Classification of Tri Pramana learning activities in virtual reality environment using convolutional neural network

I Gede Partha Sindu[1,2], Made Sudarma[2], Rukmi Sari Hartati[2], Nyoman Gunantara[2]
[1]Department of Informatics Education, Faculty of Technical and Vocational, Universitas Pendidikan Ganesha, Bali, Indonesia
[2]Faculty of Engineering, Universitas Udayana, Denpasar, Bali, Indonesia

Article Info

Article history:

Received Oct 22, 2023
Revised Feb 10, 2024
Accepted Mar 15, 2024

Keywords:

Batch size
Convolutional neural network
Learning rate
Tri Pramana
Virtual reality

ABSTRACT

Tri Pramana as the local genius of Balinese society, is now adopted in the education system. This adaptation results in a learning cycle model which essentially consists of three classes namely Sabda Pramana (theoretical study), Pratyaksa Pramana (direct observation), and Anumana Pramana (practicum). In learning activities, it is difficult for educators to fully observe individuals to find out the most suitable learning model. Through virtual environment technology, educators can observe students more freely through the recording of students' activities. However, in its implementation, manual analysis requires large resources. Deep learning approach based on convolutional neural network (CNN) is able to automate this analysis process through the classification ability of the image of the recorded learner activity. To produce a robust CNN model, this research compares four of the most commonly used architectures, namely ResNet-50, MobileNetV2, InceptionV3, and Xception. Each architecture is tuned using a combination of learning rate and batch size. Through a 512×512 resolution dataset with 70% training subset (4,541 images), 20% validation (1,296 images), and 10% test (652 images), the best ResNet model is obtained with a learning rate configuration of 1e-3 and batch size 64 with an accuracy of 99.39%, precision of 99.37%, and recall of 99.42%.

Corresponding Author:

I Gede Partha Sindu
Department of Informatics, Universitas Pendidikan Ganesha
Udayana Street No.11 Singaraja Bali 81116, Indonesia
Email: partha.sindu@undiksha.ac.id

1. INTRODUCTION

Learning approach through virtual reality (VR) is an increasingly popular method in the world of education nowadays. VR is a technology that allows users to interact with artificial environments that are immersive and interactive [1], [2]. When it is used in an educational context, VR can provide an immersive and realistic learning experience, which may be difficult to achieve with traditional methods. In a VR environment, users can be fully immersed in an artificial environment that resembles real situations [3]. It creates an immersive and memorable experience for students, which can help them in improving their understanding and retention of information. In line with this, educators must be careful in choosing which approaches and methods are appropriate for the virtual environment. A learning approach is an idea or principle of how to view and determine learning activities. These ideas or principles can be found in the noble values of a local genius in Bali Province, Indonesia called Tri Pramana. The Tri Pramana concept has previously been adopted in learning process activities in Indonesia, especially in schools and universities in Bali Province, which can be used as learning approach [4], [5].

The Tri Pramana learning activities in the previous study are very much in line with the goals and values of modern education. The term Tri Pramana is derived from Sanskrit, where "Tri" means three, and "Pramana" means the basis or means of recognition [6]. Thus, "Tri Pramana learning" refers to the concept of learning based on three principles or tools of recognition. Tri Pramana can be interpreted as a learning approach consisting of three parts, namely Sabda Pramana, Pratyaksa Pramana, and Anumana Pramana [7]. Subagia and Wiratma have developed a learning cycle model called the Tri Pramana learning cycle model [8]. The Tri Pramana learning cycle model emphasizes the implementation of learning in three stages, namely direct observation (Pratyaksa Pramana), reception of information (Sabda Pramana), and analysis of direct observation (Anumana Pramana). In total, there are six types of Tri Pramana learning cycle models used, namely two cycles starting with Pratyaksa Pramana activities, two cycles starting with Sabda Pramana activities, and two other cycles starting from Anumana Pramana activities [9]. The six learning cycles of Tri Pramana are PSA learning cycle model, PAS learning cycle model, SAP learning cycle model, SPA learning cycle model, APS learning cycle model, and ASP learning cycle model. In this case, "S" stands for Sabda which means listening or reading, "P" stands for Pratyaksa which means direct observation, and "A" stands for Anumana which means mind. The Tri Pramana learning cycle model can be used as an alternative to improve the scientific approach applied in the current learning curriculum in Indonesia.

The Tri Pramana learning cycle model is adopted to improve the quality of teaching in higher education, especially at Ganesha University of Education, Bali Province, Indonesia. The application of the Tri Pramana learning cycle incorporates immersive VR technology. In this research, specifically practicum courses that use standard operating procedures that apply the Tri Pramana learning cycle model based on immersive VR technology. The process of observing learning activities with the Tri Pramana learning cycle model so far in the VR environment is still conducted in the form of video recordings. Lecturers only obtained general observations of individual student learning outcomes in the VR environment so they could not determine the activities of the six Tri Pramana learning cycle models used. This certainly makes the lecturers feel difficult to classify student learning activities in the VR environment in accordance with the Tri Pramana learning cycle model. In overcoming these problems, this research applies convolutional neural network (CNN) to classify Tri Pramana learning activities so that it can determine the Tri Pramana learning cycle model for each individual student accurately and automatically. The input data from the CNN classifier is a video of each student's learning activity in the VR environment.

The CNN method is a derivative of deep learning, one of its purposes is to handle image classification cases. With its remarkable achievements in deep learning research, especially CNN, is widely applied to image classification and action recognition. There are several CNN architectures developed by previous research and often used nowadays, namely ResNet-50, MobileNetV2, InceptionV3, and Xception. Research by Li *et al.* [10] related to the use of ResNet-50 architecture is used to get better and more discriminative feature extraction in disposable learning motion. Research by Yuan *et al.* [11] uses the MobileNetV2 architecture for visual image processing to address time efficiency and resource consumption limitations. Research by Taspinar *et al.* [12] uses InceptionV3 architecture for end-to-end classification and feature extraction of dried bean images quickly and accurately. Research by Sharma *et al.* [13] using Xception CNN for the recognition and classification of Windows malware image data shows effective results and low computational costs. This research aims to compare four CNN architectures in the case of image classification so as to find the architecture with the best performance. Deep learning models, especially CNN methods, have been implemented in VR environments. Miller *et al.* [14] discussing problems in identifying and authenticating VR using the CNN method. This research incorporates real-world constraints on user behavior into a learning algorithm for VR behavior-based identification and authentication using deep networks. On average, this study observed an increase of 0.63% to 0.73% for 10-fold, 0.55% to 0.85% for 5-fold, and 0.93% to 2.08% higher identification success observed for 36 out of 42 combinations of user sets and VR system pairs, indicating the promise of methods that explicitly represent spatial relationships in the input.

Zhang [15] determined the landscape design classification of deep neural network (DNN) development environments based on VR. Percentage-wise, through the investigation results obtained, the accuracy of the proposed DNN is 96.7%; higher than other methods such as principal component analysis (PCA) 85.8% and support vector machines (SVM) 91.5%. The use of CNN in VR is also applied by Qin and Qin in their research [16]. Based on DenseNet, an improved shallow layer dense CNN (L-DenseNet) is proposed, which can compress the network parameters and improve the feature extraction ability of the network. The experimental results show that the L-DenseNet method can effectively improve the classification accuracy of image images better. Moreover, it has good application value with the highest accuracy of 95.3%. Meissler *et al.* [17] visualized the CNN method in VR with the aim of giving users an introduction to its function. Bibbò and Morabito [18] used the CNN method to collect data, train and optimize the recognition of human activities in the VR environment. Research by Tran *et al.* [19] proposed a deep learning strategy for finger vein recognition based on CNN anti-aliasing method and double exposure fusion algorithm. Experimental results show that the proposed CNN method outperforms the current method (Densenet-161), improving 97.66% accuracy on the FVUSM data set, 99.94% accuracy on the SDUMLA data set, and 88.19% accuracy on the THUFV2 data set. The strength of the CNN method with these architectural variations is fundamental to classifying Tri Pramana learning activities.

Previous research has mainly focused on the performance of CNN architectures in image classification tasks in virtual environments. However, these studies have not directly addressed the challenge of developing a model capable of observing and monitoring the learning cycle model for individual students. As a result, teachers may experience difficulties in customizing appropriate learning activities for students, due to the lack of an automated system capable of comprehensively assessing and understanding students' learning cycles in virtual environments. This gap underscores the need for research that specifically addresses the development of models capable of monitoring and analyzing students' learning activities in virtual settings, thus allowing educators to better tailor learning experiences to individual needs. This research aims to compare 4 CNN architectures in the case of image classification, thus finding the architecture with the best performance.

2. METHOD

Our approach in this study consists of a series of processes ranging from data acquisition to model performance evaluation. The stages include data preparation which includes data collection, cleaning, pre-processing, and model building which involves architecture selection, training, and model parameter tuning. Afterwards, the model is evaluated using relevant metrics to measure its performance. The visualization of the details of the proposed approach is shown in Figure 1.

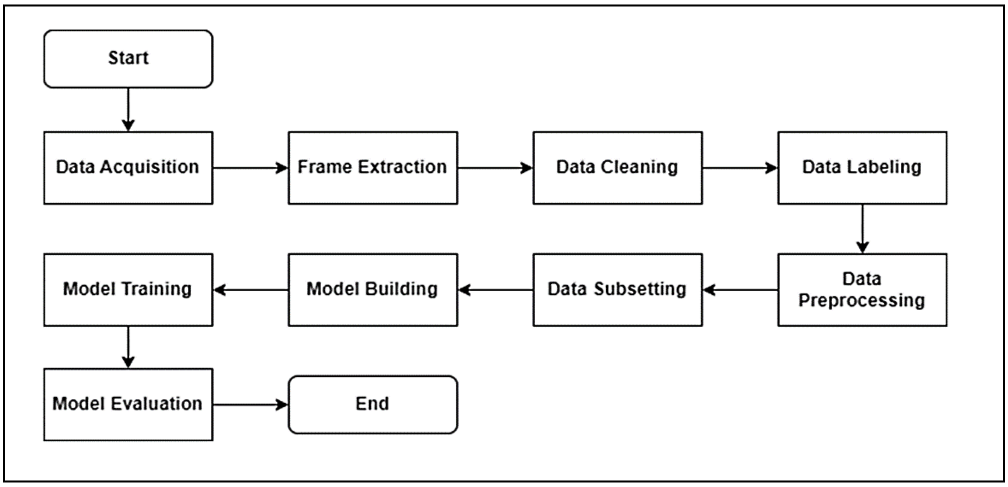


Figure 1. Proposed approach method

Referring to Figure 1, this research begins with data acquisition in the form of recording learners' activities in a VR environment. The recorded video is then extracted into frames and then cleaned by eliminating frames outside of the classified activity class. The next process is labelling the data according to its class, and each frame is resized and segmented to increase the variety and number of samples. The pre-processed dataset is then divided into three subsets: the train set, validation set, and test set. Model building is a step to define model layers for the training process. Once the dataset and model are ready, we proceed with the training process that leads to the evaluation model using the test set data. The environment used is Python-based with detailed specifications of Intel Core i5 Processor, 8 GB Nvidia RTX 3050 GPU, 1 TB SSD, and 32 GB RAM. The details of the dataset in the proposed approach are presented in the following sub-sections.

2.1. Data acquisition and frame extraction

This research utilizes a self-acquired dataset by recording learners' activities in the Tri Pramana learning in VR environment. The VR device used in the acquisition process was oculus quest 2. Oculus quest 2 was connected to Unity App to build the virtual classroom environment. The documentation of the data acquisition process is shown in Figure 2.

Figure 2. Documentation of the acquisition process

This acquisition process resulted in five videos. All videos were in MP4 format. Each video is RGB channelled with a frame count of 25 in one second. Each 1024×1024 resolution video was extracted by taking only 5 frames out of a total of 25 frames in one second. This aimed to obtain a significant sequence of frames. The specifications of the data acquisition and frame extraction results are shown in Table 1.

Table 1. Specifications of the results of data acquisition and frame extraction

Video	Duration	Total Extracted Frame
Video1.mp4	07:48	2125
Video2.mp4	05:43	1622
Video3.mp4	06:03	1726
Video4.mp4	02:13	625
Video5.mp4	02:36	645

Referring to Table 1, the total frames generated amounted to 6,743 frames. The details of the frames from each class are: i) Sabda Pramana class amounted to 884 frames, ii) Pratyaksa Pramana class amounted to 1,011 frames; and iii) Anumana Pramana class amounted to 1,005 frames. This data provides a clear picture of the distribution of frames in the dataset used.

2.2. Data cleaning and data labelling

The extracted frames were then filtered to eliminate frames that do not contain activities from the three classes. The data cleaning process was done manually by dropping frames that are considered noise or outliers. After going through data cleaning, it is labelled based on the directory of the frame it was stored in, resulting in 3 folders representing 3 classes. The detailed results of data cleaning and data labelling are shown in Table 2.

Table 2. Result of data cleaning and data labelling

Class	Frame total
Sabda Pramana	654
Pratyaksa Pramana	756
Anumana Pramana	753

2.3. Data preprocessing and data subsetting

Before the dataset was ready to be trained, pre-processing and data sub setting were required to match the data specifications with the layers of the model that had been defined. In this case, all frames that originally had a resolution of 1024×1024 were resized to 256×256. The purpose of reducing the resolution was to streamline the processing so that the training and inference time became faster. After the image was resized, it was then augmented to add variety to the data. The augmentation process used the albumentations package with two types of augmentation, namely flip horizontal and rotate. Flip horizontal with probability=1 parameter indicated that flip horizontal augmentation was applied to the entire image. The same also applied to rotate with a probability value=1 with a degree value between -15° to -15°. The rotate process applied a border mode parameter of 0 which states that the extrapolation uses constant pixels of value 0 (black). The total image after undergoing augmentation amounted to 6,489 frames. The augmented data was then divided into three subsets with different allocations. The three subsets in this dataset had a composition of train 70% with a total of 4,541 images, validation 20% with a total of 1,296 images, and test 10% with a total of 652 images. The visualization of the final dataset in this study is shown in Figure 3.

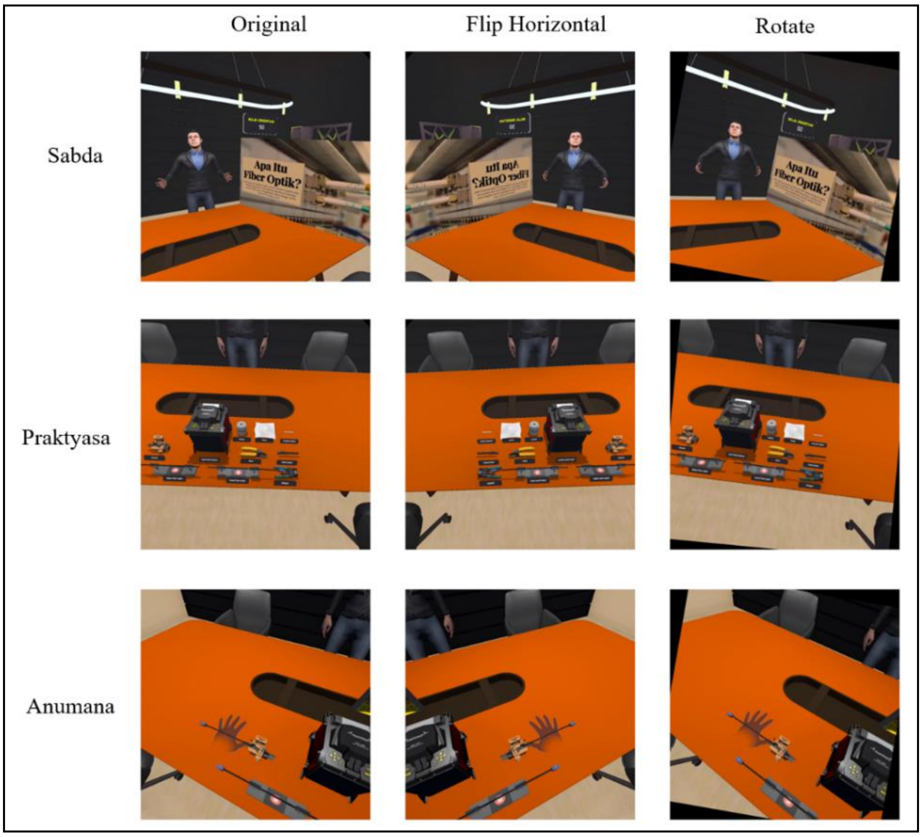


Figure 3. Research dataset visualization

2.4. Convolutional neural network architecture ResNet-50

ResNet stands for residual network, a well-known structure in the context of image classification and recognition. Developed by a Microsoft research team led by Kaiming, this architecture aims to create models with higher accuracy. ResNet consists of more and deeper layers than the visual geometry group (VGG) network, but still has a lower level of complexity. The main innovation of this architecture is the introduction of a residual learning framework that is able to overcome the performance degradation problem that often occurs as the layer depth increases [20]. The layer view of this architecture is shown in Figure 4, and the residual layer is shown in Figure 5.

Based on Figure 4, it can be seen that the ResNet-50 architecture has a total of 50 layers consisting of one 7×7 convolution layer, one max poling layer with stride=2, nine convolution layers with output size 56×56, twelve convolution layers with output size 28×28, eighteen convolution layers with output layer 14×14, and nine convolution layers with output size 7×7. The total of 50 layers boils down to average pooling and softmax activation functions to provide classification probabilities in 3 classes, namely Sabda Pramana, Pratyaksa Pramana, and Anumana Pramana. Referring to Figure 5, increasing the depth of the layer can be an advantage because the features generated in the previous layer have shortcut connections that result in the merging of the features F(x) and x [20]. This led to the development of deeper CNN architectures such as ResNet-50, which effectively handles the deep training problem by applying residual learning blocks. By deepening the architecture and applying the residual concept, ResNet-50 is able to maintain good performance even with the addition of deeper layers. The equation for the residual learning block is shown (1).

$$y = \mathcal{F}(x, \{W_i\}) + W_s x \tag{1}$$

In (1) illustrates the process in the residual learning block in the CNN architecture as used in ResNet-50. The variable y represents the output layer, while x refers to the previous input layer. The residual mapping $\mathcal{F}(x, \{W_i\})$ describes the changes that the input will undergo in the training process, where W_i are the parameters involved in this transformation. In addition, the shortcut connection W_s allows the original information from input x to be passed directly to the output layer, helping to speed up and strengthen the learning process [21].

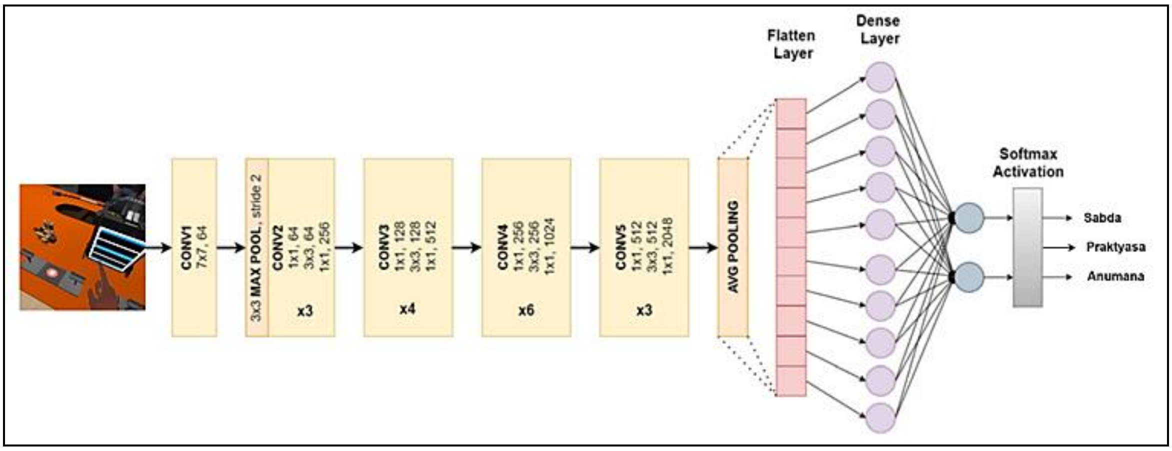


Figure 4. ResNet-50 architecture design for multiclass classification

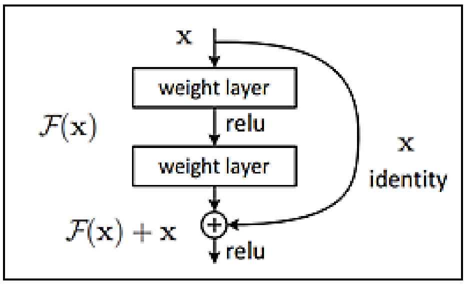


Figure 5. Residual learning block

2.5. Convolutional neural network architecture MobileNetV2

This architecture was proposed by Sandler *et al.* [22] from team Google to present the simplest neural network architecture [23]. Besides focusing on portability, the development of MobileNetV2 aims to address the vanishing gradient problem that often occurs in previous generations. MobileNetV2 comes with linear bottlenecks feature that addresses information decay on non-linear layers in the convolution block. Another feature that plays a role in preserving the extracted information is the inverted residual. The block layers of MobileNetV2 are shown in Figure 6.

Referring to Figure 6, there are two main building blocks of this architecture. The blocks with stride 2 and stride 1 go hand in hand (interrelated) to achieve convergence more easily despite the low complexity. The block with stride 2 focuses on reducing the feature size, while the block with stride 1 focuses more on handling the features to achieve convergence so that the input and output sizes of these layer blocks are equal [23].

2.6. Convolutional neural network architecture InceptionV3

The InceptionV3 architecture is a further development of InceptionV2 with some significant optimizations. One of the main optimizations is the use of convolution factorization to reduce the filter size from 5×5 to two 3×3 convolutions. In addition, InceptionV3 also makes use of convolution asymmetry which allows convolution filters with unsymmetrical matrix row and column sizes to be implemented [24]. The visualization of the InceptionV3 architecture is shown in Figure 7.

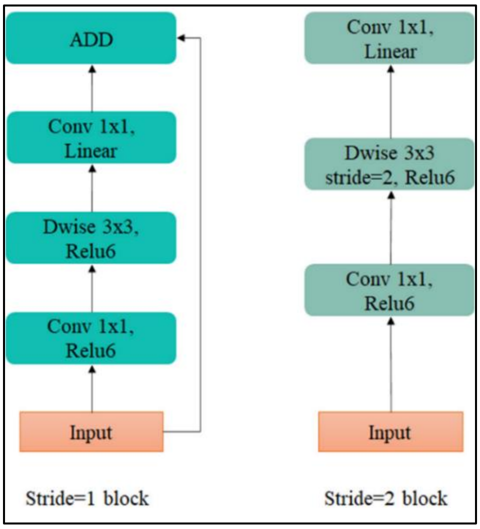


Figure 6. Building block layer of MobileNetV2

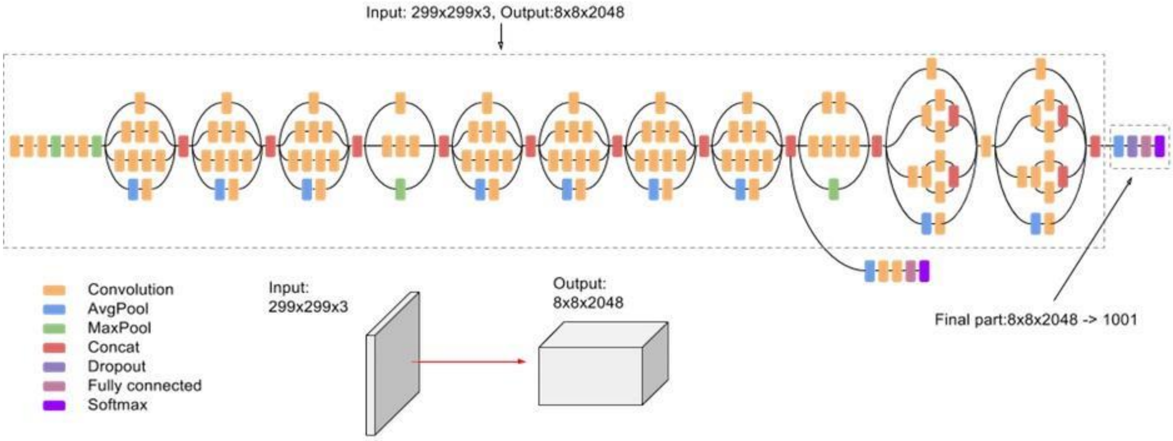


Figure 7. Visualization of the InceptionV3 architecture [25]

2.7. Convolutional neural network architecture xception

A follow-up to the InceptionV3 architecture is this one that Google has proposed, which is Xception. It stands for extreme inception introduced the term depthwise separable convolutions. Unlike conventional convolution, the convolution process is reduced by applying a 1×1 convolution performed on isolated image channels [26]. The visualization of the Xception architecture is shown in Figure 8.

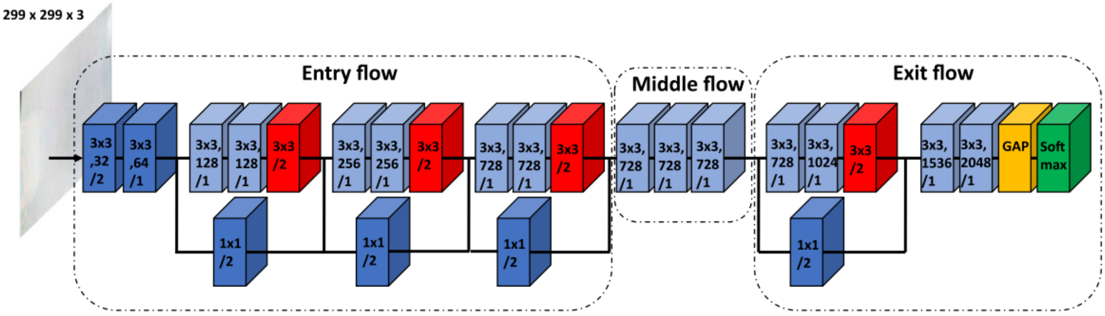


Figure 8. Visualization of Xception architecture [27]

Each CNN architecture discussed has its own advantages and features, including computational efficiency, recognition accuracy, and adaptability to various image classification tasks. Through a series of careful experiments and tests, the key parameters of each architecture have been analyzed in detail to provide a comprehensive overview of the characteristics and performance of each model. This detailed information is presented in Table 3.

Table 3. Specification of each architecture's parameters

CNN architecture	Input size	Parameters
ResNet-50	256×256×3	24,638,339
MobileNetV2	256×256×3	2,261,827
InceptionV3	256×256×3	21,808,931
Xception	256×256×3	20,867,627

2.8. Hyperparameter tuning

Before the process entered the training phase, it was necessary to define the hyperparameter values to achieve convergence level. Hyperparameters greatly affect the complexity of model calculations during training, so care is needed in defining them [28]. Optimizer is a method used to update the weights of each iteration so that the model converges. In this research, the optimizer algorithm used was adaptive momentum estimation (Adam). Adam was proposed to overcome two predecessor algorithms, namely AdaGrad and RMSProp. Adam has a tendency to excel in computation time and specification requirements, so it is suitable for training on large architectures [29], [30]. In the tensorflow framework, Adam has a configuration of β_1=0.9 and β_2=0.999. β_1 is the first moment exponential decay rate and β_2 the second moment exponential decay rate.

Learning rate is one of the hyperparameters that most determines the performance of model training. The learning rate controls the rate at which the training converges. As of now, determining the learning rate has been of particular interest because its treatment depends on the architecture, other hyperparameter components, and the dataset. In general, researchers conduct a grid search to find the appropriate learning rate [31]. The learning rate used in this research consists of 1e-3, 1e-4, 1e-5, 1e-6. In the case of learning activity classification, the comparison of four learning rate scales aims to obtain the most appropriate learning rate and batch size combination configuration.

Determining the number of samples learned in one iteration greatly affects the weighting in model training. The number of samples is often referred to as the batch size. Technically, the greater the number of samples learned in one iteration, the more representative the weights updated in that iteration. Researchers agreed that the range of batch sizes used is between 64 and 512. The variation in the number of batch sizes is generally a multiple of 2 [32]. In this research, the batch variations used were 32, 64, 128, and 256. It should be noted that the larger the batch size, the larger the memory specifications on the training device.

Complex classification patterns are getting away from linearity. In deep learning, the focus of classification refers to non-linear learning patterns. Input data in linear form requires a method that transforms the input linear values into non-linear ones. This task was performed by the activation function by calculating the input value into a non-linear form, so that it matches the feed required in the next neuron or classifier. This research used rectified linear unit (ReLU) activation and Softmax activation functions. The ReLU activation function was used in the convolution layer because of its speed in calculation. This activation function will return its own value if it is positive and return the value 0 if it is negative so that it is free from vanishing gradient conditions [33]. The ReLu activation function equation is shown in (2).

$$f'(x) = \{ \begin{smallmatrix} 1 \; for \; x \geq 0 \\ 0 \; for \; x < 0 \end{smallmatrix} \tag{2}$$

The Softmax activation function is used in the output layer to calculate the probability distribution in the case of multiclass classification. Compared to ReLU, Softmax is specifically designed to produce outputs in the form of probability distributions, allowing the model to provide confident predictions for each possible class. Therefore, Softmax is suitable in the context of multiclass classification to interpret the model results more clearly. The equation for this activation function is shown in (3).

$$f(x_i) = \frac{\exp(x_i)}{\sum_j^K \exp(x_j)} \tag{3}$$

Based on equation (x), exp (x_i) is the standard exponential function for the input vector and exp (x_j) the standard exponential function for the output vector. K represents the number of classes that are classified [34]. Loss function is a method used to calculate model performance by comparing the actual value with the predicted value. This research uses categorical cross entropy (CCE) loss function in measuring training performance. In the training process, CCE weights the minority class so that it can perform well even on unpredictable samples [35]. The equation of CCE is shown in (4).

$$CCE = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{K} y_{i,j}\log(p_{i,j}) \tag{4}$$

Variable n contains the total sample, K indicates the number of classes, $y_{i,j}$ is a vector of actual values and $p_{i,j}$ is the probability of the model prediction. All architectures including ResNet-50, MobileNetV2, InceptionV3, and Xception were trained using 16 hyperparameter variations to get the best performing model. The details of the hyperparameters used in all training variations are shown in Table 4.

Table 4. Details of hyperparameters used in all training variations

Hyperparameters	Value
Epochs	200
Optimizer	Adam
Learning rate	(1e-3, 1e-4, 1e-5, 1e-6)
Batch size	(32, 64, 128, 256)
Loss function	Categorical cross entropy
Metric	Accuracy

Based on Table 4, the variation is emphasized on the application of learning rate and batch size. The learning rate hyperparameter greatly affects the movement of the derived equation in achieving global minimum convergence. The batch size variation concerns the number of samples learned in one training iteration. The number of samples affects the representation rate of each weight update. The match between learning rate and batch becomes a variation that is explored using the grid search method. This method is an experiment to obtain the optimum hyperparameter configuration by exploring the hyperparameter pair completely [36].

2.9. Metrics evaluation

To test the performance of the built prediction model, we introduce four evaluation metrics derived from the confusion matrix. The confusion matrix consists of the prediction class and the true class. Its components consist of true positive and true negative which reflect the true class, and false positive and false negative which represent false positive classification and false negative classification. Furthermore, the details of the four evaluation metrics used are as follows: i) accuracy, which calculates the ratio of correct predictions (true positive and false negative) to total predictions, ii) precision, which evaluates the positive rate by dividing the correct positive predictions by the sum of true positive and false negative, and iii) recall, which evaluates the correctly identified positive patterns [37].

The selection of evaluation metrics such as accuracy, precision, and recall is important to provide a comprehensive understanding of the model's capabilities in the context of multiclass classification [38]. By taking into account the results of the model's inference calculations on each predefined class, these metrics provide a deep insight into how well the model can classify the data into the right class. This allows for a more holistic evaluation of the model's performance in dealing with the different complexities and distributions of data across classes. The accuracy, precision, and recall metric equations are given in (5) to (7), respectively.

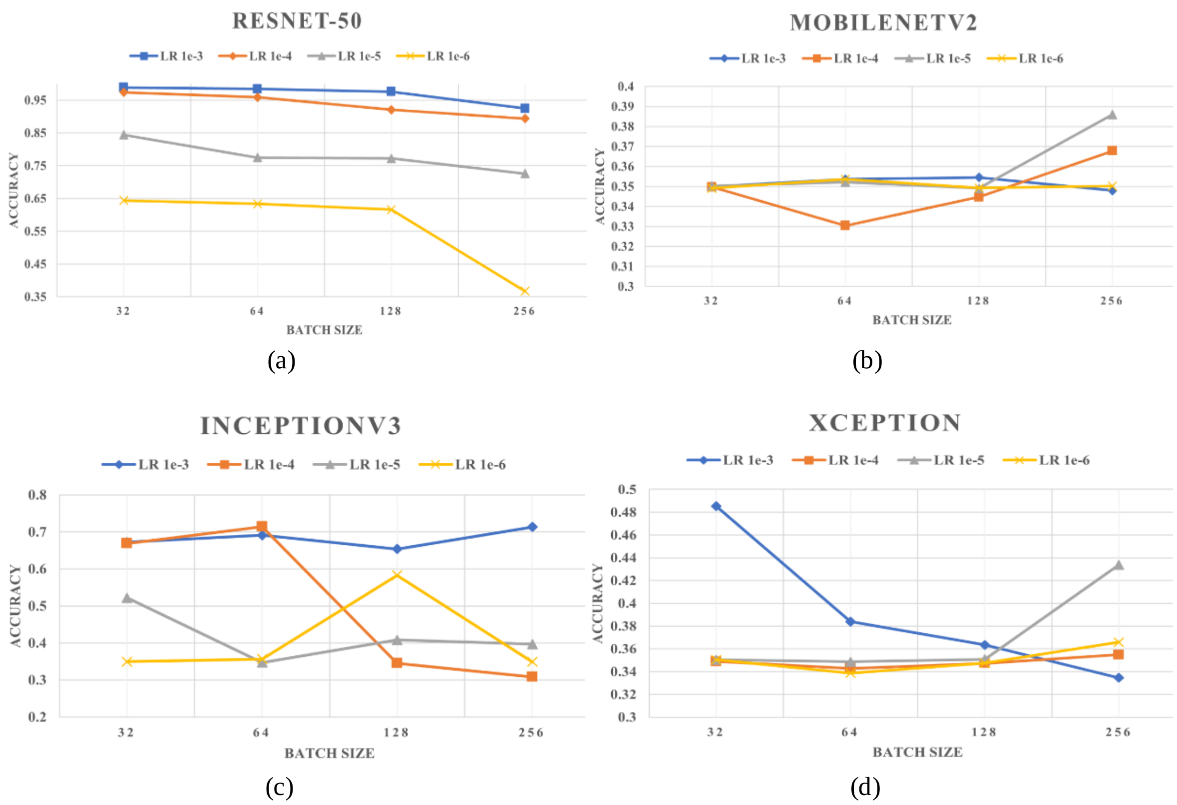
$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

3. RESULTS AND DISCUSSION

This section contains a comprehensive discussion of the experiments and findings. As mentioned earlier, there is a gap between previous research that only performs classification in a virtual environment without classifying the learner learning cycle model. Through the grid search approach, the gap between each architecture is compared with the same treatment, especially emphasizing hyperparameter tuning on the combination of learning rate and batch size. The training results on ResNet-50, MobileNetV2, InceptionV3, and Xception architectures are shown in Figures 9(a) to 9(d).



(a) (b)

(c) (d)

Figure 9. Training results of: (a) ResNet-50 architecture, (b) MobileNetV2 architecture, (c) InceptionV3 architecture, and (d) Xception architecture

Referring to Figures 9(a) to 9(d), the highest accuracy value of 0.98 during training is obtained in the ResNet-50 architecture with a learning rate variation of 1e-3 and a batch size of 32. While the lowest accuracy of 0.31 was obtained by InceptionV3 architecture with a learning rate variation of 1e-4 and batch size of 256. These results were obtained from training from scratch without the addition of a pretrained model. Model performance was also analyzed based on its ability to classify on new data that has never been learned. This process is called model evaluation by measuring the generalization ability of the model to the data in the test subset. Each architecture produces 16 models consisting of a combination of learning rates 1e-3, 1e-4, 1e-5, 1e-6 with a combination of batch sizes of 32, 64, 128, 256. The evaluation results for each combination are shown in Tables 5 to 8.

Referring to Tables 5 to 8, the best performance is generated by the model with ResNet-50 architecture with a combination of learning rate 1e-3 and batch size of 64. The second position is followed by InceptionV3 architecture with learning rate 1e-4 and batch size of 64. The third position is obtained by Xception with a learning rate configuration of 1e-5 and batch size of 32. While the last position is generated from MobileNetV2 architecture with a variation of learning rate 1e-3 and batch size of 32.

Table 5. ResNet-50 architecture evaluation results

LR	ResNet-50 Accuracy			
	BS 32	BS 64	BS 128	BS 256
1e-3	99.08	99.39	97.85	95.55
1e-4	97.39	98.47	91.87	89.11
1e-5	84.66	80.98	81.11	79.91
1e-6	64.26	63.50	60.74	34.97

Table 6. MobileNetV2 architecture evaluation results

LR	MobileNetV2 Accuracy			
	BS 32	BS 64	BS 128	BS 256
1e-3	34.97	33.33	34.97	34.97
1e-4	34.97	34.82	34.97	34.97
1e-5	34.97	34.97	34.97	34.97
1e-6	34.97	34.97	34.82	34.82

Table 7. InceptionV3 architecture evaluation results

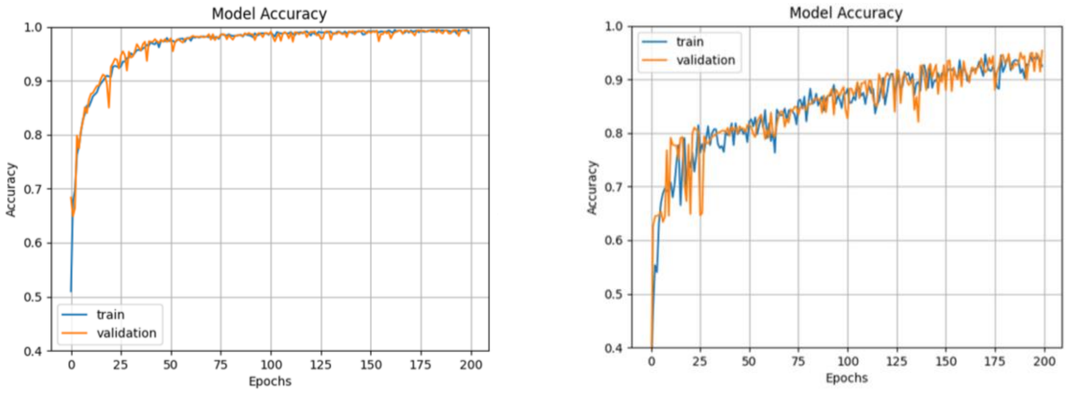
LR	InceptionV3 Accuracy			
	BS 32	BS 64	BS 128	BS 256
1e-3	66.72	67.18	57.21	65.18
1e-4	65.95	70.40	34.97	34.82
1e-5	50.31	34.97	38.96	45.40
1e-6	34.97	34.82	56.90	34.82

Table 8. Xception architecture evaluation results

LR	Xception Accuracy			
	BS 32	BS 64	BS 128	BS 256
1e-3	47.09	49.23	36.04	34.97
1e-4	34.97	34.97	34.97	34.97
1e-5	50.31	34.97	34.97	47.92
1e-6	34.97	34.82	34.82	34.97

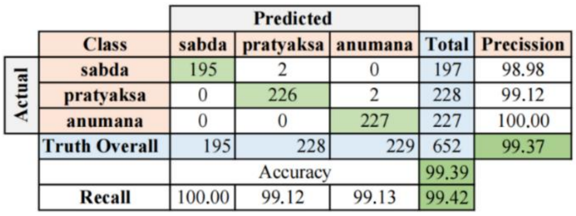
The learning rate hyperparameter with an extremely small category of 1e-6 makes it difficult for the model to achieve the convergent level on this dataset. While the learning rate of 1e-3 in each architecture tends to converge faster when compared to smaller learning rates such as 1e-5 or 1e-6. This is because the task of the learning rate is to determine the step size of the weight update [39]. In addition to the learning rate, the use of batch size also has a significant effect. The batch sizes that provide the best accuracy are batch sizes 32 and 64 [40]. This is in line with the best training result obtained by ResNet-50 which is 0.98 at a learning rate of 1e-3 with a batch of 32, and the best testing result is 99.39 with a batch of 64. The batch size analysis of the training process is shown in Figure 10.

Referring to Figure 10, batch size 256 in Figure 10(b) gives fluctuating accuracy at each epoch due to the varying number of samples. Meanwhile, in Figure 10(a), the accuracy weighting of each epoch looks smooth. Although a larger batch size speeds up the training process, the fluctuating weight updates can reduce the performance of the model [41], [42]. An in-depth analysis was conducted on the model with the highest accuracy. Through the test set data totaling 652 images, the ResNet-50 model with a learning rate variation of 1e-3 and a batch size of 64 is able to provide an accuracy of 99.39, precision 99.37, and recall 99.42. The confusion matrix on the results of this model evaluation is shown in Figure 11.



(a) (b)

Figure 10. Batch size analysis in the training process: (a) stable accuracy and (b) fluctuating accuracy



		Predicted				
	Class	sabda	pratyaksa	anumana	Total	Precission
Actual	sabda	195	2	0	197	98.98
	pratyaksa	0	226	2	228	99.12
	anumana	0	0	227	227	100.00
	Truth Overall	195	228	229	652	99.37
		Accuracy			99.39	
	Recall	100.00	99.12	99.13	99.42	

Figure 11. Confusion matrix on model evaluation results

Referring to Figure 11, the test data in the Sabda Pramana class has two false positives. The two data were classified as Pratyaksa Pramana. After identifying the two false positive data points in the Sabda Pramana class, further analysis was conducted to understand the characteristics leading to misclassification. A visualization of the false positive data for the Pratyaksa Pramana class is shown in Figure 12.

Referring to Figure 12, it can be seen that students mostly direct their gaze to the table, which gives the model perspective that students are doing direct observation or Pratyaksa Pramana class. During the Sabda Pramana activity, there are several buttons to answer the quiz on the table. This gives perspective model that the learners do direct observation or Pratyaksa Pramana class. The next analysis is on the Pratyaksa Pramana class data which also gives two false positives whose input is considered as Anumana Pramana class or practicum. The visualization of data that experienced false positive is shown in Figure 13. As shown in Figure 13, the direct observation activities in the Pratyaksa Pramana class and the practicum activities in the Anumana Pramana class are similar. The Anumana Pramana class has a tendency for learners to hold and operate tools, so when viewed in Figure 13, there are similarities between direct observation data (Pratyaksa Pramana class) and practical data (Anumana Pramana class).

Through hyperparameter tuning, 64 models were generated. The four models that were compared each implemented 16 different learning rate and batch patterns. This certainly has a significant impact in competing each model with the same hyperparameters. The author explored all learning rates and batch sizes to come up with the best model and hyperparameter conclusion for this problem [43]–[45].

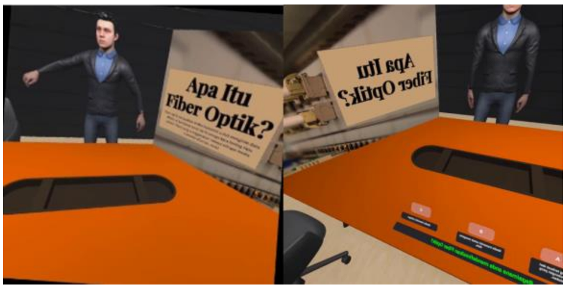


Figure 12. Data visualization of Sabda Pramana class that was misclassified as Pratyaksa Pramana class

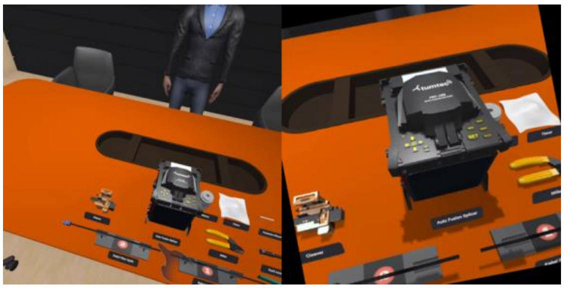


Figure 13. Data visualization of Pratyaksa Pramana class that was misclassified as Anumana Pramana class

4. CONCLUSION

This research aims to automate the classification process of image-based learning activities in virtual environments using CNN. The dataset in this research was acquired independently in PNG format with 512×512 resolution. The dataset consisted of training set 70% (4,541 images), validation 20% (1,296 images), and test 10% (652 images). Through CNN approach, this research compared ResNet-50, MobileNetV2, InceptionV3, and Xception architectures to obtain the best model. Hyperparameter tuning was obtained through grid search method with variable combination of learning rate and batch size. Through 200 epochs of training, we found that the best model of ResNet-50 architecture was obtained with a learning rate configuration of 1e-3 and batch size 64. The use of a large learning rate of 1e-3 was able to achieve convergent level faster than smaller learning rates such as 1e-5 or 1e-6. The use of a batch size that was excessively large provided fluctuating weight updates, so the performance was degraded. In this research, the recommended batch sizes were 32 and 64. Further future work on this research is to add datasets from different virtual environments and deployment in the form of applications. Therefore, the activity classification application can provide predictions of the learning cycle as a whole.

ACKNOWLEDGEMENTS

We express our appreciation for the support and facilities from the Department of Informatics Engineering, Faculty of Engineering and Vocational Studies, Ganesha University of Education and Faculty of Engineering, Udayana University towards the publication process of this research.

REFERENCES

[1] M. H. -Chávez *et al.*, "Development of virtual reality automotive lab for training in engineering students," *Sustainability*, vol. 13, no. 17, 2021, doi: 10.3390/su13179776.

[2] X. Huang, D. Zou, G. Cheng, and H. Xie, "A systematic review of AR and VR enhanced language learning," *Sustainability*, vol. 13, no. 9, 2021, doi: 10.3390/su13094639.

[3] P. Sajjadi, J. Zhao, J. O. Wallgrun, P. L. Femina, and A. Klippel, "Influence of HMD type and spatial ability on experiences and learning in place-based education," *Proceedings of 2021 7th International Conference of the Immersive Learning Research Network, iLRN 2021*. IEEE, 2021, doi: 10.23919/iLRN52045.2021.9459405.

[4] I. P. W. Ariawan, W. Sugandini, I. M. Ardana, and D. G. H. Divayana, "Simulation of the weighted product method in the tri pramana-based formative-summative evaluation application," *Proceedings - International Conference on Education and Technology, ICET*, IEEE, pp. 8–13, 2022, doi: 10.1109/ICET56879.2022.9990611.

[5] I. P. W. Ariawan, W. Sugandini, I. M. Ardana, I. M. S. D. Arta, and D. G. H. Divayana, "Design of formative-summative evaluation model based on tri pramana-weighted product," *Emerging Science Journal*, vol. 6, no. 6, pp. 1476–1491, 2022, doi: 10.28991/esj-2022-06-06-016.

[6] W. Paramartha, N. L. Sustiawati, N. M. Sukrawati, and G. A. D. Sugiharni, "Tri pramana values in educational pedagogy," *Academic Journal of Interdisciplinary Studies*, vol. 11, no. 3, pp. 199–212, 2022, doi: 10.36941/ajis-2022-0078.

[7] I. G. Astawan, D. N. Sudana, N. Kusmariyatni, and I. G. N. Japa, "The STEAM integrated panca pramana model in learning elementary school science in the industrial revolution era 4.0," *International Journal of Innovation, Creativity and Change*, vol. 5, no. 5, pp. 26–39, 2019.

[8] I. N. Susrawan, I. M. Sutama, and I. W. Rasna, "Learning speaking skills through the mikir method based on local genius tri pramana," *Proceedings of the First Jakarta International Conference on Multidisciplinary Studies Towards Creative Industries, JICOMS 2022, 16 November 2022, Jakarta, Indonesia*. EAI, 2022, doi: 10.4108/eai.16-11-2022.2326109.

[9] D. Made and D. Putra Nugraha, "Tri pramana concept as learning approach to develop a lifelong learner," *The International Conference on Multi-Disciplines Approaches for The Sustainable Development*, 2023, pp. 228-236.

[10] L. Li, S. Qin, Z. Lu, K. Xu, and Z. Hu, "One-shot learning gesture recognition based on joint training of 3D ResNet and memory module," *Multimedia Tools and Applications*, vol. 79, no. 9–10, pp. 6727–6757, 2020, doi: 10.1007/s11042-019-08429-9.

[11] H. Yuan, J. Cheng, Y. Wu, and Z. Zeng, "Low-res MobileNet: an efficient lightweight network for low-resolution image classification in resource-constrained scenarios," *Multimedia Tools and Applications*, vol. 81, no. 27, pp. 38513–38530, 2022, doi: 10.1007/s11042-022-13157-8.

[12] Y. S. Taspinar, M. Dogan, I. Cinar, R. Kursun, I. A. Ozkan, and M. Koklu, "Computer vision classification of dry beans (Phaseolus vulgaris L.) based on deep transfer learning techniques," *European Food Research and Technology*, vol. 248, no. 11, pp. 2707–2725, 2022, doi: 10.1007/s00217-022-04080-1.

[13] O. Sharma, A. Sharma, and A. Kalia, "Windows and IoT malware visualization and classification with deep CNN and Xception CNN using markov images," *Journal of Intelligent Information Systems*, vol. 60, no. 2, pp. 349–375, 2023, doi: 10.1007/s10844-022-00734-4.

[14] R. Miller, N. K. Banerjee, and S. Banerjee, "Combining real-world constraints on user behavior with deep neural networks for virtual reality (VR) biometrics," *Proceedings - 2022 IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2022*. IEEE, pp. 409–418, 2022, doi: 10.1109/VR51125.2022.00060.

[15] T. Zhang, "Research on environmental landscape design based on virtual reality technology and deep learning," *Microprocessors and Microsystems*, vol. 81, 2021, doi: 10.1016/j.micpro.2020.103796.

[16] G. Qin and G. Qin, "Virtual reality video image classification based on texture features," *Complexity*, vol. 2021, pp. 1–11, 2021, doi: 10.1155/2021/5562136.

[17] N. Meissler, A. Wohlan, N. Hochgeschwender, and A. Schreiber, "Using visualization of convolutional neural networks in virtual reality for machine learning newcomers," *Proceedings - 2019 IEEE International Conference on Artificial Intelligence and Virtual Reality, AIVR 2019*. IEEE, pp. 152–158, 2019, doi: 10.1109/AIVR46125.2019.00031.

[18] L. Bibbò and F. C. Morabito, "Neural network design using a virtual reality platform," *Global Journal of Computer Science and Technology*, vol. 22, no. D1, pp. 45–61, 2022, doi: 10.34257/gjcstdvol22is1pg45.

[19] N. C. Tran, J. H. Wang, T. H. Vu, T. C. Tai, and J. C. Wang, "Anti-aliasing convolution neural network of finger vein recognition for virtual reality (VR) human–robot equipment of metaverse," *Journal of Supercomputing*, vol. 79, no. 3, pp. 2767–2782, 2023, doi: 10.1007/s11227-022-04680-4.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[21] D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia, "Deep learning in image classification using residual network (ResNet) variants for detection of colorectal cancer," *Procedia Computer Science*, vol. 179, pp. 423–431, 2021, doi: 10.1016/j.procs.2021.01.025.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[23] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "MobileNetV2 model for image classification," *Proceedings - 2020 2nd International Conference on Information Technology and Computer Application, ITCA 2020*. IEEE, pp. 476–480, 2020, doi: 10.1109/ITCA52113.2020.00106.

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

[25] F. Chulu, J. Phiri, P. O. Y. Nkunika, M. Nyirenda, M. M. Kabemba, and P. H. Sohati, "A convolutional neural network for automatic identification and classification of fall army worm moth," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 7, pp. 112–118, 2019, doi: 10.14569/ijacsa.2019.0100717.

[26] F. Chollet, "Xception: deep learning with depthwise separable convolutions," *30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017, pp. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.

[27] E. Westphal and H. Seitz, "A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks," *Additive Manufacturing*, vol. 75, 2023, doi: 10.1016/j.addma.2023.103739.

[28] H. Jin, "Hyperparameter importance for machine learning algorithms," *ArXiv-Statistics*, pp. 1-8, 2022.

[29] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," *ArXiv-Computer Science*, pp. 1-15, 2015.

[30] R. Tiwari, "Stabilizing the training of deep neural networks using adam optimization and gradient clipping," *Interantional Journal of Scientific Research in Engineering and Management*, vol. 7, no. 1, 2023, doi: 10.55041/ijsrem17594.

[31] J. Jepkoech, D. M. Mugo, B. K. Kenduiywo, and E. C. Too, "The effect of adaptive learning rate on the accuracy of neural networks," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 8, pp. 736–751, 2021, doi: 10.14569/IJACSA.2021.0120885.

[32] P. M. Radiuk, "Impact of training set batch size on the performance of convolutional neural networks for diverse datasets," *Information Technology and Management Science*, vol. 20, no. 1, 2018, doi: 10.1515/itms-2017-0003.

[33] R. Lapid and M. Sipper, "Evolution of activation functions for deep learning-based image classification," *GECCO 2022 Companion-2022 Genetic and Evolutionary Computation Conference*. ACM, pp. 2113–2121, 2022, doi: 10.1145/3520304.3533949.

[34] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: comparison of trends in practice and research for deep learning," *ArXiv-Computer Science*, pp. 1-20, 2018.

[35] L. Feng, S. Shu, Z. Lin, F. Lv, L. Li, and B. An, "Can cross entropy loss be robust to label noise?," *IJCAI International Joint Conference on Artificial Intelligence*, pp. 2206–2212, 2020, doi: 10.24963/ijcai.2020/305.

[36] D. M. Belete and M. D. Huchaiah, "Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results," *International Journal of Computers and Applications*, vol. 44, no. 9, pp. 875–886, 2022, doi: 10.1080/1206212X.2021.1974663.

[37] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 01–11, 2015, doi: 10.5121/ijdkp.2015.5201.

[38] S. D. A. Bujang *et al.*, "Multiclass prediction model for student grade prediction using machine learning," *IEEE Access*, vol. 9, pp.

95608–95621, 2021, doi: 10.1109/ACCESS.2021.3093563.

[39] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00444-8.

[40] B. Liu, W. Shen, P. Li, and X. Zhu, "Accelerate mini-batch machine learning training with dynamic batch size fitting," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2019, 2019, doi: 10.1109/IJCNN.2019.8851944.

[41] S. S. S. Palakodati, V. R. R. Chirra, Y. Dasari, and S. Bulla, "Fresh and rotten fruits classification using CNN and transfer learning," *Revue d'Intelligence Artificielle*, vol. 34, no. 5, pp. 617–622, 2020, doi: 10.18280/ria.340512.

[42] L. Yuwen, S. Chen, and X. Yuan, "G2Basy: a framework to improve the RNN language model and ease overfitting problem," PLoS One, vol. 16, no. 4, Apr. 2021, doi: 10.1371/journal.pone.0249820.

[43] A. Bhattacharjee, R. Murugan, B. Soni, and T. Goel, "Ada-GridRF: a fast and automated adaptive boost based grid search optimized random forest ensemble model for lung cancer detection," *Physical and Engineering Sciences in Medicine*, vol. 45, no. 3, pp. 981–994, 2022, doi: 10.1007/s13246-022-01150-2.

[44] S. Srinivasan, D. Francis, S. K. Mathivanan, H. Rajadurai, B. D. Shivahare, and M. A. Shah, "A hybrid deep CNN model for brain tumor image multi-classification," *BMC Medical Imaging*, vol. 24, no. 1, Jan. 2024, doi: 10.1186/s12880-024-01195-7.

[45] H. K. Ravikiran, J. Jayanth, M. S. Sathisha, and K. Bindu, "Optimizing sheep breed classification with bat algorithm-tuned CNN hyperparameters," *SN Computer Science*, vol. 5, no. 2, 2024, doi: 10.1007/s42979-023-02544-z.

BIOGRAPHIES OF AUTHORS

I Gede Partha Sindu is a lecturer in the field of Informatics Education at Department of Informatics Education, Faculty of Technical and Vocational, Universitas Pendidikan Ganesha. He is interested in researching in the field of Informatics in education, virtual reality, augmented reality, deep learning and instructional design. He can be contacted at email: partha.sindu@undiksha.ac.id

Made Sudarma is a Professor of Electrical Enggineering and a Postgraduate lecturer at Universitas Udayana. He is interested in researching in the field of information technology, artificial intelligence, cognitive ergonomic, human computer interaction, data mining, and data warehouse. He can be contacted at email: msudarma@unud.ac.id

Rukmi Sari Hartati is a Professor of Electrical Enggineering and a Postgraduate lecturer at Universitas Udayana. Her research interests are in Electrical Power System and Engineering Science. She can be contacted at email: rukmisari@unud.ac.id

Nyoman Gunantara is a Professor of Electrical Enggineering and a Postgraduate lecturer at Universitas Udayana. His research interests include wireless communications, ad-hoc network, quality network, and optimization. He is a member of the IEEE. He can be contacted at email: gunantara@unud.ac.id