A Survey on Light-weight Convolutional Neural Networks: Trends, Issues and Future Scope

Abdul Mueed Hafiz

Department of Electronics & Communication Engineering, Institute of Technology, University of Kashmir, Srinagar, J&K, 190006, India
E-mail: mueedhafiz@uok.edu.in

Received 08 February 2023; Accepted 01 July 2023;
Publication 11 August 2023

Abstract

Today with the substantial increase in the computing power of small devices and systems new challenges are emerging. For example, how to control a small handheld device which has the computing capabilities of a desktop Personal computer (PC) used five years ago. Devolving decision-making power to the device in order to make it more intelligent e.g. in the case of autonomous driving, is an interesting area. Deep learning has paved the way for this task due to its reliable decision-making capabilities which are quite popular. However for small devices there are constraints like availability of limited computation hardware, less power due to small batteries, need for real-time as well as accurate decision-making abilities, etc. In this regard, light-weight Convolutional Neural Networks (CNNs) are a valuable tool. Lightweight CNNs like MobileNets, ShuffleNets, CondenseNets, etc. are deep networks which have a much lesser number of layers and a much smaller number of parameters as compared to their larger CNN counterparts like GoogLeNet, Inception, ResNets, etc. Due to their unique advantages for small stand-alone systems, light-weight CNNs are used in these systems. In this

Journal of Mobile Multimedia, Vol. 19_5, 1277–1298.
doi: 10.13052/jmm1550-4646.1957

literature survey the notable light-weight CNNs along with their architecture, design features, performance metrics, advantages, etc are discussed. The trends, issues and future scope in the area are also discussed. It is hoped that by studying this survey, the reader will engage in research in this interesting area.

Keywords: Lightweight CNNs, deep learning, survey, convolutional neural networks, limited-hardware devices.

1 Introduction

Deep learning has emerged as a popular approach for machine vision applications [5,9,10,14,19,42,44,46,51,60]. It is also used for other applications like Natural Language Processing (NLP) [1]. The basics of deep learning include concepts like convolution, weight sharing, using the Rectified Linear Unit (ReLU) activation functions, etc. Ascribed to convolution, the Convolutional neural networks (CNNs) are invariant to translation, rotation, scaling, etc. in the input and this feature makes them robust. CNNs are pre-trained on large datasets like ImageNet [6] and are later fine-tuned for two important reasons [29]: (i) The features learned by CNNs from the large-data help them generalize better, and (ii) The pre-trained CNNs are expert at avoiding over-fitting for smaller downstream applications during fine-tuning. The performance accuracy of CNNs depends on the architecture [11, 12] and their training approach [15]. There are many CNNs which have a very large number of parameters. As mentioned earlier, for training these CNNs large datasets are required. Some notable CNNs include AlexNet [32], VGG [47], GoogLeNet [48], ResNet [21], and DenseNet [26]. In this regard notable datasets for computer vision include ImageNet [6] and OpenImage [33]. In deep learning neural networks with large numbers of layers are used for classification, prediction and regression. CNNs were introduced by LeCun et. al [34] and they rose to popularity with the introduction of the AlexNet CNN [32] which gave extraordinary classification accuracy on the ImageNet challenge [6]. Since then deep learning has broken many performance records on tasks like computer vision [10, 13, 14, 16, 19, 20, 35, 46, 50, 51], speech recognition [3,7], financial market forecasting [2,61], etc. However all of the above mentioned CNNs are computationally exhausting due to which they are not suitable for the implementation in the embedded systems or limited hardware systems.

For object-detection, a system like a drone, car, missile, etc. collects data from its sensors like cameras, etc. and sends the data to an offline processing unit for analysis [4]. By doing this, the unit is able to save power by offloading its computational tasks. However, wireless transmission is slower due to which an additional cost of latency is added to the system. The need for compute-intensive hardware for video processing is a challenge in fitting deep-learning based techniques on low cost and low power computation platforms. In many applications like robotics, autonomous cars, autonomous drones and virtual reality, the video recognition algorithms need to be run quickly on a low compute-capable hardware platform [24]. For this purpose, CNNs that are more suitable for on-board object detection in real-time need to be developed. Here it is vital to reduce the number of model parameters and use faster calculations in the CNN while saving power by reducing the computations. Hence the need arises for light-weight CNN architectures which are to be used on limited compute-capability hardware.

The new generation of lightweight CNNs are used for embedded systems in units like drones, cars, defence systems, etc. Some advantages of using light-weight CNNs e.g. in the case of drones, are that the battery life and flight time could be increased. In the computing hardware an additional 0.5 to 1 W power is required for cooling for each watt of power dissipated [41]. Also, a low-power computation system may reduce thermal problems and cooling requirements, which is an important issue for systems like autonomous drones, etc. Hence the need for light-weight CNNs comes to the fore.

In previous survey papers like [4] which was published in 2019 the early architectures of a limited number of lightweight CNNs were discussed briefly. My paper not only considerably expands the argument but also discusses the latest versions of the notable lightweight CNNs. It also compares their performance on state-of-the-art image databases for computer vision. This survey paper also discusses the latest trends in the area. Hence a gentle introduction is given to light-weight CNNs while mentioning their trends, the major issues and the future scope of the area. Through the medium of this survey it is hoped that the reader will develop a decent insight into the field of light-weight CNNs. It is also hoped that the reader will engage in research in this interesting field.

The main contributions of the work are given below:

- An overview of the notable state-of-the-art light-weight CNNs, their features and unique advantages is given
- The current trends for light-weight CNNs are given

- The performance comparisons, major issues and future scope of light-weight CNNs are also discussed

The rest of the paper is organized as follows. Section 2 discusses the works related to light-weight CNNs. Section 3 briefly discusses the current trends, issues and future scope in the area. The conclusion is given in Section 4.

2 Related Work

In this section, I discuss the notable light-weight CNNs used so far and I also discuss their important features along with their advantages.

2.1 MobileNet

The MobileNet CNN series is a popular light-weight CNN series used for computer vision applications. Its variants are discussed below.

2.1.1 MobileNet-V1

MobileNet-V1 [24] was the first light-weight CNN in the MobileNet series used for mobile and embedded vision systems. It used the technique of splitting the convolution into depth wise-separable convolution and pointwise convolution respectively for building a light-weight CNN. Also, it introduced two hyper-parameters which were used to build smaller and lower latency models for mobile and embedded vision tasks. One of these hyper-parameters was the width-multiplier which allowed the thinning of the number of channels. The second hyper-parameter was the resolution-multiplier which reduced the spatial dimensions of the features. Although MobileNet-V1 was not generally as accurate as the heavier CNNs, however it fared much better in the resource v/s accuracy trade-off. It also gave high accuracy with limited hardware resources. In [24], the authors proposed MobileNet-SSD which used depth-wise separable convolutions. They used MobileNet-V1 as the backbone CNN. MobileNet-SSD achieved a notable accuracy on the MS COCO dataset [37].

2.1.2 MobileNet-V2

MobileNet-V2 [45] was an updated version of MobileNet-V1 with more efficiency in terms of speed and accuracy. For example, for the MS COCO dataset [37] MobileNet-V2 was 2x faster than MobileNet-V1 and also slightly more accurate in terms of performance. MobileNet-V2 was a much

faster model making it suitable for real-time tasks. In their work [45], the authors claimed that a MobileNet-V2 with a width-multiplier of 0.25 and a resolution-multiplier of 0.714 achieved 28.1 frames per second (fps) on an Nvidia Jetson TX2 GPU, 31.5 fps on an Intel Core i5-6200U CPU and 164 fps on a K40 Desktop GPU. The authors of [45] proposed SSDLite which was based on the MobileNet-V2 backbone. SSDLite outperformed YOLO-V2 on the MS COCO [37] dataset with 20x more efficiency and 10x lesser size.

2.1.3 MobileNet-V3

MobileNet-V3 [23] is the latest version in the MobileNet series. Its main contribution is the use of the AutoML technique [22] for finding the best possible CNN architecture for a particular problem. This is in contrast to the handcrafted design of previous versions of the MobileNet architecture. MobileNet-V3 leverages two AutoML techniques viz., MnasNet [49] and NetAdapt [57]. MobileNetV3 first searches coarsely for a possible architecture using MnasNet, which in turn uses Reinforcement learning (RL) for selection of the optimum configuration. Next, the CNN is fine-tuned by using NetAdapt which is a complementary technique used for trimming underutilized channels with small decremental steps.

MobileNet-V3 also uses a squeeze-and-excitation block [25] in its architecture. The squeeze-and-excitation block improves the representation quality of the network by modelling the inter-channel feature interdependencies. The CNN uses feature recalibration by which it learns the use of global information for selective emphasis of informative features and for suppressing the ones which are less useful. MobileNet-V3 extends MobileNet-V2 by incorporating the squeeze-and-excitation blocks in the search space. This technique gives a more robust CNN architecture.

Another optimization of MobileNet-V3 is the redesigning of some 'expensive' layers in the CNN model. Some layers in MobileNet-V2 were foundational for the accuracy of the model, but they also introduced latency. By using the optimization techniques, MobileNet-V3 removes three expensive layers in MobileNet-V2 without sacrificing any performance accuracy. Table 1 shows the overall architecture of MobileNet-V3.

Table 2 shows the performance of MobileNet-V3 against it previous versions on the MS COCO dataset.

The code for the MobileNet CNN series is available at: https://github.com/tensorflow/models/tree/master/research/slim/nets/mobilenet.

Table 1 Overall architecture of MobileNet-V3 (Large version) [23]. SE denotes if there is a squeeze and excite module in the block. NL is the non-linearity type. HS is the h-swish activation function, and RE is is the ReLU activation function. s denotes the stride

Input	Operator	Exp Size	$\#out$	SE	NL	s
224×224×3	conv2d	–	16	–	HS	2
112×112×16	bneck 3×3	16	16	–	RE	1
112×112×16	bneck 3×3	64	24	–	RE	2
56×56×24	bneck 3×3	72	24	–	RE	1
56×56×24	bneck 5×5	72	40	✓	RE	2
28×28×40	bneck 5×5	120	40	✓	RE	1
28×28×40	bneck 5×5	120	40	✓	RE	1
28×28×40	bneck 3×3	240	80	–	HS	2
14×14×80	bneck 3×3	200	80	–	HS	1
14×14×80	bneck 3×3	184	80	–	HS	1
14×14×80	bneck 3×3	184	80	–	HS	1
14×14×80	bneck 3×3	480	112	✓	HS	1
14×14×112	bneck 3×3	672	112	✓	HS	1
14×14×112	bneck 5×5	672	160	✓	HS	2
7×7×160	bneck 5×5	960	160	✓	HS	1
7×7×160	bneck 5×5	960	160	✓	HS	1
7×7×160	conv2d 1×1	–	960	–	HS	1
7×7×960	pool 7×7	–	–	–	–	1
1×1×960	conv2d 1×1	–	1280	–	HS	1
1×1×1280	conv2d 1×1	–	k	–	–	1

Table 2 Performance of MobileNet-V1 [24], MobileNet-V2 [45] and MobileNet-V3 [23] on the MS COCO dataset [37]. mAP is *Mean Average Precision*. With channel reduction, MobileNet-V3 is faster than MobileNet-V2 by 27% with almost same mAP

Backbone	mAP	Latency (msec)	Parameters (M)
MobileNet-V1	22.2	228	5.1
MobileNet-V2	22.1	162	4.3
MobileNet-V3	22.0	119	3.22
MobileNet-V3 Small	16.1	43	1.77

2.2 SqueezeNet

SqueezeNet [28] is a small CNN which achieves the same performance accuracy of the AlexNet CNN on the ImageNet dataset with 50x lesser parameters as shown in Table 3. SqueezeNet can also have 500x lesser parameters than AlexNet by using deep compression methods [18]. Table 3 shows the performance of SqueezeNet as compared to AlexNet.

Table 3 Performance of AlexNet and SqueezeNet on the ImageNet dataset [28]

CNN Model	Model Size (MB)	Size Reduction v/s AlexNet	Top-1 Accuracy (%)	Top-5 Accuracy (%)
AlexNet	240	1x	57.2	80.3
SqueezeNet (32 bit)	4.8	35x	57.5	80.3
SqueezeNet (8 bit) Compressed	.66	363x	57.5	80.3
SqueezeNet (6 bit) Compressed	.47	510x	57.5	80.3

The SqueezeNet CNN has three important aspects:

1. It uses (1×1) filters instead of (3×3) filters, since the former has 9x fewer parameters than the latter.
2. It uses lesser number of input channels for the (3×3) filters by using squeeze layers.
3. It downsamples the later stages for keeping a large feature map.

The novel Fire module is the basic building block of the SqueezeNet CNN which consists of two layers viz., a squeeze-convolution layer having (1×1) filters, and an expansion layer having a mix of (1×1) and (3×3) convolution filters.

SqueezeNet has a stand-alone convolution layer (conv1) followed by eight Fire modules (fire2 to fire9) and lastly another final convolution layer (conv10). Table 4 shows the overall architecture of SqueezeNet.

Inspired by YOLO [**?**] using a SqueezeNet backbone, Wu et al. [54] proposed the SqueezeDet CNN for autonomous cars. Zhang et al. in their work [58] have developed a modified SqueezeNet integrated into the Attention based U-Net [43] and have used their novel lightweight network for detecting forest fires. They call their lightweight CNN model *ATT Squeeze U-Net*.

2.3 ShuffleNet

I now discuss another notable light-weight CNN series viz. the ShuffleNet series.

2.3.1 ShuffleNet-V1

ShuffleNet-V1 [59] was an efficient light-weight CNN architecture for mobiles having limited computing power. The CNN gave better performance than MobileNet on the ImageNet and MS COCO dataset tasks. The

Table 4 Overall architecture of SqueezeNet [28]

Input	Output Size	Filter Size/Stride	Depth	$s_{1\times1}$ Squeeze	$e_{1\times1}$ Expand	$e_{3\times3}$ Expand
input	224×224×3					
conv1	111×111×96	7×7/2 (×96)	1			
maxpool1	55×55×96	3×3/2	0			
fire2	55×55×128		2	16	64	64
fire3	55×55×128		2	16	64	64
fire4	55×55×256		2	32	128	128
maxpool4	27×27×256	3×3/2	0			
fire5	27×27×256		2	32	128	128
fire6	27×27×384		2	48	192	192
fire7	27×27×384		2	48	192	192
fire8	27×27×512		2	64	256	256
maxpool8	13×12×512	3×3/2	0			
fire9	13×13×512		2	64	256	256
conv10	13×13×1000	1×1/1 (×1000)	1			
avgpool	1×1×1000	13×13/1	0			

ShuffleNet-V1 architecture was composed of a stack of novel ShuffleNet units grouped in three stages. The first block in every stage was applied with a stride of 2. The outputs were the same in every stage but were doubled for the next stage. ShuffleNet architecture used two new operations for reducing the computation cost viz., point-wise group convolution and channel shuffling. The channel shuffling operation allowed division of the CNN channels into several sub-groups and then fed every group in the next layer with different sub-groups. ShuffleNet-V1 achieved 13x speed-up over the AlexNet CNN on an ARM mobile phone while achieving similar accuracy.

2.3.2 ShuffleNet-V2

The authors of [40] have conducted several empirical studies and base the improved ShuffleNet-V2 on the following experimental outcomes:

1. Using balanced convolutions with equal channel-width.
2. Being aware of the cost of using group-convolutions.
3. Reducing the fragmentation degree.
4. Reducing the element-wise operations.

They note that the above properties also depend on the platform characteristics like memory manipulation and code optimization and should be taken into account for the practical CNN design. Accordingly they introduce

Table 5 Overall architecture of ShuffleNet-V2 [40]

Layer	Output Size	KSize	Stride	Repeat	Output Channels
Image	224×224				3
Conv1	112×112	3×3	2		24
MaxPool	56×56	3×3	2	1	
Stage2	28×28		2	1	116
	28×28		1	3	
Stage3	14×14		2	1	232
	14×14		1	7	
Stage4	7×7		2	1	464
	7×7		1	3	
Conv5	7×7	1×1	1	1	1024
GlobalPool	1×1	7×7			
FC					1000
No. of Weights					2.3M

Table 6 Performance of ShuffleNet-V2 on the MS COCO dataset for 500 MFlops [40]

Model	mAP (%)	GPU Speed (Images/sec)
MobileNet-V2	30.6	72
ShuffleNet-V1	32.9	60
ShuffleNet-V2	**33.3**	**83**

a simple operator called *channel split* where in the input of the feature channels is split into two branches. After convolution, the two branches are concatenated. They also remove the "Add" operation in ShuffleNet-V1. The ReLU element-wise operations and depth-wise convolution operations are used only in one branch. Three successive element-wise operations are used viz., *Concat, Channel Shuffle* and *Channel Split*, and these are merged into one element-wise operation. For down-sampling, the module is modified slightly. The channel splitting operation is removed which leads to doubling of the number of output channels. Table 5 shows the overall architecture of ShuffleNet-V2.

Using their improved ShuffleNet-V2 architecture the authors obtain better performance as compared to Xception, MobileNet-V1, MobileNet-V2 and ShuffleNet-V1 on the MS COCO dataset. They obtain a top mAP of 34.2% with a slightly more GPU Speed of 87 Images/sec for these CNNs for the object detection task. Table 6 shows the performance of ShuffleNet-V2 compared to those of MobileNet-V2 and ShuffleNet-V1 on the MS COCO dataset.

The code for the ShuffleNet CNN series is available at: https://github.com/megvii-model/ShuffleNet-Series/tree/master/ShuffleNetV2%2B

2.4 L-Net

A light-weight object detection CNN with the ShuffeNet-V2 [40] based backbone, called the L-Net is presented in the work [17]. It has a backbone which is obtained by modifying the depth convolution from (3×3) to (5×5) and also reducing the number of channels in the input. L-Net is more image discriminative with the help of the Pyramid-pooling module and Attention-pyramid module which are both used after the backbone. Experimentation shows that the L-Net CNN uses only 1.54M Flops (Floating point operations) and achieves 70.2% mAP (mean average precision) on the PASCAL VOC 2007 task, and 21.8% mAP on the MS COCO task.

2.5 CondenseNet

Another notable light-weight CNN series is the CondenseNet series.

2.5.1 CondenseNet-V1

In their paper [27], the authors developed the CondenseNet light-weight CNN with good efficiency. Their CNN combined dense connectivity in their novel convolution module called learned group-convolution. The dense connectivity was used for feature map re-use in the CNN and the learned group-convolutions removed those inter-layer connections for which the feature re-use was superfluous. For testing, the CNN was implemented using group-convolutional operations which led to efficient practical computation. The authors of CondenseNet-V1 showed that it was much more efficient than CNNs like ShuffleNet. They obtained a better Top-5 classification error of 8.3% with a 4.8M parameter CNN under 529 MFlops.

2.5.2 CondenseNet-V2

CondenseNet-V1 [27] showed that feature-reuse in deep networks through dense connections achieved high computational efficiency by removing redundant features. In their work [56], the authors propose a novel approach called Sparse feature reactivation (SFR) used for increasing the feature-utility. In their CNN called CondenseNet-V2 every layer is able to simultaneously learn:

1. Selective reuse of the set of most important features from previous layers.

Table 7 Overall architecture of CondenseNet-V2 [56]. Squeeze and excite (SE) or Hard-swish non-linearity function (HS) are applied to the respective dense layers wherever indicated

Input	Operator
224×224	Conv2D 3×3 (Stride 2)
112×112	Dense
112×112	AvgPool 2×2 (Stride 2)
56×56	Dense
56×56	AvgPool 2×2 (Stride 2)
28×28	Dense (HS)
28×28	AvgPool 2×2 (Stride 2)
14×14	Dense (SE,HS)
14×14	AvgPool 2×2 (Stride 2)
7×7	Dense (SE,HS)
1×1	AvgPool 7×7
1×1	Conv2D 1×1
1x1	FC

Table 8 Performance of CondenseNet-V2 on the MS COCO dataset. The detection framework used for all these CNNs here is RetinaNet [36]. A variant of CondenseNet-V2 has been used [56]

Backbone CNN	Backbone FLOPs	mAP (%)
MobileNet-V2	300M	29.7
ShuffleNet-V2 1.5x	299M	29.1
CondenseNet-V2-C	305M	**31.7**

2. Active update of the set of previous features for increasing their utility for ensuing layers.

Table 7 shows the overall architecture of CondenseNet-V2.

The experimentation shows that CondenseNet-V2 achieves promising performance on image classification tasks like ImageNet and object detection tasks like MS COCO, both in terms of efficiency as well as speed. They achieve a Top-1 error rate of 35.6% with 2M parameters under 46 MFlops. Table 8 shows the performance of CondenseNet-V2 compared to MobileNet-V2 and ShuffleNet-V2 on the MS COCO dataset.

The code for the CondenseNet CNN series is available at: https://github.com/jianghaojun/CondenseNetV2.

2.6 Other Notable Light-weight CNNs

Other notable light-weight CNNs developed which have promising performance are PeleeNet [52], Tiny-YOLO [30], Lira-YOLO [39], ResMoNet [8],

Table 9 Performance comparison of various lightweight CNNs and some large CNNs for the video-classification task using the Kinetics-600 video dataset [38] as given in [31]. The cycles per second (cps) execution speed of the models on the task is also indicated for the Titan XP GPU [31]

Backbone CNN	Backbone MFLOPs	Parameters (M)	Speed (cps) on Titan XP GPU)	Error Rate
3D-ShuffleNet-V1-0.5x	42	0.55	398	.3551
3D-ShuffleNet-V2-0.25x	42	0.83	442	.2573
3D-MobileNet-V1-0.5x	46	1.17	290	.3174
3D-MobileNet-V2-0.2x	42	0.96	357	.2414
3D-ShuffleNet-V1-1.0x	125	1.52	269	.4531
3D-ShuffleNet-V2-1.0x	119	1.91	243	.4610
3D-MobileNet-V1-1.0x	137	3.91	164	.4007
3D-MobileNet-V2-0.45x	126	1.40	203	.3647
3D-ShuffleNet-V1-1.5x	235	2.92	204	.5275
3D-ShuffleNet-V2-1.5x	215	3.16	186	.5205
3D-MobileNet-V1-1.5x	273	8.22	116	.4824
3D-MobileNet-V2-0.7x	245	2.05	130	.4559
3D-ShuffleNet-V1-2.0x	393	4.78	161	.5684
3D-ShuffleNet-V2-2.0x	360	6.64	146	.5517
3D-MobileNet-V1-2.0x	454	14.10	88	.4853
3D-MobileNet-V2-1.0x	446	3.12	93	.5065
3D-SqueezeNet	728	2.15	682	.4052
ResNet18	5557	33.24	334	.5765
ResNet50	6782	44.24	183	.6300
ResNet101	10612	83.29	143	.6418
ResNeXt101	6932	48.34	122	.6830

E3D [53], MobileNeXT [55], etc. For additional information the reader may refer to the cited literature.

Table 9 shows the comparison of performance of some notable lightweight CNNs which have been discussed above.

As can be observed from Table 9, the lightweight CNNs like ShuffleNets, MobileNets, SqueezeNets, etc. have much lesser parameters than their large counterparts like ResNets. Also the execution speeds indicated in term of cycles per second (cps) are much higher for the lightweight CNNs as compared to their largel couterparts.

3 Trends, Issues and Future Scope

With the development of smaller more powerful handheld and stand-alone devices and systems, research is being done in making them more intelligent and capable of automatic decision-making, e.g. the systems used in autonomous vehicles, aircraft, stand-alone defence systems, drones, mobile phones, gaming devices, Internet-of-things (IOT) devices, intelligent sensor nodes, etc. Although it is becoming easier to devolve decision-making to these systems however it is also important to protect them from misuse, corruption, hacking, software attacks, etc. In this context, applications like deep learning are very useful in making these hardware-limited systems intelligent and somewhat 'self-aware'. The development of light-weight deep learning frameworks or CNNs helps in tailoring the heavy, compute-intensive and resource-hungry deep CNNs to these light-weight devices. There are some basic aspects which the design of light-weight CNNs needs to adhere to: (i) The CNNs should have lesser computing needs, (ii) The CNNs should use much lesser power, (iii) The CNNs should be fast enough to be deployed in real-time, (iv) The CNNs should be accurate enough to make quick and reliable decisions after precise classification or detection. The current generation of light-weight CNNs are able to achieve some part of these aspects as is shown by their experimental results.

The research in the above area is still in its initial stages. The main approach applied for design of the light-weight CNN is the reduction of number of layers, operators, activators, etc. or replacing these by faster and lighter versions. Techniques like $AutoML$ help craft light-weight CNNs by intelligent automatic domain search. However, no reliable auto-design technique has been proposed so far other than a handful. Also, light-weight CNN development remains primarily a hand-crafted technique with a trial-and-error procedure. This is the case with CNN design in general. Also, one more issue in this regard is that the light-weight CNN design is usually limited to modification of pre-existing heavy CNNs. The lower accuracy of light-weight CNNs is an important issue due to their lesser generalization or function-fitting capabilities.

Although light-weight CNN design is a challenging area, there is a lot of potential. Automatic light-weight CNN design is an open research area because a much smaller number of layers has to be designed and fused, instead of designing a large number of compute-intensive layers as done in traditional large CNNs. Another interesting potential area in this regard is the

use of light-weight CNN ensembles, which may distribute the computation-load of one CNN onto many smaller and lighter ones. This may unlock better performance and even lesser memory-, power- or time-footprint. Also, design of evolving and adaptive light-weight CNNs for varying battery life of the hardware unit is another interesting area.

4 Conclusion

In this survey paper, the topic of light-weight CNNs was touched. The introduction discussed the outline of the paper. Following this, the 'Related Works' section discussed some notable light-weight CNNs and gave their overview. The performance improvements alongwith performance comparisons were also discussed wherever feasible. In the next section, the trends, issues and future scope of the area were discussed. It came to fore by the discussion that using light-weight CNNs is vital for modern day limited-capability devices. It is hoped that through this survey paper, the reader will be encouraged to study and engage in the area of design of light-weight CNNs which will pave the way for automation of modern-day small and intelligent devices.

Conflict of Interest

The author declares no conflict of interest.

Funding Statement

The work is not funded.

References

[1] Amna Amanat, Muhammad Rizwan, Abdul Rehman Javed, Maha Abdelhaq, Raed Alsaqour, Sharnil Pandya, and Mueen Uddin. Deep learning for depression detection from textual data. *Electronics*, 11(5), 2022.

[2] Matin N. Ashtiani and Bijan Raahemi. News-based intelligent prediction of financial markets using text mining and machine learning: A systematic literature review. *Expert Systems with Applications*, 217:119509, 2023.

[3] Kishor Barasu Bhangale and Mohanaprasad Kothandaraman. Survey of deep learning paradigms for speech processing. *Wireless Personal Communications*, 125(2):1913–1949, 2022.

[4] Abdelmalek Bouguettaya, Ahmed Kechida, and Amine Mohammed Taberkit. A survey on lightweight cnn-based object detection algorithms for platforms with limited computational resources. *International Journal of Informatics and Applied Mathematics*, 2(2):28–44.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[7] Jean Louis K. E Fendji, Diane C. M. Tala, Blaise O. Yenke, and Marcellin Atemkeng. Automatic speech recognition using limited vocabulary: A survey. *Applied Artificial Intelligence*, 36(1):2095039, 2022.

[8] Rodolfo Ferro-Pérez and Hugo Mitre-Hernandez. ResMoNet: a residual mobile-based network for facial emotion recognition in resource-limited systems. *arXiv preprint arXiv: 2005.07649v1*, 2020.

[9] Ross Girshick. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[11] A. M. Hafiz, R. A. Bhat, and M. Hassaballah. Image classification using convolutional neural network tree ensembles. *Multimedia Tools and Applications*, pages 1–18, 2022.

[12] A. M. Hafiz and M. Hassaballah. Digit image recognition using an ensemble of one-versus-all deep network classifiers. In M. Shamim Kaiser, Juanying Xie, and Vijay Singh Rathore, editors, *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*, pages 445–455, Singapore, 2021. Springer Singapore.

[13] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey of deep learning techniques for medical diagnosis. In Milan Tuba, Shyam

Akashe, and Amit Joshi, editors, *Information and Communication Technology for Sustainable Development*, pages 161–170, Singapore, 2020. Springer Singapore.

[14] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey on instance segmentation: state of the art. *International journal of multimedia information retrieval*, 9(3):171–189, 2020.

[15] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. *Fast Training of Deep Networks with One-Class CNNs*, pages 409–421. Springer International Publishing, Cham, 2021.

[16] Abdul Mueed Hafiz, Rouf Ul Alam Bhat, Shabir Ahmad Parah, and M Hassaballah. SE-MD: a single-encoder multiple-decoder deep network for point cloud reconstruction from 2D images. *Pattern Analysis and Applications*, pages 1–12, 2023.

[17] Jin Han and Yonghao Yang. L-Net: lightweight and fast object detector-based shufflenetv2. *Journal of Real-Time Image Processing*, 18(6):2527–2538, 2021.

[18] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[19] Mahmoud Hassaballah and Ali Ismail Awad. *Deep learning in computer vision: principles and applications*. CRC Press, 2020.

[20] Mahmoud Hassaballah and Hosny Khalid M. *Recent Advances in Computer Vision: Theories and Applications*. Springer, 2019.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[22] Xin He, Kaiyong Zhao, and Xiaowen Chu. AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, 2021.

[23] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *2019 IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[24] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[25] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[26] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society.

[27] Gao Huang, Shichen Liu, Laurens van der Maaten, and Kilian Q. Weinberger. CondenseNet: an efficient densenet using learned group convolutions. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[28] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. SqueezeNet: alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[29] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4037–4058, 2021.

[30] Ivan Khokhlov, Egor Davydenko, Ilya Osokin, Ilya Ryakin, Azer Babaev, Vladimir Litvinenko, and Roman Gorbachev. Tiny-YOLO object detection supplemented with geometrical data. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5. IEEE, 2020.

[31] O. Kopuklu, N. Kose, A. Gunduz, and G. Rigoll. Resource Efficient 3D Convolutional Neural Networks. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1910–1919, Los Alamitos, CA, USA, Oct 2019. IEEE Computer Society.

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[33] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

[34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[35] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[36] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE international conference on computer vision (ICCV)*, pages 2980–2988, 2017.

[37] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *2014 European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[38] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Learning to localize actions from moments. *arXiv preprint arXiv: 2008.13705*, 2020.

[39] Zhou Long, Wei Suyuan, Cui Zhongma, Fang Jiaqi, Yang Xiaoting, and Ding Wei. Lira-YOLO: A lightweight model for ship detection in radar images. *Journal of Systems Engineering and Electronics*, 31(5):950–956, 2020.

[40] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. ShuffleNet V2: Practical guidelines for efficient cnn architecture design. In *2018 European Conference on Computer Vision (ECCV)*, September 2018.

[41] Sparsh Mittal. Power management techniques for data centers: A survey. *arXiv preprint arXiv:1404.6681*, 2014.

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[44] Faiqa Sajid, Abdul Rehman Javed, Asma Basharat, Natalia Kryvinska, Adil Afzal, and Muhammad Rizwan. An efficient deep learning framework for distracted driver detection. *IEEE Access*, 9:169270–169280, 2021.

[45] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[46] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, apr 2017.

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv: 1409.1556*, 2014.

[48] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, Los Alamitos, CA, USA, jun 2015. IEEE Computer Society.

[49] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *2019 IEEE/CVF conference on computer vision and pattern recognition*, pages 2820–2828, 2019.

[50] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[51] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, Los Alamitos, CA, USA, jun 2015. IEEE Computer Society.

[52] Robert J. Wang, Xiang Li, and Charles X. Ling. Pelee: A real-time object detection system on mobile devices. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[53] Yunfei Wang, Rong Li, Zheng Wang, Zhixin Hua, Yitao Jiao, Yuanchao Duan, and Huaibo Song. E3D: An efficient 3D CNN for the recognition of dairy cow's basic motion behavior. *Computers and Electronics in Agriculture*, 205:107607, 2023.

[54] Bichen Wu, Forrest Iandola, Peter H Jin, and Kurt Keutzer. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *2017 IEEE conference*

on computer vision and pattern recognition workshops, pages 129–137, 2017.

[55] Zhang Xiang Yan Chun-man and Wang Qingpeng. Face expression recognition based on improved MobileNeXt. *Research Square Preprint, DOI: 10.21203/rs.3.rs-2270472/v1*, 16 November 2022.

[56] Le Yang, Haojun Jiang, Ruojin Cai, Yulin Wang, Shiji Song, Gao Huang, and Qi Tian. CondenseNet V2: Sparse feature reactivation for deep networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3569–3578, June 2021.

[57] Tien-Ju Yang, Andrew Howard, Bo Chen, Xiao Zhang, Alec Go, Mark Sandler, Vivienne Sze, and Hartwig Adam. Netadapt: Platform-aware neural network adaptation for mobile applications. In *2018 European Conference on Computer Vision (ECCV)*, pages 285–300, 2018.

[58] Jianmei Zhang, Hongqing Zhu, Pengyu Wang, and Xiaofeng Ling. ATT squeeze U-Net: a lightweight network for forest fire detection and recognition. *IEEE Access*, 9:10858–10870, 2021.

[59] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[60] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, 2017.

[61] Yanli Zhao and Guang Yang. Deep learning-based integrated framework for stock price movement prediction. *Applied Soft Computing*, 133:109921, 2023.

Biography

Abdul Mueed Hafiz received the B.Tech degree in Electronics & Communication Engineering in 2005 from the National Institute of Technology, Srinagar, J&K, India; the M.Tech degree in Communication & Information Technology in 2008 from the National Institute of Technology, Srinagar; and the Ph.D in Computer Vision in 2018 from the University of Kashmir, Srinagar, J&K, India. Currently he serves as the Head of the Department, and Sr. Assistant Professor, at the Department of Electronics & Communication Engineering, Institute of Technology, University of Kashmir. He has publications in international journals, conferences and book chapters. He serves as a reviewer for journals in IEEE, IET, ACM, Springer, etc. and is a member of A.C.M. His research interests include Neural Networks, Learning Systems, and Computer Vision.