SCIENCE AND INFORMATION TECHNOLOGY
SINTECH
JOURNAL

SINTECH JOURNAL | ISSN 2598-7305 | E-ISSN 2598-9642
Vol. 5 No 1 – April 2022 | https://s.id/sintechjournal
DOI : https://doi.org/10.31598
Publishing : LPPM STMIK STIKOM Indonesia

SECURITY MONITORING USING IMPROVED MOBILENET V2 WITH FINE-TUNING TO PREVENT THEFT IN RESIDENTIAL AREAS DURING THE COVID-19 PANDEMIC

Ryandra Anditto[1], Rusdianto Roestam[2]

[1,2]Information Technology, Faculty of Computing, President University
Jl. Ki Hajar Dewantara, Kota Jababeka, Cikarang Baru, Bekasi, Indonesia

e-mail: ryandra.anditto@student.president.ac.id[1], rusdianto@president.ac.id[2]

Received : January, 2022	Accepted : April, 2022	Published : April, 2022

Abstract

In this journal, authors improved a deep learning, it was MobileNet V2 which can learn with higher speed and accuracy using the ImageNet dataset. The basic architecture of MobileNet V2 was modified using a fine-tuning technique. It did not change the entire weight of the default deep learning model, there was freezing of some layers. The modifications involved by changing the parameters on the layer. The improved model would be trained using the ImageNet dataset based on the security monitoring context. The dataset class objects selected for training were objects that usually used by suspicious people with the aim of committing crime of theft. The results of training using an improved model could increase accuracy up to 71% with a difference of 3% from the training results of the default model of MobileNet V2. Since MobileNet V2 was a lightweight deep neural network model that had few parameters compared to other neural network model parameters, this modified model could be implemented on devices with low specifications, such as mobile devices or raspberry pi in the form of real-time applications.

Keywords: *Convolutional Neural Network (CNN), Deep Neural Network (DNN), Fine-Tuning, ImageNet Dataset, MobileNet V2, Security Monitoring*

Abstrak

Dalam jurnal ini, penulis meningkatkan deep learning, yaitu MobileNet V2 yang dapat belajar dengan kecepatan dan akurasi yang lebih tinggi menggunakan dataset ImageNet. Arsitektur dasar MobileNet V2 dimodifikasi menggunakan teknik fine-tuning. Teknik tersebut tidak mengubah seluruh berat model deep learning bawaan, hanya melakukan freezing beberapa layer. Modifikasi dilakukan dengan mengubah parameter pada layer. Model yang ditingkatkan akan dilatih menggunakan dataset ImageNet berdasarkan konteks pemantauan keamanan. Objek kelas dataset yang dipilih untuk pelatihan adalah objek yang biasanya digunakan oleh orang-orang yang mencurigakan dengan tujuan untuk melakukan kejahatan pencurian. Hasil pelatihan dengan model yang ditingkatkan dapat meningkatkan akurasi hingga 71% dengan selisih 3% dari hasil pelatihan model bawaan MobileNet V2. Karena MobileNet V2 adalah model deep neural network yang ringan dan memiliki sedikit parameter dibandingkan dengan parameter model neural network lainnya, model yang dimodifikasi ini dapat diimplementasikan pada perangkat dengan spesifikasi rendah, seperti perangkat seluler atau raspberry pi dalam bentuk aplikasi real-time.

Kata Kunci: *Convolutional Neural Network (CNN), Deep Neural Network (DNN), Fine-Tuning, ImageNet Dataset, MobileNet V2, Pemantauan Keamanan*

1. INTRODUCTION

Since the outbreak of the COVID-19 pandemic in Indonesia, many impacts have been felt. The COVID-19 pandemic has resulted in an increase in the poverty rate so that the crime rate has also increased. One of the acts of theft in a residential area with various modes. Many jobs breaker is the biggest contributor to poverty so they cannot fulfill their daily economic needs, thus urging perpetrators to steal from people's homes. Quoting from Kompas.com, the crime rate increased by 10 percent in the Greater Jakarta area during the pandemic from March to April 2020 [1]. It can be proven that many who reported or were reported in this crime of theft were brought to court [2].

Cases of theft that often occur increasingly make the community restless because the method used is also continuously developing. At first the theft was carried out in conventional ways such as damaging doors, windows, jumping over the fence to the roof of the house, but in its development the theft was carried out openly and even by more than one person, no longer in a quiet place, but also in a crowd become a target for theft.

Theft crimes have developed, initially mostly carried out at night, now increasing to daytime. The equipment used has also developed from sharp weapons, transportation and communication tools, currently turning into firearms, using advanced transportation tools, and complex communication tools [3].

Although the crime of theft cannot be completely eradicated, efforts that can be taken are to suppress or reduce the number of crimes. In the context of protecting property and protecting individual property, prevention and control of theft is carried out [4]. One of the preventive steps that can be taken at home to keep the house safe from thieves is using security monitoring.

There are many forms of security monitoring tools in today's modern era, one of the most widely used is CCTV (Closed Circuit Television). CCTV is a device that has a camera in it which aims to collect videos that are installed in several places or locations and used for various purposes. Recently, the performance of CCTV has been improved by developing technology that can automatically recognize faces based on facial information that has been stored in the CCTV system [5]. Also, able to recognize various kinds of objects caught on camera. Namely objects that are usually used by suspicious people with the aim of committing the crime of theft. This technology is obtained with the help of deep learning.

Deep learning is a subset of machine learning. Machine learning consists of complex sets of mathematics implemented in the form of models and algorithms that gradually improve the performance of a task. It uses datasets that contain large amounts of data as input, then is trained into various models to make predictions. It is very broad. Therefore, it is categorized into 2 groups: supervised and unsupervised learning. Supervised learning is an algorithm that has a dataset to input and an output part that has been set. Supervised learning has a sub-section, namely semi-supervised learning. Both are similar but the difference is semi-supervised learning has missing datasets. While unsupervised learning is an algorithm that has a dataset to input but the output part is not set. Supervised learning is used as an application for classification and regression, while unsupervised learning is used for feature learning and the inverse, dimensionality reduction [6].

There are several machine learning algorithms that are widely used today, such as, Naïve Bayes Classifier Algorithm, K Means Clustering Algorithm, Support Vector Machine Algorithm, Linear Regression, Logistic Regression, Artificial Neural Networks, Decision Trees, Random Forests, and Nearest Neighbours [7].

Deep learning which is part of machine learning is a machine learning technique that teach computers to think the way humans think naturally, learning by example. In deep learning, computer models classify tasks directly in various forms: images, text, or sounds. Deep learning models can achieve accuracy that is able to level with human abilities. The models are trained using large datasets and the neural network architectures consists of many layers [8].

Based on [9], the most famous types of deep learning networks are Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Recursive Neural Networks (RvNNs).

Table 1: MobileNet V2 Layers
[Source: [10]]

Input	Operator	*t*	*c*	*n*	*s*
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1 × 1	-	1280	1	1
$7^2 \times 1280$	avgpool 7 × 7	-	-	1	-
$1^2 \times 1280$	conv2d 1 × 1	-	k	-	

From several kinds of deep learning, the authors use the lightweight CNN (Convolutional Neural Network) detection model: MobileNet V2 as a basic development. Then in training using the ImageNet dataset which contains a large variety of images and classes. MobileNet V2 model is proposed by Google in 2017 [11]. Table 1 is the architecture of MobileNet V2. Basically, MobileNet V2 has been able to achieve high accuracy like other neural network models, because it was developed from the previous model, it is MobileNet V1 which was proposed by Google in 2015 [11]. With high accuracy, MobileNet V2 has less parameters compared to other models to achieve high accuracy requires a lot of parameters too. As a comparison of research [12], using MobileNet V2 produces an accuracy of 96.20% with parameters 2,234,120, while VGG16 produces an accuracy of 92.93% with parameters 134,293,320. The research was conducted on their own dataset.

The purpose of this research is to make modifications to increase the learning speed and accuracy of the MobileNet V2 model compared to the unmodified model, by performing fine-tuning techniques on the layer. Some parameters were changed and some layers were frozen to reduce the weight of the default model changed. When training the model using an ImageNet dataset that has been selected based on the object class that is usually used by suspicious people with the aim of committing crimes, the layer that is not frozen will also be retrained to adjust the parameter changes to the weight of the frozen model.

Several research that are related and relevant to the author's research include:

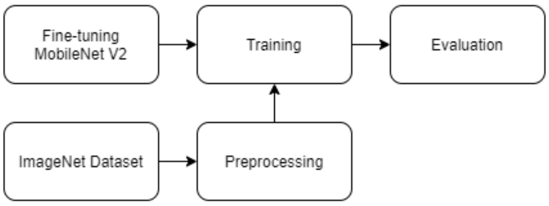
1. In research [13], the authors explain that the Faster R-CNN and Inception V2 methods get 87.58% of average and a maximum of 89.58% accuracy by using the GDB X-Ray dataset. Strengthened by doing a comparison between Inception V2 - SSD, MobileNet V2 - SSD, ResNet 101 - R-FCN, and ResNet 101 - Faster R-CNN. The number of images and objects used in the simulation is 250 for handgun, 250 for knife, 250 for razorblade, 250 for shuriken, 150 for battery, 150 for wire, and 150 for mortar. So, the total object images are 1450.

2. The authors conducted research on face recognition [14] using the VGG16 model that had been trained with ImageNet datasets against a dataset containing people's faces provided from Yale and AT&T. The Yale database consists of 11 images per subject drawn from 15 peoples with expressions stemming from different emotions. Each image has a pixel with a gray scale and a size of 480 x 640. Meanwhile, the AT&T database consists of 10 facial images with expressions originating from emotions and taken at different times from a total of 40 subjects. Each image has a size of 92 x 112. The Yale dataset is split into 90 training data and 75 testing data. The AT&T dataset is split into 240 training data and 160 testing data. Implemented with Keras using python. The number of epochs is set into 10 and batch size is 2. The learning rate of the network is set at 0.00001 resulting in an accuracy of 98.7% using the Yale dataset and 100.0% using the AT&T dataset. The curation results are strengthened by comparing the PCA method where the Yale dataset produces 82.0% and the AT&T dataset produces 96.5%.

3. Authors [15] proposed RMNv2 (Reduced MobileNet V2). The research carried out several developments on the basic architecture of MobileNet V2 including disabling down sampling layers, replacing bottlenecks with HetConv blocks, activation function is mish, and auto augmentation. The model is trained using Stochastic Gradient Descent (SGD) optimizer with different 3 learning rates, they are 0.1, 0.01, and 0.001. The number of epochs used is 200. For training the network, used a batch size of 128 and for test epochs used a batch size of 64. The results of this research are decreasing the model size reached 52.2% and increasing accuracy to 92.4%.

2. RESEARCH METHOD

The following picture is the steps in the research.



Picture 1. Research Steps

2.1 Prepare Dataset

In this research, the authors use images of the object from the ImageNet dataset to be trained on the modified MobileNet V2 model. ImageNet is built upon the hierarchical structure provided by WordNet. In the development of ImageNet has the aim to contain in the order of 50 million cleanly labeled full resolution images (500-1000 per synset) [16]. Of the total object classes on ImageNet, the authors only choose 30 object classes, namely objects that are usually used by suspicious people with the aim of committing crimes.

2.2 Preprocessing Dataset

In this step, the prepared dataset will be preprocessed using the TensorFlow library, it is ImageDataGenerator. The tensor of each dataset is only scaled from 0 to 1. In order to reduce confusion and increase efficiency during model training.

2.3 Fine-tuning MobileNet V2

Fine-tuning is a transfer learning technique that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [17]. The fine-tuning in this research was carried out in two steps, namely changing parameters and freezing layers. First, the last output is fetched on the block_16_project_BN (Batch Normalization) layer. The output will be the next input layer, Conv_1 (Conv2D), where the parameter units are changed from 1280 to 2048 and kernel size 1 to 5. The last layer added for average and prediction, the activation is softmax because the authors use more than one object class. Table 2 is the modified model layers. After that, the first layer up to the block_10_project_BN (Batch Normalization) layer is frozen to prevent weight changes of the models that have been trained with ImageNet dataset. The rest of the layers after that are retrained to adjust to parameter changes made to the lower layers.

Table 2: MobileNet V2 Layers Modification

Input	Operator	*t*	*c*	*n*	*s*
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1 × 1	-	2048	1	1
$3^2 \times 2048$	avgpool 3 × 3	-	-	1	-
$1^2 \times 2048$	conv2d 1 × 1	-	k	-	

2.4 Training Model

ImageNet dataset that has passed the preprocessing step will be trained with the modified model. Loss parameter using categorical crossentropy, because the activation model is using softmax, SGD optimizer, learning rate 0.01, and metrics accuracy.

2.5 Evaluation

In this step, the results of the training model will be evaluated based on loss, accuracy, validation loss, and validation accuracy. Then the authors made a comparison between the

MobileNet V2 model before it was modified and after it was modified to see an increase in terms of learning speed and accuracy.

3. RESULT AND DISCUSSION

3.1 Data Description

3.1.1 Hardware and Software Used

The authors conduct training on the MobileNet V2 model using the following hardware and software specifications:

- 6-Core Processor 3.70 GHz
- 32 GB RAM
- Windows 10 Home 64-bit
- Visual Studio Code 1.63.2
- Python 3.9.7
- TensorFlow 2.7.0
- Matplotlib 3.5.0

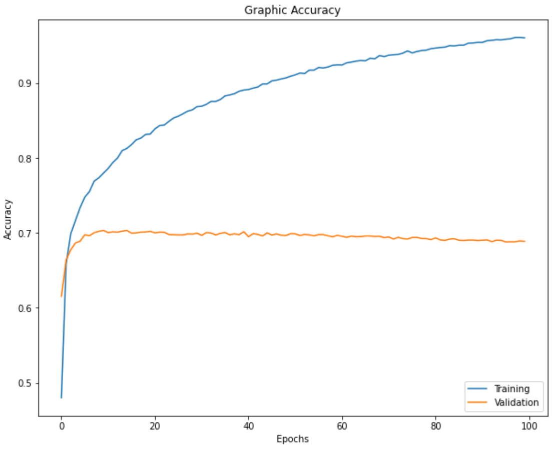
3.1.2 Dataset Description

Table 3 below is list of 30 class objects from the ImageNet dataset that are used as model training materials. Objects are selected based on objects that are usually used by people who aim to commit crimes. The total image of each object taken from the ImageNet dataset is divided into two parts, data train and data validation with a ratio of 70:30.

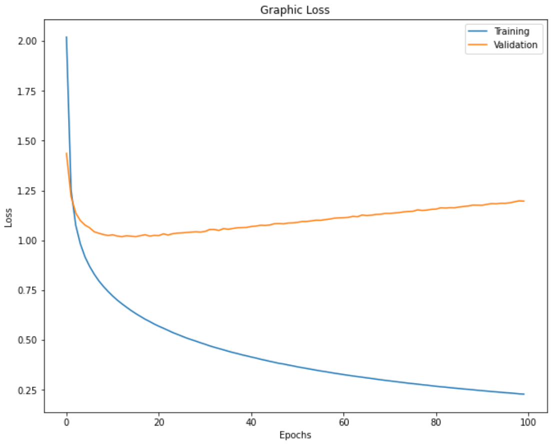
Table 3: List of Class Objects with Total Data Train and Data Validation

No	Class	Train	Validation
1	Ax	275	119
2	Bag	1060	456
3	Bicycle	495	214
4	Bomb	47	21
5	Car	1128	485
6	Cellular Phone	422	183
7	Crowbar	68	30
8	Dagger	305	132
9	Drill	232	101
10	Gun	413	179
11	Hammer	418	180
12	Handsaw	119	53
13	Hat	479	207
14	Headset	2	2
15	Helmet	534	231
16	Jacket	693	298
17	Knife	824	354
18	Machete	215	94
19	Mask	612	263
20	Motorcycle	374	161
21	Person	333	144
22	Pliers	269	117
23	Rope	411	177
24	Scissors	325	141
25	Screwdriver	374	161
26	Short Pants	320	138
27	Sickle	66	30
28	Sword	448	193
29	Walkie Talkie	280	121
30	Wrench	320	138

3.2 Discussion

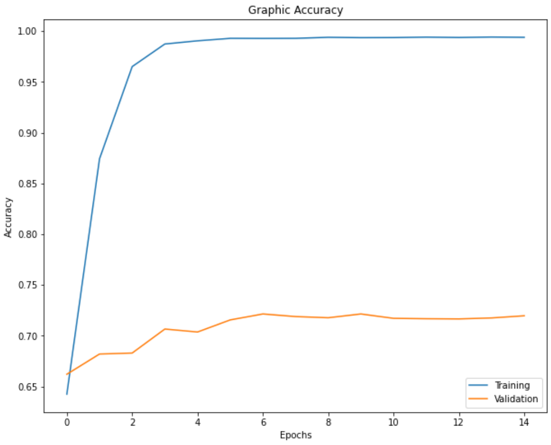


Picture 2. Graphic Accuracy of Default MobileNet V2

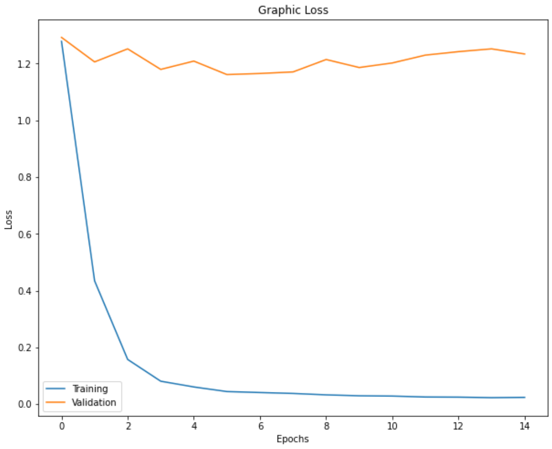


Picture 3. Graphic Loss of Default MobileNet V2

Figure 2 and 3 above is MobileNet V2 before modification. Figure 4 and 5 below is MobileNet V2 which has changed parameters and done fine-tuning techniques.



Picture 4. Graphic Accuracy of Improved MobileNet V2



Picture 5. Graphic Loss of Improved MobileNet V2

Viewed from the graph, the numbers are not too different between the graphs before and after modification. Therefore, the following table 4 provides detailed numbers from the results of each training.

Table 4: Training Results

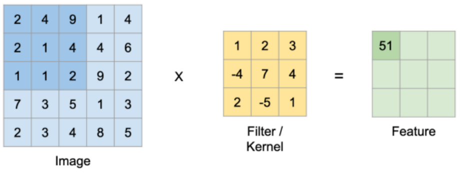
Model MobileNet V2	Epochs	Loss	Accuracy	Validation Loss	Validation Accuracy	Validation Ratio
Default	100	0.2286	0.9609	1.1970	0.6889	1.7375
Improved (Ours)	15	0.0240	0.9938	1.2341	0.7197	1.7147

From table 4 it can be seen that the improved model can learn faster than the default model. The accuracy of the default model can reach 96% at 100 epochs, while the improved model reaches 99% accuracy only need 15 epochs. The result of the calculation of the validation ratio between validation loss and validation accuracy states that the improved model is smaller than the default model. It indicates that the model is improved even though the validation loss increases compared to the validation loss of the default model, but the validation accuracy is higher than the validation accuracy of the default model. Validation of model accuracy is improved to cover the validation loss. The following is the calculation of the validation ratio.

$$Validation\ Ratio = \frac{Validation\ Loss}{Validation\ Accuracy}$$

(1)

The improved model is superior to the default model because of the larger unit and kernel size. Unit size affects the complexity of the layer, the larger the unit size, the more complex the layer learns. Then, kernel size or also called filter size. The larger the kernel size, the larger the pixels taken in the tensor calculation. More details can be seen in Figure 6.



$$\begin{bmatrix} 2 & 4 & 9 & 1 & 4 \\ 2 & 1 & 4 & 4 & 6 \\ 1 & 1 & 2 & 9 & 2 \\ 7 & 3 & 5 & 1 & 3 \\ 2 & 3 & 4 & 8 & 5 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \\ -4 & 7 & 4 \\ 2 & -5 & 1 \end{bmatrix} = \begin{bmatrix} 51 & & \\ & & \\ & & \end{bmatrix}$$

Picture 6. Example of Kernel Size 3 by 3 [Source: [18]]

The following equation to get the output layer size as input to the next layer.

$$\frac{(n+2p-f+1)}{s}$$

(2)

Description:
n = layer size
p = padding is used on layer 0 or 1
f = filter
s = strides

From equation 2, the default model has an output of 7, while the improved model has an output of 3 at the Conv_1 (Conv2D) layer.

4. CONCLUSION

The conclusion of the research that has been done is to get a faster learning speed and higher accuracy with the MobileNet V2 model, by doing fine-tuning techniques to change parameters and freezing layers. The improved model was trained using the ImageNet dataset, which only selected 30 class objects from all of them based on objects that are usually used by suspicious people with the aim of committing crimes. The training results show that the improved model is able to achieve 71% higher accuracy with a difference of 3% from the default MobileNet V2 model, which is 68%. High validation accuracy affects high validation loss as well, but based on the validation ratio, the improved model remains superior to the default model. From the results of this research, it is hoped that further research can increase accuracy higher than the improved model proposed by the authors with a lower validation loss.

REFERENCES

[1] "Waspada Aksi Pencurian, Ini Tips Agar Rumah Aman dari Pencuri." https://www.allianz.co.id/explore/waspada-aksi-pencurian-ini-tips-agar-rumah-aman-dari-pencuri1.html (accessed Feb. 26, 2022).

[2] Z. Lukman, "Faktor-Faktor Dan Upaya Penanggulangan Tindak Pidana Pencurian Sepeda Motor (Studi Kasus Polresta Banda Aceh)," *J. Justisia*, vol. 4, 2019.

[3] A. Suharsoyo, "Karakter pelaku tindak pidana pencurian dalam tipologi kejahatan pencurian di wilayah sukoharjo," *Jurisprudence*, vol. 5, no. 1, pp. 64–74, 2015.

[4] L. Mamluchah, "Peningkatan Angka Kejahatan Pencurian pada Masa Pandemi dalam Tinjauan Kriminologi dan Hukum Pidana Islam," *Huk. Pidana Islam*, vol. 6, no. 1, pp. 1–26, 2020, [Online]. Available: http://jurnalfsh.uinsby.ac.id/index.php/HPI/article/view/1037/763.

[5] B. Muthusenthil and H. sung Kim, "CCTV surveillance system, attacks and design goals," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 4, pp. 2072–2082, 2018, doi: 10.11591/ijece.v8i4.pp2072-2082.

[6] M. P. Hosseini, A. Hosseini, and K. Ahi, "A Review on Machine Learning for EEG Signal Processing in Bioengineering," *IEEE Rev. Biomed. Eng.*, vol. 14, no. c, pp. 204–218, 2021, doi: 10.1109/RBME.2020.2969915.

[7] "A guide to the types of machine learning algorithms and their applications | SAS UK." https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html (accessed Feb. 26, 2022).

[8] "What Is Deep Learning? | How It Works, Techniques & Applications - MATLAB & Simulink." https://www.mathworks.com/discovery/deep-learning.html (accessed Feb. 26, 2022).

[9] L. Alzubaidi *et al.*, *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, vol. 8, no. 1. Springer International Publishing, 2021.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.

[11] B. Dan, X. Sun, and L. Liu, "Diseases and Pests Identification of Lycium Barbarum Using SE-MobileNet V2 Algorithm," *Proc. - 2019 12th Int. Symp. Comput. Intell. Des. Isc. 2019*, vol. 1, pp. 121–125, 2019, doi: 10.1109/ISCID.2019.00034.

[12] S. Li *et al.*, "Individual dairy cow identification based on lightweight convolutional neural network," *PLoS One*, vol. 16, no. 11 November, pp. 1–13, 2021, doi:

10.1371/journal.pone.0260510.

[13] J. Koçi, A. O. Topal, and M. Ali, “Threat Object Detection in X-ray Images Using SSD, R-FCN and Faster R-CNN,” pp. 10–15, 2020.

[14] R. Meena Prakash, N. Thenmoezhi, and M. Gayathri, “Face Recognition with Convolutional Neural Network and Transfer Learning,” *Proc. 2nd Int. Conf. Smart Syst. Inven. Technol. ICSSIT 2019*, no. Icssit, pp. 861–864, 2019, doi: 10.1109/ICSSIT46314.2019.8987899.

[15] M. Ayi and M. El-Sharkawy, “RMNv2: Reduced Mobilenet V2 for CIFAR10,” *2020 10th Annu. Comput. Commun. Work. Conf. CCWC 2020*, pp. 287–292, 2020, doi: 10.1109/CCWC47524.2020.9031131.

[16] L. Fei-Fei, J. Deng, and K. Li, “ImageNet: A Large-Scale Hierarchical Image Database,” *J. Vis.*, vol. 9, no. 8, pp. 1037–1037, 2010, doi: 10.1167/9.8.1037.

[17] V. Taormina, D. Cascio, L. Abbene, and G. Raso, “Performance of fine-tuning convolutional neural networks for HEP-2 image classification,” *Appl. Sci.*, vol. 10, no. 19, pp. 1–20, 2020, doi: 10.3390/app10196940.

[18] “Convolutional Neural Networks — A Beginner’s Guide | by Krut Patel | Towards Data Science.” https://towardsdatascience.com/convolution-neural-networks-a-beginners-guide-implementing-a-mnist-hand-written-digit-8aa60330d022 (accessed Feb. 26, 2022).