Research and Development of the Pupil Identification and Warning System using AI-IoT

Trinh Luong Mien [1*], Vo Van An [2], Ngo Thi Thu Huong [3]
[1] Department of cybernetics, University of transport and communications, Hanoi, Vietnam
[2,3] Dept. of Electrical and Electronic, Campus in Ho Chi Minh City, University of transport and communications, Vietnam
Email: [1] mientl@utc.edu.vn, [2] anvo1088@gmail.com, [3] huongntt1_ph@utc.edu.vn
*Corresponding Author

Received: 10 June 2022; Revised: 29 June 2022; Accepted: 24 July 2022

***Abstract*— Currently, pupils being left in the classroom, in the house or in the car is happening a lot, causing unintended incidents. The reason is that parents or caregivers of pupils go through busy and tiring working hours, so they accidentally leave pupils in the car, indoors, or forget to pick up students at school. In this paper, we developed an algorithm to recognize students who use neural networks and warn managers, testing on a model integrated Raspberry Pi 4 kit programmed on Python in combination with cameras, sim modules, and actuators to detect and alert abandoned pupils to the manager to take timely remedial measures and avoid unfortunate circumstances. With the ability to manage students, the system collects and processes images and data on student information for artificial intelligence (AI) systems to recognize when operating. The system of executive structures serves to warn when students are left in the car, in the classroom, or in the house to avoid unintended incidents or safety risks.**

***Keywords*— AI; IoT; CNN; Raspberry; Image Processing**

I. INTRODUCTION

Nowadays, along with the continuous development and progress of science and technology, image processing is a topic that needs to be paid attention to and developed. In addition, the popularity of the raspberry pi kit with compact size, considered as a miniature computer should have many applications in image processing to offer several solutions to apply to live.

There have been numerous applied solutions for photo processing in life [1], some prominent studies in traffic such as applying photo processing to warn drowsy drivers [2] to minimize traffic accidents associated with drowsiness by drivers. In industry, image processing is also used to identify devices using Matlab [3], checking the card shape of the voice recognition operator [4] for security purposes. Many studies have also used image processing in combination with artificial intelligence for facial recognition [5] to eliminate excess facial data. Or in medicine, image processing to detect symptoms of a pupil's depressive eggs [6], analyze the emotional states of pupils in classroom environments [7], or use image processing to teach learning topics [8]. Today, forest fires are very common on earth causing a lot of damage to people, materials, and spirits. As a result, the problems of forest fire detection were in turn studied by scientists, one of the most prominent of which was the use of deep learning to detect forest fires [9] or via video to detect fire and smoke using artificial intelligence [10] or designing an image-based fire alarm system designed using a handbook computer and webcam as the main device [11]. Furthermore, the researchers used image processing technologies to accurately identify images sent back from the satellite [12, 13, 14]. Researches on applying technology of human detection, image recognition, image processing [18-20] or developing Deep Learning algorithms in embedded systems [21][22] have obtained certain results. The above studies have shown us the importance of image processing and its application in everyday life. However, currently, the above studies are mostly responsive in terms of image processing but there is not a clear system to thoroughly address common conditions in life.

This paper presents the study of identification and warning pupils to use Raspberry Pi 4 in conjunction with the artificial neural network processed via Python in image recognition to help users manage to monitor effectively. This system helps users identify and warn when pupils are left in the car, in the classroom, in the house to avoid unintended incidents or safety risks.

The main research content of the article is to develop a hardware system and a computer program to identify students in the classroom, thereby remotely alerting administrators via email or phone. The hardware program is developed on Raspberry Pi 4. The computer program is developed based on convolutional neural network with mobileNet network architecture and efficient network training factor. This article layout consists of four parts. Part I provides an overview. Part II presents the development of system hardware, student identification algorithms and integrated warning system to managers. Part II presents test scenarios and evaluates test results. Part IV is the conclusion

II. BUILDING A SYSTEM FOR WARNING AND IDENTIFYING PUPILS

A. Pupil identification and warning model

There have been many different models on the market, in this research paper we are providing a user-friendly model. Fig. 1 gives an overview of the operation of the pupil identification and alert system. Raspberry Pi's support combined with an artificial neural network in image recognition helps users manage surveillance effectively.

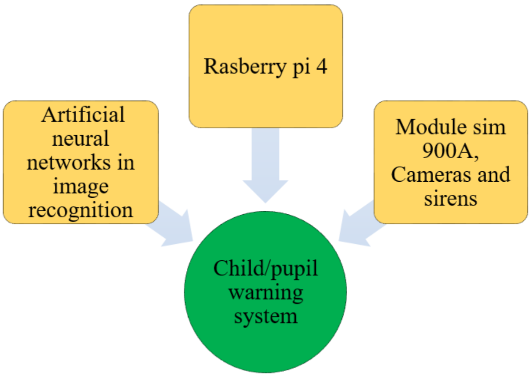

Fig. 1. Identify and warn children/pupils

B. Identification and warning system block diagram

In this section, the author presents the design of the identification and warning system. This control system was designed by the author using a Raspberry Pi4 kit [15].

The identification and warning system structure diagram is presented as shown in Fig. 2, including the following blocks.

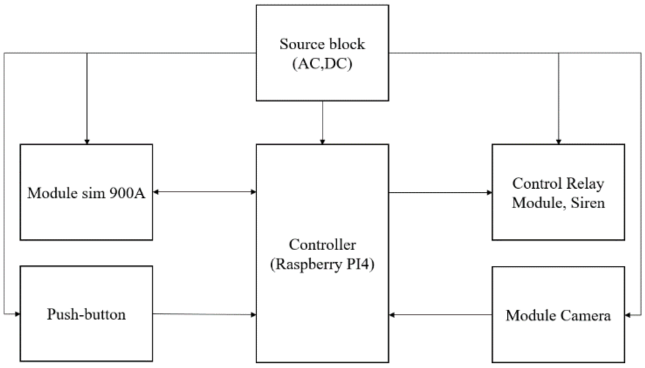

Fig. 2. Block diagram of the system.

The overall structure of the system consists of 6 main parts: A power supply for controllers (Raspberry Pi4) and execution devices (sirens). The control relay module is used to receive signals from a Raspberry embedded computer then amplifies the isolation control with the actuators such as: siren, flashing light, ... Logitech's HD C270 camera module has the function of recording information about images and videos to send signals back to Raspberry processing. Sim module 900 sends messages or phone calls to users

C. Raspberry Pi4 kit and camera module

In this article, the author uses the Raspberry Pi 4 Model B kit that receives data from the camera and then processes and gives a control signal. The Raspberry PI 4 Model B kit has a Quad-core Cortex-A72 processor core with a clock max of 1.5GHz, two micro-HDMI ports, supports 2 USB 2.0 ports and 2 USB 3.0 ports, IEEE 802.11ac standard Wi-Fi support with Dual-Band 2.4GHz and 5GHz, GPIO port, SPI, I2C, peripheral connection, Bluetooth 5.0 support, supports high-speed Ethernet ports (Gigabit Ethernet) and PoE (Power over Ethernet) support [14] (Fig. 3). The Raspberry PI 4 kit connects to the 900 sim module using the UART protocol, the RX battery on the 900 sim module connects to the TX battery on the Raspberry PI 4 Kit, and the TX battery on the 900 sim module connects to the RX battery on the Raspberry PI 4 Kit, the GND battery of the sim module 900 and the Raspberry PI 4 Kit connects

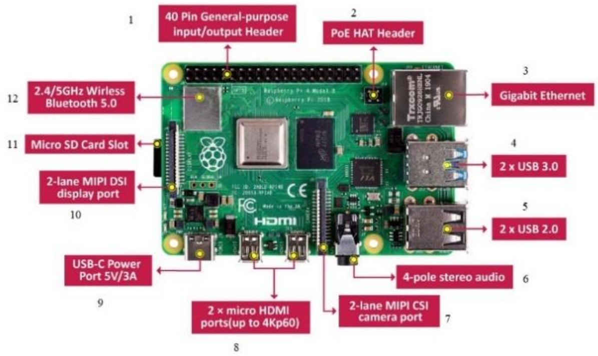

Fig. 3. Raspberry Pi4 kit

Logitech's HD C270 camera module has the function of recording information about images and videos to send signals back to raspberry Pi4 processing (Fig. 4).

Fig. 4. HD Webcam Logitech C270

D. Flowchart of the algorithm of identification and warning system

After initializing the system, the data will be read from the camera, after obtaining the data, we compare it with the characteristics of the data generated from the previous train process, if detected the person will turn on the notification to the alarm, signal to the driver or the people in charge that there are still people in the car. When the alarm goes off, the driver or manager must go to the end of the car to press the button to turn off the bell. This process will ensure the accurate detection of the person in the car. In the process of ringing the bell without pressing the off button, after a period of preset, the system will send a message or call to the preset contact number or send an email message as Fig. 5.

First, we'll set up a camera to track the subject. To find out if there are people or not, we use the technique of marginal detection and image partition. Once there is a marginal detection and image partition, we can detect the neglected person and alert the driver or manager in time, as shown in Fig. 6.

E. Building the python identification and warning system

In this paper, the author uses the Python language based on a built-in neural network based on the Mobile Net network architecture and the Open CV library to accurately identify objects. Algorithm of neuron network programming steps on python is shown in Fig. 7.

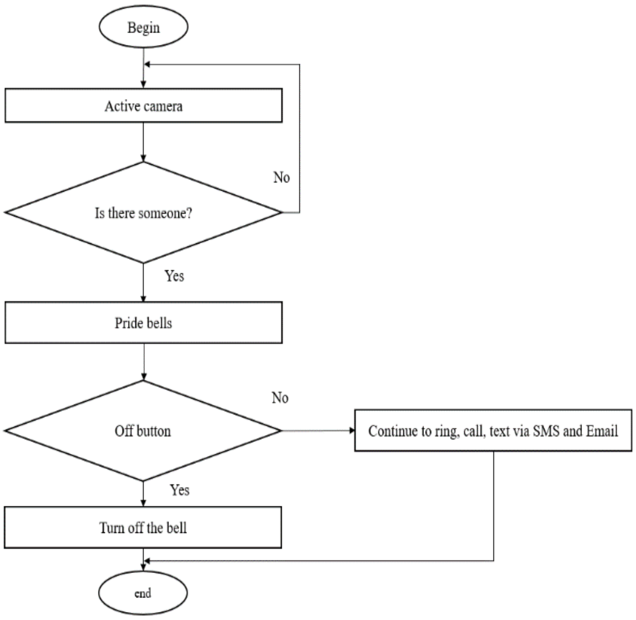

Fig. 5. System algorithm flowchart

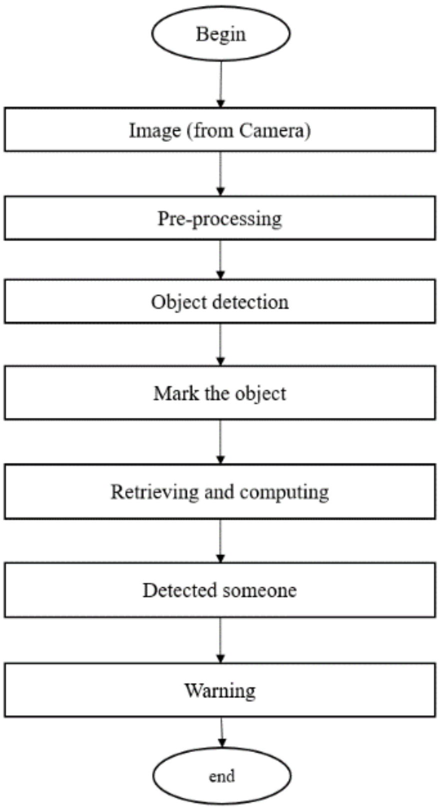

Fig. 6. Python program flowchart

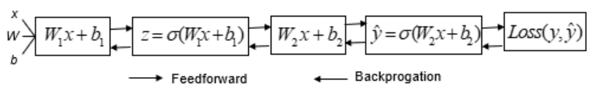

Fig. 7. CNN's calculation process

Step 1: Determine the output value: output layer, $\hat{y}$, It's called Feedforward

$$\hat{y} = \sigma(W_2\sigma(W_1x + b_1) + b_2) \tag{1}$$

Step 2: Update the weights and decision thresholds (blockers), called Backpropagation. There are many ways to help us identify the loss function, which depends on our choice.

$$Sum\ of\ Square\ Error\ = \sum_{i=1}^{n}(y - \hat{y})^2 \tag{2}$$

To adjust W and b, we need the derivative relationship of the loss function with W and b. However, the equation of the loss function does not contain these two coefficients, so we need to build a formula to calculate with weights.

$$\begin{gathered} Loss(y,\hat{y}) = \sum_{i=1}^{n}(y - \hat{y})^2 \\ \frac{\partial Loss(y,\hat{y})}{\partial W} = \frac{\partial Loss(y,\hat{y})}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial z} * \frac{\partial z}{\partial W} \end{gathered} \tag{3}$$

Where

$$\begin{aligned} z = Wx + b = 2(y - \hat{y}) \\ &* derivative\ of\ sigmoid\ function * x \\ &= 2(y - \hat{y}) * z(1 - z) * x \end{aligned}$$

Convolutional Neural Network (CNN) is one of the advanced Deep Learning models [23]. It helps us build smart systems with high accuracies like today such as large image processing systems such as Facebook, Google, or Amazon have introduced into our products smart functions such as facial recognition, Develop self-driving cars or autonomous delivery drones. CNN is used in many of the problems of identifying objects in images [24][25].

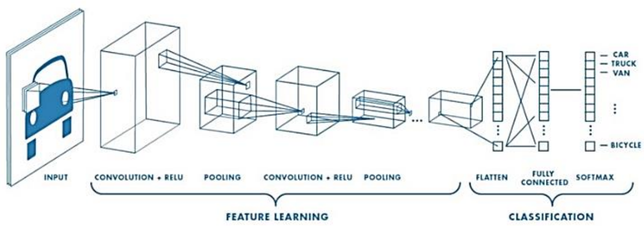

Fig. 8. Integrated neural network structure [16]

CNN's basic structure consists of convolution layers, nonlinear layers, and pooling layers like Fig. 8. The built-in layers combined with nonlinear layers use nonlinear functions such as ReLU or tanh to generate more abstract/higher-level information for subsequent classes.

MobileNet Architecture [16]

In this study, the model uses a convolution method named DSC (Depthwise Separable Convolution) [26] to reduce the model size and reduce computational complexity.

In MobileNet [27], there are 2 classes of Conv used: separableConv and DepthwiseConv. Instead of convolution, as usual, separableConv will perform depth-of-space convolution followed by pointwise Convolution. DepthwiseConv will implement Depthwise Separable Convolution (no directional convolution). This convolution greatly reduces the network volume and number of parameters. With this change, MobileNet can work smoothly even on hardware with a low configuration [28][29][30].

Depthwise separable convolution is a depthwise convolution followed by a pointwise convolution (Fig. 9) [31][32]. Depthwise convolution is a channel-wise $D_K \times D_K$ spatial convolution and Pointwise convolution is simply the convolution measuring 1×1.

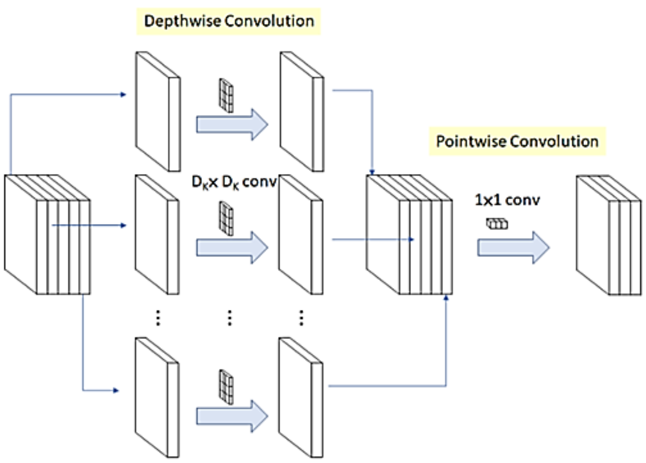

Fig. 9. Structure of a Depthwise Separable Convolution

With M being the number of input channels, N is the number of output channels, D_K is Kernel size, and D_F is a feature map size, we can calculate:

The calculated cost of depthwise convolution is:

$$D_K \times D_K \times M \times D_F \times D_F \quad (4)$$

The calculated cost of Pointwise convolution is:

$$M \times N \times D_F \times D_F \quad (5)$$

The total computing cost of Depthwise separable convolution is:

$$D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F \quad (6)$$

If we do not use Depthwise separable convolution but use convolution as usual, the calculation cost is:

$$D_K \times D_K \times M \times N \times D_F \times D_F \quad (7)$$

Therefore, the cost of calculation will decrease:

$$\frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_K^2} \quad (8)$$

MobileNet selects Kernel size $D_K = 3$, we will reduce the multiplication by 8 to 9 times thereby reducing the cost of calculation by a lot.

Depending on the number and order of layer arrangements in the CNN network, we can get different network architectures. The MobileNet network structure is built on a split in-depth (as in Table I).

The first class is the standard built-in class (used only once). After the standard initial initials, the standard head is a layer of depth-splitting. All the positive layers are followed by the ReLU activation function.

The next layers use a depth-splitting split: a layer of 3×3 in-depth and a point-by-point layer. Layers that accumulate in-depth and accumulate 1×1 by point are arranged alternately. Such a split of the calculus greatly reduces the computing volume and number of parameters of the network.

TABLE I. STRUCTURE OF LAYERS IN A MOBILENET NETWORK

Type/Stride	Filter Shape	Input Size
Conv / s2	3×3×3×32	224×224×3
Conv dw / s1	3×3×32 dw	112×112×32
Conv / s1	1×1×32×64	112×112×32
Conv dw / s2	3×3×64 dw	112×112×64
Conv / s1	1×1×64×128	56×56×64
Conv dw / s1	3×3×128 dw	56×56×128
Conv / s1	1×1×128×128	56×56×128
Conv dw / s2	3×3×128 dw	56×56×128
Conv / s1	1×1×128×256	28×28×128
Conv dw / s1	3×3×256 dw	28×28×256
Conv / s1	1×1×256×256	28×28×256
Conv dw / s2	3×3×256 dw	28×28×256
Conv / s1	1×1× 256 × 512	14×14×256
Conv dw / s1 5 × Conv / s1	3×3×512 dw 1×1×512×512	14×14×512 14×14×512
Conv dw / s2	3× 3 × 512 dw	14×14×512
Conv / s1	1×1×512×1024	7×7×512
Conv dw / s2	3×3×1024 dw	7×7×1024
Conv / s1	1×1×1024×1024	7×7×1024
Avg Pool / s1	Pool 7×7	7×7×1024
FC / s1	1024×1000	1×1×1024
Softmax / s1	Classifier	1×1×1000

From the data collected the objects will be labeled, configured, trained, and exported as sample files to apply to detection and identification (Fig. 10).

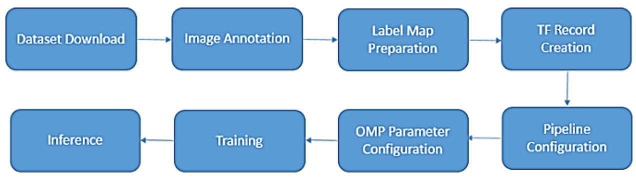

Fig. 10. Process of building object detection and recognition system

Image data is used to extract featured content, then through deep learning creates datasets. The process of training creates datasets through the popular API TensorFlow API [17][33]. In addition, the OpenCV powerful image processing library supports image processing [34]. With a built-in hardware and software system, we collect sample data in classrooms with different postures of pupils. From the images collected, we categorize them into two parts: test (20%) and train (80%) to perform the labeling of the object.

To perform "Image Annotation," the author uses the LabelImg tool to label objects (As Fig. 12). The number of photos used for labeling is 160. The TF Record file is a file that contains input data for Tensorflow (Fig. 13). Depending on the purpose of use, we configure with different parameters (Fig. 14). After we have set the parameters, we conduct training with the TensorFlow API set powered by GPU (Fig. 15). After running the training successfully (about 4 hours depending on the computer configuration), we extract the .pb file. This file is a model for us to apply to the identification. (Fig. 16).

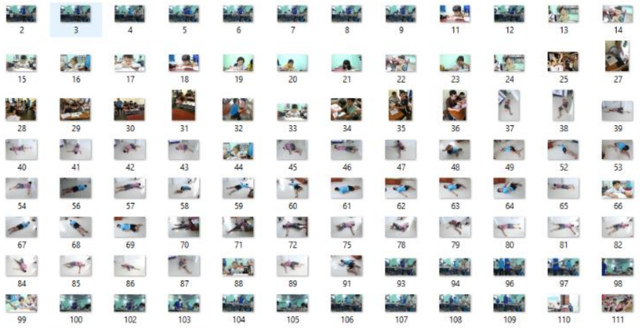

Fig. 11. Data sets used to train user identification

Fig. 12. Use LabelImg labeling for objects as Person

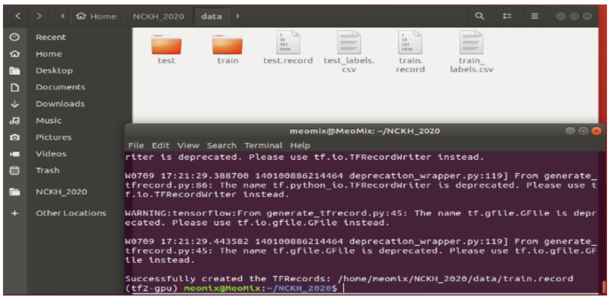

Fig. 13. File TFRecord

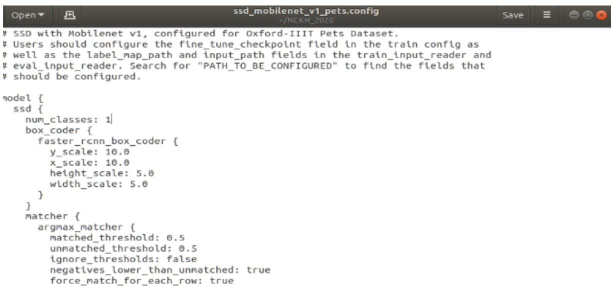

Fig. 14. Pipeline file configuration

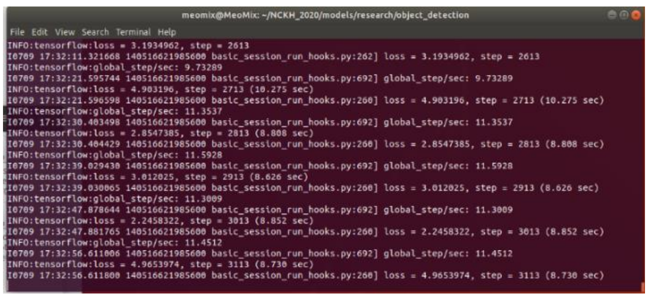

Fig. 15. Conduct training

Fig. 16. Export sample files for testing

The entire model is trained on AMD Ryzen 5 1600x CPU configuration, 16GB Ram, Nvidia Gefore GTX 1050 GPU when trained can handle 0.003s/image on GPU and 0.02s/image on CPU.

III. Experimentation and Evaluation

A. *Results of the experimental running of identification and warning system model*

Fig. 17 shows the level of accuracy when testing a random photo taken from the data set, the results are quite accurate, and the absolute success rate is 100% (in the class there are 3 students, detected all 3 students). Fig. 18 shows that the results still detect the full number of people with a photo in the classroom taken by the phone.

Fig. 17. Pupil identification results

Fig. 18. Test results detect pupils in the classroom

The system will email the driver to notify the driver if it detects someone being left inside the vehicle (Fig. 19) or

SMS and call the supervisor based on the phone number already set in the system (Fig. 20)

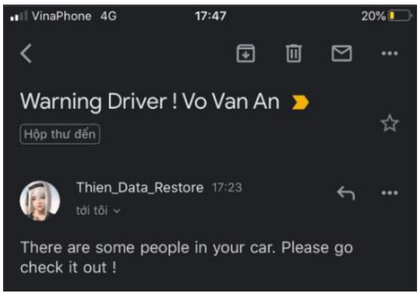

Fig. 19a. Warning to drivers via email

Fig. 19b. Warning to drivers via email

Fig. 20. Warning to the driver over the phone.

B. Evaluation of the results

The experimental results of the entire model worked well, the system of identifying and alerting pupils using Python identification is highly accurate with different camera positions: 100% if the image is captured, about 87.5% with images obtained from fixed cameras (Table II).

TABLE II. THE CAMERA'S ABILITY TO IDENTIFY IN DIFFERENT PLACES

Characteristics of where to spin	**Actual number**	**Counted numbers**	**Reverse identification rate %**
The camera is located at a high altitude, scanning the entire campus in the room	400	400	100%
The camera is located at a high altitude, scanning the entire campus in the car	150	150	100%
The camera is located at a fixed angle	400	350	87.5%
Total/Average	950	900	94.74%

IV. CONCLUSION

In this paper, we developed an algorithm to recognize pupils who apply neural networks and alert managers, testing on the raspberry Pi4 integrated model programmable on Python. This is a highly viable application solution and can be applied to pupil identification and alert models. The above tests show that the proposed architecture and components have met the actual needs, processing time, and alerts fast enough to meet for a real-time request application. Help users closely manage pupils to avoid unfortunate cases occurring. The test results show that the algorithm and computer program developed in the system when the network training factor is added, allowing the recognition system to be more efficient. The system also allows alerts and information to managers to respond in real time with high accuracy. In the future, the research team focuses on perfecting hardware products and application software so that they are easy to use, fully functional and operate reliably.

REFERENCES

[1] V. Wiley and T. Lucas, “Computer Vision and Image Processing: A Paper Review,” International Journal of Artificial Intelligence Research, vol. 2, no. 1, pp. 28-36, 2018.

[2] S. H. Parmar, M. Jajal, and Y. P. Brijbhan, “Drowsy Driver Warning System Using Image Processing,” International Journal of Engineering Development and Research, pp. 78-83, 2013.

[3] H. Takanashi and S. Adachi, “Development of a GUI for a system identification device using Matlab,” IFAC Proceedings Volumes, vol 39, no. 1, 2006, pp. 925-930.

[4] Md. B. Hossain and M. S. Rahman, “A 2D-DCT Image Processing in Matlab and Voice Informatics Based Remote Home Monitoring and Security System,” International Journal of Smart Home, vol 9, no 2, pp. 69-80, 2015.

[5] J. Nagi, S. K. Ahmed, and F. Nagi, “A MATLAB based Face Recognition System using Image Processing and Neural Networks,” 4th International Colloquium on Signal Processing and its Applications, 2008, pp. 83-88.

[6] S. P. Namboodiri and D. Venkataraman, “A computer vision based image processing system for depression detection among students for counseling,” Indonesian Journal of Electrical Engineering and Computer Science, vol. 14, no. 1, pp. 503-512, 2019.

[7] Ashwin T. S, Ram Mohana Reddy Guddeti, Automatic detection of students’ affective states in classroom environment using hybrid convolutional neural networks, Education and Information Technologies, vol 25, 2020, pp. 1387-1415.

[8] R. Wicentowski and T. Newhall, “Using image processing projects to teach CS1 topics,” Association for Computing Machinery, vol. 37, no. 1, pp. 287-291, 2005.

[9] Q. Zhang, J. Xu, L. Xu and H. Guo, “Deep convolutional neural networks for forest fire detection,” in 2016 International Forum on Management, Education and Information Technology Application (IFMEITA 2016), 2016, pp. 568-575.

[10] S. Frizzi, R. Kaabi, M. Bouchouicha, J. Ginoux, E. Moreau and F. Fnaiech, "Convolutional neural network for video fire and smoke detection," *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 877-882.

[11] R. P. Sadewa, B. Irawan and C. Setianingsih, "Fire Detection Using Image Processing Techniques with Convolutional Neural Networks," *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2019, pp. 290-295.

[12] B. Sreenivas and B. Narasimha Chary, “Processing of Satellite Image Using Digital Image Processing,” A world forum on Geospatial, 2011, pp. 18-21.

[13] B. Basavaprasad and M. Ravi, “A Study on the Importance of Image Processing and Its Applications,” International Journal of Research in Engineering and Technology, vol 3, pp. 2321-7308, 2014.

[14] Huynh Phuoc Hai, Comparison of deep learning model with other automation methods in gene expression data classification Microarray, FAIR 2017, DaNang, 2017, pp. 841-850.

[15] Raspberry Pi Trading Ltd, Raspberry Pi4 computer Model B, https://static.raspberrypi.org, 2019.

[16] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Arxiv, 2017.

[17] EdjeElectronics, "Tutorial to set up TensorFlow Object Detection API on the Raspberry Pi," https://github.com/EdjeElectronics/TensorFlow-Object-Detection-on-the-Raspberry-Pi.

[18] D. T. Toan and T. L. Vo, "A driver drowsiness and distraction warning system based on raspberry Pi 3 Kit," Transport and Communications Science Journal, vol. 70, no. 3, pp. 184-192, 2019.

[19] M. Sharif, M. A. Khan, T. Akram, M. Y. Javed, T. Saba, and A. Rehman, "A framework of human detection and action recognition based on uniform segmentation and combination of Euclidean distance and joint entropy-based features selection," EURASIP Journal on Image and Video Processing, vol. 89, 2017.

[20] E. P. Ijjina, D. Chand, S. Gupta, and G. K, Computer vision-based accident detection in traffic surveillance, in Proc. of Int. Conf. Comput. Commun. Netw. Technol., 2019

[21] W. Rahmaniar and A. Hernawan, "Real-Time Human Detection Using Deep Learning on Embedded Platforms: A Review," Journal of Robotics and Control (JRC), vol. 2, no. 6, 2021.

[22] H. I. K. Fathurrahman, A. Ma'arif, and L.-Y. Chin, "The Development of Real-Time Mobile Garbage Detection Using Deep Learning," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 7, no. 3, p. 472, Jan. 2022.

[23] D. Sharma and N. Agrawal, "Development of Modified CNN Algorithm for Agriculture Product: A Research Review," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 8, no. 1, pp. 167–174, May 2022.

[24] S. Aulia and S. Hadiyoso, "Tuberculosis Detection in X-Ray Image Using Deep Learning Approach with VGG-16 Architecture," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 8, no. 2, pp. 290–297, Jul. 2022.

[25] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 24–49, Mar. 2021, doi: 10.1016/J.ISPRSJPRS.2020.12.010.

[26] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1800-1807, doi: 10.1109/CVPR.2017.195.

[27] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, "A novel image classification approach via dense-mobilenet models," *Mobile Information Systems*, vol. 2020, 2020, doi: 10.1155/2020/7602384.

[28] P. N. Srinivasu, J. G. Sivasai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, "Classification of Skin Disease Using Deep Learning Neural Networks with MobileNet V2 and LSTM," *Sensors*, vol. 21, no. 8, p. 2852, Apr. 2021, doi: 10.3390/S21082852.

[29] Y. Li, H. Huang, Q. Xie, L. Yao, and Q. Chen, "Research on a Surface Defect Detection Algorithm Based on MobileNet-SSD," *Applied Sciences*, vol. 8, no. 9, p. 1678, Sep. 2018, doi: 10.3390/APP8091678.

[30] L. Bai, Y. Zhao, and X. Huang, "A CNN Accelerator on FPGA Using Depthwise Separable Convolution," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 10, pp. 1415–1419, Oct. 2018, doi: 10.1109/TCSII.2018.2865896.

[31] Z. Y. Khan and Z. Niu, "CNN with depthwise separable convolutions and combined kernels for rating prediction," *Expert Systems with Applications*, vol. 170, p. 114528, May 2021, doi: 10.1016/J.ESWA.2020.114528.

[32] K. KC, Z. Yin, M. Wu, and Z. Wu, "Depthwise separable convolution architectures for plant disease classification," *Computers and Electronics in Agriculture*, vol. 165, p. 104948, Oct. 2019, doi: 10.1016/J.COMPAG.2019.104948.

[33] B. Pang, E. Nijkamp, and Y. N. Wu, "Deep Learning With TensorFlow: A Review," *Journal of Educational and Behavioral Statistics*, vol. 45, no. 2, pp. 227–248, Sep. 2019, doi: 10.3102/1076998619872761.

[34] J. Sigut, M. Castro, R. Arnay, and M. Sigut, "OpenCV Basics: A Mobile Application to Support the Teaching of Computer Vision Concepts," *IEEE Transactions on Education*, vol. 63, no. 4, pp. 328–335, Nov. 2020, doi: 10.1109/TE.2020.2993013.