Original Research Article

DIGITAL HEALTH

DIGITAL HEALTH
Volume 10: 1–16
© The Author(s) 2024
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/20552076241271639
journals.sagepub.com/home/dhj
Sage

Lightweight convolutional neural network (CNN) model for obesity early detection using thermal images

Hendrik Leo[1], Khairun Saddami[2,3], Roslidar[2], Rusdha Muharar[2], Khairul Munadi[2] and Fitri Arnia[2,3]

Abstract

Objective: The presence of a lightweight convolutional neural network (CNN) model with a high-accuracy rate and low complexity can be useful in building an early obesity detection system, especially on mobile-based applications. The previous works of the CNN model for obesity detection were focused on the accuracy performances without considering the complexity size. In this study, we aim to build a new lightweight CNN model that can accurately classify normal and obese thermograms with low complexity sizes.

Methods: The DenseNet201 CNN architectures were modified by replacing the standard convolution layers with multiple depthwise and pointwise convolution layers from the MobileNet architectures. Then, the depth network of the dense block was reduced to determine which depths were the most comparable to obtain minimum validation losses. The proposed model then was compared with state-of-the-art DenseNet and MobileNet CNN models in terms of classification performances, and complexity size, which is measured in model size and computation cost.

Results: The results of the testing experiment show that the proposed model has achieved an accuracy of 81.54% with a model size of 1.44 megabyte (MB). This accuracy was comparable to that of DenseNet, which was 83.08%. However, DenseNet's model size was 71.77 MB. On the other hand, the proposed model's accuracy was higher than that of MobileNetV2, which was 79.23%, with a computation cost of 0.69 billion floating-point operations per second (GFLOPS), which approximated that of MobileNetV2, which was 0.59 GFLOPS.

Conclusions: The proposed model inherited the feature-extracting ability from the DenseNet201 architecture while keeping the lightweight complexity characteristic of the MobileNet architecture.

Keywords

Obesity, deep learning, convolutional neural network, thermal imaging

Submission date: 14 December 2023; Acceptance date: 5 June 2024

Introduction

Obesity is a global health concern with a rising prevalence that has far-reaching implications for individuals and healthcare systems.[1] The presence of a lightweight convolutional neural network (CNN) model with a high-accuracy rate and small complexity can be useful in building an early obesity detection system. Such models are expected to have a balanced accuracy and complexity to ensure real-time and resource-efficient diagnosis.[2] However, some previous works[3–5] focused

[1]Postgraduate School of Engineering, Universitas Syiah Kuala, Banda Aceh, Indonesia
[2]Department of Electrical and Computer Engineering, Universitas Syiah Kuala, Banda Aceh, Indonesia
[3]Telematics Research Center, Universitas Syiah Kuala, Banda Aceh, Indonesia

Corresponding author:
Fitri Arnia, Department of Electrical and Computer Engineering, Universitas Syiah Kuala, Banda Aceh, 23111, Indonesia.
Email:f.arnia@usk.ac.id

on the accuracy of the CNN models without considering the complexity of the size.

In this study, we aim to build a new lightweight CNN model that can accurately classify normal and obese thermograms while having low complexity size so it could be embedded as a mobile-based early obesity detection system. We propose a customized CNN architecture derived from the fusion of DenseNet201 and MobileNet architectures to achieve this objective. We modify the standard convolution layer structure from DenseNet201 and then factorize it into separable depthwise convolution layers from MobileNet. Then, we also reduced the network depth in the dense blocks. The novel architecture is designed to inherit the feature-extracting capability of DenseNet201 while incorporating the lightweight complexity characteristics of MobileNet.

This research comprehensively evaluates the proposed customized CNN model, comparing its performance with state-of-the-art CNN models such as MobileNet[6] and DenseNet201.[7] The comparison is based on several key metrics, including classification performances, computational cost measured in terms of floating-point operations per second (FLOPS), model size quantified by the number of parameters, and memory consumption or model size (in megabyte (MB)).

Related work

CNNs use in medical image analysis has recently significantly advanced, especially in detecting and diagnosing diseases.[8] A CNN algorithm could be applied to classify and analyze various medical image modalities such as X-rays,[9] computer tomography (CT),[10,11] magnetic resonance imaging (MRI),[12] and positron emission tomography[13] scans. A CNN model will extract the given input images, learn the features, and then classify or segment the output images.[14]

In the context of obesity detection, CNN has been employed to diagnose obesity by segmenting subcutaneous and visceral adipose fat from CT scans and MRI images.[15–17] A different approach to diagnosing obesity is also made by quantifying subcutaneous and visceral fat areas from CT images through the CNN algorithm.[18,19] Some studies use bio-modeling techniques of the finite element model for analyzing the stress on knee joints between normal and obese subjects, which has the potential to be applied to diagnosing obesity by feeding up the measured model into machine learning or neural network classifiers.[20] However, the adoption of these technologies in terms of public access still needs to be improved and challenging. An obesity detection method that the public could easily access remains a topic of ongoing research.

Some studies, such as,[21–23] have explored the potential of thermal imaging, also known as infrared thermography, as an alternative approach to obesity detection. This technique involves measuring the temperature of brown adipose fat in the human body.[24–26] The studies reported that there were significant temperature pattern differences between obese and normal individuals, and these features could be extracted and learned by CNN models.[27,28] Recent studies[3–5] have proposed a customized CNN model, drawing inspiration from existing architectures, to classify thermal images into obese and normal classes. However, these studies did not consider the computation cost and model size metrics of their proposed model. These complexity metrics of computation cost and model size would be useful to increase the efficiency and applicability of the CNN model in mobile-based applications.

The implementation of CNN applications is rapidly expanding, with a noteworthy trend being the integration of CNN models into mobile devices. However, for seamless deployment on these devices, the imperative is a lightweight CNN model characterized by minimal memory consumption, low computation cost, and reduced complexity.[2,29] Various studies have been undertaken to achieve such lightweight CNN models. These include approaches such as combining straightforward CNN architectures with diverse machine learning classifiers to reduce learning parameters,[30] customizing basic CNN architectures with techniques such as overlapping pooling and increased convolution stride,[31] adapting the LiteFlowNet architecture and incorporating deep skip connection gated recurrent units,[32,33] and modifying state-of-the-art lightweight CNN architectures such as ShuffleNet[34–36] and MobileNet.[37,38] Collectively, these studies contribute to developing mobile-friendly CNN models with enhanced efficiency and resource utilization.

In the pursuit of developing an efficient CNN model for obesity detection, our previous study, as detailed in Leo et al.,[39] involved fine-tuning established models, namely VGG19,[40] DenseNet201,[7] ResNet152V2,[41] and MobileNet.[6] Traditional CNN architectures such as VGG19 and DenseNet201 show better accuracy in classifying obese thermograms. However, their drawback lies in their high memory consumption and high computational cost. On the other hand, lightweight CNN models, such as MobileNet and ResNet, show a lower accuracy, yet their advantages lie in significantly reduced computation cost and model size.

The existing body of literature underscores the importance of developing CNN models for obesity diagnosis, especially in thermal imaging. The challenges lie in achieving a delicate balance between accuracy, computational efficiency, and model size, which was advantageous for deploying such models on mobile platforms. This study builds upon these insights by proposing a novel lightweight CNN architecture to address the specific requirements of mobile-based early obesity diagnosis systems.

Method

The nature of this study involves the development and evaluation of a novel CNN architecture specifically designed for detecting obesity using thermal image data. The study was conducted in the signal processing and multimedia laboratory at the Electrical and Computer Engineering Department of Syiah Kuala University in

Indonesia. The dataset collection and the experimental setup were conducted from April to December 2023. In this section, we will explain the method used to design the proposed novel CNN architectures.

The proposed CNN architecture is designed by choosing the DenseNet201 architecture as the baseline architecture.[7] Then, it is followed by modifying the convolutional layers into separable depthwise convolution layers from MobileNet architecture,[6] modifying the network depth in the dense blocks and the fully connected layers to our desired output classes. This section helps explain which ideas have been adopted in developing the proposed CNN architecture.

Load baseline architecture: DenseNet201

The first step in designing the proposed model was to load the baseline architecture of DenseNet201 as shown in Figure 1.[7] The DenseNet201 architecture consists of single convolution layers with kernel dimension of 7 × 7, a single max pool, four dense blocks, three transition blocks, and followed with fully connected layers. In a single dense block, multiple sequences of convolution and convolution layers exist. In transition blocks, only a single convolution layer and average pooling layers exist. While the feature extraction is in the bottom layer, the features are next fed into 1000 fully connected layers for 1000 classes of classification outputs.

The main concept of DenseNet lies in the densely connected layers in the dense blocks for combining the features maps in concat layers, as shown in Figure 2. The features map F has a size of $D_F \times D_F \times C_F$, features map G has a size of $D_G \times D_G \times C_G$, and features map H has a size of $D_H \times D_H \times C_H$. Instead of combining the feature maps F and features map G by using the connection layer of feature summation in ResNet,[41] the DenseNet architecture concatenated them into a new features map of H, which the dimension size could be calculated as follows:

$$D_H = D_F = D_G \tag{1}$$

$$C_H = C_F + C_G \tag{2}$$

where the size of D_H equal with the size of D_F and D_G, while the size of C_H equals to the summation of C_F and C_G. To do the concatenated operation, the size of the

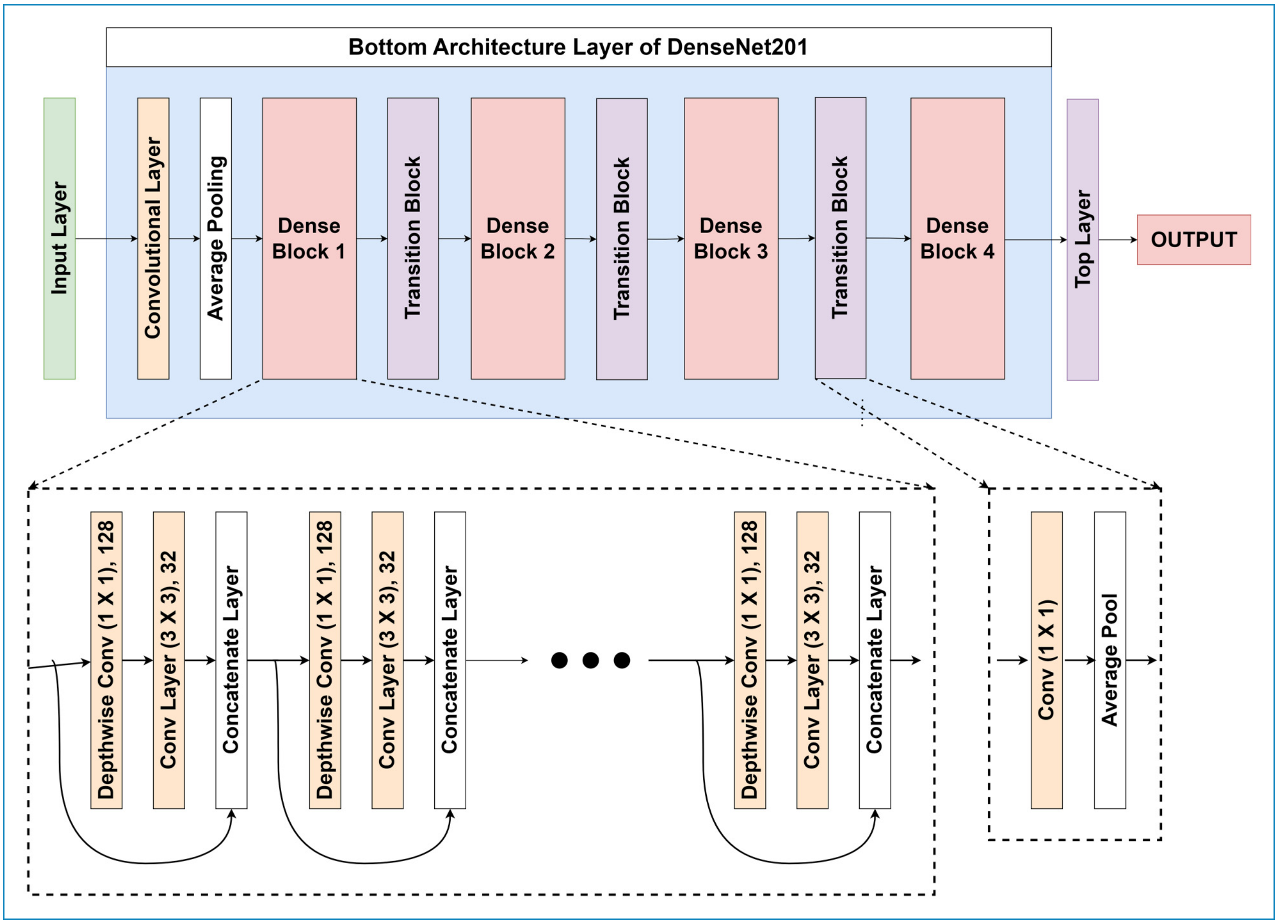


Figure 1. DenseNet201 architecture.

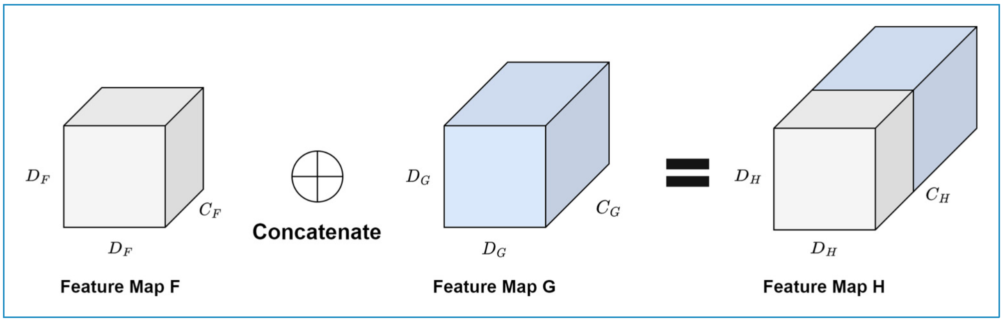


Figure 2. Concatenate operation in DenseNet201 dense blocks.

operands, in this case D_F and D_G, should be the same size. The features map from the previous convolution process is added and convoluted in the next convolution layers by performing a concatenate operation. By performing the concatenate connection multiple times in dense blocks, the flow of information and gradients of the networks was improved and became easier to train.[7]

To downsampling the features map, an 1×1 convolution layer and a 2×2 global average pooling layer were added in each transition block. The 1×1 convolution layer added to changes the features map's channel size by half of its original size. Suppose a features map F with sizes of $D_F \times D_F \times C_F$ is convoluted with a kernel with sizes of $D_K \times D_K \times C_K$. The size of D_K is 1 and the the size of C_K is computed as:

$$C_K = \frac{C_F}{2} \tag{3}$$

where the size of C_K is adjusted into half of the size of C_F. By performing the 1×1 convolution layers, the output features map size will become $D_F \times D_F \times C_K$. While the 2×2 average pool is added to downsampling the features map's width and height into half of the original size. Therefore, when a feature map was fed into the transition blocks, the output size of the feature map became half of the input size.

Modify convolution layers

After loading the base model of DenseNet architectures, we modify the convolution layers in the dense blocks into separable depthwise convolution layers from MobileNet architecture. MobileNet is a lightweight CNN model due to its separable depthwise convolution layer, significantly reducing computation cost and learning parameters.[6] The MobileNet architecture factorizes a standard convolution into depthwise and pointwise convolution. A standard convolution filters the input feature map and combines them into a new feature map output, as shown in Figure 3(a). While the separable depthwise convolutions consist of a single kernel of depthwise convolution as shown in 3(b) and 1×1 pointwise convolution as shown in Figure 3(c). The depthwise convolution filters the input feature map while the pointwise convolution combines them into a new feature map output, which has an equivalent result to the standard convolution.

In the standard convolution layer, a $D_F \times D_F \times C_F$ feature map F is convoluted by a $D_K \times D_K \times C_K$ kernel K as shown in Figure 3(a). Where D_F is the spatial width and height of feature map F, C_F is the number channel of feature map F, D_K is the kernel size of kernel K, and C_K is the number channel of kernel K. The output feature map for the standard convolution by assuming with 2×2 padding and single stride is $D_F \times D_F \times C_K$ feature map output with a computation cost of:

$$D_F \cdot D_F \cdot C_F \cdot D_K \cdot D_K \cdot C_K \tag{4}$$

where the computation cost depends on the multiplication of all the features map and the kernel size.

While in the depthwise convolution layer, a $D_F \times D_F \times C_F$ feature map F was convoluted by a single $D_K \times D_K$ kernel K as shown in Figure 3(b). The depthwise convolution operation filters the input map and produces a $D_F \times D_F \times 1$ feature map output. Where the output channel size is equal to the size of the kernel channel with computation cost of:

$$D_F \cdot D_F \cdot C_F \cdot D_K \cdot D_K \tag{5}$$

Then continued with the pointwise convolution layer where the $D_F \times D_F \times 1$ feature map is convoluted with 1×1 kernel with the size of C_K as shown in Figure 3(c). The pointwise convolution operation combines the feature map and produces a $D_F \times D_F \times C_K$ feature map output. The pointwise convolution operation produces a computation cost of:

$$D_F \cdot D_F \cdot C_K \tag{6}$$

The separable convolution operation from the depthwise and pointwise operation results in a $D_F \times D_F \times C_K$ output features map, which is equal to the standard convolution operations output. However, the separable

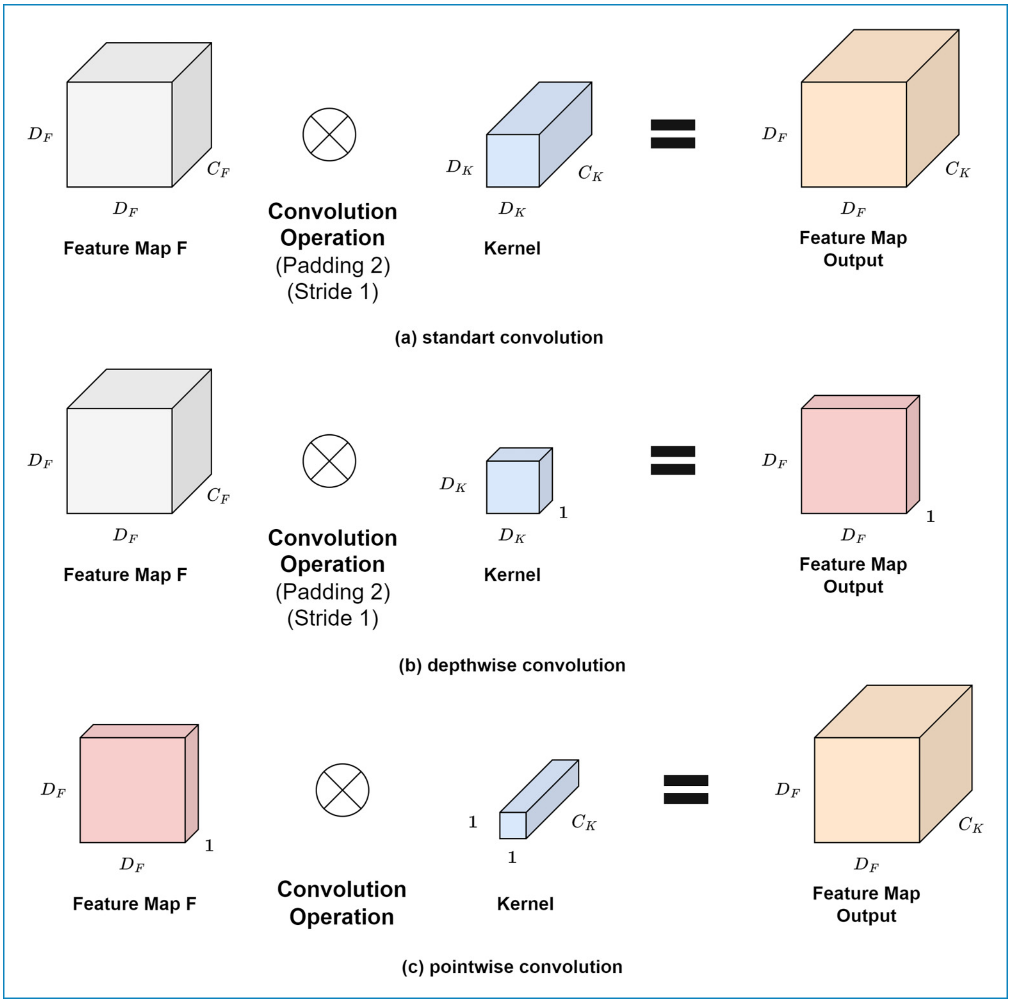


Figure 3. Convolution operations: (a) standard convolution, (b) depthwise convolution, and (c) pointwise convolution.

convolution operations significantly reduce the computation cost and model size. The computation cost of both depthwise and pointwise convolutions is calculated by following:

$$D_F \cdot D_F \cdot C_F \cdot D_K \cdot D_K + D_F \cdot D_F \cdot C_K \tag{7}$$

where the computation cost is the sum of the depthwise and pointwise convolution computation cost instead of multiplying all the spatial elements in standard convolution operation.

In this study, after we loaded the DenseNet201 base model, we modified the standard 3 × 3 convolution layer into a sequence of 3 × 3 depthwise and 1 × 1 pointwise convolution as shown in Figure 4 and Table 1. The 7 × 7 convolution layer after the input layer, 1 × 1 convolution layer in the transition block, and 1 × 1 convolution layer in the dense block remain untouched, only the 3 × 3 convolution layer in dense blocks are factorized into separable depthwise convolution layer.

Modify dense blocks

To reduce the computational cost of the proposed CNN model, we modified the repetition sequence of convolution operation in the dense block as shown in Table 1. Table 1 shows the CNN architecture comparison between the state-of-the-art DenseNet201[7] with our proposed CNN architecture, namely modified DenseNet (MD). We named our proposed CNN architecture based on its sequence repetition in the dense blocks MD 6-12-48-32, MD 6-12-12-6, and MD 2-4-4-2. In our proposed architecture, instead of using standard convolution in the dense blocks, we implement the depthwise separable convolution in each repetition sequence of the dense blocks. Note that each “Conv” operation shown in the table is the sequence of the batch normalization layer, rectified linear unit (ReLU) activation function, and a convolution layer.

The DenseNet201 architecture originally consisted of four dense blocks to extract the feature maps and three

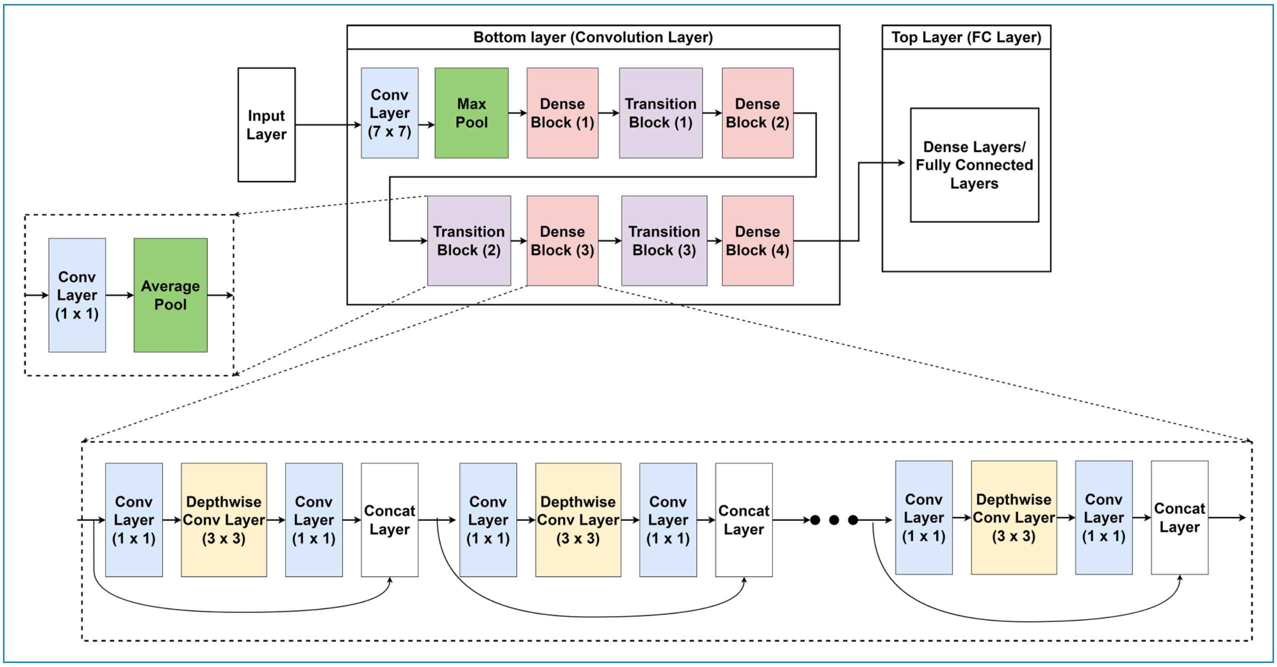


Figure 4. Proposed convolutional neural network (CNN) architecture.

transition blocks to reduce the feature maps' dimensions. Each dense block has repetition sequences of 1 × 1 conv layer, 3 × 3, and a concatenate layer. In DenseNet201, the repetition in the dense blocks was 6-12-48-32, which has very high computation operations. Then we modify the 3 × 3 standard convolution layer into a separable depthwise convolution technique from MobileNet and named MD 6-12-48-32, where the repetition in the dense blocks was the same as the DenseNet201. We also try to reduce the repetition sequence from 6-12-48-32 into 6-12-12-6 and 2-4-4-2 to evaluate the trade-off between the computation cost and the classification performance. As we see in Table 1, the width and height of the output feature map size of the MD are equal to the DenseNet201 architecture. The depthwise and pointwise convolution produce the same output size as the standard convolution with the reduced computation cost. However, the dimension channels between all the DenseNet201, MD 6-12-48-32, MD 6-12-12-6, and MD 2-4-4-2 are different due to the concatenate operation in the dense blocks. The denser the repetition of the dense blocks, the larger the channel output. Modifying the dense blocks was expected to achieve comparable classification performances with reduced computation costs.

Modify fully connected layers

After the feature extraction operation in the bottom layer, which consists of the input layer, dense blocks, and transition blocks, the feature maps are then classified through the top layer, which consists of a pooling layer and a fully connected layer. The state-of-the-art DenseNet201 top layer uses a global average pool layer and 1000 fully connected layers to classify 1000 classes of the ImageNet dataset.[42] In this study, we applied a global average pool layer to averaging and resizing the feature map into a single matrix. We added a dropout layer of 0.5 to regularize the proposed model. Then, forward to a fully connected layer with a size of 256 (with ReLU activation function) and fully connected with a size of 2 (with softmax function)

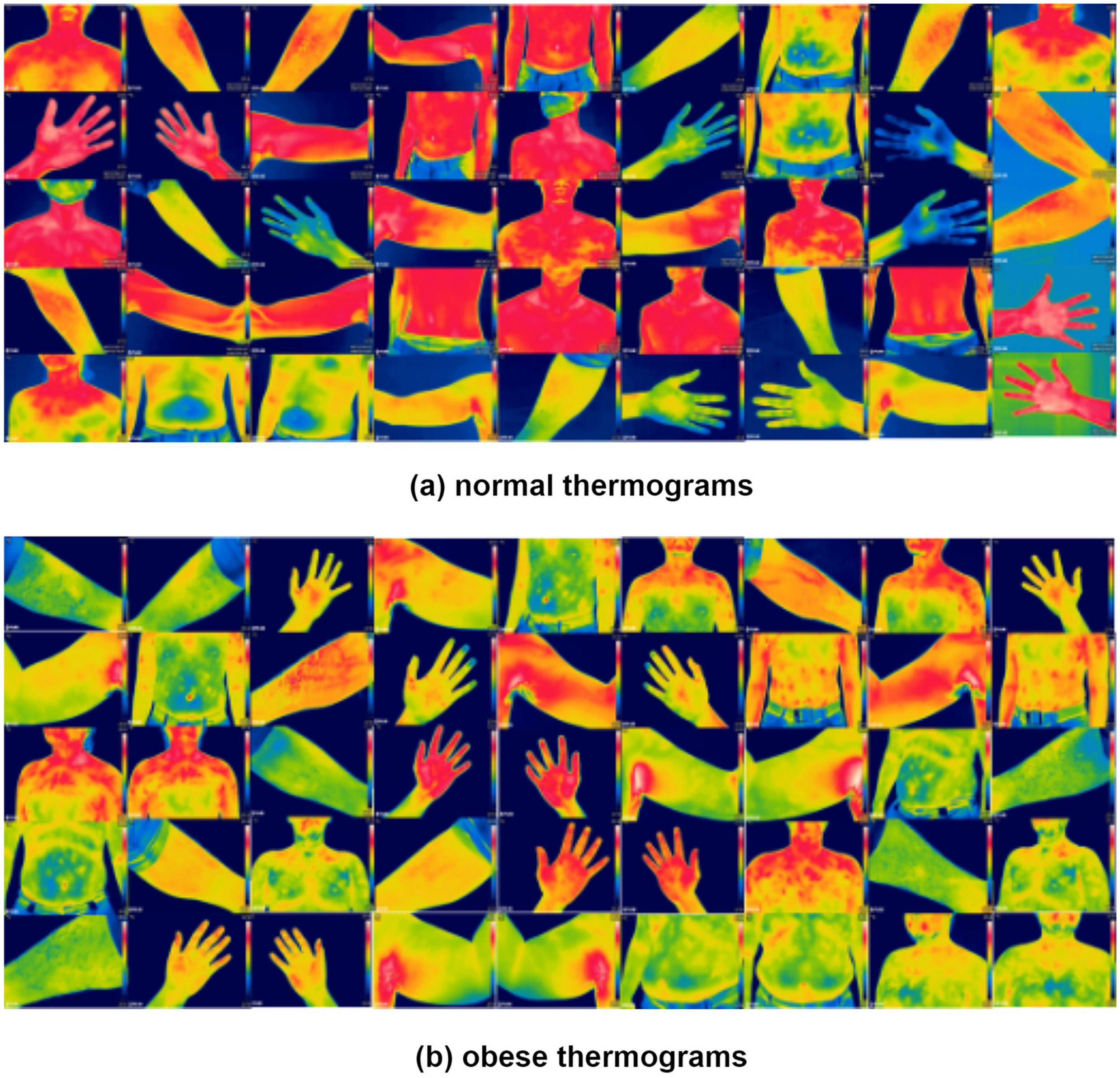
Experimental setup

This section will describe the detailed experimental setup we used to develop the lightweight CNN model for obesity detection. We describe the dataset used in this study and the training platform used to train, validate, and test the proposed model.

Dataset

The dataset we used in this study was already published and can be accessed at Leo et al.[43] The subject inclusion & exclusion criteria, image acquisition protocol, thermal camera setups, and image pre-processing procedures were explained in detail in Leo et al.[43]

The image dataset was captured from 126 male participants in Indonesia, aged 18 to 25 years. Participants' images with body mass index (BMI) <25 were labeled as normal images, while participant images with BMI more than or equal to 25 were labeled as obese images. The dataset consists of five body regions: supraclavicular (SCV), abdomen, forearm, palm, and shank regions as


(a) normal thermograms

(b) obese thermograms

Figure 5. Obesity thermogram dataset of supraclavicular, abdomen, forearm, palm, and shank regions: (a) normal thermograms and (b) obese thermograms.

shown in Figure 5, with 1260 images with two classes: obese and normal as shown in Table 2. We set the training with 1000 images, then held the 130 images for the validation set and 130 images for the testing set. Each five-body region contains 252 images, which is 20% of the total images. For the train set, each body region contains 114 normal images and 86 obese images. In each validation and testing set, each body region contains 14 normal images and 12 obese images.

The dataset images were three-channel red–green–blue colored images with the size of 224 × 224 × 3. The images were already pre-processed with a "signal linear" color mapping configuration (within FLIR Tools software configurations) and a temperature scale of 27 °C to 37 °C.[43] The pre-processing procedures were applied to normalize the color mapping of the thermal images to obtain an objective evaluation of obesity and normal images. The details of the pre-processing procedures were explained in Leo et al.[43]

We also apply image augmentation on the training set to help the models generalize the features. We used the TensorFlow library of the ImageDataGenerator function

Table 1. CNN architecture comparison between DenseNet201 and the proposed MD.

		CNN architecture			
Layer	Output size	DenseNet201	MD-6-12-48-32	MD-6-12-12-6	MD-2-4-4-2
Input	224 × 224	Thermal images [224 × 224 × 3]			
Conv layer	112 × 112	Conv [7×7, 64, strides 2]			
Max pooling	56 × 56	Max pooling [2×2]			
Dense block (1)	56 × 56	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{Conv}[3 \times 3] \\ \text{Concat} \end{bmatrix} \times 6$	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{DwConv}[3 \times 3] \\ \text{PwConv}[1 \times 1] \\ \text{Concat} \end{bmatrix} \times 6$	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{DwConv}[3 \times 3] \\ \text{PwConv}[1 \times 1] \\ \text{Concat} \end{bmatrix} \times 6$	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{DwConv}[3 \times 3] \\ \text{PwConv}[1 \times 1] \\ \text{Concat} \end{bmatrix} \times 2$
Transition	56 × 56	Conv [1 × 1]			
block (1)	28 × 28	Average pooling [2 × 2, strides 2]			
Dense block (2)	28 × 28	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{Conv}[3 \times 3] \\ \text{Concat} \end{bmatrix} \times 12$	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{DwConv}[3 \times 3] \\ \text{PwConv}[1 \times 1] \\ \text{Concat} \end{bmatrix} \times 12$	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{DwConv}[3 \times 3] \\ \text{PwConv}[1 \times 1] \\ \text{Concat} \end{bmatrix} \times 12$	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{DwConv}[3 \times 3] \\ \text{PwConv}[1 \times 1] \\ \text{Concat} \end{bmatrix} \times 4$
Transition	28 × 28	Conv [1 × 1]			
block (2)	14 × 14	Average pooling [2 × 2, strides 2]			
Dense block (3)	14 × 14	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{Conv}[3 \times 3] \\ \text{Concat} \end{bmatrix} \times 48$	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{DwConv}[3 \times 3] \\ \text{PwConv}[1 \times 1] \\ \text{Concat} \end{bmatrix} \times 48$	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{DwConv}[3 \times 3] \\ \text{PwConv}[1 \times 1] \\ \text{Concat} \end{bmatrix} \times 12$	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{DwConv}[3 \times 3] \\ \text{PwConv}[1 \times 1] \\ \text{Concat} \end{bmatrix} \times 4$
Transition	14 × 14	Conv [1 × 1]			
block (3)	7 × 7	Average pooling [2 × 2, strides 2]			
Dense block (4)	7 × 7	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{Conv}[3 \times 3] \\ \text{Concat} \end{bmatrix} \times 32$	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{DwConv}[3 \times 3] \\ \text{PwConv}[1 \times 1] \\ \text{Concat} \end{bmatrix} \times 32$	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{DwConv}[3 \times 3] \\ \text{PwConv}[1 \times 1] \\ \text{Concat} \end{bmatrix} \times 6$	$\begin{bmatrix} \text{Conv}[1 \times 1] \\ \text{DwConv}[3 \times 3] \\ \text{PwConv}[1 \times 1] \\ \text{Concat} \end{bmatrix} \times 2$
Top layer	1 × 1	Global average pooling			
		Dropout [0.5]			
		Fully connected layer [256, ReLU]			
		Fully connected layer [2, Softmax]			

CNN: convolutional neural network; MD: modified DenseNet; ReLU: rectified linear unit. Note that each "Conv" operation shown in the table are the sequences of batch normalization–ReLU–convolution layer.

to randomly augment the images in each batch according to the specified augmentation parameters. Therefore, during each epoch, the model sees different augmented versions of the original images. The augmentation parameters we used in this study are rotation 5°, width, and height shift with a range of 0.1, and zoom shift with a range of 0.1.

Training, validation, and testing setup

All the networks were trained and validated on Google Cloud Engine Virtual Machine with NVIDIA V100 Tensor Core GPU and Tensorflow framework. We train the model with an obesity thermogram dataset for 150 epochs with a batch size of 64. The prediction loss was calculated using a sparse

Table 2. Dataset compositions.

Body region	Train		Validation		Test		Total
	Normal	Obese	Normal	Obese	Normal	Obese	
Supraclavicular (SCV)	114	86	14	12	14	12	252
Abdomen	114	86	14	12	14	12	252
Forearm	114	86	14	12	14	12	252
Palm	114	86	14	12	14	12	252
Shank	114	86	14	12	14	12	252
Total	**570**	**430**	**70**	**60**	**70**	**60**	**1260**

categorical cross-entropy loss function in the backward propagation process. We train the model with Adam optimizer with a scheduled learning rate as follows:

$$lr = \begin{cases} 0.0001 & \text{if} 1 \leq \text{epoch} < 50 \\ 0.00001 & \text{if} 50 \leq \text{epoch} < 100 \\ 0.000001 & \text{epoch} \geq 100 \end{cases} \tag{8}$$

where the learning rate is adjusted for every 50 epochs, starting from 0.0001, then stepping down to 0.00001 at epoch 50, and stepping down to 0.000001 at epoch 100. By applying the following scheduled learning rate settings, the model will become easier to converge into the global minimum.

We trained the model five times to determine the best hyperparameter of the proposed MD and evaluated the top-1 and top-5 validation loss to evaluate the classification performances. To obtain insight into the impact of modification of the convolution layers between the model's performance and complexity, we compared the proposed MD with the state-of-the-art DenseNet201, DenseNet169, and DenseNet201 models. The model size, learning parameters, and computation cost were evaluated to determine the weight of the CNN model. The model size was evaluated by measuring the memory consumption of the proposed model in byte units. Learning parameters are evaluated by calculating the learning parameters produced from convolution operation in the bottom layer and neural network operation in the top layers. The computation cost of the proposed model is evaluated by measuring the floating operations (FLOPS).

After choosing the most comparable proposed model, we test the proposed model with a testing dataset of images. Classification performances of the proposed model of accuracy, specificity, sensitivity (recall), precision, and f1-score were evaluated and measured by the following equations:

$$\text{Accuracy}(\%) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100 \tag{9}$$

$$\text{Specificity}(\%) = \frac{\text{TN}}{\text{FP} + \text{TN}} \times 100 \tag{10}$$

$$\text{Sensitivity}(\%) = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100 \tag{11}$$

$$\text{Precision}(\%) = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100 \tag{12}$$

$$\text{F1-score}(\%) = 2 \times \frac{\text{Sensitivity} \times \text{Precision}}{\text{Sensitivity} + \text{Precision}} \times 100 \tag{13}$$

where TP is the true positive rate, TN is the true negative rate, FP is the false positive rate, and FN is the false negative rate. The obese images are determined as the positive label, while normal images are determined as the negative label. Therefore, accuracy metrics represented the model's performance in accurately predicting both obese and normal images. Specificity represented the model's performance in correctly predicting the normal images. Sensitivity or recall represents the model's performance in correctly predicting the obese images. Precision represented the correctness of the model in predicting positive prediction. F1-score represents the balance between sensitivity and precision. The higher the accuracy, specificity, sensitivity, precision, and F1-score of a model, the better performance it has in classifying obesity thermal images.

Then, the proposed model will be compared with DenseNet,[7] MobileNetV1,[6] MobileNetV2,[44] and Snekhalata et al.[3] models in classification performances, model size, computation cost, and the confusion matrix. The next section will discuss how the proposed model performs and whether it is suitable to be called a lightweight CNN model for obesity detection.

Results and discussion

This section will explain and discuss the results of the experiments conducted. The discussion will start with the trade-off between layer depth and validation loss and then proceed to compare the testing results with other models.

Trade-off between layer depth and validation loss

In this section, the validation results were discussed to evaluate the trade-off between the layer's depth configuration and the validation loss. We trained our proposed model in different depths of dense blocks and compared its classification performances and complexity with the baseline DenseNet as shown in Table 3 and visualized it in Figure 6.

In this section, the validation results were discussed to evaluate the trade-off between the layer's depth configuration and the top-1 validation loss. We trained the model five times and listed the validation loss as top-1 and top-5 loss metrics. We compared the top-1 validation lost, learning parameters, model size, and computations cost in FLOPS of our proposed model in different depths of dense blocks with the DenseNet variants as shown in Table 3 and in Figure 6. We compared our proposed model of MD model of MD-6-12-48-32, MD-6-12-12-6, and MD-2-4-4-2 with the DenseNet variant model of DenseNet201, DenseNet169, and DenseNet121.[7]

The baseline DenseNet201 model was the state-of-the-art DenseNet201 model from[7] with modified fully connected to adjust the output class prediction. The DenseNet201 model had 18 million learning parameters, a model size of 71.77 MB, and a computation cost of 8.58 GFLOPS, considered a heavy-weight CNN model. The variants of the DenseNet model of DenseNet169 and DenseNet121 both had lower learning parameters compared with DenseNet201. DenseNet169 architectures were the reduction of the dense blocks of DenseNet201 from 6-12-48-32 into 6-12-48-32-32, while DenseNet121 is the reduction of dense blocks into 6-12-24-16.[7] The DenseNet169 has the lowest top-1 loss of 0.4005 compared with the DenseNet variants.

Therefore, we modified the baseline DenseNet201 to obtain lower computation costs but still inherited its classification performances. In the MD-6-12-48-32 model, we modified the standard 3 × 3 convolution layer into sequences of 3 × 3 depthwise and 1 × 1 pointwise convolution layer without modifying the depth of the dense blocks. The MD-6-12-48-32 achieved 3 million parameters and 2 GFLOPS lower than the DenseNet201, which indicated the depthwise separable convolution layers proven could reduce the computation cost of the proposed model. However, there is a trade-off between the reduced computation cost and the classification performances. The MD-6-12-48-32 model achieved higher validation loss than the DenseNet201 model.

We modified the MD-6-12-48-32 depth of repetition convolution sequences to reduce computation costs from 6-12-48-32 to 6-12-12-6. The MD-6-12-12-6 achieved a significant reduction in learning parameter number and computation cost. It achieved 13 million parameters and 4 GFLOPS lower than the MD-6-12-48-32. By reducing the depth of the dense blocks, the concatenated features map output also reduced significantly and made the model more generalized in learning the features. The MD-6-12-12-6 achieved a better classification performance than MD-6-12-48-32, representing top-1 and top-5 errors. However, a smaller model size and computation cost are required to embed a stand-alone CNN model into limited computation devices. Then, we reduced the depth of the model to 2-4-4-2 and achieved significantly lower complexity. MD-2-4-4-2 achieved 377 thousand learning parameters, model size of 1.44 MB, and a computation cost of 0.69 GFLOPS, the lowest complexity of the other models. However, the validation top-5 loss shows better results than the MD-6-12-12-6 but was still lower than the DenseNet201.

Figure 6 compared all the model's performances and shows the trade-off between learning parameters and

Table 3. Validation results: comparison between the top-1 and top-5 validation loss and model depth (learning parameters, model size, and computation cost) on the validation set.

Model	Parameters (model size)	Computation cost (GFLOPS)	Top-1 loss	Top-5 loss
DenseNet201	18,814,274 (71.77 MB)	8.58	0.4142	0.4537
DenseNet169	13,069,634 (49.86 MB)	6.72	0.4005	0.4084
DenseNet121	7,300,418 (27.85 MB)	5.67	0.4228	0.4381
MD-6-12-48-32	15,781,762 (60.20 MB)	6.15	0.6165	0.7612
MD-6-12-12-6	2,346,050 (8.95 MB)	2.81	0.4546	0.6879
MD-2-4-4-2 (Proposed model)	**377,346 (1.44 MB)**	**0.69**	0.4742	0.5549

GFLOPS: billion floating-point operations per second; MB: megabyte; MD: modified DenseNet.

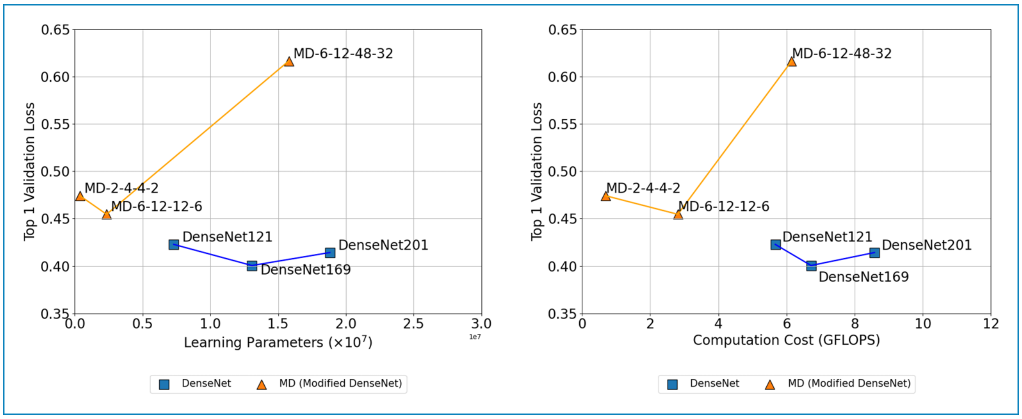


Figure 6. Trade-off between top-1 validation loss respective with networks depth represented by learning parameters numbers and computation cost (in billion floating-point operations per second (GFLOPS)) metrics. (a) Trade-off chart between learning parameters and loss; (b) trade-off chart between computation cost and loss.

computation cost with the top-1 loss. There are two groups of comparisons, which are the DenseNet variant and the MD variants, with both reduced weight. The trade-off chart shows how the impacts of reducing the model depths with respect to its loss validations. In both groups, the higher learning parameters and computation cost, the trends of validation loss have similar patterns. When DenseNet201 layers were reduced to DenseNet169, the validation loss was lower, but when it was reduced to DenseNet121, the loss got higher back, surpassing the DenseNet201. Among the DenseNet group, DenseNet169 has the lowest validation loss even though the parameter and computation costs are not the lowest. Among the MD group, when the MD-6-12-48-32 depth layers were reduced into 6-12-12-6, the validation loss became lower, but when it was reduced more to MD-2-4-4-2, the validation loss got worse.

Therefore, we choose the MD-2-4-4-2 model as the proposed model in this study. Compared with the other models with a heavier depth of dense blocks, the proposed MD-2-4-4-2 achieved better classification performances and lower complexity among the MD groups. Although the MD-2-4-4-2 is not the best model in terms of top-1 loss validation results, it has the lowest complexity of learning parameters, model size, and computation cost among all the models. It was expected that the validation loss wouldn't be lower than the DenseNet variants in situations where we significantly reduced the MD-2-4-4-2 dense blocks. This trade-off resulted in a reduced ability to classify exchanges, but with the benefit of its lightweight ability.

Testing results

After choosing the proposed model based on its validation loss, we tested it with the testing set. We compared its classification performances and complexity with other state-of-the-art CNN models. We compared the proposed model MD-2-4-4-2 with DenseNet201,[7] MobileNetV1,[6] MobileNetV2,[44] and previous works models of Snekhalata et al.[3] We evaluated the learning parameters, the model size, the computation cost in GFLOPS, and classification performances: accuracy, specificity, sensitivity, precision, and F1-score metrics as shown in Tables 4 and 5.

The DenseNet201[7] model achieved the best accuracy of 83.08%, specificity of 82.86%, sensitivity of 83.33%, precision of 80.65 %, and the best F1-score of 81.97%. The densely connected proved that the temperature features in the obesity thermograms could be learned effectively by feeding each input feature map through concatenated layers. Table 6 shows that the DenseNet201 model correctly predicts 58 of 68 normal thermograms and 50 of 62 obese thermograms. The DenseNet201 achieved a balanced ability to distinguish the difference between normal and obese thermograms. However, the DenseNet201 is considered a heavy-weight CNN model with 18 million learning parameters and a model size of 71.77 MB. The standard convolution operation in the dense blocks has a computation cost of 8.58 GFLOPS.

The MobileNetV1[6] model was known as a lightweight CNN model, which became the main idea in proposing the separable depthwise convolution operation in the proposed models. The MobileNetV1 architecture consists of multiple depthwise and pointwise convolution layers, making the model size significantly smaller than conventional CNN models. It has 3 million learning parameters, a model size of 13.32 MB, and a computation cost of 1.14 GFLOPS, significantly smaller than the DenseNet201 model. However, there was a trade-off in the classification performances with the computation cost where the MobileNet achieved lower accuracy, specificity, and sensitivity than the DenseNet201 model, as shown in Table 7. The MobileNetV2[44] improves the MobileNetV1 model, which utilized inverted residuals and linear bottlenecks network. It achieved a better

Table 4. Testing results: complexity comparison of learning parameters and computation cost.

Model	Parameters (model size)	Computation cost (GFLOPS)
DenseNet201	18,814,274 (71.77 MB)	8.58
MobileNetV1	3,491,778 (13.32 MB)	1.14
MobileNetV2	2,586,434 (9.87 MB)	0.59
Snekhalata et al.[3]	51,407,178 (196.10 MB)	2.56
MD-2-4-4-2 (proposed model)	**377,346 (1.44 MB)**	**0.69**

GFLOPS: billion floating-point operations per second; MB: megabyte; MD: modified DenseNet.

Table 5. Testing results: classification performances in accuracy, specificity, sensitivity, precision, and F1-score.

Model	Accuracy (%)	Specificity (%)	Sensitivity (%)	Precision (%)	F1-score (%)
DenseNet201	83.08	82.86	83.33	80.65	81.97
MobileNetV1	80.00	81.43	78.33	78.33	78.33
MobileNetV2	79.23	82.86	75.00	78.95	76.92
Snekhalata et al.[3]	74.62	84.29	63.33	77.55	69.72
MD-2-4-4-2 (proposed model)	**81.54**	**84.29**	**78.33**	**81.03**	**79.66**

MD: modified DenseNet.

complexity but poorly classified the testing dataset, as shown in Table 8.

The customized CNN model from the previous works of Snekhalata et al.[3] was also trained and tested to classify obese thermograms. The CNN architectures consist of multiple sequences of standard convolution layers based on VGG19 architectures.[40] The standard structures of convolution cause the model to have very large training parameters, model sizes, and computation costs compared with other CNN models. The CNN model is designed for computer-aided diagnosis systems, which are not designed as lightweight CNN models. As shown in Table 9, the Snekhalata et al.[3] model achieved better performance predicting normal images but failed to predict obese images correctly.

Our proposed model of MD-2-4-4-2 has achieved 377,000 learning parameters and a model size of 1.44 MB, which are the smallest compared with other models. Modifying the standard convolution layers into depthwise and pointwise layers and modifying the depth of dense blocks causes a significant reduction in the proposed model complexity. Therefore, the computation cost of the proposed model was the second lowest, 0.69 GFLOPs, and was not smaller than MobileNetV2 due to the standard convolution layers in the transition blocks. However, despite its small size, the proposed model achieved a comparable classification performance on the test set. The proposed model achieved an accuracy of 81.54%, which was the second highest after the DenseNet201, and achieved the highest specificity of 84.29% and precision of 81.03%. As shown in Table 10, the proposed model achieved better performances in predicting normal images correctly. When the model predicts images as normal images, there is a high probability that the actual images are normal.

High classification performance, small computation cost, and small model size were the expected requirements for building a lightweight CNN model. However, the proposed model MD-2-4-4-2 has been proven to achieve a CNN model that can be concluded as a comparable lightweight model. It has inherited the feature extraction ability from the DenseNet architecture and the lightweight characteristic from the MobileNet architecture. Modifying the standard convolution layers into separable depthwise convolution layers could reduce the model complexity and maintain its classification performances even though the depth networks in the dense blocks have been reduced. All these results show that the proposed model

Table 6. Confusion matrix of DenseNet201.

	Prediction negative	Prediction positive
Actual negative	**True negative = 58**	**False positive = 12**
	SCV = 13	SCV = 1
	Abdomen = 13	Abdomen = 1
	Forearm = 12	Forearm = 2
	Palm = 10	Palm = 4
	Shank = 10	Shank = 4
Actual positive	**False negative = 10**	**True positive = 50**
	SCV = 2	SCV = 10
	Abdomen = 1	Abdomen = 11
	Forearm = 1	Forearm = 11
	Palm = 2	Palm = 10
	Shank = 4	Shank = 8

SCV: supraclavicular.

Table 7. Confusion matrix of MobileNetV1.

	Prediction negative	Prediction positive
Actual negative	**True negative = 57**	**False positive = 13**
	SCV = 12	SCV = 2
	Abdomen = 12	Abdomen = 2
	Forearm = 14	Forearm = 1
	Palm = 9	Palm = 5
	Shank = 9	Shank = 3
Actual positive	**False negative = 13**	**True positive = 47**
	SCV = 4	SCV = 8
	Abdomen = 2	Abdomen = 10
	Forearm = 1	Forearm = 11
	Palm = 2	Palm = 10
	Shank = 4	Shank = 8

SCV: supraclavicular.

Table 8. Confusion matrix of MobileNetV2.

	Prediction negative	Prediction positive
Actual negative	**True negative = 58**	**False positive = 12**
	SCV = 11	SCV = 3
	Abdomen = 13	Abdomen = 1
	Forearm = 12	Forearm = 2
	Palm = 11	Palm = 3
	Shank = 11	Shank = 3
Actual positive	**False negative = 15**	**True positive = 45**
	SCV = 3	SCV = 9
	Abdomen = 2	Abdomen = 10
	Forearm = 2	Forearm = 10
	Palm = 3	Palm = 9
	Shank = 5	Shank = 7

SCV: supraclavicular.

was considered a lightweight CNN model suitable for application on limited computation devices such as mobile applications.

The proposed model in this study has the potential to be embedded in real-world applications such as health mobile apps to monitor obesity. However, our currently proposed model was limited only to classifying the obesity thermal images from Indonesian subjects due to the limited obesity datasets of other continents or ethnicities. Thermal images were very dependent on the temperatures and environments of the subjects when the images were captured. Different areas would have different temperature environments and produce different body

Table 9. Confusion matrix of Snekhalata et al.[3]

	Prediction negative	Prediction positive
Actual negative	**True negative = 59**	**False positive = 11**
	SCV = 13	SCV = 1
	Abdomen = 12	Abdomen = 2
	Forearm = 12	Forearm = 2
	Palm = 11	Palm = 3
	Shank = 11	Shank = 3
Actual positive	**False negative = 22**	**True positive = 38**
	SCV = 4	SCV = 8
	Abdomen = 3	Abdomen = 9
	Forearm = 3	Forearm = 9
	Palm = 6	Palm = 6
	Shank = 6	Shank = 6

SCV: supraclavicular.

Table 10. Confusion matrix of MD-2-4-4-2.

	Prediction negative	Prediction positive
Actual negative	**True negative = 59**	**False positive = 11**
	SCV = 11	SCV = 3
	Abdomen = 12	Abdomen = 2
	Forearm = 12	Forearm = 2
	Palm = 12	Palm = 2
	Shank = 12	Shank = 2
Actual positive	**False negative = 13**	**True positive = 47**
	SCV = 2	SCV = 10
	Abdomen = 3	Abdomen = 9
	Forearm = 0	Forearm = 12
	Palm = 4	Palm = 8
	Shank = 4	Shank = 8

MD: modified DenseNet; SCV: supraclavicular.

temperature distribution; for example, in the four seasons, obese European subjects would have different temperature environments with two seasons continents, such as the Asian and Middle Eastern continents. To obtain a balanced view to discuss the potential real-world applications of the proposed models, the diversity and variation datasets from different areas and continents would solve this limitation in the future.

Conclusion

In this study, we have constructed a new lightweight CNN model for obesity early detection based on DenseNet201 CNN architectures by modifying the standard convolution layers into separable depthwise convolution layers from MobileNet CNN architectures. The proposed MD-2-4-4-2 model achieved a classification performance comparable to the DenseNet201 model and achieved complexity comparable to the MobileNet model. The proposed model has been successfully designed to inherit the feature-extracting ability from the DenseNet201 architecture with an accuracy rate of 81.54% and the lightweight complexity characteristic of the MobileNet architecture with 377,000 learning parameters, model size of 1.44 MB and computation cost of 0.69 GFLOPS. The proposed lightweight CNN model for obesity early detection systems showed comparable performances and is suitable for embedding on mobile devices.

For future works, integrating obesity thermograms from diverse ethnicities across various continents would enhance the model's ability to detect obesity more effectively on a global scale. Additionally, we intend to analyze the most representative body regions for obesity detection, aiming to optimize the performance of the proposed CNN models by eliminating non-representative body region images.

Acknowledgements: We would like to express our gratitude and appreciation to all those participants who gave us the possibility to complete this study. We also would like to thank Tsunami and Disaster Mitigation Research Center Universitas Syiah Kuala (TDRMC USK) for providing the equipment for the thermal camera.

Contributorship: HL contributed to the methodology, software operation, investigation, validation, and writing the original draft. KS was involved in software operation and investigation, and Rosllidar was involved in data curation and formal analysis. RMr contributed to methodology and formal analysis. KM conceptualized the study and provided the resources and supervision. FA contributed to methodology, conceptualization, supervision, and funding acquisition. All authors reviewed and

edited the manuscript and approved the final version of the manuscript.

Declaration of conflicting interests: The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Ethical approval: The study was approved by Fakultas Kedokteran Universitas Syiah Kuala Ethical Clearance Committee (Number: 167/EA/FK/2023).

Funding: The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) Universitas Syiah Kuala, Indonesia, under the scheme of Program Riset Unggulan USK Percepatan Doktor (PRUUPD) 2023 funding with contract number 532/UN11.2.1/PT.01.03/PNBP/2023.

Guarantor: FA

ORCID iD: Fitri Arnia https://orcid.org/0000-0001-6020-1275

References

1. Chooi YC, Ding C and Magkos F. The epidemiology of obesity. *Metabolism* 2019; 92: 6–10.
2. Zhou Y, Chen S, Wang Y, et al. Review of research on lightweight convolutional neural networks. In: *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Chongqing, China, 12 June–14 June 2020, pp.1713–1720. Piscataway, NJ, USA: IEEE.
3. Snekhalatha U, Palani Thanaraj K and Sangamithirai K. Computer aided diagnosis of obesity based on thermal imaging using various convolutional neural networks. *Biomed Signal Process Control* 2021; 63: 102233.
4. Rashmi R, Umapathy S and Krishnan PT. Thermal imaging method to evaluate childhood obesity based on machine learning techniques. *Int J Imag Syst Technol* 2021; 31: 1752–1768.
5. Rashmi R, Snekhalatha U, Krishnan PT, et al. Fat-based studies for computer-assisted screening of child obesity using thermal imaging based on deep learning techniques: a comparison with quantum machine learning approach. *Soft Comput* 2023; 27: 13093–13114.
6. Howard AG, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:170404861*, 2017.
7. Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, USA, 21 July–26 July, 2017; pp.4700–4708. Piscataway, NJ, USA: IEEE.
8. Salehi AW, Khan S, Gupta G, et al. A study of CNN and transfer learning in medical imaging: advantages, challenges, future scope. *Sustainability* 2023; 15: 5930.
9. Thakur S and Kumar A. X-ray and CT-scan-based automated detection and classification of COVID-19 using convolutional neural networks (CNN). *Biomed Signal Process Control* 2021; 69: 102920.
10. Al-Yasriy HF, Al-Husieny MS, Mohsen FY, et al. Diagnosis of lung cancer based on CT scans using CNN. *IOP Conf Ser: Mater Sci Eng* 2020; 928: 022035.
11. Alakwaa W, Nassef M and Badr A. Lung cancer detection and classification with 3D convolutional neural network (3D-CNN). *Int J Adv Computer Sci App* 2017; 8: 409–417.
12. Farooq A, Anwar S, Awais M, et al. A deep CNN based multi-class classification of Alzheimer's disease using MRI. In: *2017 IEEE International Conference on Imaging Systems and Techniques (IST)*, Beijing, China, 18 October–20 October 2017; pp.1–6. Piscataway, NJ, USA: IEEE.
13. Khagi B and Kwon GR. 3D CNN design for the classification of Alzheimer's disease using brain MRI and PET. *IEEE Access* 2020; 8: 217830–217847.
14. Alzubaidi L, Zhang J, Humaidi AJ, et al. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J Big Data* 2021; 8: 1–74.
15. Devi BS and Misbha D. Hybrid convolutional neural network based segmentation of visceral and subcutaneous adipose tissue from abdominal magnetic resonance images. *J Ambient Intell Human Comput* 2023; 14: 13333–13347.
16. Wang Z, Meng Y, Weng F, et al. An effective CNN method for fully automated segmenting subcutaneous and visceral adipose tissue on CT scans. *Ann Biomed Eng* 2020; 48: 312–328.
17. Micomyiza C, Zou B and Li Y. An effective automatic segmentation of abdominal adipose tissue using a convolution neural network. *Diabetes Metabolic Syndrome: Clin Res Rev* 2022; 16: 102589.
18. Shen N, Li X, Zheng S, et al. Automated and accurate quantification of subcutaneous and visceral adipose tissue from magnetic resonance imaging based on machine learning. *Magn Reson Imag* 2019; 64: 28–36.
19. Wang Y, Qiu Y, Thai T, et al. Applying a deep learning based CAD scheme to segment and quantify visceral and subcutaneous fat areas from CT images. *Proc SPIE* 2017; 10134: 101343G.
20. Al Khatib F, Gouissem A, Mbarki R, et al. Biomechanical characteristics of the knee joint during gait in obese versus normal subjects. *Int J Environ Res Public Health* 2022; 19: 989.
21. Law J, Morris DE, Budge H, et al. *Infrared thermography: Brown adipose tissue*. Cham, Switzerland: Springer, 2019; pp.259–282.
22. El Hadi H, Frascati A, Granzotto M, et al. Infrared thermography for indirect assessment of activation of brown adipose tissue in lean and obese male subjects. *Physiol Meas* 2016; 37: N118.
23. Jimenez-Pavon D, Corral-Perez J, Sánchez-Infantes D, et al. Infrared thermography for estimating supraclavicular skin temperature and bat activity in humans: A systematic review. *Obesity* 2019; 27: 1932–1949.
24. Brasil S, Renck AC, de Meneck F, et al. A systematic review on the role of infrared thermography in the brown adipose tissue assessment. *Rev Endocr Metab Disord* 2020; 21: 37–44.
25. Jang C, Jalapu S, Thuzar M, et al. Infrared thermography in the detection of brown adipose tissue in humans. *Physiol Rep* 2014; 2: e12167.

26. Law J, Morris DE, Izzi-Engbeaya C, et al. Thermal imaging is a noninvasive alternative to PET/CT for measurement of brown adipose tissue activity in humans. *J Nucl Med* 2018; 59: 516–522.
27. Rashmi R and Snekhalatha U. Thermal imaging method in the evaluation of obesity in various body regions—a preliminary study. *IOP Conf Ser: Mater Sci Eng* 2020; 912: 062022.
28. Sangamithirai S, Snekhalatha U, Sanjeena R, et al. Thermal imaging of abdomen in evaluation of obesity: A comparison with body composition analyzer—-a preliminary study. In: *Proceedings of the international conference on ISMAC in computational vision and bio-engineering 2018 (ISMAC-CVB)*, Palladam, India, 16 May–17 May 2018, 2019; pp.79–87. Cham, Switzerland: Springer.
29. Bouguettaya A, Kechida A and TABERKIT AM. A survey on lightweight CNN-based object detection algorithms for platforms with limited computational resources. *Int J Inf Appl Math* 2019; 2: 28–44.
30. Gayathri S, Gopi VP and Palanisamy P. A lightweight CNN for diabetic retinopathy classification from fundus images. *Biomed Signal Process Control* 2020; 62: 102115.
31. Haque WA, Arefin S, Shihavuddin A, et al. Deepthin: A novel lightweight CNN architecture for traffic sign recognition without GPU requirements. *Expert Syst Appl* 2021; 168: 114481.
32. Hui TW, Tang X and Loy CC. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City, Utah, USA, 18 June –23 June 2018; pp.8981– 8989. Piscataway, NJ, USA: IEEE.
33. Ullah A, Muhammad K, Ding W, et al. Efficient activity recognition using lightweight CNN and DS-GRU network for surveillance applications. *Appl Soft Comput* 2021; 103: 107102.
34. Roslidar R, Syaryadhi M, Saddami K, et al. Breacnet: A high-accuracy breast thermogram classifier based on mobile convolutional neural network. *Math Biosci Eng* 2022; 19: 1304–1331.
35. Tesfai H, Saleh H, Al-Qutayri M, et al. Lightweight ShuffleNet based CNN for arrhythmia classification. *IEEE Access* 2022; 10: 111842–111854.
36. Wang J, Ji P, Xiao W, et al. Lightweight convolutional neural networks for vehicle target recognition. In: 2020 IEEE 5th International Conference on Intelligent Transportation Engineering (ICITE), Beijiing, China, 12 June–14 June 2020; pp. 245–248. Piscataway, NJ, USA: IEEE.
37. Anisimov D and Khanova T. Towards lightweight convolutional neural networks for object detection. In: *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)*, Lecce, Italy, 29 August–September 1, 2017, pp.1–8. Piscataway, NJ, USA: IEEE.
38. Leo H, Saddami K, Roslidar R, et al. A mobile application for obesity early diagnosis using CNN-based thermogram classification. In: *2023 international conference on artificial intelligence in information and communication (ICAIIC)*, Bali, Indonesia, 8 February–10 February 2023, pp. 514–520. Piscataway, NJ, USA: IEEE.
39. Leo H, Arnia F and Munadi K. Fine tuning CNN pre-trained model based on thermal imaging for obesity early detection. *J Rekayasa Elektrika* 2022; 18: 53–60.
40. Simonyan K and Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:14091556*. 2014.
41. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, Nevada, USA. 27 June–30 June 2016, pp. 770–778. Piscataway, NJ, USA: IEEE.
42. Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. *Int J Comput Vis* 2015; 115: 211–252.
43. Leo H, Saddami K, Roslidar R, et al. An improved thermogram dataset for obesity detection research. In: *2023 international workshop on artificial intelligence and image processing (IWAIIP)*, Bali, Indonesia, 8 February&ndash;10 February 2023, pp.353&ndash;358. Piscataway, NJ, USA: IEEE.
44. Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City, Utah, USA, 18 June–22 June 2018, pp.4510–4520. Piscataway, NJ, USA: IEEE.