An application of MoblieNet on distinguishing river otter and sea otter

Zekai Wang

School of Computer Engineering and Science, Shanghai University, 99 Shangda Road, Baoshan District, Shanghai, China

Riosted@shu.edu.com

Abstract. Sea otters are in great danger nowadays while river otters are free of worrying about predators, and they occupy some common habitats. Hence, it is meaningful to develop a Convolutional Neural Network (CNN) classifier aiming at distinguishing these two species when rescuers are confused. This paper illustrates a developed model of MobileNetV2 with the support of the Convolutional Block Attention Module (CBAM) attention module, and the model is well-performed in recognizing accuracy, memory usage, time consumption and portability. The result shows that the accuracy rises by 2.6% compared with the original version though at the cost of 61% growth in both inferencing time and flash usage. The final model is capable to be deployed on any smart device which can install a browser and call the service of a camera, enabling users to tell sea otters from river otters under wild circumstances rapidly, accurately, and immediately. This study serves as a reference for a better understanding of animal recognition.

Keywords: MoblieNet, Sea otters, Convolutional Neural Network.

1. Introduction

As is known to all, two major species of otters are under public sights: the sea otter and the river otter. Most times, we can easily find a fluffy river otter in a zoo which is no more than two hours of drive from a normal residence in a city, or even beside a river next to the resident block. However, on contrary, its similar friend, the sea otter which only lives in specific parts of the world like the coastline of Alaska is now in great danger. The number of sea otters that lived in nearshore habitats might once reach 300,000 surrounding the Pacific Ocean before they had been hunted by merchants who were eager at their shiny leathers for about 170 years. Though having realized the fact, the preservation project started at the end of the 19th century hardly saw noticeable progress in keeping their habitats, leading to the miserable consequence that fewer than 125,000 sea otters are remaining nowadays, which figure is recently lower than it was [1].

The differences between them are, however, a bit hard for humans to distinguish. Firstly, sea otters are bigger than river otters in most conditions. Another distinction is that a sea otter's tail is short and flattened while that part of a river otter is often long and pointed [2]. But it's foolish to tell them apart just by the size and tails as sometimes there are exceptions. What's more, if it comes to extreme circumstances like categorizing a huge number of otters, doing it manually (common in most countries) seems impossible. That is to say, a method to distinguish these two species rapidly and accurately is necessary.

Trying to combine this task with machine learning since it's now feasible to distinguish any species provided that there is an abundance of their photos, this paper aims to illustrate a CNN classifier based on transfer learning with the frame of MobileNetV2 and CBAM attention module on the website platform Edge Impulse. As a consequence of the well-selected frame and deliberately decided hyperparameters, the model of this project can tell quite clearly if a picture contains a sea otter or a river otter with a considerably little time consumption and an acceptable peak Random Access Memory (RAM) usage. To make the model available for anyone and anywhere, the classifier can be deployed on any device which is capable to install a browser and linking to the Internet due to the support of the Edge Impulse platform, though certain wireless data traffic usage is indispensable when users need to classify outdoors.

2. Related work

There are several ways for animal recognition in recent years, and most of the well-behaved ones are based on CNN models.

2.1. Statistical Machine Learning

In 2017, Trnovszky and his coresearchers compared some classical recognition methods, namely Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Local Binary Pattern Histogram (LBPH), and Support Vector Machine (SVN) with the CNN network which was widely used to distinguish objects [3]. Their study emphasizing animal distinction shows that the CNN frame undoubtedly prevails over those outdated ones at performance by approximately 15% to 20%, and in other words, proves that deep learning surpasses old simple statistical machine learning means in animal recognizing fields.

2.2. Convolutional Neural Network (CNN)

In the same year, H Nguyen and his collaborators released a systematic paper on applying deep learning frameworks to recognizing animals [4]. In the paper, they deploy 4 kinds of frameworks on a specific device which can automatically supervise endangered animals like bandicoots, red foxes and some specific categories of rabbits. After comparing the recognition effect of AlexNet, VGG-16, GoogLeNet, and ResNet-50, they conclude that all three frameworks perform similarly but ResNet-50 does slightly better (less than 5%). The result reveals that it has become harder to improve the accuracy of the framework than to reduce its time, memory, and power consumption, thus the latter is what scientists nowadays are focusing on, whose improvement aims at increasing portability as well.

2.3. MobileNet

In 2021, a group of researchers at Yunnan University of Finance and Economics trying to detect wildlife proposed a network based on You Only Look Once (YOLO) in which the backbone features extraction module is replaced by a redesigned MobileNet framework [5]. The reason why they do the replacement is a requirement of deploying the model on a portable tiny device which needs to be placed in forests for about one week successively without any human intervention. Their research makes it clear that the improved model gets an average precision of 93.6% and 3.8 FPS computed by the CPU, which is 7.7% higher in accuracy and incredibly 475% better in Frame Per Second (FPS). The consequence indicates that MobileNet is playing a more and more important role in animal recognition for peoples' convenience since most observations and distinguishments are taken by devices which do not have the capability to do some complicated calculations. Hence, a model for distinguishing animals (in this case, otters) rapidly and accurately is required.

3. Methods

This chapter will be about the application of MobileNetV2 with the adapted attention mechanism, and it is specified for otter recognition. It consists of two parts, namely the structure of MobileNetV1 and V2 as well as the application of the CBAM attention mechanism on MobileNetV2.

3.1. MobileNet and MobileNetV2

MobileNet, a newly proposed model, grabbed the researchers' attention in recent years. It was first developed on 17 Apr 2017 by a group of Google researchers including Andrew G. Howard and his collaborators [6]. Like what is illustrated in its name, MobileNet which is designed for TensorFlow only at first is specialized for mobile usage circumstances. What's of significant difference against other CNN models is the notion of depthwise separable convolutions, which is actually a combination of two separate convolution operations: a depthwise convolution follows with a pointwise convolution. Unlike normal convolutions, the depthwise convolution coming at first convolutes R, G, and B respectively instead of mixing them together, leading to output with 3 channels with individual weights. This operation is supposed to act as a filter. What follows is the pointwise convolution with a 1×1 kernel accumulating all results of three channels. These two operations noticeably reduce the computing work by 8-9 times by providing the pointwise convolution module a high dimension working condition and require less weight learning process. As a spinoff, it's now possible to pose two ReLU6 functions to one complete convolution [7].

With the depthwise separable convolution still taking effect, residual connections and expand/projection layers are introduced in MobileNetV2.

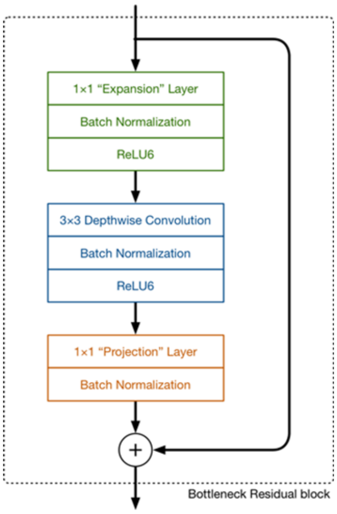

Figure 1. Structure of bottleneck residual block.

As the structure of MobileNetV2 shows above, the new layer is called the expansion layer whose purpose is to expand the number of channels, and the parameter called the expansion factor decides how much the data is expended. On the contrary, the layer responsible for pointwise convolution called the projection layer now changes to make the number of channels smaller instead of keeping them stable or doubled, and that is the reason why it is known as the projection layer (projecting more figures to fewer ones). These layers are also called bottleneck layers because they make the data amount smaller. Another new concept named residual connection is added to the model, reversing relevant operations in ResNet to help with the flow of gradients in case it is too time-wasting to adjust all weights of each layer to a proper extent in a restricted time. That is to say, if the first several layers are enough to describe the standard result function, the rest layers will just be "skipped" and all of them will be regarded as a simple $f(x) = x$ function [8][9].

Table 1. Structure of MobileNetV2.

Input	Operator	t	c	n	s
224×224×3	conv2d	-	32	1	2
112×112×32	bottleneck	1	16	1	1
112×112×16	bottleneck	6	24	2	2
56×56×24	bottleneck	6	32	3	2
28×28×32	bottleneck	6	64	4	2
14×14×64	bottleneck	6	96	3	1
14×14×96	bottleneck	6	160	3	2
7×7×160	bottleneck	6	320	1	1
7×7×320	conv2d 1×1	-	1280	1	1
7×7×1280	avg pool 7×7	-	-	1	-
1×1×1280	conv2d 1×1	-	k	-	-

As a result, the MobileNetV2 taken in the project consists of 11 kinds of layers as the table 1 followed shows, with “t” meaning expansion factor, “c” meaning the number of output channels, “n” meaning the repeated times of the layer, and “s” meaning the stride of learning.

In overview, the two new techniques are helpful to noticeably increase the processing speed and enable people to do classifications on portable devices though there is a cost of lower feature generating accuracy.

3.2. MobileNetV2 based on the CBAM attention mechanism

The classical MobileNetV2 performs well enough to win itself a high status in mobile device deployment, however, it can do better. Attention mechanism, which prevails in machine learning in recent years, can make the performance excellent with regard to picture recognition, especially image classification tasks. So, let us come to the attention mechanism which acts like the attention of humans. When people look at an image, they do not look at every pixel and remember them in their brains as ordinary conventional neural networks do, and instead, they focus on some “highlights” of it. Taking a picture of a baby as an instance, when people take a look at it, what they pay attention to first is the face, especially the eyes and expressions, not the whole picture and irrelevant details like its background.

According to the attempts of applying the attention mechanism to the CNN network by a group of Korean experts including S Woo and his coresearchers, an elaborately designed attention sub-module called CBAM is proposed by them [10].

The module has two sub-modules, one called the channel attention module is responsible for knowing “what” is significant while another one named the spatial attention module is designed to find “where” the important part is in the picture. Since the input of the task of distinguishing otters are otters’ portraits probably with some irrelevant obstacles, we need both modules, namely the channel attention module and the spatial attention module. So, the procedure is that after an image is put into the model, the two attention modules will calculate the attention parameter and be responsible for ‘what’ and ‘where’ the object is respectively. As a result of their study, they find that putting the two modules in sequential order is more reasonable than keeping them in a parallel one. Bolei Zhou and his team find that the average-pooling layer is useful in understanding the context of the object [11], and apart from that, S Woo argues that the max-pooling layer is also of great importance, so this improvement takes them all into practice.

Firstly, a max-pooling layer and an average-pooling layer take place after the convolution module calculates, which aims at aggregating space information from the feature map for the channel attention module. This operation is able to generate two context describers. After the procession of a shared Multilayer Perceptron (MLP), the channel attention map of it is formed before undergoing an element-wise accumulation.

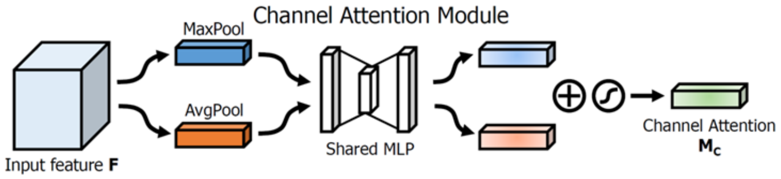

Figure 2. Structure of CBAM channel module [10].

As for the spatial attention module which is a complement to the former, there is a following max-pooling layer followed with an average-pooling layer. Then they are combined and sent to a convolution with a 7×7 filter, and finally, a sigmoid function as the activation function. After that, what needs to be done is to connect the attention module to the MobileNetV2, thus fully-connected layers are required.

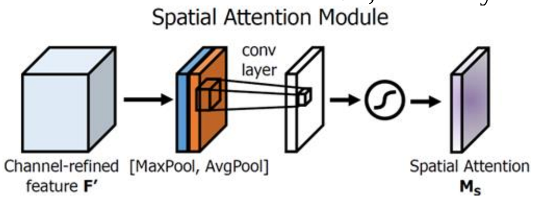

Figure 3. Structure of CBAM spatial module [10].

When the combination referred to above is finished, a large number of fully-connected layers are used to transform tensors into vectors and the output of each convolution layer of MobileNetV2 is linked with a channel attention module followed by a spatial attention module so that the improved MobileNet obtains the ability to focus on the most important part of an otter image, namely its appearance and size. What’s more, the residual connection part is adapted to fit the improved model by connecting the output of convolution module and the accumulator together to help the model skip useless parts.

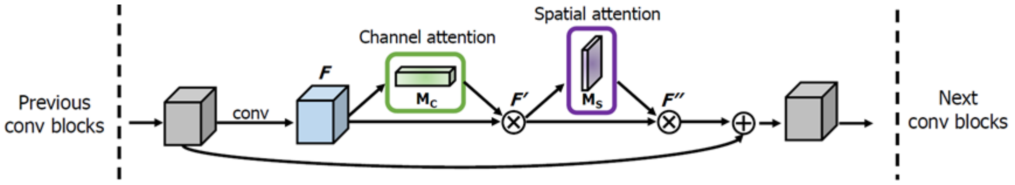

Figure 4. Combination of MobileNetV2 and CBAM [10].

As Shahi, T. B. recommends in his attempt to apply CBAM to CNN [12], 128 units of the dense layer are pretty good according to his experiment since too many dense layers will noticeably and exponentially increase the amount of computation while too few layers cannot describe the whole feature completely. They think that a figure between 100 and 200 is reasonable and 128, which is a power of 2 is suitable for calculation in expansion and projection. In addition, the ReLU6 (As for input larger than 6, the output is fixed to 6 instead of rising as the normal ReLU function does) as its activation function is adopted. However, the dropout rate is set at 0.3 to keep the model robust, which is a conservative figure in case a too high rate causes an overfitting problem or a too small one leading to poor performance. When the dropout rate is adjusted to over 0.5, the model can hardly learn anything from the rest information which means that it is unacceptable.

4. Results and Discussion

This part consists of several sections, including prepossession of recognition, experimental details, ablation experiment, and comparisons with other state-of-art methods.

4.1. Prepossession of recognition

All images are ought to be resized into equal dimensions. The MobileNetV2 accepts two kinds of image sizes: 96×96 and 160 × 160 and for the sake of accuracy, a 160 × 160 one is adopted. Another problem is the method of resizing. In order not to lose the main feature of the picture (the otters) and destroy the image information, fitting the longest side is adopted. This method will fill the outside part of the long side with dark to enlarge the picture into a square avoiding twisting it, and then compression of pixels takes place.

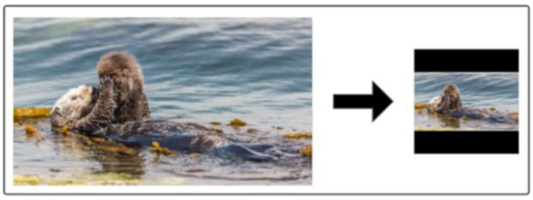

Figure 4. Image resizing.

After prepossessing, the extracting module of MobileNetV2 will generate the features of each picture. The images keep their raw colour depth, RGB, to maintain the information. By procession of Digital Signal Processing (DSP), the raw information of each pixel is converted to specified formats; the number of features varies by picture, and for the picture in Figure 5, there are 134,318 rows of data.

4.2. Experimental details

It’s hard to find datasets specified for otters, so all the pictures are collected from Google Pictures manually. The size of the dataset is 250, which consists of 125 pieces of river otter images and 125 pieces of that of sea otters. All images are elaborately selected and handled to fit the requirements MobileNetV2 needs: jpg format, below 2 MB, and only selecting those with a whole figure of otters as we are absolutely capable of taking such a photo under supposed using circumstances. In addition, the channel of the pictures is set to RGB instead of grey because three channels carry more information and colour is significant to distinguish otters as they have leathers of different colours varying from deep brown to light brown. What's more, the proportion of the training set and testing set is regulated to the ratio of 4:1 to achieve the best training effect since too high a rate leads to an overfitting problem while a too low one cannot learn enough features.

As there are not so many pictures in the dataset, it may be necessary to expand the variety of images through a data processing method called data argumentation which is able to change the training data randomly during the process and prevent the model from taking shortcuts by remembering the superficial features of the image. What needs to be mentioned is that the effect of this technique is not definite and in common, the smaller the dataset is, the more unpredictable the result is. After taking comparisons, it is clear that expanding the dataset in this way can increase the performance by less than 1% and drop the loss rate to a little extent as well without any noticeable side effects, so this procedure is adopted.

As for learning rate and training epochs, they are set to 0.0005 and 20 rounds respectively after comparing with slightly altered ones. What’s more, the final dense layers are omitted because the MobileNetV2 is good enough to cover the using occasion.

4.3. Ablation experiment

After prepossessing and feature extraction, the developed models of different parameters and versions are applied to the dataset respectively.

All the models are trained and validated under such parameters: RGB channels, a size at 160×160, 20 rounds of training, a stride at 0.0005, dropout rate at 0.3, with data argumentation on.

Table 2. Ablation Experiment Results.

Model used	Model version	Recognition Accuracy (%)	Loss	River Otter Accuracy (%)	Sea Otter Accuracy (%)	F_1 Score
MobileNetV2 1.0	Quantized (int8)	92.5	0.23	94.4	90.9	0.925
	Unoptimized (float32)	95.0	0.08	94.4	95.5	0.945
MobileNetV2 1.0 + CBAM module	Quantized (int8)	97.5	0.09	100	95.5	0.975
	Unoptimized (float32)	97.5	0.08	100	95.5	0.975

Thanks to the residual bottleneck block of MobileNetV2 which helps the model to avoid too many layers to learn, the model is both well-performed and portable. What's more, it is clear that CBAM modules can improve the performance of the models, especially the river otter distinguishment part (Table 2). The CBAM module enables the model to concentrate on important parts and avoid being distracted by obstacles. Let us take a look at the consumption performance then.

Table 3. Consumption Performance.

Model used	Model version	Inferencing time (ms)	Peak RAM usage (MB)	Flash usage (MB)
MobileNetV2 1.0	Quantized (int8)	431	1.4	1.8
	Unoptimized (float32)	1336	4.9	5.4
MobileNetV2 1.0 + CBAM module	Quantized (int8)	287	1.5	2.5
	Unoptimized (float32)	2149	4.9	8.7

On quantized occasions, the CBAM module, which undoubtedly makes the model larger by 39%, improved the overall accuracy by 5% while getting the loss rate reduced by half, and excels in otter recognition. However, the quantization procession decreases the storage usage and processing time by making the precision lower and inevitably get a reduction in performance, so it can do better when it comes to an unoptimized one. As for unoptimized circumstances, the accuracy is a little bit higher than those without a CBAM module, at 2.6%. The processing time is raised by 61%, from 1336 ms to 2149 ms while the flash consumption grows by 60 as well, from 5.4 MB to 8.7MB (Table 3).

In order to get better accuracy, an unoptimized one is adopted finally. It is easy to understand why the time and memory consumption rise after adding the CBAM module, but what is strange is that the attention module does not reduce the loss rate under unoptimized occasions. This problem is actually caused by the existence of the dropout rate which will always block random parts of the image and let the rest of it trained in case the model is learning too much and overfit, and it will prevent the model from completely converging [13]. In other cases, this problem can also be caused by a too high learning rate making the weights just jump around the converge point horizontally and repetitively, which will definitely make the performance poor. Fortunately, these problems have never been observed in this model as its accuracy is absolutely higher than normal models.

4.4. Comparisons with other state-of-art methods

When talking about deep learning, some CNN networks are always mentioned, namely LeNet-5, AlexNet, VGG and so on. MobileNetV2 is always compared with its ancestor, MobileNetV1 and its prototype, ResNet. As for the former one, the existence of the pointwise convolution module enables

V2 to increase the dimension and help the depth wise convolution module to work on a higher dimension, which can help make the performance better. What's more, linear bottlenecks prevent the model from vanishing gradient problems which may lead to poor performance since those 0-gradient units are "dead" and can never get back to life again [14]. With regard to the latter, ResNet takes features of the picture by standard convolution while V2 uses D-W convolution pairs. V2 rise the dimension and then decreases it as ResNet does a complete reversed one like a sand clock, and this improvement can noticeably make the recognition faster and more accurate (Table 4).

Table 4. Performance between V1 and V2.

Model used	Model version	Recognition Accuracy (%)	Loss	River Otter Accuracy (%)	Sea Otter Accuracy (%)	F_1 Score
MobileNetV1 0.25	Quantized (int8)	75	0.42	77.8	72.7	0.75
	Unoptimized (float32)	82.5	0.38	83.3	82.8	0.825
MobileNetV2 1.0 + CBAM module	Quantized (int8)	97.5	0.09	100	95.5	0.975
	Unoptimized (float32)	97.5	0.08	100	95.5	0.975

The comparison experiment clearly shows that MobileNetV2 undoubtedly prevails its ordinary version, V1 at accuracy in both quantized and unoptimized occasions. In overview, MobileNetV2 takes the advantages of pioneers and improves their detrimental parts and undoubtedly makes better performance with a more attractive point: portability.

5. Conclusion

This essay aims at illustrating an improvement in applying the CBAM attention module to the existing MobileNetV2 and helps make the performance better at the cost of memory usage and time consumption to help distinguish between river otters and sea otters. The MobileNetV2 used is better in performance because of the improvement of residual bottleneck block and depth wise convolution module. A CBAM module which consists of a channel attention sub-module followed by a special attention sub-module is linked to each convolution module in order to make the model know what is important.

The model trained which is specified for otters distinguishment can be deployed on any phone or computer with a browser by just scanning a specific Quick Response (QR) code generated by the Edge Impulse platform which means it is completely rational, portable, and convenient.

Due to the open-source Edge Impulse website, it's feasible to get the recognizing Application Programming Interface (API) and then pack the website into a mini application. That can definitely get the processing speed faster than just running it on a simple website which can only call a few resources of our device, being a rational improvement direction.

References

[1] Estes, J. A. Catastrophes and conservation: lessons from sea otters and the Exxon Valdez. 1991 Sci, 254(5038), 1596-1596.

[2] https://www.seattleaquarium.org/animals/sea-otters

[3] Trnovszky, T., Kamencay, P., Orjesek, R., Benco, M., & Sykora, P. Animal recognition system based on convolutional neural network. 2017 Adv. Elec. Eng, 15(3), 517-525.

[4] Nguyen, H., Maclagan, S. J., Nguyen, T. D., Nguyen, T., Flemons, P., Andrews, K., & Phung, D.. Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. 2017 Int. conf. data sci. adv. Anal. 40-49.

[5] Zhao, T., Yi, X., Zeng, Z., & Feng, T. MobileNet-Yolo based wildlife detection model: A case

study in Yunnan Tongbiguan Nature Reserve, China. 2021 J. Intel. Fu. Sys., 1-11.

[6] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., & Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. 2017 arXiv preprint arXiv:1704.04861.

[7] Howard, A. G., Zhu, M. Mobilenets: Open-source models for efficient on-device vision. 2017 Google AI blog.

[8] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. Mobilenetv2: Inverted residuals and linear bottlenecks. 2018 Conf. Comput. Vis. Pat. Rec. 4510-4520).

[9] He, K., Zhang, X., Ren, S., & Sun, J. Deep residual learning for image recognition. 2016 Conf. Comput. Vis. Pat. Rec. 770-778.

[10] Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. Cbam: Convolutional block attention module. In Proceedings of the Euro Conf. Comput. Vis. 3-19.

[11] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. Learning deep features for discriminative localization. 2021 Conf. Comput. Vis. Pat. Rec. 2921-2929.

[12] Shahi, T. B., Sitaula, C., Neupane, A., & Guo, W. Fruit classification using attention-based MobileNetV2 for industrial applications. Plos one, 17(2), e0264586.

[13] Dickson, M. C., Bosman, A. S., & Malan, K. M. Hybridised Loss Functions for Improved Neural Network Generalization. 2021 Ar. Intel. Smart Sys. Conf. 169-181.

[14] Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. 1998 Int. J. Un., 6(02), 107-116.