is not real-world data. However, a military fighter jet may need a flipped sky image. Therefore, these methods should be considered in the domain of the CV product.

Mix-up is a technique that generates new samples through linear interpolations from two data samples. This technique not only produces a smooth decision boundary and good prediction performance, but also shows robustness against adversarial attacks [40]. It is still being used in various research fields because it guarantees good generalization performance even with simple operating principles [14, 42].

CV tends to become overconfident as its prediction performance improves [15]. This makes it difficult to control product quality during development. Studies have shown that calibrating with Mix-up is effective against over-confidence.

Thulasidasan et al. [43] explain that Mix-up help improve performance because they learn new samples each time. Zhang et al. [15] proved mathematically that Mix-up can reduce the upper bound of adversarial loss. Despite the effectiveness of Mix-up, the technique has limitations. In the process of utilizing the data, unnecessary information can be used for training [44]. To improve this limitation, researchers are actively working on finding information in the important regions of the data [45-48].

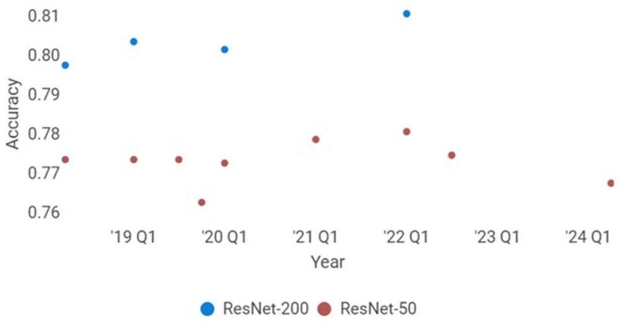

Fig. 4. Data augmentation performance graph [58].

Ensuring the safety of AI systems requires adversarial training, which improves their ability to resist attacks by incorporating adversarial examples during the learning process [49]. However, since adversarial examples are data that does not exist in the real world, they show distributional mismatches with clean data, leading to poor performance [50], i.e., they can be less accurate compared to using only clean data [50-52].

Xie et al. [53] developed AdvProp using auxiliary batch normalization to control the distribution of different inputs. This is the first benchmark that improves the performance of the model using only adversarial examples. Despite this improvement, the increase in training computation makes adversarial training unsuitable for industrial-scale production [54].

Recently, researchers have been investigating ways to improve generalization and robustness simultaneously. To reduce the bottleneck caused by batch normalization, it has also been proposed to remove the batch normalization and proceed with adversarial training [55].

Mix-up and adversarial training, as well as the basic methods, require an understanding of the nature of the training data and the domain context to perform well.

AutoAugment is the first attempt to study how to effectively enrich data without domain knowledge [56]. The disadvantage of this approach is the increase in computation. It is difficult to expect a significant performance improvement for many computations.

To overcome this problem, advances such as PBA, Fast AA, RandAugment, and Uniform-Augment have been made. However, there is no significant performance improvement compared to TrivialAugment, which fixes the variants in a single operation [57].

Fig. 4 shows the improvement of data augmentation using ImageNet data [58]. Dividing by model shows a slow rate of improvement in accuracy. Realistically, data augmentation in CV requires domain knowledge.

4) Continuous Training

CV suffer from poor inference performance on untrained data. Since the trained model does not accumulate knowledge about new data, it is difficult to respond to the reality.

To solve these problems, Incremental learning is used. Incremental learning refers to maintaining or improving the performance of multiple sequential tasks with one

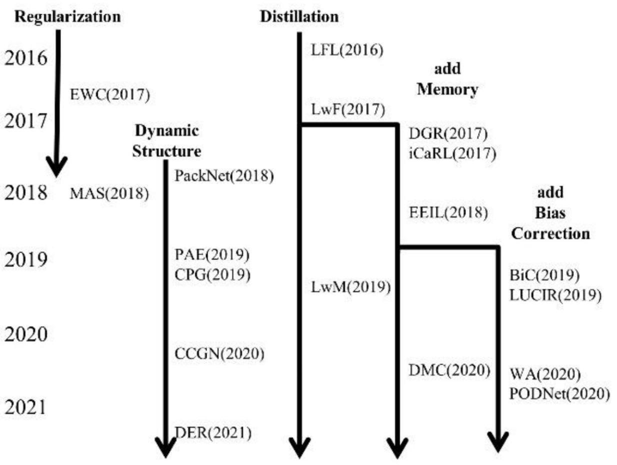

Fig. 5. Evolution and taxonomy of incremental learning.

network. It accepts a data stream and accumulates it in a knowledge base [59].

The challenge of Incremental learning is to minimize forgetting, which is the loss of information about previous tasks [16]. The methods of incremental learning are being developed by dividing them into regularization, distillation, and dynamic structure as shown in Fig. 5.

Incremental learning was first developed as a regularization method. When learning a new task, it regulates the change of model parameters to avoid forgetting the existing knowledge.

The first paper, elastic weight consolidation (EWC) [60], uses the fisher information to reduce the amount of change in the parameters when learning a new task, and as a result, it preserves existing knowledge. Memory aware synapses (MAS) [61] uses a gradient to identify important knowledge and regulate the change.

Distillation methods transfer knowledge from past tasks to new tasks. The first paper, less-forgetting learning (LFL) [62], applies Euclidean loss so that the old and new models have the same features. This method has a memory issue: as the number of tasks increases, the size of the train set that needs to be stored increases. To solve this problem, the Distillation+Memory method has been proposed, which stores and uses only some important data.

iCaRL [63] uses parts of previous tasks as exemplars to learn new tasks. In classification, classes are separated by the closest distance to the mean of the class feature vector. Deep generative replay (DGR) [64] utilizes generative adversarial networks to implement previous task data. This data is called exemplar, which is found to be highly biased when learning new tasks. BiC finds that removing the bias alone improves performance. Methods to correct the bias were subsequently studied [65].

Dynamic structure is a method of learning by manipulating the model, such as pruning or masking. PackNet [66] prunes the model to leave parameters for learning the next task. Packing-and-expanding (PAE) [67] expands after pruning to accommodate more parameters.

In the early days, regularization-based methods were mainly studied, and more recently, a mix of methods has been developed [68]. DualNet [69] introduces a concept from brain science to distinguish between fast-net and slow-net. Slow-net extracts general features from the data and passes them to fast-net. The fast-net learns by utilizing the received features and the current task.

Despite various attempts and achievements, incremental learning studies only assume simple scenarios. However, more diverse scenarios are needed to continuously train CV products in reality.

B. Space Requirements

As giant multimodal services such as GPT-4 [70] have become a trend in the industry, developers are increasingly concerned about inference latency, server load, and power. In addition, as the use of IoT devices such as drones and robots increases, CV products that require limited computational power and memory power are required [17, 18].

Currently applicable lightweight technologies are light weight architecture, pruning, and quantization. Although there has been a lot of progress, academic results still only suggest possibilities and are not realizable by themselves, so multiple attempts are needed to develop ML products.

1) Light Weight Architecture

CV uses convolution for feature extraction. Light weight architecture is to lighten the network itself by using convolution. CV models like AlexNet and ResNet utilized convolution to significantly improve performance, but suffered from the problem of requiring a lot of computation.

Therefore, various structures have been studied to reduce the number of networks. Techniques for changing the initial model structure have been extended to the study of reducing the amount of computation and the number of variables while training channels separately.

SqueezeNet [71] proposed the fire module structure, which reduced the number of parameters by nearly 50 times compared to AlexNet. Xception [72] uses a depth-wise separable convolution layer to compute cross-channel correlations and spatial correlations independently. In MobileNet [73], a conventional convolutional filter is first depth-wise convolved on a channel-by-channel basis, and the result is divided into a pointwise convolution performed on a single point. When the length of the filter is 3, the gain is about 9 times.

The choice of architecture determines the space requirements of the product, so developers need to choose a good architecture. The problem is that the optimal architecture depends on the dataset and task. CV products that use neural networks have a slow learning speed, so it is difficult to find the optimal architecture.

To solve this problem, neural architecture search directly searches for network structures in a defined search space. MNasNet [74] explicitly includes speed information in the main reward function of the search algorithm in a mobile environment for architecture exploration to find a model that balances accuracy and speed, and can find a model that runs 2.4 times faster than the existing NasNet. These lightweight architecture methods are constantly being researched and developed because they can solve fundamental computational challenges in networks.

2) Quantization and Binarization

Quantization and binarization are aimed at reducing the number of floating points in a traditional neural network.

Quantization reduces the size of the model by expressing weight values as 16-bit floating point or 8-bit integer instead of 32-bit floating point by lowering the bit band [75]. Quantization can be used in conjunction with pruning techniques as they do not harm each other's accuracy [76]. Most deep learning frameworks, such as TensorFlow and MXNet, support 16-bit floating point and 8-bit integer quantization, making it easy and fast to use in the industry.

Binarization is a technique that converts the inputs between the weights and layers of a neural network into binary values of -1 or +1 depending on the sign, which greatly compresses the capacity and computation compared to conventional neural networks using floating points [77]. On the other hand, the low bit-width greatly limits the range of numbers that can be represented, which leads to loss of accuracy. Therefore, the goal in performing quantization is to minimize the loss of accuracy.

Stock et al. [78] proposed a binarization technique for the weights of a model and then introduced a method to binarize both the weights and activation outputs, which allows to replace matrix multiplication operations which are essential for model training and inference, with bitwise operations.

3) Pruning

Pruning starts with the observation that not all parameters in a model have the same impact on inference. Han et al. [79] proposed to reduce the total number of parameters by setting a certain threshold, removing neurons with lower values and their connections layer by layer, and repeating the retraining. The result was an accuracy loss of only 0.1%p (42.78% vs. 42.77%), but a nearly 9-fold reduction in the number of parameters, from 61M to 6.7M.

The accuracy of over-parameterized networks can be improved by pruning [80]. Tung and Mori [81] applied a quantization technique that prunes unnecessary parameters, clusters them, and replaces the existing parameters with the centroid of the cluster. This improved the accuracy of the original ResNet-50 model by only 0.6% while reducing the size of the original model from 102.5 MB to 6.7 MB, which is about 15 times lighter.

Subsequent studies have focused on fine-tuning the neural network in such a way that the accuracy can be increased through a retraining process after weight pruning. In addition to the general weighting approach, researchers are also working on compressing the model by selecting channels and pruning unnecessary channels [76].

V. THE KEY TO EFFICIENCY

In RQ2, techniques for improving efficiency requirements were analyzed. Various approaches are being attempted and many achievements are being made. However, there is a limit to performance improvement, including the case with automation. From this, it is clear that quality improvement is only possible with domain knowledge.

A. Domain Knowledge

To improve the cost-effectiveness of CV products, it is important to make it easy to collect the right data. To

Table 4. Need for domain knowledge

	Explain the need for each technology
Data cleaning	-Unsupervised anomaly detection does not require a labeling process, but it is less accurate. -In order to apply realistic Out-of-Model-Scope detection, it is necessary to study the environment and users of CV products. -When using generative-based methods, the dataset must be as realistic as possible to improve the performance.
Data augmentation	-For semantic segmentation or stereo matching that requires information about location, utilize appropriate techniques that take into account the domain. -Mix-up uses information about important areas of the data. -Adversarial training is a trade-off between accuracy and robustness, so generalization is necessary by checking image information.
Incremental learning	-Requires domain knowledge of the CV product to train in various scenarios.
Light weight architecture	-Lightweight models by choosing the right architecture for their domain without sacrificing performance.

train an CV with the desired performance, it is necessary to preprocess the collected data to create high-quality training data. To do this, it is important to know the meaning contained in the data and how to reorganize it to achieve the desired goal. Domain knowledge is expertise and experience in a domain where the CV product is used. Table 4 describes the need for domain knowledge.

Utilizing domain knowledge in the CV development process has several benefits. First, it can be used by transforming the data into a form that better represents its features. It can also reduce the variance of the input data through preprocessing. By utilizing a variety of pre-trained models, model development time can be reduced.

B. Future for CV Product

It's no exaggeration to say that 2023 was the year of generative AI, with leading AI models such as ChatGPT, a conversational AI service developed by OpenAI, and Stable Diffusion, which advances the performance of image AI, gaining public attention.

It's 2025 now, and AI is becoming a part of human life by being fully adopted and commercialized in businesses. In recent years, "human-centered AI" has become a buzzword in business. AI should be a means of putting humanity first, expanding human capabilities and improving the well-being of individuals and society.

Meanwhile, requirements should be documented and synthesized from stakeholder opinions. There are many different modeling languages to express them. It is necessary to develop a modeling language that reflects the characteristics of AI and humans. In the future, human-computer interaction/user experience (HCI/UX) professionals will play an increasingly important role, and more research is needed on user testing.

VI. CONCLUSION

Despite the growing societal interest and expectations on AI, there is a lack of research on the quality of CV products, so we conducted a study to find out how to improve the quality of CV products. We studied what we should focus on to improve the quality of CV products and concluded that developers should be aware of the domain knowledge related to the product.

This paper contributes to future research in this under-researched area by suggesting what is needed to manage CV product quality. We expect that as CV products are utilized more and more in the future, research on quality will become increasingly interesting. We hope that this study will provide important preliminary research on improving the quality of ML products.

CONFLICT OF INTEREST

The authors have declared that no competing interests exist.

REFERENCES

1. A. Cam, M. Chui, and B. Hall, "Global AI Survey: AI proves its worth, but few scale impact," 2019 [Online]. Available: https://www.mckinsey.com/featured-insights/artificial-intelligence/global-ai-survey-ai-proves-its-worth-but-few-scale-impact.
2. B. Kitchenham, "Procedures for performing systematic reviews," Keele University Technical Report TR/SE-0401, Keele, UK, 2004.
3. A. Vogelsang and M. Borg, "Requirements engineering for machine learning: perspectives from data scientists," in *Proceedings of 2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, Jeju, South Korea, 2019, pp. 245-251. https://doi.org/10.1109/REW.2019.00050
4. A. Gjorgjevikj, K. Mishev, L. Antovski, and D. Trajanov, "Requirements engineering in machine learning projects," *IEEE Access*, vol. 11, pp. 72186-72208, 2023. https://doi.org/10.1109/ACCESS.2023.3294840
5. M. H. Guo, T. X. Xu, J. J. Liu, Z. N. Liu, P. T. Jiang, T. J. Mu, et al., "Attention mechanisms in computer vision: a survey," *Computational Visual Media*, vol. 8, no. 3, pp. 331-368, 2022. https://doi.org/10.1007/s41095-022-0271-y
6. Y. Bi, B. Xue, P. Mesejo, S. Cagnoni, and M. Zhang, "A survey on evolutionary computation for computer vision and image analysis: past, present, and future trends," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 1, pp. 5-25, 2023. https://doi.org/10.1109/TEVC.2022.3220747
7. S. Masuda, K. Ono, T. Yasue, and N. Hosokawa, "A survey of software quality for machine learning applications," in *Proceedings of 2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, Vasteras, Sweden, 2018, pp. 279-284. https://doi.org/10.1109/ICSTW.2018.00061
8. F. Kumeno, "Software engineering challenges for machine learning applications: a literature review," *Intelligent Decision Technologies*, vol. 13, no. 4, pp. 463-476, 2019. https://doi.org/10.3233/IDT-190160
9. I. Atoum, M. K. Baklizi, I. Alsmadi, A. A. Otoom, T. Alhersh, J. Ababneh, J. Almalki, and S. Alshahrani, "Challenges of software requirements quality assurance and validation: a systematic literature review," *IEEE Access*, vol. 9, pp. 137613-137634, 2021. https://doi.org/10.1109/ACCESS.2021.3117989
10. Z. Pei, L. Liu, C. Wang, and J. Wang, "Requirements engineering for machine learning: a review and reflection," in *Proceedings of 2022 IEEE 30th International Requirements Engineering Conference Workshops (REW)*, Melbourne, Australia, 2022, pp. 166-175. https://doi.org/10.1109/REW56159.2022.00039
11. R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: a survey," 2019 [Online]. Available: https://arxiv.org/abs/1901.03407.

12. G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: a review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, article no. 38, 2021. https://doi.org/10.1145/3439950
13. L. Zhang, Z. Deng, K. Kawaguchi, A. Ghorbani, and J. Zou, "How does mixup help with robustness and generalization?," 2020 [Online]. Available: https://arxiv.org/abs/2010.04819v1.
14. C. Park, S. Yun, and S. Chun, "A unified analysis of mixed sample data augmentation: a loss function perspective," *Advances in Neural Information Processing Systems*, vol. 35, pp. 35504-35518, 2022.
15. C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," *Proceedings of Machine Learning Research*, vol. 70, pp. 1321-1330, 2017.
16. G. M. Van de Ven and A. S. Tolias, "Three scenarios for continual learning," 2019 [Online]. Available: https://arxiv.org/abs/1904.07734.
17. R. Sanchez-Iborra and A. F. Skarmeta, "Tinyml-enabled frugal smart objects: challenges and opportunities," *IEEE Circuits and Systems Magazine*, vol. 20, no. 3, pp. 4-18, 2020. https://doi.org/10.1109/MCAS.2020.3005467
18. P. P. Ray, "A review on TinyML: state-of-the-art and prospects," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 4, pp. 1595-1623, 2022. https://doi.org/10.1016/j.jksuci.2021.11.019
19. C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, London, UK, 2014, pp. 1-10. https://doi.org/10.1145/2601248.2601268
20. P. Haindl and R. Plosch, "Focus areas, themes, and objectives of non-functional requirements in DevOps: a systematic mapping study," in *Proceedings of 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Portoroz, Slovenia, 2020, pp. 394-403. https://doi.org/10.1109/SEAA51224.2020.00071
21. J. M. Zhang, M. Harman, L. Ma, and Y. Liu, "Machine learning testing: survey, landscapes and horizons," *IEEE Transactions on Software Engineering*, vol. 48, no. 1, pp. 1-36, 2020. https://doi.org/10.1109/TSE.2019.2962027
22. D. Kreuzberger, N. Kuhl, and S. Hirschl, "Machine learning operations (MLOps): overview, definition, and architecture," *IEEE Access*, vol. 11, pp. 31866-31879, 2023. https://doi.org/10.1109/ACCESS.2023.3262138
23. I. Sommerville, *Software Engineering*, 9th ed. Boston, MA: Pearson, 2011.
24. D. Zha, Z. P. Bhat, K. H. Lai, F. Yang, Z. Jiang, S. Zhong, and X. Hu, "Data-centric artificial intelligence: a survey," 2023 [Online]. Available: https://arxiv.org/abs/2303.10158.
25. Y. Liang, J. Zhang, S. Zhao, R. Wu, Y. Liu, and S. Pan, "Omni-frequency channel-selection representations for unsupervised anomaly detection," *IEEE Transactions on Image Processing*, vol. 32, pp. 4327-4340, 2023. https://doi.org/10.1109/TIP.2023.3293772
26. J. Yoon, K. Sohn, C. L. Li, S. O. Arik, and T. Pfister, "SPADE: semi-supervised anomaly detection under distribution mismatch," 2022 [Online]. Available: https://arxiv.org/abs/2212.00173.
27. T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: a patch distribution modeling framework for anomaly detection and localization," in *Pattern Recognition*. Cham, Switzerland: Springer, 2021, pp. 475-489. https://doi.org/10.1007/978-3-030-68799-1_35
28. L. Mariani, M. Pezze, and D. Zuddas, "Recent advances in automatic black-box testing," *Advances in Computers*, vol. 99, pp. 157-193, 2015. https://doi.org/10.1016/bs.adcom.2015.04.002
29. D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," 2016 [Online]. Available: https://arxiv.org/abs/1610.02136v1.
30. P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, C. "MVTec AD: a comprehensive real-world dataset for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 9592-9600. https://doi.org/10.1109/CVPR.2019.00982
31. J. Liu, G. Xie, J. Wang, S. Li, C. Wang, F. Zheng, and Y. Jin, "Deep industrial image anomaly detection: a survey," *Machine Intelligence Research*, vol. 21, no. 1, pp. 104-135, 2024. https://doi.org/10.1007/s11633-023-1459-z
32. K. Batzner, L. Heckler, and R. Konig, "EfficientAD: accurate visual anomaly detection at millisecond-level latencies," 2023 [Online]. Available: https://arxiv.org/abs/2303.14535v1.
33. P. Bergmann, S. Lowe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," 2018 [Online]. Available: https://arxiv.org/abs/1807.02011v1.
34. S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," 2017 [Online]. Available: https://arxiv.org/abs/1706.02690v1.
35. K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," *Advances in Neural Information Processing Systems*, vol. 31, pp. 7167-7177, 2018.
36. D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," 2018 [Online]. Available: https://arxiv.org/abs/1812.04606v1.
37. H. Choi, E. Jang, and A. A. Alemi, "Waic, but why? Generative ensembles for robust anomaly detection," 2019 [Online]. Available: https://arxiv.org/abs/1810.01392.
38. J. Serra, D. Alvarez, V. Gomez, O. Slizovskaia, J. F. Nunez, and J. Luque, "Input complexity and out-of-distribution detection with likelihood-based generative models," 2019 [Online]. Available: https://arxiv.org/abs/1909.11480v1.
39. N. Karunanayake, R. Gunawardena, S. Seneviratne, and S. Chawla, "Out-of-distribution data: an acquaintance of adversarial examples-a survey," 2024 [Online]. Available: https://arxiv.org/abs/2404.05219.
40. H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: beyond empirical risk minimization," 2017 [Online]. Available: https://arxiv.org/abs/1710.09412v1.
41. C. Summers and M. J. Dinneen, "Improved mixed-example data augmentation," in *Proceedings of 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, 2019, pp. 1262-1270. https://doi.org/10.1109/WACV.2019.00139
42. J. Liu and Z. Liu, "YOLOv5s-BC: an improved YOLOv5s-based method for real-time apple detection," 2023 [Online]. Available: https://arxiv.org/abs/2311.05811.
43. S. Thulasidasan, G. Chennupati, J. A. Bilmes, T. Bhattacharya, and S. Michalak, "On mixup training: improved calibration

and predictive uncertainty for deep neural networks," *Advances in Neural Information Processing Systems*, vol. 32, pp. 13888-13899, 2019.
44. S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, South Korea, 2019, pp. 6022-6031. https://doi.org/10.1109/ICCV.2019.00612
45. J. Park, J. Y. Yang, J. Shin, S. J. Hwang, and E. Yang, "Saliency grafting: Innocuous attribution-guided mixup with calibrated label mixing," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, pp. 7957-7965, 2022. https://doi.org/10.1609/aaai.v36i7.20766
46. D. Walawalkar, Z. Shen, Z. Liu, and M. Savvides, "Attentive CutMix: an enhanced data augmentation approach for deep learning based image classification," 2020 [Online]. Available: https://arxiv.org/abs/2003.13048.
47. J. H. Kim, W. Choo, and H. O. Song, "Puzzle mix: exploiting saliency and local statistics for optimal mixup," *Proceedings of Machine Learning Research*, vol. 119, pp. 5275-5285, 2020.
48. S. Lee, M. Jeon, I. Kim, Y. Xiong, and H. J. Kim, "Sagemix: saliency-guided mixup for point clouds," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23580-23592, 2022.
49. A. Mustafa, S. Khan, M. Hayat, R. Goecke, J. Shen, and L. Shao, "Adversarial defense by restricting the hidden space of deep neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, South Korea, 2019, pp. 3384-3393. https://doi.org/10.1109/ICCV.2019.00348
50. P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: protecting classifiers against adversarial attacks using generative models," 2018 [Online]. Available: https://arxiv.org/abs/1805.06605.
51. C. Xie, Y. Wu, L. V. D. Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 501-509. https://doi.org/10.1109/CVPR.2019.00059
52. A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017 [Online]. Available: https://arxiv.org/abs/1706.06083v1.
53. C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, pp. 816-825. https://doi.org/10.1109/CVPR42600.2020.00090
54. X. Mao, Y. Chen, R. Duan, Y. Zhu, G. Qi, S. Ye, X. Li, R. Zhang, and H. Xue "Enhance the visual representation via discrete adversarial training," *Advances in Neural Information Processing Systems*, vol. 35, pp. 7520-7533, 2022.
55. H. Wang, A. Zhang, S. Zheng, X. Shi, M. Li, and Z. Wang, "Removing batch normalization boosts adversarial training," *Proceedings of Machine Learning Research*, vol. 162, pp. 23433-23445, 2002.
56. E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation policies from data," 2018 [Online]. Available: https://arxiv.org/abs/1805.09501v1.
57. S. G. Muller and F. Hutter, "TrivialAugment: tuning-free yet state-of-the-art data augmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Montreal, Canada, 2021, pp. 754-762. https://doi.org/10.1109/ICCV48922.2021.00081
58. ImageNet, "Data Augmentation," 2023 [Online]. Available: https://paperswithcode.com/sota/data-augmentation-on-imagenet.
59. S. Mazumder and B. Liu, "Lifelong and continual learning dialogue systems," 2022 [Online]. Available: https://arxiv.org/abs/2211.06553v1.
60. J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, et al., "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521-3526, 2017. https://doi.org/10.1073/pnas.1611835114
61. R. Aljundi, F. Babiloni, M. Elhoseiny, M., Rohrbach, and T. Tuytelaars, "Memory aware synapses: learning what (not) to forget," in *Computer Vision – ECCV 2018.* Cham, Switzerland: Springer, 2018, pp. 144-161. https://doi.org/10.1007/978-3-030-01219-9_9
62. H. Jung, J. Ju, M. Jung, and J. Kim, "Less-forgetting learning in deep neural networks," 2016 [Online]. Available: https://arxiv.org/abs/1607.00122.
63. S. A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: incremental classifier and representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 5533-5542. https://doi.org/10.1109/CVPR.2017.587
64. H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *Advances in Neural Information Processing Systems*, vol. 30, pp. 2990-2999, 2017.
65. Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 374-382. https://doi.org/10.1109/CVPR.2019.00046
66. A. Mallya and S. Lazebnik, "PackNet: adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 7765-7773. https://doi.org/10.1109/CVPR.2018.00810
67. S. C. Hung, J. H. Lee, T. S. Wan, C. H. Chen, Y. M. Chan, and C. S. Chen, "Increasingly packing multiple facial-informatics modules in a unified deep-learning model via lifelong learning," in *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, Ottawa, Canada, 2019, pp. 339-343. https://doi.org/10.1145/3323873.3325053
68. G. M. Van de Ven, T. Tuytelaars, and A. S. Tolias, "Three types of incremental learning," *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1185-1197, 2022. https://doi.org/10.1038/s42256-022-00568-3
69. Q. Pham, C. Liu, and S. Hoi, "DualNet: continual learning, fast and slow," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16131-16144, 2021.
70. A. Koubaa, "GPT-4 vs. GPT-3.5: a concise showdown," 2023 [Online]. Available: https://doi.org/10.20944/preprints202303.0422.v1.
71. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016 [Online]. Available: https://arxiv.org/abs/1602.07360.
72. F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition, Honolulu, HI, USA, 2017, pp. 1800-1807. https://doi.org/10.1109/CVPR.2017.195

73. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017 [Online]. Available: https://arxiv.org/abs/1704.04861.
74. M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: platform-aware neural architecture search for mobile," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 2820-2828. https://doi.org/10.1109/CVPR.2019.00293
75. P. Stock, A. Joulin, R. Gribonval, B. Graham, and H. Jegou, "And the bit goes down: revisiting the quantization of neural networks," 2019 [Online]. Available: https://arxiv.org/abs/1907.05686v1.
76. S. Han, H. Mao, and W. J. Dally, "Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015 [Online]. Available: https://arxiv.org/abs/1510.00149v1.
77. M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: ImageNet classification using binary convolutional neural networks," in *Computer Vision – ECCV 2016*. Cham, Switzerland: Springer, 2016, pp. 525-542. https://doi.org/10.1007/978-3-319-46493-0_32
78. M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: training deep neural networks with weights and activations constrained to+ 1 or-1," 2016 [Online]. Available: https://arxiv.org/abs/1602.02830.
79. S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in Neural Information Processing Systems*, vol. 28, pp. 1135-1143, 2015.
80. N. Narodytska, S. Kasiviswanathan, L. Ryzhyk, M. Sagiv, and T. Walsh, "Verifying properties of binarized deep neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp. 6615-6624, 2018. https://doi.org/10.1609/aaai.v32i1.12206
81. F. Tung and G. Mori, "CLIP-Q: deep network compression learning by in-parallel pruning-quantization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 7873-7882. https://doi.org/10.1109/CVPR.2018.00821

Hyeon Choe https://orcid.org/0009-0007-0371-1587

Hyeon Choe completed M.C.A. at the Christ University, Bangalore, India, and is currently pursuing Ph.D. in Computer Engineering at the Korea National Defense University, South Korea. His current research interests include AI-based weapon systems.

Dongsu Kang https://orcid.org/0000-0001-6481-5071

Dongsu Kang is currently a professor of Computer Science and Engineering and the Director of the Department of Defense Science, Korea National Defense University. His main areas of expertise are software security testing, penetration testing, AI-based systems testing, naval cyber security, weapon system software, North Korea software, interoperability of defense system, machine learning, defense modeling and simulation, and defense acquisition. He was also the Director of Defense Science Center at the Research Institute for National Security Affairs.