Jurnal Rekayasa Elektrika Vol. 18, No. 1, Maret 2022, hal. 53-60
ISSN. 1412-4785; e-ISSN. 2252-620X, Terakreditasi RISTEKDIKTI No. 30/E/KPT/2018
DOI: 10. 17529/jre.v18i1.25100

Fine Tuning CNN Pre-trained Model Based on Thermal Imaging for Obesity Early Detection

Hendrik Leo, Fitri Arnia, dan Khairul Munadi
Dept. of Electrical and Computer Engineering, Faculty of Engineering, Universitas Syiah Kuala
Banda Aceh, Indonesia, 23111
e-mail: hendrik@mhs.unsyiah.ac.id

***Abstract*—Obesity is a complex disease that causes serious impact health, such as diabetes mellitus, cardiovascular disease, cancer, and stroke. An early obesity diagnosis/ detection method is required to prevent the increasing number of obese people. This study aims to: (i) fine-tune the pre-trained Convolutional Neural Network (CNN) models to build an early detection of obesity and (ii) evaluate the model performance in terms of classifying performance, computation speed, and learning performance. The thermal images acquisition procedure was conducted with 18 normal subjects and 15 obese subjects to build a thermal images dataset of obesity. Pre-trained CNN models: VGG19, MobileNet, ResNet152V2, and DenseNet201 were modified and trained using the acquired dataset as the input. The training results show that the DenseNet201 model outperformed other models regarding classifying accuracy: 83.33 % and learning performances. At the same time, the MobileNet model outperformed other models in terms of computation speed with training elapsed time: 12 seconds/epoch. The proposed DenseNet201 model was suitable for implementation as an early screening system of obesity for health workers or physicians. Meanwhile, the proposed MobileNet model was suitable for mobile applications' early detection/diagnosis of obesity.**

Keywords: ***obesity, deep learning, transfer learning, pre-trained CNN, thermal imaging***

***Abstrak*—Obesitas merupakan penyakit yang dapat mengakibatkan dampak serius bagi kesehatan, seperti penyakit diabetes mellitus, *cardiovascular disease*, *cancer* dan *stroke*. Sehingga dibutuhkan suatu metode deteksi dini penyakit obesitas untuk mencegah bertambahnya jumlah pengidap penyakit obesitas. Studi ini bertujuan untuk (i) melakukan fine-tune dari model *pre-trained Convolutional Neural Network* (CNN) yang melakukan deteksi dini penyakit obesitas (ii) mengevaluasi performa model dalam melakukan klasifikasi, waktu yang dibutuhkan dalam komputasi dan performa model dalam melakukan learning. Telah dilakukan prosedur akuisisi citra terhadap 18 peserta normal dan 15 peserta obesitas untuk membangun dataset citra termal obesitas. Dataset yang dibangun dijadikan input dalam proses training dari model *pre-trained* CNN yang telah dimodifikasi yaitu *VGG19*, *MobileNet*, *ResNet152V2* dan *DenseNet201*. Hasil training menunjukkan model DenseNet201 merupakan model dengan akurasi tertinggi: 83.33 % dan performa learning terbaik dibandingkan dengan model lainnya. Sedangkan model *MobileNet* merupakan model dengan kecepatan komputasi tercepat yaitu 12 detik/*epoch* dibandingkan dengan model lainnya. Model *DenseNet201* yang dibangun dapat diimplementasikan sebagai sistem deteksi dini penyakit obesitas untuk petugas kesehatan. Sedangkan model *MobileNet* yang dibangun lebih sesuai untuk diimplementasikan pada aplikasi mobile dalam melakukan deteksi dini penyakit obesitas.**

Kata kunci: ***obesitas, deep learning, transfer learning, pre-trained CNN, citra termal***

I. Introduction

Obesity is a complex disease that causes serious impact health, such as diabetes mellitus, cardiovascular disease, cancer, and stroke [1], [2]. According to World Health Organization (WHO) statistics report 2021, obesity among adults has risen over the decades, with 650 million obese adults in 2016 [3]. The prevalence of obesity in 2000 was 8.7 % and increased to 13,1 % in 2016 [3]. An early obesity diagnosis/ detection method is required to prevent the increasing number of obese people.

Therefore, several researchers have developed methods of diagnosing obesity based on machine learning algorithms such as Random Forest [4], [5], Logistic Regression [6], [7], Classification and Regression Trees (CART) [6], Random Tree [5], XGBoost [8], Naïve Bayess [5], Artificial Neural Network (ANN) [7], etc. Dataset input features of machine learning algorithms include statistical features such as age, weight, height, gender, location, clinical history, food and drinks consumption, race, ethnic [4]–[8] and give prediction/ classification output of Body Mass Index (BMI) value, obesity level, and obesity risk factor [4]–[8]. Machine Learning Algorithm represents a powerful algorithm that could learn the dataset features and predict the output of obesity conditions accurately. However, an obese diagnosis based on a Machine Learning algorithm could only predict the user's condition based on statistical features and did not represent the actual quantity of the fat.

Thermogram or thermal images can represent the fat

Received 05 March 2022; Revised 21 March 2022; Accepted 22 March 2022

in the human body or known as Brown Adipose Tissue (BAT) [9]. BAT fat has a role in heat production in the human body, and those heat are translated to thermogram using a thermal camera [9], [10]. This technique is also known as Infrared Thermography (IRT) or Thermal Imaging technique. The study of Rashmi et al. [11] stated significant differences in temperature between normal and obese subjects. Figure 1 shows the differences in temperature between normal and obese subjects in several body regions. Those temperature distribution patterns in the human body became the main feature and potential of the Thermal Imaging technique in building an early diagnosis of obesity [12].

Several studies have developed an obese diagnosis based on machine learning and thermal imaging. Those studies extracted the feature of thermal images with Scale-invariant feature transformation (SIFT) and speeded up robust features (SURF) algorithm [13], [14]. They used Support Vector Machine (SVM), Naïve Bayes, and Random Forest Algorithm to classify the extracted feature. However, the Machine Learning algorithm required additional treatment in manually extracting the thermal images' feature. Meanwhile, by using Deep Learning Convolutional Neural Network (CNN) algorithm, we don't need to design feature extractors manually. Instead, the CNN algorithm learns the feature and builds a trainable classifier [15].

In this study, we built a dataset of obesity through the images acquisition process and trained the obese diagnosis model with a pre-trained CNN model. This study aims to: (i) fine-tune the pre-trained CNN models to build an early detection of obesity, and (ii) evaluate the model performance in classifying performance in accuracy, computation speed in training elapsed time, and learning performance in each epoch.

II. Method

A. Subjects

Thirty-three subjects agreed and volunteered to participate in the study. Eighteen of them were normal subjects (BMI $\leq$ 25), and fifteen of them were obese subjects (BMI > 25) were enrolled in the study. The mean age of subjects was 22 ± 2 years with a range of 19 – 27 years old. Subjects with growth hormone deficiency/ dwarfism, polycystic ovarian syndrome, glandular hormone disorder: hypothyroidism, pituitary, and metabolic diseases hyperglycemia were excluded from the study [12]. The subjects were not allowed to take medications, drink caffeine, smoke, and exercise 6 hours before the image acquisition. The image acquisition was conducted in the morning (9 a.m – 12 a.m), and the subjects should do overnight fasting before the image acquisition process.

Table 1. Collected thermal images dataset

Dataset	Training	Validation/ Testing	Total
Obese	88	32	120
Normal	104	40	144

B. Thermal Images Acquisition Procedure

Thermal images acquisition was conducted in a temperature-controlled room with 25 ^{o}C - 27 ^{o}C and humidity of 50 %. The participants were instructed to sit in the room with comfortable clothes for 30 minutes to acclimatize their body temperature with the room's temperature. Before the thermal images acquisition process, all metallic ornaments and electronic devices were suggested to be removed from the room. Participants' thermal images have been acquired with a thermal camera (FLIR E95) with the emissivity of 0,98 (human skin), camera distances of 1 meter from participants, temperature scaling of 27 ^{o}C to 37 ^{o}C, and images mode of rainbow mode. We acquired four body regions of the normal and obese participants: supraclavicular (SCV), abdomen, forearm, and shank region, as shown in Fig. 1.

C. Dataset

We have acquired an obesity dataset from the thermal images acquisition consisting of 120 obese thermal images and 144 normal thermal images, as shown in Table 1. From 120 obese thermal images, 88 images were set for the training dataset, and the remaining 32 images were set for the validation dataset. From 144 normal thermal images, 104 images were set for the training dataset, and the remaining 40 images were set for the validation dataset.

D. Pre-trained CNN Model Architecture

In designing CNN model architecture, there are 2 phases in processing the input dataset: feature learning and image classifying [16], [17]. The feature learning phase consists of several convolutional layers and a pooling layer followed by ReLU and max-pooling activation function. While the image classifying phase consists of a flatten layer and fully connected layer/ dense layer followed by a softmax activation function. After the model produces an output of classifying images, the backpropagation procedure is executed [17]. The loss of the output prediction was calculated through a loss function, and the weight of each filter or unit was updated through an optimizer.

To build an early diagnosis CNN model of obesity, we imported and modified a pre-trained CNN model, known as the Transfer Learning method, as shown in Fig. 2 [18]. The pre-trained CNN model imported the convolution layer architecture and kernel's weight. While the fully connected layer was modified and adjusted to our desired output class (k=2, obese and normal). When the backpropagation procedure is executed, the kernel's weight of the convolutional layer was not updated, and the model updated only the unit's weight of the dense layer and fully connected. In this study, Pre-trained CNN Model: VGG19 [19], MobileNet [20], ResNet152V2 [21], and DenseNet201 [22] were imported and trained using acquired thermal images dataset of obesity as the input.

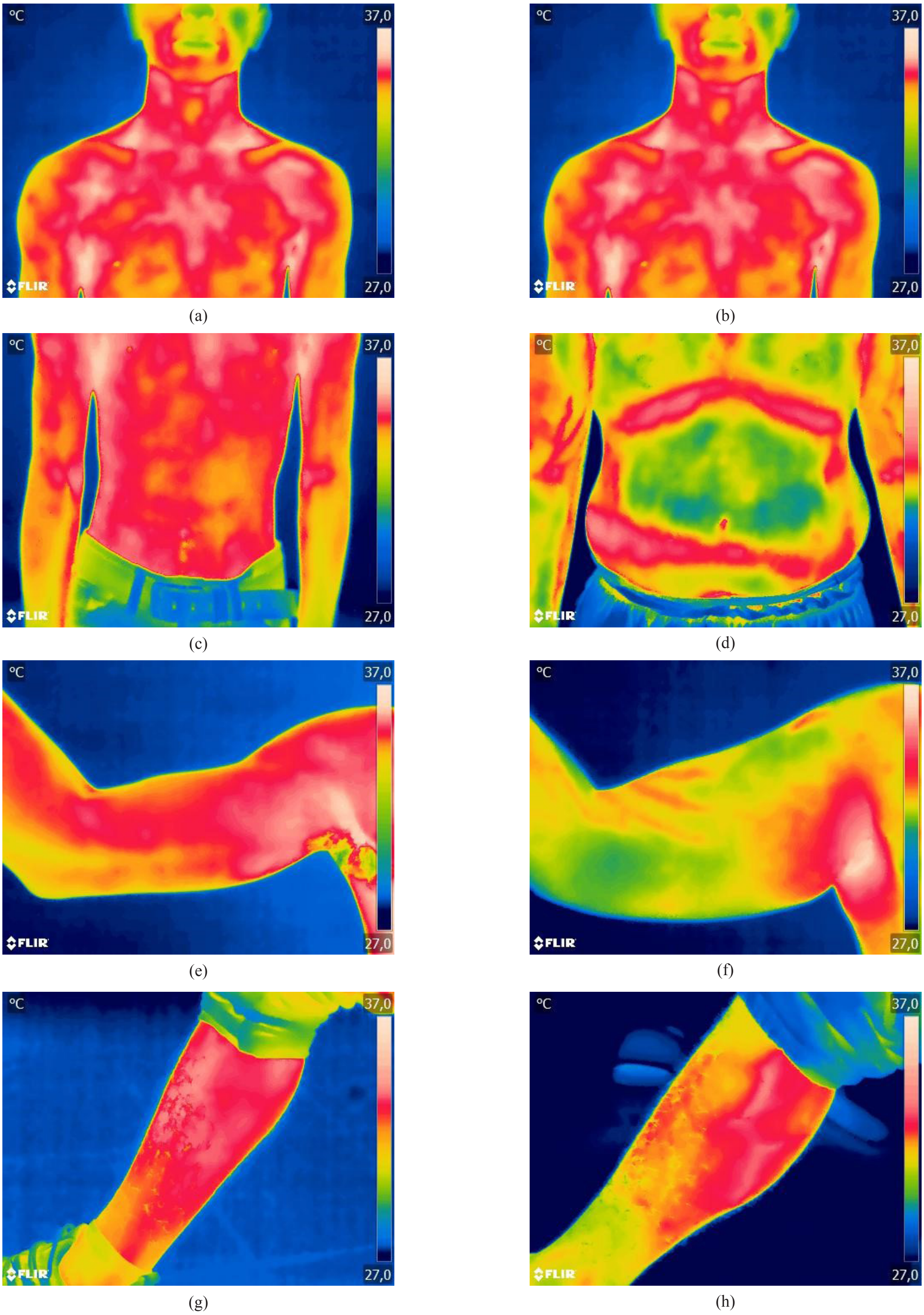

Fig. 1. Thermal Images of Normal and Obese Subjects (a) Normal Supraclavicular Region (b) Obese Supraclavicular Region (c) Normal Abdomen Region (d) Obese Abdomen Region (e) Normal Forearm Region (f) Obese Forearm Region (g) Normal Shank Region (h) Obese Shank Region

1. VGG19

VGGNet architecture is designed with the same principles of AlexNet [19], [23]. VGGNet increases the convolution layers from 8 layers (AlexNet) to 16 layers (VGG16) and 19 layers (VGG19). The convolution layers were added with 3 × 3 filters and followed by a max-pooling layer with 2 × 2 filters. The increased depth of VGGNet layers is designed to train the model to generalize better in a wide range of tasks and datasets [19]. However, the increased depth of convolution layers requires a lot of computation and increased the time required for the training process.

2. MobileNet

Mobile Convolutional Network (MobileNet) is designed as an efficient deep neural network to match with design requirements for mobile and embedded devices [20]. Mobile devices require very small and low latency models to do classification tasks due to their limited computation ability [20]. MobileNet architectures are based on depthwise separable filters, which factorize standard convolution into depthwise and pointwise convolutions [20]. Depthwise convolution is consists of a single filter for each input channel, while pointwise convolution consists of multiple simple 1 × 1 filters. Then the depthwise (DW) and pointwise (PW) convolution are followed with Batch Normalization (BN) and ReLU activation function for both layers. For example, standard convolutional layers consist of 3×3 convolution filter – BN – ReLU, while depthwise separable convolution consists of 3×3 DW filter – BN – ReLU – 1×1 PW filter –BN – ReLU. By implementing depthwise separable filters, the computation cost was reduced and optimizing the classification speed. However, MobileNet architecture was trading off the accuracy performance to reduce size and latency.

3. ResNet152V2

Residual Convolutional Network (ResNet) is designed based on VGGNet architecture [21]. Deep residual learning is introduced to solve the degradation problem of saturated accuracy due to network depth increasing. A shortcut connection is added to perform identity mapping by adding the previous layer weight output to the next stacked layers output. The shortcut connection was parameter-free and led to more efficient models for bottlenecks designs. ResNet with 152 layers (ResNet152) was proved to have lower complexity than VGG16/VGG19 according to their Floating Point Operations (FLOPs). ResNet152 has 11.3 billion FLOPs, while VGG16/VGG19 has 15.3/19.6 billion FLOPs [21].

4. DenseNet201

Dense Convolution Network (DenseNet) is designed to ensure maximum information flow between layers by connecting all the layers directly to each other [22]. In contrast to ResNet [21], the features of connected layers were concatenated instead of combined through summation. With those connected layers, the flow of information and gradients throughout the network is improved and makes the model easier to train. DenseNet has fewer parameters and less computation than the state of art ResNet.

E. Training and Validation

The training and validation process was run on Google Compute Engine backend Virtual Machine (GCE VM) with RAM of 16 GB and accessed with Google Colab editor. The CNN model was designed and developed on an open-source deep learning framework: TensorFlow 2.0 in Python.

III. Result and Discussion

This section will discuss: (i) the results of the fine-tuned hyperparameter we used in building an early detection model of obesity and (ii) evaluate the model's performances results.

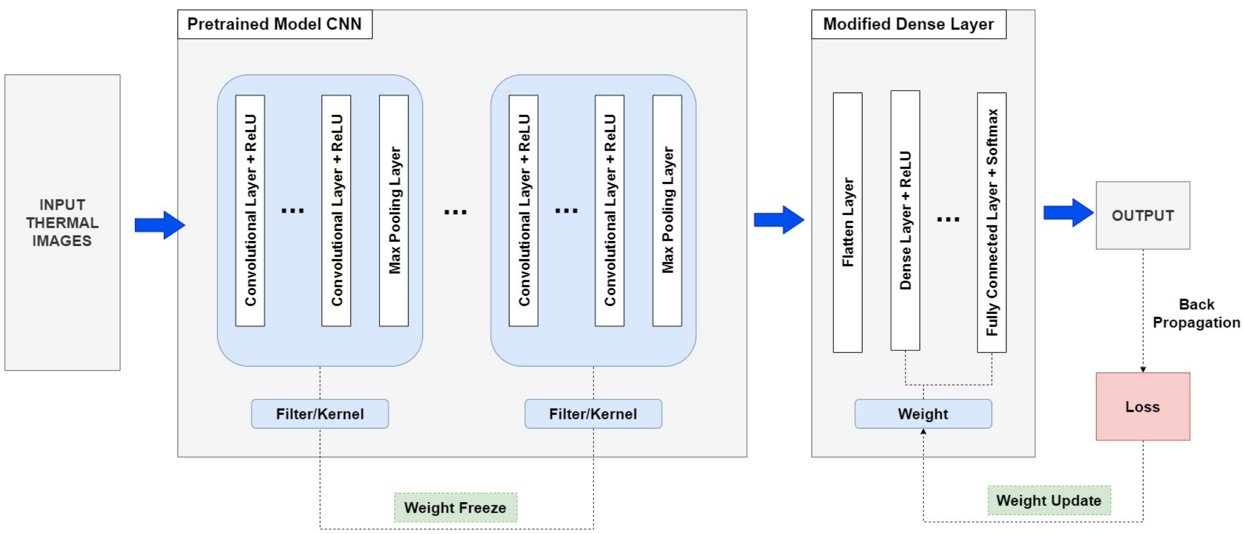

Fig. 2. Pre-trained CNN Model Architecture

Table 2. Tuned Hyper Parameters of Pre-trained CNN Model

Hyper Parameter	Value
Pre-trained CNN Model	VGG19, MobileNet, ResNet152V2, DenseNet201
Dataset Size	192 training images, 72 validation/testing images
Image Size	224 × 224 x 3 (RGB)
Augmentation Metrics	rotation_range= 20 width_shift_range= 0.1 height_shift_range= 0.1 shear_range= 0.1 zoom_range= 0.2 horizontal_flip= True
Batch Size	32
Modified Dense Layer Architecture	Flatten Layer, Dense Layer (64, activation = ReLu) Dense Layer (32, activation = ReLu) Dense Layer (16, activation = ReLu) Dense Layer (8, activation = ReLu) Fully Connected Layer (2, activation = Softmax)
Output Class	Normal, Obese (k = 2)
Training Epoch	50 Epochs
Loss Function	Categorical Cross Entropy
Optimizer	Adam Optimizer: α (learning rate) = 0.001, β1 (1st moment of exponential decay rate) = 0.9, β2 (2nd moment of exponential decay rate) = 0.999, ε (Numerical stability constant) = 10-7 (0.0000001)

A. Tuned Hyper Parameters

Fine Tuning is a concept of transferring weight from existing pre-trained model CNN to obtain the best result of the classification task [15], [16]. In the fine-tuning process, we tuned and identified the hyper parameters of the pre-trained CNN model, as shown in Table 2.

The pre-trained CNN models were tuned with VGG19 [19],MobileNet [20], ResNet152V2 [21], and DenseNet201 [22] networks. We chose the four pre-trained models because they have different depth layers, computation cost, and quantity of parameters. VGG19, ResNet152V2, and DenseNet201 were considered a very deep network, while MobileNet was considered a lightweight network. Different deep networks models were required to evaluate the trade-off between accuracy and computation time required in classifying thermal images of obesity.

We also tuned the hyperparameters of the dataset to achieve the best result in the training and validation process. The input thermal images dataset was configured with 192 images for training and 72 images for validation. The image size was resized to 224 × 224 pixels and contained 3 color channels of Red-Green-Blue (RGB). The dataset was augmented with rotation, width shifting, height shifting, shear mapping, zooming, and horizontal flipping to increase the dataset's variations and prevent the model from overfitting. Considering the total of the thermal images collected of 264 images, we tuned the batch size to 32 images for each batch.

The dense layers of the pre-trained model were modified to extract the images feature from the convolutional layer. Images from convolutional layers were flattened into matrix value by passing through the flattened layer. After the images were flattened, the images passed through several dense layers in the classification layer. The classification layer was tuned by adding four dense layers with the size of 64-32-16-8 and the ReLU activation function. The fully connected layer was adjusted with a size of 2 and followed with softmax activation function to produce the output classification of obese and normal class.

For the training and validation hyperparameters, we tuned the iterations/epochs required, loss function, and optimizer to achieve the best performance of the model. We tuned the iterations/epochs to 50 epochs, so the models had enough time for converging. The back propagations or learning phase is executed for each epoch or iteration. The output prediction was measured with the true labels of the dataset and defined as prediction loss. We used categorical cross-entropy as the loss function to measure the loss. While to optimize the weight value of each dense layer, an optimizer was needed to optimize the weight update. We used adam optimizer with α (learning rate) of 0.001, β_1 (1st moment of exponential decay rate) of 0.9, β_2 (2nd moment of exponential decay rate) of 0.999 and ε (Numerical stability constant) of 10^{-7} (0.0000001).

B. Evaluating Model Performances

To evaluate the model's performances, some metrics are recorded to present the performance of the models in classifying thermal images of obesity. The metrics consist of: training loss, training accuracy, validation loss, validation accuracy, trainable parameters, and elapsed training time (1 epoch), as shown in Table 3. While Figure 3 shows the training and validation accuracy due to the epochs required. In evaluating the model performances, we will discuss it based on the terms of (i) Classification performances, (ii) Computation speed, and (iii) Learning performances.

1. Evaluating Classification Performances

In evaluating the classification performances of the models, we compared the loss and accuracy metrics of the training and validation results, as shown in Table 3. The higher the model's accuracy and the lower the model's loss, the better its performance in classifying thermal images of obesity.

In terms of the model's training classification performances, MobileNet achieved the highest training accuracy of 97.92 % and the lowest training loss of 0.06. While the ResNet152V2 achieved the lowest training accuracy of 95.83 % and the highest training loss of 0.09. But the overall training classification performances of accuracy were above 95 %, and the loss was lower than 0.09. These results show that the models learn well in detecting temperatures feature in the thermal images dataset.

In terms of the model's validation classification performances, DenseNet201 achieved the highest validation accuracy of 83.33 % and the lowest validation loss of 0.35. While ResNet152V2 achieved the lowest validation accuracy of 70.83 % and the highest validation loss of 1.05. However, all models were overfitting because the validation accuracy was lower than the training accuracy. It shows that all models classify obesity accurately in training but degrade when validated with another image/dataset.

2. Evaluating Computation Speed

In evaluating the computation speed of the models, we compared the time elapsed for training of each model, as shown in Table 3. The time elapsed for training will represent the model's performance in computing speed. The quicker the time elapsed needed for the training, the higher models in computation speed. Even though the recorded elapsed time was a training elapsed time and not the real computation time when the model was embedded in an obesity diagnosis device, we could use the training elapsed time to represent the model's computation ability.

In terms of training elapsed time, MobileNet achieved the fastest elapsed training time of 12 seconds than other models. In contrast, VGG19 achieved the longest elapsed training time of 149 seconds/ epoch even though it had the smallest trainable parameters (1.6 million) than MobileNet (3.2 million) and other models. VGG19 architecture was consisted of multiple convolution layers and required more computation in the convolution process [19]. The elapsed time needed for train VGG19 was longer than other models. While MobileNet used depthwise convolution and pointwise convolution layers to minimize the computation needed in the convolution process [20]. The less computation is needed in the convolutions process, the quicker the elapsed time is needed for the classification.

3. Evaluating Learning Performances

In evaluating the learning performance of the models, we compared the training and validation accuracy graph of each model, as shown in Fig. 3. The curve between training and validation accuracy due to each epoch will determine the learning performance. The smaller the gap between the training and validation accuracy, the better its performance in learning features.

Figure 3 (a) shows that VGG19 has a small gap between training and validation accuracy at the 1st to 10th epochs. But the longer the training epochs passed, the bigger gap occurred after the 10th epoch. The training accuracy increases from the 11th to the 50th epochs, while the validation accuracy starts converging at the 11th to 50th epochs.

Figure 3 (b) shows that MobileNet has a small gap between its training and validation at the 1st to 18th epochs, while the gap gets bigger after the 19th to 50th

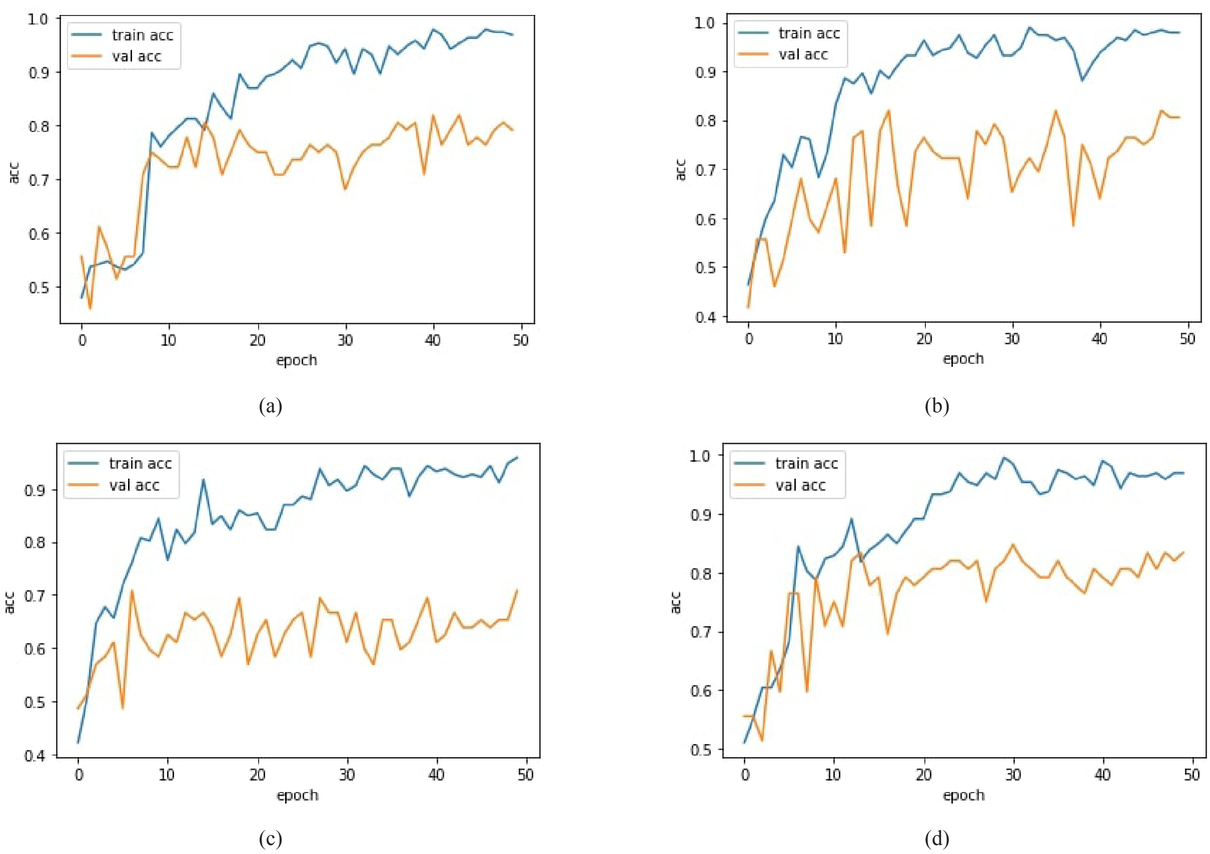

Fig. 3. Accuracy of Trained CNN Model (a) VGG19 (b) MobileNet (c) ResNet152V2 (d) DenseNet201

Table 3. Training and Validation Results

Model	Training		Validation/ Testing		Trainable Params	Training Elapsed Time (1 Epoch)
	Loss	Acc (%)	Loss	Acc (%)		
VGG19	0.07	96.88	0.87	79.17		
MobileNet	0.06	97.92	0.57	80.56	3,214,090	12 s
ResNet152V2	0.09	95.83	1.05	70.83	6,425,354	105 s
DenseNet201	0.08	96.88	0.35	83.33	6,023,946	52 s

epoch. The training accuracy increases from the 1st to 20th epoch and starts converging after the 21st epoch. In contrast, the validation accuracy keeps fluctuating from the 1st to 40th epoch and increases at the 41st epoch.

Figure 3 (c) shows that ResNet152V2 has a small gap that occurred between the 1st to 8th epoch, but the gap gets bigger after the 9th to 50th epoch. The training accuracy keeps increasing from the 1st to 30th epoch and converges at the 31st to 50th epoch. In contrast, the validation accuracy starts converging and fluctuating at the 10th to 50th epoch.

Figure 3 (d) shows that DenseNet201 has a small gap that occurred between the 1st to 15th epoch, but the gap gets bigger after the 16th to 50th epoch. The training accuracy keeps increasing from the 1st to 15th epoch and converges at the 16th to 50th epoch. In contrast, the validation accuracy increases from the 1st to the 20th epoch and starts converging at the 21st to the 50th epoch.

After we evaluated all of the models, we could conclude that ResNet152V2 was considered the biggest gap between its training and validation accuracy. While DenseNet201 was considered as the smallest gap between its training and validation accuracy. Even though all models were overfitting, DenseNet201 achieved the smallest gap of 13.33 % between its training accuracy (96.88 %) and validation accuracy (83.33 %). DenseNet201 was also considered the most stabilized learning, compared to MobileNet and ResNet152V2, which have fluctuating learning.

IV. CONCLUSION

In this study, we fine-tuned the pre-trained CNN model in classifying the thermal images dataset of obesity. DenseNet201 model outperformed other models in terms of classifying accuracy (validation accuracy of 83.33 %) and learning performance (the closest gap of 13.33 % between training and validation accuracy). While the MobileNet model outperformed other models in terms of computation speed (training elapsed time of 12 seconds/ epoch). The proposed DenseNet201 model was suitable to be implemented as an early screening system of obesity for the health workers or physicians due to the model's performance in classification performance. Meanwhile, the proposed MobileNet model was suitable for early detection/diagnosis of obesity for mobile applications due to the model's performance in computation speed terms. Further research will focus on increasing the dataset size and the output classification to 3 classes (obese, overweight, normal) to avoid over. An efficient CNN model with a balanced tradeoff between accuracy and computation time was also required to develop further.

REFERENCE

[1] Y. C. Chooi, C. Ding, and F. Magkos, "The epidemiology of obesity," *Metabolism.*, vol. 92, pp. 6–10, 2019.

[2] J. Berglund, "The future of fat: As obesity grows to epidemic proportions, researchers are trying to halt the trend," *IEEE Pulse*, vol. 9, no. 5, pp. 8–11, 2018.

[3] WHO, "World Health Statistics 2021: Monitoring Health For The SDGs," Geneva, 2021. [Online]. Available: https://www.who.int/data/gho/publications/world-health-statistics.

[4] K. Jindal, N. Baliyan, and P. S. Rana, "Obesity prediction using ensemble machine learning approaches," in *Advances in Intelligent Systems and Computing*, 2018, vol. 708, pp. 355–362.

[5] T. M. Dugan, S. Mukhopadhyay, A. Carroll, and S. Downs, "Machine learning techniques for prediction of early childhood obesity," *Appl. Clin. Inform.*, vol. 6, no. 3, pp. 506–520, 2015.

[6] S. A. Thamrin, D. S. Arsyad, H. Kuswanto, A. Lawi, and S. Nasir, "Predicting Obesity in Adults Using Machine Learning Techniques: An Analysis of Indonesian Basic Health Research 2018," *Front. Nutr.*, vol. 8, no. June, pp. 1–15, 2021.

[7] K. W. DeGregory *et al.*, "A review of machine learning in obesity," *Obes. Rev.*, vol. 19, no. 5, pp. 668–685, 2018.

[8] X. Pang, C. B. Forrest, F. Le-Scherban, and A. J. Masino, "Understanding early childhood obesity via interpretation of machine learning model predictions," *Proc. - 18th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2019*, pp. 1438–1443, 2019.

[9] M. E. Symonds and H. Budge, "How promising is thermal imaging in the quest to combat obesity?," *Imaging Med.*, vol. 4, no. 6, pp. 589–591, 2012.

[10] S. Brasil, A. C. Renck, F. de Meneck, M. L. Brioschi, E. F. Costa, and M. J. Teixeira, "A systematic review on the role of infrared thermography in the Brown adipose tissue assessment," *Rev. Endocr. Metab. Disord.*, vol. 21, no. 1, pp. 37–44, 2020.

[11] R. Rashmi and U. Snekhalatha, "Thermal imaging method in the evaluation of obesity in various body regions–A preliminary study," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 912, pp. 062022, 2020.

[12] H. Leo, K. Saddami, R. Roslidar, M. Oktiana, F. Arnia, and K. Munadi, "A Study of Parameters Required for a Thermogram Dataset in Obesity Detection : A Systematic Review," 2021, no. Dl.

[13] R. Rashmi, S. Umapathy, and P. T. Krishnan, "Thermal imaging method to evaluate childhood obesity based on machine learning

techniques," *Int. J. Imaging Syst. Technol.*, no. March, pp. 1–17, 2021.

[14] U. Snekhalatha, K. Palani Thanaraj, and K. Sangamithrai, "Obesity Detection in Thermal Imaging Using Convolution Neural Network: A Comparison with Machine Learning Models," in *Proceedings of the International e-Conference on Intelligent Systems and Signal Processing*, 2022, pp. 583–592.

[15] X. Yin, W. Chen, X. Wu, and H. Yue, "Fine-tuning and visualization of convolutional neural networks," *Proc. 2017 12th IEEE Conf. Ind. Electron. Appl. ICIEA 2017*, vol. 2018-Febru, pp. 1310–1315, 2018.

[16] R. Roslidar, K. Saddami, F. Arnia, M. Syukri, and K. Munadi, "A study of fine-tuning CNN models based on thermal imaging for breast cancer classification," *Proc. Cybern. 2019 - 2019 IEEE Int. Conf. Cybern. Comput. Intell. Towar. a Smart Human-Centered Cyber World*, pp. 77–81, 2019.

[17] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[18] A. C. Ian Goodfellow, Yoshua Bengio, *Deep learning*. MIT Press, 2016.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[20] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017.

[23] A. Elhassouny and F. Smarandache, "Trends in deep convolutional neural Networks architectures: A review," *Proc. 2019 Int. Conf. Comput. Sci. Renew. Energies, ICCSRE 2019*, no. 1, pp. 1–8, 2019.