

# Systematic review of a lightweight convolutional neural network architectures on edge devices

Muhammad Abbas Abu Talib<sup>1</sup>, Samsul Setumin<sup>1</sup>, Siti Juliana Abu Bakar<sup>1</sup>, Adi Izhar Che Ani<sup>1</sup>, Denis Eka Cahyani<sup>2</sup>

<sup>1</sup>Centre for Electrical Engineering Studies, Universiti Teknologi MARA Cawangan Pulau Pinang, Permatang Pauh, Malaysia

<sup>2</sup>Department of Mathematics, Faculty of Mathematics and Natural Science, Universitas Negeri Malang, Malang, Indonesia

## Article Info

### Article history:

Received Nov 15, 2024

Revised May 8, 2025

Accepted Jun 10, 2025

### Keywords:

Edge device

Image classification

Lightweight convolutional neural network

Resource-constrained

Systematic literature review

## ABSTRACT

A lightweight convolutional neural network (CNN) has become one of the major studies in machine learning field to optimize its potential for employing it on the resource-constrained devices. However, a benchmark for fair comparison is still missing and thus, this paper aims to identify the recent studies regarding the lightweight CNN architectures including the types of CNN, its applications, edge devices usage, evaluation types and matrices, and performance comparison. The preferred reporting items for systematic reviews and meta-analysis (PRISMA) framework was used as the main approach to collect and interpret the literature. In the process, 37 papers were identified as meeting the criteria for lightweight CNNs aimed at image classification or regression tasks. Of these, only 20 studies explored the use of these models on edge devices. To conclude, MobileNet appeared as the most used architecture, while the types of CNN focused on image classification for the general-purpose application. Following that, the NVIDIA Jetson Nano was the most utilized edge device in recent research. Additionally, performance evaluation commonly included measures like accuracy and time, along with metrics such as recall, precision, F1-Score, and other similar indicators. Finally, the average accuracy for performance comparison can serve as threshold value for future research in this scope of study.

This is an open access article under the [CC BY-SA](#) license.



## Corresponding Author:

Samsul Setumin

Centre for Electrical Engineering Studies, Universiti Teknologi MARA Cawangan Pulau Pinang

Permatang Pauh, 13500 Pulau Pinang, Malaysia

Email: samsuls@uitm.edu.my

## 1. INTRODUCTION

Researchers these days are primarily focused on the advancement of artificial intelligence (AI) technology in order to enhance society's quality of life and facilitate the industrial revolution. Although this discipline was introduced back in the 1950s, it has gone through rapid development in the past decades, which has covered both inside and outside of the computer science field [1], [2]. It can be seen that many technologies and non-technology-based journals have published articles related to AI [1], [2]. AI has progressed from simple rule-based systems to more complicated algorithms that can make autonomous decisions and solve problems. The primary idea underlying AI is to develop systems capable of doing activities that would normally need human intellect, such as visual perception, speech recognition, decision-making, and language translation [1], [2]. Figure 1 shows the inter-relation of data science to artificial neural network through AI, machine learning (ML), and deep learning (DL) [1].

Furthermore, ML is one of the most common subfields in AI, where it takes a different approach from a classical programming method. So, instead of using an algorithm for a specific problem or function, ML use a certain dataset for its algorithm to learn, predict, and decide the outcome [1], [2]. Primarily, ML is typically categorized into four major types such as supervised learning, which involves training models on labelled data, unsupervised learning, which involves searching for patterns in unlabeled data, semi supervised learning, which includes both supervised and unsupervised learning, and reinforcement learning, which teaches models to make decisions based on trial and error [1], [2]. In addition, the application of ML is commonly divided into object classification or regression (i.e., prediction). Some typical examples of algorithms in ML include artificial neural network, decision trees, linear regression, and support vector machine [1], [2].

Moreover, convolutional neural network (CNN) is one of ML's artificial neural network algorithms that is specialized for image-based tasks [1], [3]-[6]. In other words, CNN is fundamental in many computer vision tasks such as image detection, recognition, classification, regression, and segmentation. CNN is made up of three main layers, including convolutional, pooling, and fully connected layers [1], [3]-[6]. Figure 2 depicts the basic CNN architecture and its training process [1], [3]-[6]. First, convolutional layers apply filters to incoming data, capturing spatial hierarchies and local patterns that are necessary for applications such as image identification [1], [3]-[6]. Second, pooling layers lowers the dimensionality of the data, increasing computing efficiency and resilience [1], [3]-[6]. Third, fully connected layers make high-level decisions based on the extracted characteristics [1], [3]-[6]. Simply put, the first two main layers perform feature extraction from the input data and the third main layer maps the extracted features to decide or predict the output data [1], [3]-[6].

Nevertheless, the development of CNN's applications usually involves with big data which relies heavily on cloud infrastructure and resources for high computation complexity, memory and load power consumption [4]-[8]. In recent years, the widespread use of cloud computing in many fields of CNN's applications has raised some concern regarding strict latency requirements, strained network capacity, as well as privacy and security issues [4]-[8]. Ultimately, in order to overcome these problems and optimize CNN's applications, an increasing demand for deploying DL models directly onto edge devices to enable real-time inference and decision-making has been introduced. Edge computing includes processing data at or near the source of data creation which is called edge devices, such as IoT devices, smartphones, or sensors, rather than using a centralized cloud infrastructure.

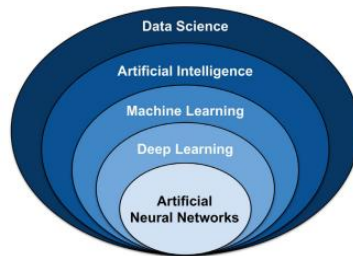


Figure 1. The inter-relation of data science to artificial neural network through AI, ML, and DL [1], [2]

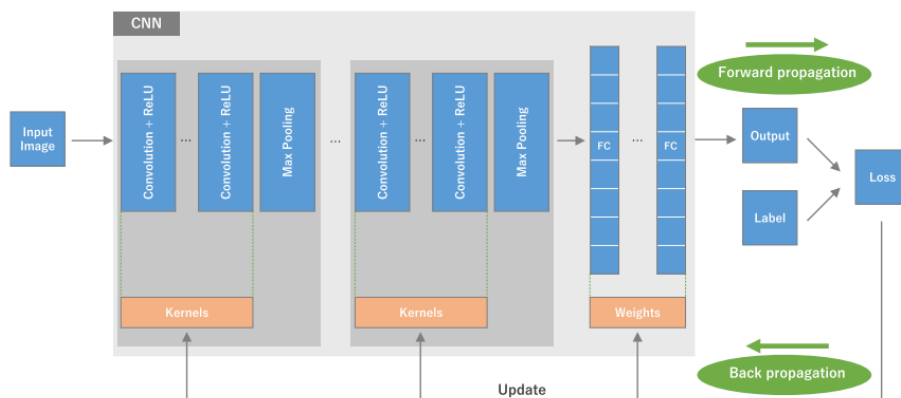


Figure 2. The basic CNN architecture and its training process [1], [2]

Many researchers have employed edge computing to gain various benefits, including shorter latency, lower bandwidth use, and greater privacy and security [4]-[8]. For that, edge devices equipped with strong central processing units (CPUs) and specialized hardware accelerators like graphical processing units (GPUs), tensile processing units (TPUs), and neural processing units (NPUs) can execute complicated AI and ML models locally. However, optimizing CNN architecture's efficiency for edge devices deployment poses a critical challenge since their applications vary from each other and due to the limited computational resources and power constraints of the edge devices. To date, most study in this subject have used different method of optimization to produce their lightweight CNN for edge device deployment and the benchmark for a fair comparison is still missing [8]-[12].

Now, this systems literature review (SLR) aims to collect, analyze, and interpret the current or recently published articles on the lightweight CNN architectures for edge devices and categorized them in terms of the specific architecture or based-model, CNN's types (i.e., classification or regression), and applications (i.e., the subject or purpose of each CNN's architectures) used for their research. Next, based on the types of edge device, the evaluation criteria (e.g., time and accuracy), evaluation matrices (e.g., accuracy, precision, F1-Score, and root mean squared error (RMSE)), and performance comparison between each study in the first question will be recorded and discussed. In short, the research questions in this paper are as follows and can be seen as illustrated in Figure 3:

- RQ1: What is the current lightweight CNN architectures used on the limited computational resources or edge devices?
- RQ2: What is the current performance of the lightweight CNN architecture used on the limited computational resources or edge devices?

This SLR is divided into four major sections. In the first section, the introduction is reviewed regarding the background knowledge of AI, ML, CNN, and edge device, the focused limitations, and the research questions of this literature. For the second section, most of the related articles on the lightweight CNN on edge devices will be collected and mapped by using the preferred reporting items for systematic reviews and meta-analysis (PRISMA) framework as the methodology part of this study. Thirdly, all the results will be analyzed and discussed in order to answer the research questions of the current lightweight CNN architectures used on the limited computational resources or edge devices and its performances, as the third section of this paper. Finally, this SLR will be concluded in the fourth section by providing the summarized findings and an insight for future recommendations on this scope of study.

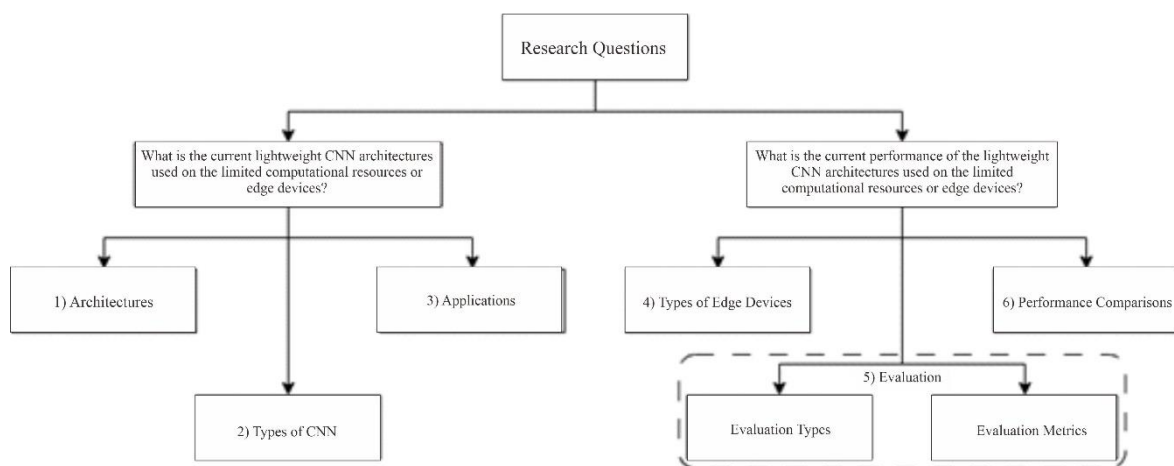


Figure 3. Mind map illustration of the research questions

## 2. RESEARCH METHOD

In this SLR, a detailed technique for identifying relevant research on the subject regarding lightweight CNN architectures on edge devices was conducted. This approach was adopted based on the PRISMA framework [13], and the modified flowchart in Figure 4 shows the practical view of each step of this SLR's methodology. Basically, there were three major phases involved in completing this paper. Initially, the identification phase determined the records acquired from the search strategy used in any kind of academic research database. Secondly, the initial part of the screening phase involves executing the selection criteria in order to only consider the necessary categories for the descriptive analysis. Thirdly, the quality assessment is also included as the second part of the screening phase in order to find out the eligibility of

each article for this scope of study. Finally, the included phase shows the final number of articles that will be used in literature classification in order to satisfy all the research questions stated beforehand.

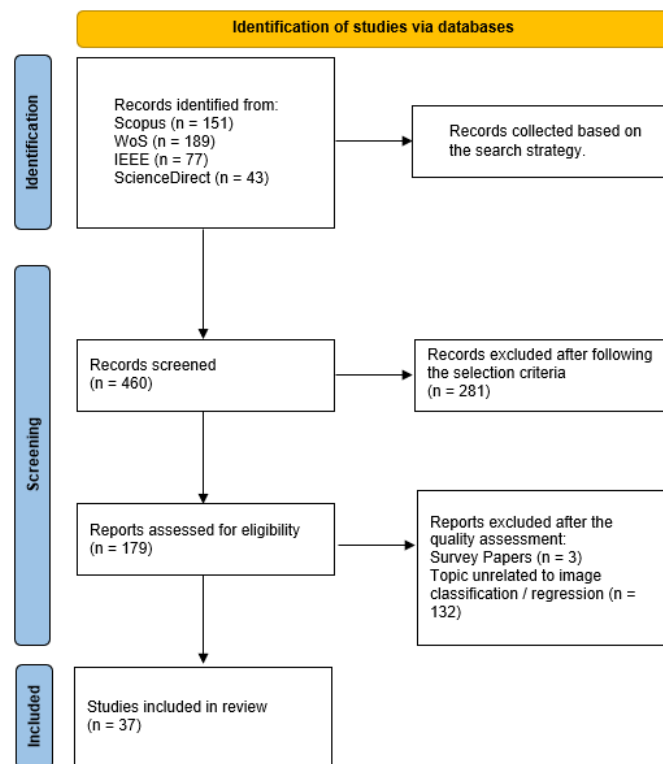


Figure 4. The modified PRISMA framework [13] with all the records for this SLR's methodology

### 2.1. Search strategy

First of all, the search strategy was specifically executed in two academic database indexers, such as Scopus, and web of science (WoS), as well as two published article databases, such as IEEE and ScienceDirect. Moreover, the search only included journal articles, review papers, and conference papers or proceedings for Scopus and WoS, while only records for journal articles from IEEE and ScienceDirect were extracted. Next, the keywords used for all the database searches were “lightweight CNN” AND “edge devices” in the search fields of title, abstract, and author's keyword. Then, in order to focus more on the recent and updated papers, the publications' years in the databases were limited from 2020 to 2024. Also, the search focused on papers published only in English. By applying these terms, the search was narrowed down to a specific area and scope related to this study. At this stage, a total record of 460 articles' metadata was obtained throughout the search.

### 2.2. Selection criteria

For the selection criteria, all the recorded metadata was combined in a single spreadsheet for the screening process. The major goal was to map the available literature on the use of lightweight CNN in edge devices according to the source title, journal publisher, year of publications, research field, and number of citations, as these categories will be used in the descriptive analysis of the result and discussion section. All data for other categories was excluded and removed. For the next step, all the papers' digital object identifiers (DOIs) were sorted out in order to remove duplicate records easily using the spreadsheet's tool. Last but not least, review papers and conference proceedings were also excluded in order to keep the records more relevant. Due to these criteria, 281 research publications were rejected during the initial screening process, and only 179 records were left for further assessment.

### 2.3. Quality assessment

Following the initial screening phase, a quality assessment was performed on each research paper in order to further ensure that only the most eligible studies were included in this SLR for a critical review.

Each article's title and abstract were scrutinized thoroughly to ensure their relevancy and contribution to the topic under review. By executing this process, it helps to purify the selection, ensuring only pertinent and high-quality academic literature is included in the review process. As a result, a total of 132 articles were removed from the records, with three of them being a review papers and others being articles that were not related to image classification or regression (i.e., signal classification, sound classification, object detection, segmentation, and localization). At this stage, there were only 37 research papers left in the record for the final process of data extraction.

#### 2.4. Data extraction

During the data extraction phase, 37 publications were carefully selected for their relevance and capacity to address the research questions given in the preceding section. For that, by understanding the current trend of lightweight CNN on edge devices, all the data will be analyzed as a literature classification in the latter part of the result and discussion section. With that, various lightweight CNN architectures that have been conducted in previous research with resource-constrained devices will be highlighted, including their task-based categories, applications, as well as the types of edge devices and their specifications. Then, the key performance of lightweight CNN architectures on these low-resource devices will be examined in terms of evaluation types, matrices, and performance comparisons as reported by various researchers.

### 3. RESULTS AND DISCUSSION

In this section, all the results obtained after conducting the approach discussed in the previous section will be observed and analyzed. This section is divided into two subsections. The first subsection will focus on the descriptive analysis to see the general trend of the research, and the second subsection, literature classifications, will further discuss the content in order to fulfil the research questions in this paper.

#### 3.1. Descriptive analysis

From the methodology conducted, the obtained literature for this systematic literature review has a total of 37 papers that are specifically related to the research of lightweight CNN architectures implemented or were designed for resource-constrained edge devices. Based on Table 1, all the papers were classified according to the year of publication, journal publishers, and number of citations. Then, the number of related papers published in the following year, 2021–2024, is depicted in Figure 5, publisher classification in Figure 6, and the number of citations from each paper in Figure 7.

Table 1. Research database descriptive analysis

Ref. Number	Year	Publisher	Cited	Ref. Number	Year	Publisher	Cited
[14]	2024	Elsevier	1	[33]	2023	Elsevier	2
[15]	2024	Elsevier	0	[34]	2023	Elsevier	9
[16]	2024	Wiley	0	[35]	2022	IEEE	40
[17]	2024	Elsevier	1	[36]	2022	Springer	11
[18]	2024	Elsevier	4	[37]	2022	MDPI	0
[19]	2024	MDPI	0	[38]	2022	KIPS	6
[20]	2024	MDPI	3	[39]	2022	Wiley	3
[21]	2023	CSIR-NIScPR	2	[40]	2022	Springer	2
[22]	2023	CSIR-NIScPR	1	[41]	2022	Elsevier	22
[23]	2023	Springer	3	[42]	2022	Springer	1
[24]	2023	Wiley	1	[43]	2021	MDPI	3
[25]	2023	IEEE	0	[44]	2021	IEEE	12
[26]	2023	IEEE	7	[45]	2021	Wiley	16
[27]	2023	Elsevier	3	[46]	2021	KSS	2
[28]	2023	Elsevier	4	[47]	2021	IEEE	33
[29]	2023	Elsevier	16	[48]	2021	Elsevier	41
[30]	2023	CSIR-NIScPR	1	[49]	2021	MDPI	5
[31]	2023	CSIR-NIScPR	5	[50]	2021	MDPI	18
[32]	2023	IEEE	9				

In Figure 5, the pie chart shows that in the year 2021, the number of published papers was 8, which was also the same number produced in 2022. However, in 2023, the number was almost twice the previous published paper, which was 14. Moreover, by the mid-year of 2024, the already-published articles were 7. Hence, it can be seen and predicted that by the end of 2024, the number of published papers will be twice as large.

Next, Figure 6 shows the number of research articles by publishers. According to this pie chart, the highest number of papers were published by Elsevier, which is 11 papers and 29.7% of the total papers

reviewed in this article. Following that, IEEE and MDPI published six papers, with 16.2% from each publisher. Other than that, 4 papers and 10.8% were published by each of Wiley, Springer, and CSIR-NIScPR, leaving 1 paper and 2.7% of the total papers being published by KIPS and KSS, respectively.

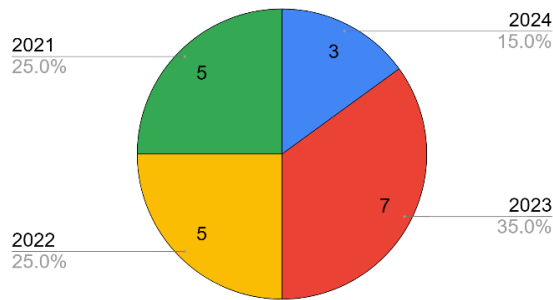


Figure 5. Number of papers published from each year within 2021 until 2024

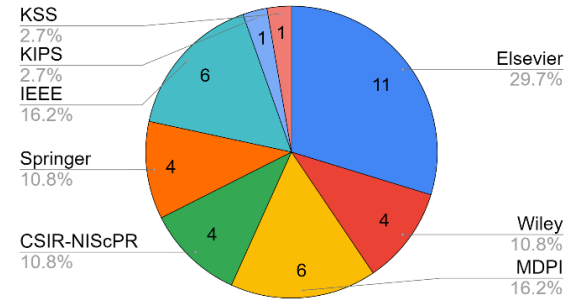


Figure 6. Number of papers published by each publisher

After that, Figure 7 analyzed the number of citations from each paper in this SLR. The horizontal axis denotes the reference number of each article, and the vertical axis shows the number of its citation. Based on the bar graph, the highest number of citations is 41, followed by 40, 33, and 22. Only 4 papers from the total article have the number of citations above 20; other than that, most papers have the number of citations below 20, which range from 0 to 18. In summary, by observing and analyzing these simple literature classifications, it suggests that the research focused on this field is still currently in the beginning phase. Therefore, further research is needed in order to contribute more novelty and a state-of-the-art approach to the study of lightweight CNN on edge devices.

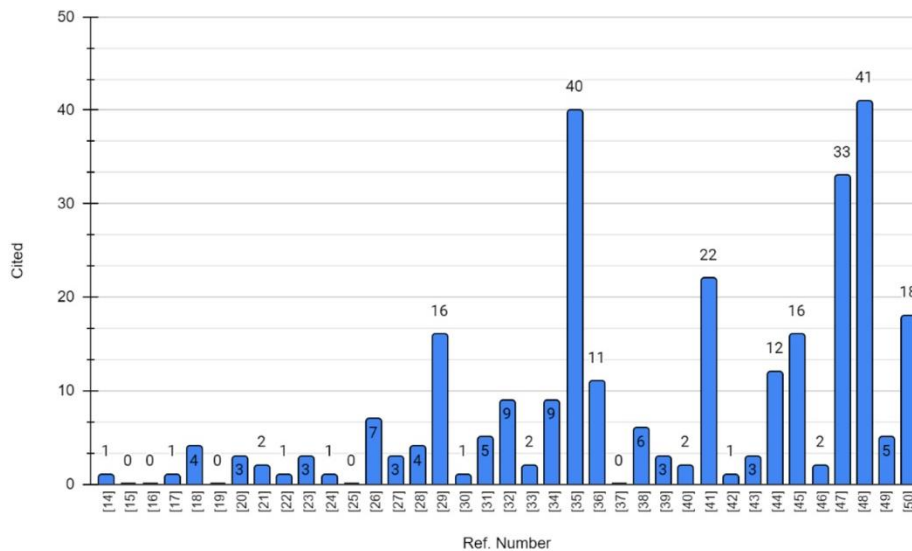


Figure 7. Number of citations from each paper

### 3.2. Literature classifications

For the literature classification, the reviewed articles were classified accordingly by referring to the lightweight CNN architectures for the purpose of edge device implementation. Table 2 summarize all those lightweight CNN architectures or based models, the types of the CNN, and its applications. Meanwhile, Table 3 focused on those lightweight CNN architectures that have been experimented on edge devices, which includes its performance evaluation in terms of their evaluation types, matrices, and performance comparison in terms of average accuracy.

Table 2. Research database literature classifications

Ref.	Architectures/models	Types of CNN	Applications
[14]	Lite-MDC	Classification types of pigeon pea's diseases	Plant disease detection for pigeon pea
[15]	VGG-16	Classification of cracked/non-cracked surfaces	Automated crack detection in building inspection and maintenance
[16]	SDLN	Classification of cataract/non-cataract eye	Cataract eye detection
[17]	ShuffleNetV2_YOLOv5s	Classification types of canola kernel grades	Real-time canola damage detection
[18]	OnDev-LCT	Classification	General purpose for image classification
[19]	Zero-FVeinNet	Classification types of fingers vein	Finger vein recognition
[20]	YOLOv5	Classify the maturity of blueberry fruit	Blueberry fruit maturity detection
[21]	MobileNetV2, CondenseNetV2, ShuffleNetV2	Classification of fabric, surface, and casting defect	Product defect detection in MFG industries
[22]	MobileNetV2	Classification types of fabric defect	Fabric defect detection in textile MFG
[23]	CCNNet	Classification types of traditional Chinese medicine (TCM)	Traditional Chinese medicine image classification
[24]	MobileNet	Classification	General purpose for image classification
[25]	VGG16_BN, ResNet-50, RegNet-X	Classification	General purpose for image classification
[26]	CH-CNN	Classification	General purpose for image classification
[27]	MobileViTFace	Classification of different breeds of sheep	Sheep face recognition
[28]	VGGNet-16, ResNet-50/56/110, GoogLeNet, DenseNet-40	Classification	General purpose for image classification
[29]	LiteCNN	Classification of different types of plant diseases	Plant disease identification
[30]	ShuffleNetv2	Classification of defective/nondefective casting	Casting defect detection
[31]	CondenseNetV2	Classification types of surface defect	Surface defect detection in industrial intelligent production
[32]	CNN	Classification types of facial emotion	Facial emotion recognition for VIP
[33]	EBNAS	Classification	General purpose for image classification
[34]	YOLOv5s-BiFPN	Classification of pig body region of interest (RoI), regression of pig body temperature	Pig body temperature automatic detection for early disease warning
[35]	EdgeFireSmoke	Classify the occurrence of forest fires	Fire-smoke detection of forest fires
[36]	TripleNet	Classification	General purpose for image classification
[37]	CondenseNetV2	Classification	General purpose for image classification
[38]	InceptionV3, MobileNet, VGG16	Classification types of face with mask/without mask	Face mask classification
[39]	HFENet	Classification of defective/nondefective ceramic tile surface	Ceramic tile surface defect detection
[40]	ShuffleNet	Classification of 3D object images	3D object recognition for 3D scanning technology
[41]	MobileNetv2	Regression of crowd density estimation	Estimating crowd density for public security management
[42]	RDPNet	Classification	General purpose for image classification
[43]	Ensemble Binarized DroNet (EBDN)	Classification task for collision-avoidance, regression task for prediction of desired steering angle	Autonomous driving for unmanned autonomous vehicles (UAV)
[44]	BC-Net	Classification	General purpose for image classification, speech recognition of keyword spotting, facial expression recognition
[45]	MobileNet-v2	Classification types of solid waste	Waste classification for solid waste management
[46]	DenseNet	Classification	General purpose for image classification
[47]	MobileNetV3	Classification types of icing grades	Icing monitoring of transmission lines
[48]	SparkNet	Classification	General purpose for image classification
[49]	MobileNetV2 & SqueezeNet	Classification types of waste	Reverse vending machine for types of waste recycles
[50]	ASIR-Net	Classification types of different ground vehicle target	Automatic target recognition (ATR) in synthetic aperture radar (SAR) images for military surveillance

Table 3. Research database edge devices' performance classifications

Ref.	Edge device types	Evaluation types	Evaluation matrices	Ave. accuracy (%)
[15]	Raspberry Pi 3B+	- Accuracy - Model size - Robustness	- Accuracy - Recall - Precision - F1-Score	95.30
[16]	Android Smartphone	- Accuracy - Model size - Time	- Accuracy - Inference time - Parameters	95.63
[17]	NVIDIA Jetson Nano	- Speed - Sensitivity	- Precision - Recall - F1-Score - Inference speed	-
[21]	NVIDIA Jetson Nano	- Accuracy - Sensitivity - Specificity	- Accuracy - Recall - Precision - F1-Score	97.00
[22]	NVIDIA Jetson Nano	- Accuracy - Sensitivity	- Accuracy - Recall - Precision - F1-Score	96.52
[25]	- NVIDIA AGX Xavier - NVIDIA Jetson Nano	- Accuracy - Computations complexity - Time - Model size	- Top-1 accuracy - MACs - Latency - Parameters	75.57
[27]	NVIDIA Jetson Nano	- Accuracy	- Accuracy - Precision - Recall	97.13
[29]	ZYNQ Z7-Lite 7020 FPGA	- Accuracy - Speed - Time	- Accuracy - Inference speed - Latency	95.71
[30]	NVIDIA Jetson Nano	- Accuracy - Sensitivity	- Precision - Recall - F1-Score - Accuracy	99.58
[31]	NVIDIA Jetson Xavier Nx	- Accuracy - Sensitivity	- Accuracy - Recall - Precision - F1-Score	91.40
[35]	NVIDIA Jetson Nano	- Accuracy - Sensitivity - Time	- Accuracy - Recall - Precision - F1-Score - Hamming loss	98.97
[36]	Raspberry Pi 4	- Time - Computations complexity	- Latency - FLOPS	-
[37]	NXP BlueBox 2.0	- Model Size - Accuracy - Time - Computations complexity	- FLOPS - Parameters - Top-1 accuracy - Inference time	84.55
[38]	Raspberry Pi 4	- Accuracy - Speed - Loss	- Accuracy - Precision - Recall - F1-Score	95.51
[37]	Raspberry Pi 4	- Speed - Computations complexity - Time	- Mean absolute error (MAE) - RMSE - Inference speed and time - FLOPS	-
[43]	Xilinx Zynq 7Z100 FPGA	- Accuracy - Precision - Speed	- RMSE - F1-Score - FPS	95.60
[44]	NUCLEO-F767ZI with STM32H743	- Time	- MSE - Latency	-
[47]	Huawei Atlas 200 DK	- Accuracy - Precision - Time	- Accuracy - Time - FPS	74.50
[48]	Intel Arria 10 GX1150 FPGA	- Time - Speed	- Inference time - Speed up	-
[49]	- NVIDIA Jetson Nano - NVIDIA Jetson TX1	- Accuracy - Time	- Accuracy	95.00



### 3.2.1. Lightweight convolutional neural network architectures

Firstly, Figure 8 shows the number of architectures used from all the research in Table 2. Based on the architectures or based model, it can be seen that there were several similar networks that were being used as their approaches, such as MobileNet with 9 (19.1%), ShuffleNet with 4 (8.5%), VGG-16 with 4 (8.5%), CondenseNet with 3 (6.4%), respectively in terms of number of their usage. Meanwhile, the Others with 21 (44.7%), represent the number of different models with only one usage. All these are some of the state-of-the-art approaches that are currently being used by researchers in this field. While some of the researchers use them as benchmarks, there are also several others that modify these original networks with various versions to improve their performances. Aside from that, there were a few with hybrid models or multimodal which combined two or more networks together by using novel approaches. For example, research in [27] combined a lightweight CNN architecture's MobileNet model with a Vision Transformer architecture. All in all, MobileNet including its various version is the most used lightweight CNN architecture for edge devices implementation.

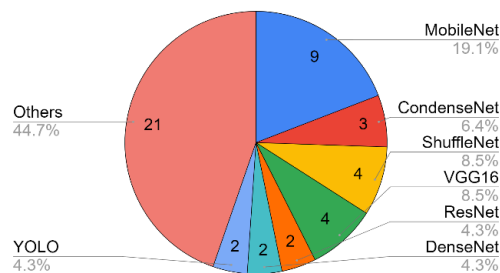


Figure 8. Several different common lightweight CNN architectures

### 3.2.2. Types of convolutional neural network

Next, as for the types of lightweight CNN architectures in Table 2, most of the research used for classification with a few of them used for regression, and some combined both the classification and regression. Based on Figure 9, it suggests that the classification type was the main approach that was being researched with 34 (91.9%) and the regression type with 1 (2.7%) was much more complicated to be researched on. Since the regression types of CNN requires continuous data for prediction, its implementation for resource-constrained devices may require a higher computational usage compared to the classification types and thus, the result indicated that only minor research has been done for regressions task. However, there were still a few studies that applied the regression lightweight CNN, and some also used it in hybrid models with classification and regression with 2 (5.4%). In short, most of the research in lightweight CNN was leaning towards the classification types of CNN.

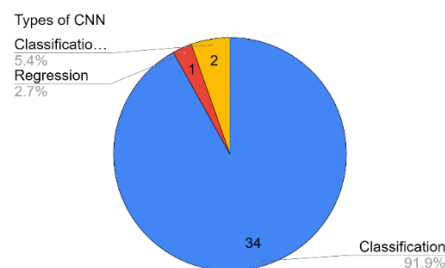


Figure 9. The common types of CNN

### 3.2.3. Models' applications

From the aspect of applications, Table 2 depicted that this field of research is very comprehensive. The studies were conducted from various field including industrial manufacturing (MFG) with 8 (21.6%), public surveillance and safety with 8 (21.6%), health with 4 (10.8%), waste management with 2 (5.4%), agriculture (i.e., animal and plant) disease detection with 2 (5.4%), and military with 1 (2.7%) as illustrate in

Figure 10. Aside from that, the highest number of applications were for general-purpose (GP) uses with 12 (32.4%) which shows the flexibility and reliability of the network to be used with many different applications.

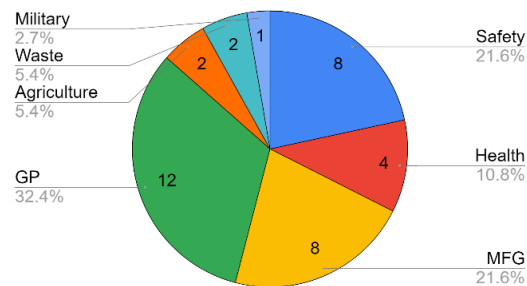


Figure 10. Various applications of lightweight CNN use for different sector of industry

### 3.2.4. Types of edge devices

On the other hand, based on Table 3, 20 study were filtered out for the performance analysis of lightweight CNN architecture deployed in edge device. Aside from these 20 research, the other 17 research earlier did not continue using edge devices while only evaluating their lightweight CNN with some of the benchmark lightweight models. Figure 11 summarizes the types and numbers of edge devices based on the research in Table 3. Firstly, it can be seen that edge devices can be categorized into two types which are on-the-shelf devices including several system-on-a-chip (SoC) (e.g., NVIDIA Jetson series and Raspberry Pi series), microcontroller (MCU) with its development board such as the NUCLEO-F767ZI with STM32H743, and Android Smartphone. Meanwhile, the off-the-shelf devices include those being implemented as FPGAs (e.g., Xilinx Zynq 7Z100, ZYNQ Z7-Lite 7020, and Intel Arria 10 GX1150). Hence, most of the recent study shows that NVIDIA Jetson Nano as the most used edge device with 7 (31.8%), followed by Raspberry Pi 4 with 3 (13.6%), and FPGAs also with 3 (13.6%) while others (SoC) with 7 (31.8%) represent the other types of SoC edge devices that were used only once.

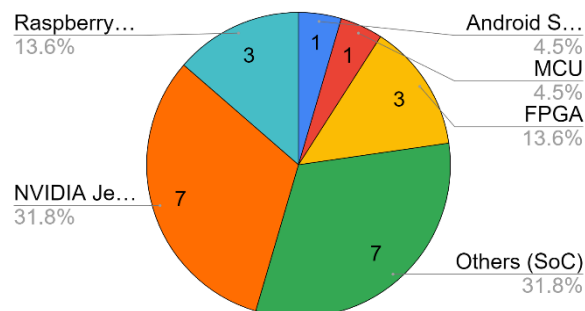


Figure 11. Some of the general edge devices used to embed a lightweight CNN architecture

### 3.2.5. Evaluation types and matrices

Last but not least, for the evaluation types and matrices, most of these studies focused on accuracy, time, speed, sensitivity, model size, computation complexity, and others. Moreover, several matrices, such as accuracy, recall, precision, F1-Score, inference time and speed, latency, parameters, floating point operations (FLOPs), and others, were always being used in order to ensure that the lightweight CNN for edge devices performance was optimized. Some of the evaluation matrices represent each of the evaluation types as the matrices are the specific assessment of each of the evaluation types. For example, accuracy and top-1 accuracy are the evaluation matrices for evaluation type of accuracy, inference time and latency are the evaluation matrices for evaluation type of time, and FLOPs and multiply-accumulate operations (MACs) are the evaluation matrices for evaluation type of computation complexity. Table 3 describes some of the

common evaluation types and matrices that were used in those studies. Furthermore, Figures 12(a) and (b) pointed out the numbers of those common evaluation types and matrices. Generally, accuracy with 15 (29.4%) and 12 (17.6%) was the most used evaluation type and matrix, respectively, based on the recent studies.

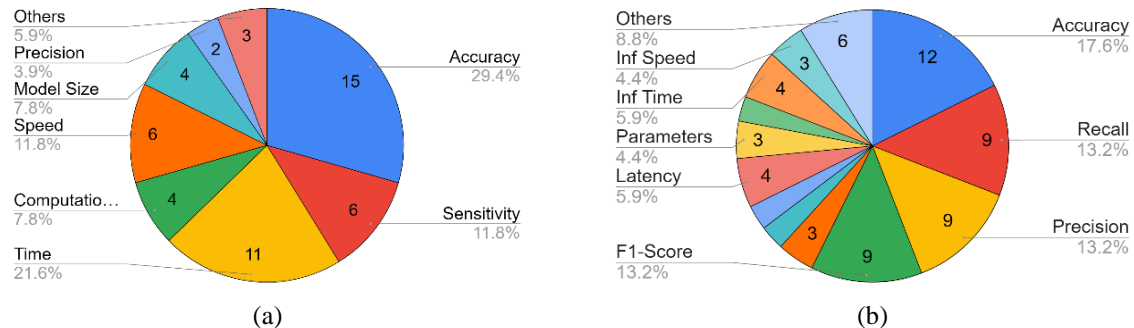


Figure 12. Numbers of (a) the evaluation types and (b) the evaluation matrices used frequently in the recent studies

### 3.2.6. Performance comparison

Finally, the performance of all the lightweight CNN that were deployed in an edge devices was compared in terms of its respective average accuracy as one of the typical evaluation parameters and summarized in Table 3. Despite that, there were still a few research that did not evaluate the accuracy of the model on edge device while focusing more on the precision, time, and others. Figure 13 shows the illustration of each edge device performance in average accuracy. While most of them have an average accuracy around 95% and above, a few of them only have an average accuracy of around 75%. This is because taking into account of many factors such as different specifications of each edge device and input database of the lightweight CNN used in their respective applications influenced the performance comparison. As such, to compare each of those edge devices' performance is not relevant, however it can still be served for expected performance in terms of average accuracy threshold value for future research of a lightweight CNN in resource-constrained device.

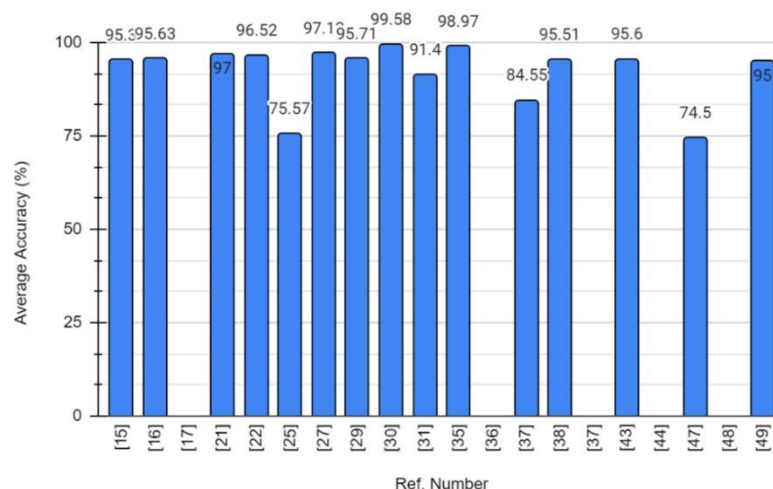


Figure 13. Performance comparison in terms of average accuracy for each edge device

## 4. CONCLUSION

In conclusion, the study of this systematic literature review was written in order to understand the current trend of lightweight CNN architectures used in edge devices. With that, several lightweight CNN architectures and models were identified including the types of the CNN, their applications, edge devices

used, evaluation types and matrices, as well as a simple performance comparison regarding their average accuracy. As the scope of this field of study is still in the early stage, many contributions and novelty approaches are required to ensure a comprehensive advancements of edge device technology with the utilization of lightweight CNN.

## ACKNOWLEDGEMENTS

The authors would like to sincerely thank Universiti Teknologi MARA Cawangan Pulau Pinang for their invaluable support in this research, providing access to essential resources and materials. We are also deeply grateful to our fellow researchers, especially the dedicated members of the Machine Learning Research Group (MLRG) at the Centre for Electrical Engineering Studies, whose collaboration and insights were crucial in achieving the best possible outcomes for this study.

## FUNDING INFORMATION

This research article was financially supported by University Teknologi MARA and Institute of Postgraduate Studies UiTM.

## REFERENCES

- [1] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang, and J. P. Campbell, "Introduction to machine learning, neural networks, and deep learning," *Translational Vision Science & Technology*, vol. 9, no. 2, 2020.
- [2] N. Sharma, R. Sharma, and N. Jindal, "Machine learning and deep learning applications-a vision," *Global Transitions Proceedings*, vol. 2, no. 1, pp. 24–28, Jun. 2021, doi: 10.1016/j.gltp.2021.01.004.
- [3] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, pp. 611–629, 2018, doi: 10.1007/s13244-018-0639-9.
- [4] X. Hou, Y. Guan, T. Han, and N. Zhang, "DistrEdge: speeding up convolutional neural network inference on distributed edge devices," *arXiv*, Feb. 2022, doi: 10.48550/arXiv.2202.01699.
- [5] S. Naveen, M. R. Kounte, and M. R. Ahmed, "Low latency deep learning inference model for distributed intelligent IoT edge clusters," *IEEE Access*, vol. 9, pp. 160607–160621, 2021, doi: 10.1109/ACCESS.2021.3131396.
- [6] R. Stahl, A. Hoffman, D. Mueller-Gritschneider, A. Gerstlauer, and U. Schlichtmann, "DeeperThings: Fully Distributed CNN Inference on Resource-Constrained Edge Devices," *International Journal of Parallel Programming*, vol. 49, no. 4, pp. 600–624, Aug. 2021, doi: 10.1007/s10766-021-00712-3.
- [7] S. P. Baller, A. Jindal, M. Chadha, and M. Gerndt, "DeepEdgeBench: benchmarking deep neural networks on edge devices," *arXiv*, Aug. 2021, doi: 10.48550/arXiv.2108.09457.
- [8] J. Liu, S. Tripathi, U. Kurup, and M. Shah, "Pruning algorithms to accelerate convolutional neural networks for edge applications: a survey," *arXiv*, May 2020, doi: 10.48550/arXiv.2005.04275.
- [9] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing*, vol. 126, Jun. 30, 2022, doi: 10.1016/j.dsp.2022.103514.
- [10] F. Wang, M. Zhang, X. Wang, X. Ma, and J. Liu, "Deep learning for edge computing applications: a state-of-the-art survey," *IEEE Access*, vol. 8, pp. 58322–58336, 2020, doi: 10.1109/ACCESS.2020.2982411.
- [11] V. Kamath and A. Renuka, "Deep learning based object detection for resource constrained devices: Systematic review, future trends and challenges ahead," *Neurocomputing*, vol. 531, pp. 34–60, Apr. 28, 2023, doi: 10.1016/j.neucom.2023.02.006.
- [12] H. Hussain, P. S. Tamizharasan, and C. S. Rahul, "Design possibilities and challenges of DNN models: a review on the perspective of end devices," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5109–5167, Oct. 2022, doi: 10.1007/s10462-022-10138-z.
- [13] M. J. Page *et al.*, "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *BMJ Publishing Group*, vol. 372, Mar. 29, 2021, doi: 10.1136/bmj.n71.
- [14] S. Bhagat *et al.*, "Advancing real-time plant disease detection: A lightweight deep learning approach and novel dataset for pigeon pea crop," *Smart Agricultural Technology*, vol. 7, Mar. 2024, doi: 10.1016/j.atech.2024.100408.
- [15] S. Chang and B. Zheng, "A lightweight convolutional neural network for automated crack inspection," *Construction and Building Materials*, vol. 416, Feb. 2024, doi: 10.1016/j.conbuildmat.2024.135151.
- [16] D. Neogi, M. A. Chowdhury, M. M. Akter, and M. I. A. Hossain, "Mobile detection of cataracts with an optimised lightweight deep Edge Intelligent technique," *IET Cyber-Physical Systems: Theory and Applications*, vol. 9, no. 3, pp. 269–281, 2024, doi: 10.1049/cps2.12083.
- [17] A. Thakuria and C. Erkinbaev, "Real-time canola damage detection: An end-to-end framework with semi-automatic crusher and lightweight ShuffleNetV2 YOLOv5s," *Smart Agricultural Technology*, vol. 7, p. 100399, Mar. 2024, doi: 10.1016/j.atech.2024.100399.
- [18] C. M. Thwal, M. N. H. Nguyen, Y. L. Tun, S. T. Kim, M. T. Thai, and C. S. Hong, "OnDev-LCT: On-Device lightweight convolutional transformers towards federated learning," *Neural Networks*, vol. 170, pp. 635–649, Feb. 2024, doi: 10.1016/j.neunet.2023.11.044.
- [19] N. C. Tran *et al.*, "Zero-FVeinNet: optimizing finger vein recognition with shallow CNNs and zero-shuffle attention for low-computational devices," *Electronics (Basel)*, vol. 13, no. 9, p. 1751, May 2024, doi: 10.3390/electronics13091751.
- [20] F. Xiao, H. Wang, Y. Xu, and Z. Shi, "A lightweight detection method for blueberry fruit maturity based on an improved YOLOv5 algorithm," *Agriculture (Switzerland)*, vol. 14, no. 1, Jan. 2024, doi: 10.3390/agriculture14010036.
- [21] J. Bonam, S. S. Kondapalli, N. L. V. Prasad, and K. Marlapalli, "Lightweight CNN models for product defect detection with edge computing in manufacturing industries," *Journal of Scientific & Industrial Research (JSIR)*, vol. 82, no. 4, pp. 418–425, Apr. 2023, doi: 10.56042/jsir.v82i04.72390.




- [22] L. R. Burra, A. Karuna, S. Tumma, K. Marlapalli, and P. Tumuluru, "MobileNetV2-based transfer learning model with edge computing for automatic fabric defect detection," *Journal of Scientific & Industrial Research (JSIR)*, vol. 82, no. 1, pp. 128–134, Jan. 2023, doi: 10.56042/jsir.v82i1.69928.
- [23] H. Gang, S. Guanglei, W. Xiaofeng, and J. Jinlin, "CCNNet: a novel lightweight convolutional neural network and its application in traditional Chinese medicine recognition," *Journal of Big Data*, vol. 10, no. 1, Dec. 2023, doi: 10.1186/s40537-023-00795-4.
- [24] C. Y. Kim, K. S. Um, and S. W. Heo, "A novel MobileNet with selective depth multiplier to compromise complexity and accuracy," *ETRI Journal*, vol. 45, no. 4, pp. 666–677, Aug. 2023, doi: 10.4218/etrij.2022-0103.
- [25] H. Kong *et al.*, "EdgeCompress: coupling multidimensional model compression and dynamic inference for EdgeAI," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 12, pp. 4657–4670, Dec. 2023, doi: 10.1109/TCAD.2023.3276938.
- [26] S. Li, Y. Sun, G. G. Yen, and M. Zhang, "Automatic design of convolutional neural network architectures under resource constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 3832–3846, Aug. 2023, doi: 10.1109/TNNLS.2021.3123105.
- [27] X. Li, Y. Xiang, and S. Li, "Combining convolutional and vision transformer structures for sheep face recognition," *Computers and Electronics in Agriculture*, vol. 205, Feb. 2023, doi: 10.1016/j.compag.2023.107651.
- [28] Y. Liu, K. Fan, D. Wu, and W. Zhou, "Filter pruning by quantifying feature similarity and entropy of feature maps," *Neurocomputing*, vol. 544, Aug. 2023, doi: 10.1016/j.neucom.2023.126297.
- [29] Y. Luo, X. Cai, J. Qi, D. Guo, and W. Che, "FPGA-accelerated CNN for real-time plant disease identification," *Computers and Electronics in Agriculture*, vol. 207, Apr. 2023, doi: 10.1016/j.compag.2023.107715.
- [30] L. V. N. Prasad, D. B. Dokku, S. L. Talasila, and P. Tumuluru, "Hyper Parameter optimization for transfer learning of ShuffleNetV2 with Edge Computing For Casting Defect Detection," *Journal of Scientific & Industrial Research (JSIR)*, vol. 82, no. 2, pp. 171–177, 2023, doi: 10.56042/jsir.v82i2.70250.
- [31] S. D. Rani, L. R. Burra, G. Kalyani, and N. K. B. Rao, "Edge intelligence with light weight CNN model for surface defect detection in manufacturing industry," *Journal of Scientific & Industrial Research (JSIR)*, vol. 82, no. 2, pp. 178–184, 2023, doi: 10.56042/jsir.v82i2.69945.
- [32] D. Shehada, A. Turkey, W. Khan, B. Khan, and A. Hussain, "A lightweight facial emotion recognition system using partial transfer learning for visually impaired people," *IEEE Access*, vol. 11, pp. 36961–36969, 2023, doi: 10.1109/ACCESS.2023.3264268.
- [33] C. Shi, Y. Hao, G. Li, and S. Xu, "EBNAS: Efficient binary network design for image classification via neural architecture search," *Engineering Applications of Artificial Intelligence*, vol. 120, Apr. 2023, doi: 10.1016/j.engappai.2023.105845.
- [34] Q. Xie *et al.*, "A deep learning-based detection method for pig body temperature using infrared thermography," *Computers and Electronics in Agriculture*, vol. 213, Oct. 2023, doi: 10.1016/j.compag.2023.108200.
- [35] J. S. Almeida, C. Huang, F. G. Nogueira, S. Bhatia, and V. H. C. de Albuquerque, "EdgeFireSmoke: a novel lightweight CNN model for real-time video fire–smoke detection," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7889–7898, Nov. 2022, doi: 10.1109/TII.2021.3138752.
- [36] R.-Y. Ju, T.-Y. Lin, J.-H. Jian, and J.-S. Chiang, "Efficient Convolutional Neural Networks on Raspberry Pi for Image Classification," *arXiv*, Apr. 2022, doi: 10.48550/arXiv.2204.00943.
- [37] P. Kalgaonkar and M. El-Sharkawy, "CondenseNeXtV2: Light-Weight Modern Image Classifier Utilizing Self-Querying Augmentation Policies," *Journal of Low Power Electronics and Applications*, vol. 12, no. 1, Mar. 2022, doi: 10.3390/jlpea12010008.
- [38] E. Kristiani, Y. T. Tsan, P. Y. Liu, N. Y. Yen, and C. T. Yang, "Binary and multi-class assessment of face mask classification on edge AI using CNN and transfer learning," *Human-centric Computing and Information Sciences*, vol. 12, 2022, doi: 10.22967/HICIS.2022.12.053.
- [39] F. Lu *et al.*, "HFENet: A lightweight hand-crafted feature enhanced CNN for ceramic tile surface defect detection," *International Journal of Intelligent Systems*, vol. 37, no. 12, pp. 10670–10693, Dec. 2022, doi: 10.1002/int.22935.
- [40] M. Song and Q. Guo, "Efficient 3D object recognition in mobile edge environment," *Journal of Cloud Computing*, vol. 11, no. 1, Dec. 2022, doi: 10.1186/s13677-022-00359-6.
- [41] S. Wang, Z. Pu, Q. Li, and Y. Wang, "Estimating crowd density with edge intelligence based on lightweight convolutional neural networks," *Expert Systems with Applications*, vol. 206, Nov. 2022, doi: 10.1016/j.eswa.2022.117823.
- [42] J. Xu, Y. Zhao, and F. Xu, "RDPNet: a single-path lightweight CNN with re-parameterization for CPU-type edge devices," *Journal of Cloud Computing*, vol. 11, no. 1, Dec. 2022, doi: 10.1186/s13677-022-00330-5.
- [43] W. He, D. Yang, H. Peng, S. Liang, and Y. Lin, "An efficient ensemble binarized deep neural network on chip with perception-control integrated," *Sensors*, vol. 21, no. 10, May 2021, doi: 10.3390/s21103407.
- [44] L. Mocerino and A. Calimera, "Fast and accurate inference on microcontrollers with boosted cooperative convolutional neural networks (BC-Net)," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 1, pp. 77–88, Jan. 2021, doi: 10.1109/TCSI.2020.3039116.
- [45] L. W. Qin *et al.*, "Precision measurement for industry 4.0 standards towards solid waste classification through enhanced imaging sensors and deep learning model," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–10, May 2021, doi: 10.1155/2021/9963999.
- [46] A. K. Sharma, A. Rana, and K. K. Kim, "Lightweight image classifier for CIFAR-10," *Journal of Sensor Science and Technology*, vol. 30, no. 5, pp. 286–289, Sep. 2021, doi: 10.46670/JSST.2021.30.5.286.
- [47] B. Wang, F. Ma, L. Ge, H. Ma, H. Wang, and M. A. Mohamed, "Icing-EdgeNet: a pruning lightweight edge intelligent method of discriminative driving channel for ice thickness of transmission lines," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2021, doi: 10.1109/TIM.2020.3018831.
- [48] M. Xia *et al.*, "SparkNoC: An energy-efficiency FPGA-based accelerator using optimized lightweight CNN for edge computing," *Journal of Systems Architecture*, vol. 115, May 2021, doi: 10.1016/j.sysarc.2021.101991.
- [49] T. Yoo, S. Lee, and T. Kim, "Dual image-based cnn ensemble model for waste classification in reverse vending machine," *Applied Sciences (Switzerland)*, vol. 11, no. 22, Nov. 2021, doi: 10.3390/app112211051.
- [50] J. Yu, G. Zhou, S. Zhou, and J. Yin, "A lightweight fully convolutional neural network for sar automatic target recognition," *Remote Sensing*, vol. 13, no. 15, Aug. 2021, doi: 10.3390/rs13153029.

## BIOGRAPHIES OF AUTHORS






**Muhammad Abbas Abu Talib**    received the B.Eng. degree (Hons.) in Electrical and Electronic Engineering from the Universiti Teknologi MARA (UiTM), in 2023. He is currently working as Test and Assembly Engineer at Filpal (M) Sdn. Bhd. He can be contacted at email: mat.abbas.talib@gmail.com.






**Samsul Setumin**    received the B.Eng. degree (Hons.) in Electronic Engineering from the University of Surrey, in 2006, and the M.Eng. degree in Electrical-Electronic and Telecommunication from the Universiti Teknologi Malaysia, in 2009. He obtained his Ph.D. degree from Universiti Sains Malaysia in 2019 in the imaging field. Since 2010, he has been a Lecturer with the Universiti Teknologi MARA, Malaysia. He was a Test Engineer with Agilent Technologies (M) Sdn. Bhd., and Intel Microelectronics (M) Sdn. Bhd., for a period of one year. His research interests include computer vision, image processing, pattern recognition, and embedded system design. He can be contacted at email: samsuls@uitm.edu.my.






**Siti Juliana Abu Bakar**    received her B.Eng. degree (Hons.) in Electronic Engineering from UTeM, Malaysia in 2019 and Master ESDE (Electronic System Design Engineering) from USM, Malaysia, in the year 2015. Completed her Ph.D. in the field of Automation and Control System from Universiti Sains Malaysia, Engineering Campus in 2020. She is currently working as senior lecturer at Centre for Electrical Engineering Studies, Universiti Teknologi MARA Campus Pulau Pinang Malaysia. Prior working as a senior lecturer at UiTM, she worked at Intel Product (M) Sdn.Bhd more than 5+ years as Validation Engineer. She can be contacted at email: siti Juliana@uitm.edu.my.



**Adi Izhar Che Ani**    is a Senior Lecturer at the Centre for Electrical Engineering, Universiti Teknologi MARA, Cawangan Pulau Pinang (UiTM CPP), with a Master's Degree in Engineering from Universiti Malaya Malaysia (2012). He obtained his Bachelor's Degree in Electrical and Electronics Engineering from the University of Miyazaki (Japan) in 2007. His research interests are the fields of artificial intelligence. He can be contacted at email: adiizhar@uitm.edu.my.



**Denis Eka Cahyani**    holds a Bachelor of Computer Science (S.Kom.) in Computer Science, Master of Computer Science (M.Kom.) in Computer Science, Universitas Indonesia in 2015 besides several professional certificates and skills. She holds a Bachelor of Informatics degree from Universitas Sebelas Maret, Indonesia in 2013. She is currently lecturing with the department of Mathematics at Universitas Negeri Malang, Malang, Indonesia. She is a member of the Engineers and the Institute of Electrical and Electronics Engineers (IEEE) Indonesia Section. Her research areas of interest include data science, natural language processing, and artificial intelligent. She can be contacted at email: denis.eka.cahyani.fmipa@um.ac.id.