LONTAR KOMPUTER VOL. 15, NO. 3 DECEMBER 2024 **p-ISSN 2088-1541**
DOI : 10.24843/LKJITI.2024.v15.i03.p01 **e-ISSN 2541-5832**
Accredited Sinta 2 by RISTEKDIKTI Decree No. 158/E/KPT/2021

Training VGG16, MobileNetV1 and Simple CNN Models from Scratch for Balinese Inscription Recognition

Ida Ayu Putu Febri Imawati[a1,2], Made Sudarma[b3], I Ketut Gede Darma Putra[c4], I Putu Agung Bayupati[c5], Minho Jo[d6]

[a]Study Program of Doctoral Engineering Science, Faculty of Engineering, Udayana University, Bukit Jimbaran Campus, Bali, Indonesia and
Faculty of Science and Technology, University of PGRI Mahadewa Indonesia
[1]imawati.2291011018@student.unud.ac.id
[2]febri@mahadewa.ac.id

[b]Departement of Electrical Engineering, Faculty of Engineering, Udayana University
Bukit Jimbaran Campus, Bali, Indonesia
[3]msudarma@unud.ac.id (Corresponding author)

[c]Departement of Information Technology, Faculty of Engineering, Udayana University
Bukit Jimbaran Campus, Bali, Indonesia
[4]ikgdarmaputra@unud.ac.id
[5]bayupati@unud.ac.id

[d]Departement of Computer and Information Science, Korea University,
Sejong Metropolitan City, South Korea 339-700
[6]minhojo@korea.ac.kr

Abstract

Many inscriptions in Bali are damaged. Damage to these inscriptions can be caused by natural disasters, overgrown with moss, algae and bacteria. Damage can also be caused by warfare, or deliberately erased. This inscription contains the knowledge and civilization of the ancestors so it is very important to be able to read its contents. Based on these problems, this research conducted training from scratch on 3 CNN models namely VGG16, MobileNetV1 and Simple CNN. The purpose of this research is to choose one recognition model that has the best performance and produces the highest recognition rate to proceed to the inscription restoration stage. The dataset used is Balinese inscription: Isolated Character Recognition of Balinese Script in Palm Leaf Manuscript Images in Challenge-3-ForTrain.zip. The training process of three models with five different training files resulted in the finding that VGG16 has the highest accuracy in the training, testing, and validation process with the least number of epochs. This research contributes to specific datasets, such as the Isolated Character Recognition of Balinese Script using the training process from the beginning of VGG16, involving all stages of the process. It will produce the best model performance compared to the other four training models.

Keywords: *Image Processing, Convolutional Neural Network, VGG16, MobileNetV1, Balinese Inscription, Training Model from Scratch*

1. Introduction

Inscriptions are cultural objects that contain ancient scripts and numbers. In Sanskrit, inscription means praise and then undergoes development until it becomes a charter, decree, edict, law, or writing. Many lay people call inscriptions lettered stones or writing stones. Inscriptions are the meaning of historical sources of past relics written on the surface of hard material objects such as stone, metal, wood, horns, and bones made based on the orders of the ruler or leader of an area. The main contents of the inscription can be [1]: 1) regarding legal decisions, 2) debt or pawn problems, 3) conquest or victory over certain areas, and can also be 4) the validity of a sale and purchase transaction of goods or land. The medium of inscription can be stone (utpala inscription),

rontal (ripta inscription), and copper (tamra inscription). There are two types of inscription damage, namely, damage caused by nature and also damage caused by human actions[2]. Damage caused by natural disasters causes inscriptions to be lost, broken, or worn out. Damage by nature can also be in the form of being overgrown with moss, algae, or bacteria, making the inscription soft and damaged. Damage caused by humans can be done intentionally, for example, due to warfare, deliberately erased or even destroyed because it is considered not by the times. For example, when the inscription was discovered, unintentional damage was hit by a hoe, causing the letters to disappear.

Many of the inscriptions are damaged, and the characters cannot be recognized and read, which makes it challenging to know the contents of the inscriptions. In Bali, there are about 271 groups of inscriptions[3], both lontar, plate, and stone inscriptions. If we can help read these, then it saves knowledge and civilization. Only now are there still stone inscriptions that cannot be read about the contents, so the owner needs clarification about care, maintenance, giving treatment (giving offerings), and mentioning. Why do we need to recognize inscriptions? The reason is first the difficulty of script and language; not everyone can read, especially the archaic script or language script that has no speakers [4]; the problem is the level of fragility of the media where the inscription text is written, besides that the sacredness of the inscription or inscription itself. Inscriptions in some areas, especially in Bali, in most areas are considered an ancestral heritage that should not be removed from its place, held until it is read by someone who is not authorized or entitled [3]. With the sacredness of the inscriptions, a recognition of the inscription script is needed, allowing us to understand the knowledge and civilization of the past to guide us in the future.

Many studies have been related to feature extraction, detection, and recognition of scripts in inscriptions. Research by Darma [5] recognizes the Wresastra script with its feature extraction using the zoning method and classifier with k-Nearest Neighbors (k-NN). The results show that the increase in recognition accuracy rate is strongly influenced by the number of references used when training data models and the number of K values. Accuracy will decrease the higher the K value. Conversely, a higher number of references will also increase the accuracy rate.

Research conducted by Indrawan [6] states that the recognition process of Balinese script is proposed using the Tesseract OCR Engine using datasets obtained from the web scrapping method. The previous version of Tesseract used JtessBox tools, which required much time to generate datasets. With the latest technology, the Tesseract OCR training method uses a new LSTM (RNN) engine. The initial stage of connected component analysis, outlined, is collected and converted into Blob data type. The second stage will be organized into proportional text lines, divided into definite words and fuzzy spaces. The third character recognition stage usually recognizes each word and then validates the hypothesis to get lowercase text using fuzzy spaces. The results of this study state that by using Tesseract, OCR version 5 uses three experiments and different dataset hierarchical structures; the first hierarchical dataset uses a random dataset combination with a coincidence rate of 25%, the second dataset with a hierarchical dataset per character, which produces a coincidence rate of 40%, and finally, a combination dataset of characters, word sentences, and paragraphs produces a coincidence rate of 66.67%. So, it can be concludedthat the more different datasets and the more structured the dataset hierarchy used, the higher the coincidence rate.

The selection of CNN, VGG models, and their architecture modifications is based on several studies on ancient manuscript recognition, such as in [7], [8], [9] and [10]. The author takes this problem mainly because of the fundamental thought of which models and techniques are most suitable for the recognition process for specific datasets, such as Isolated Character Recognition of Balinese Script. The literature survey related to this research can be presented in Table 1. Most research on inscriptions in the recognition process chooses CNN with training techniques from the beginning [7], [11], [8], [10], [12], not its variations. However, most transfer learning techniques use CNN model variations such as VGG16 and MobileNetV1 [13], [14]. Kesiman's research uses the same dataset as the dataset used in this study and uses training from the beginning but does not involve the feature extraction stage[11]. This research still uses every stage, from feature extraction to classification. However, some studies use CNN variations with training techniques from the beginning[15], [9], [16]. Therefore, This research shows the training process from the beginning on the VGG16, MobileNetV1, and Simple CNN models. Thus, researchers can also

consider that VGG16, MobileNetV1, and other CNN variations can be trained from scratch within the same research scope.

Kesiman [11] focused on feature extraction methods. This research proposes a combination of features to perform feature extraction. Kesiman offers 29 scenarios of feature extraction and classification combinations using k-NN or SVM. The last scenario in this study also uses CNN as one of the recognition methodologies without using feature extraction again. Of the 29 scenarios, a combination of Method HoG + NPW-Kirsch (Gray) + Zoning (Binary) with k-NN as a classifier achieves a recognition rate of 85.1557%. This combination methodology was robust enough to beat the Convolutional Neural Network, reaching only 84.3086%.

The focus of Sutramiani's research [13] is to propose an augmentation method suitable for a set of handwritten images with a limited number of strokes and high noise to improve the performance of Balinese script recognition. According to this research, the proposed augmented method performs well when using the VGG19 and MobileNetV2 models. VGG19 is intended for models with many parameters, and MobileNetV2 is for models with few parameters. The research in Kesiman and Sutramiani uses CNN and other CNN-derived model architectures as recognition models, but the research focus is different. Kesiman focuses on the combination of feature extraction using SVM or k-NN classifiers. At the same time, Sutramiani emphasizes the process of selecting or determining the best augmentation method for handwriting-based datasets. CNN in Kesiman's research, the feature extraction part is omitted, and CNN is used only as a comparison for k-NN or SVM classifiers. Sutramiani CNN is implemented with transfer learning techniques and uses a pre-trained model.

This research uses the dataset from Kesiman, namely the Isolated Character Recognition of Balinese Script in Palm Leaf Manuscript Images dataset, namely in Challenge-3-ForTrain.zip[17]. This research applies VGG16, MobileNetV1, and Simple CNN. The CNN layer used starts from feature extraction training to recognition. The author does not use pre-trained, tuning, or transfer learning models. The author tries to apply CNN architecture or its varitions by training from scratch. The goal is to compare the VGG16, MobileNetV1, and Simple CNN models using datasets on the AMADI Lontar Set so that by comparing these models, a model with an optimal level of accuracy will be obtained to recognize Balinese inscription datasets. Section 1 of this paper presents the research's background, reasons, and objectives. In section 2, the author presents the research methodology that the author used; section 3 is related to the results and discussion of the outcomes obtained from this research, and the last part in section 4 is related to the results and conclusions of this research.

Table 1. Literature survey

Authors	CNN	CNN Variations
Paulus et al, 2019	Training from scratch and architecture modification	-
Kesiman et al, 2016	Training from scratch but feature extraction step is not required	-
Hidayat et al, 2021	Training from scratch	-
Sutramiani et al, 2021	-	Pretrained of five CNN variations namely: Inception RestNetV2, DenseNet169, ResNet152V2, VGG19, MobileNetV2
Ravi, 2024	Training from scratch and architecture modification (THAC-CNN1, THAC-CNN2, THAC-CNN3)	-
Avadesh and Goyal , 2018	Training from scratch and architecture modification	-
Raharja et al, 2022	-	Pretrained of MobileNet
Septianto et al, 2018	-	Training from scratch of two CNN variations namely: LeNet and VGG

Nair et al, 2022	-	VGG-16 model training from scratch
Sudana and Gunaya , 2020	-	VGG19 training from scratch

2. Reseach Methods

The stages of feature extraction to train the recognition model in this study are : 1) preparation of the dataset, 2) preprocessing the dataset, 3) creating a model, 4) training the model, and finally, 5) evaluating the training results. The research flow is presented in Figure 1.

Each detailed stage of this research method is explained in Section 2.1 to Section 2.5. The model training stage is an important part of the recognition process (Section 2.4). The author compares the models by developing the three models into five different training files (Table 2). From the five different training files, this research will provide a proposed model that is most suitable for the recognition process of the Isolated Character Recognition of the Balinese Script dataset. The novelty of this research is that the VGG16 file that was trained from the beginning has the highest train accuracy, test accuracy, and validation accuracy compared to the other four training files.

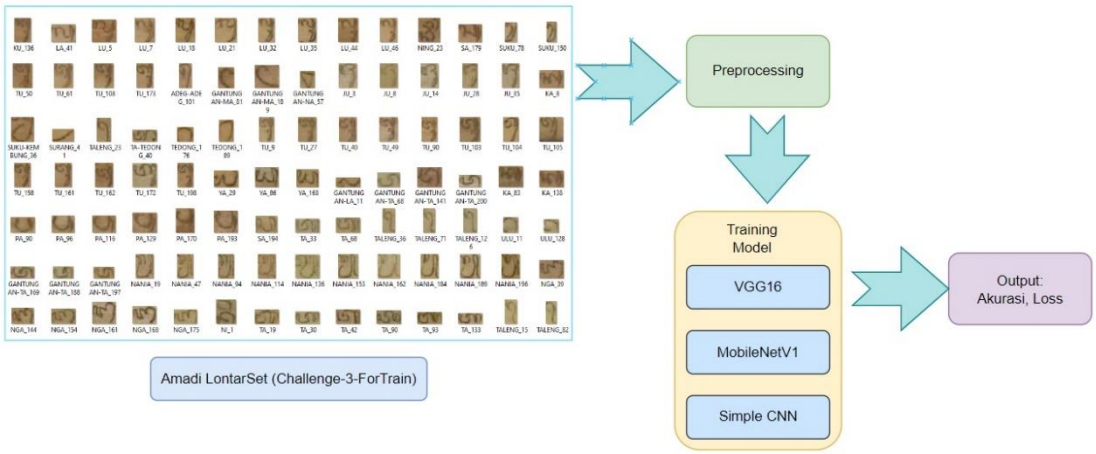

Figure 1. Training process flow of recognition model

2.1. Dataset Preparation

The author uses the dataset of Isolated Character Recognition of Balinese Script in Palm Leaf Manuscript Images in Challenge-3-ForTrain.zip. This dataset comprises 23 collections from 5 locations or regions: 2 museums and three private families with 393 palm leaf manuscript pages[17]. The stages in preparing the dataset are as follows: The first stage uses aggregation techniques. Dataset aggregation is needed because the data has reached hundreds of thousands and even millions of records, requiring minutes of looping. The second stage converts the character class into a sequence of numbers. This dataset has 133 characters and classes, each of which is coded into numbers. Third is the merging process, which combines the file name, character, and character class. The fourth stage is splitting the dataset, which is dividing the dataset into a train dataset, testing dataset, and validation dataset. The last stage is setting the dataset quota, which is the percentage of each quota of train data, testing data, and validation data that will be used

2.2. Preprocessing Dataset

The class dataset is divided into the plain and BW datasets (Figure 2). Class DatasetPlain converts the image to RGB color space, while Class DatasetBW is a dataset that converts its color space into RGB and applies the Otsu operator. Each dataset class can be given transform treatments such as T.Resize, which changes the size of the image into a particular dimension; T.CenterCrop, which is the process of cropping the image into a specific size; and also T.Normalize, which changes all pixels 0 to 255 to be worth 0 to 1. The experimental scenario in this research is in Table 1.

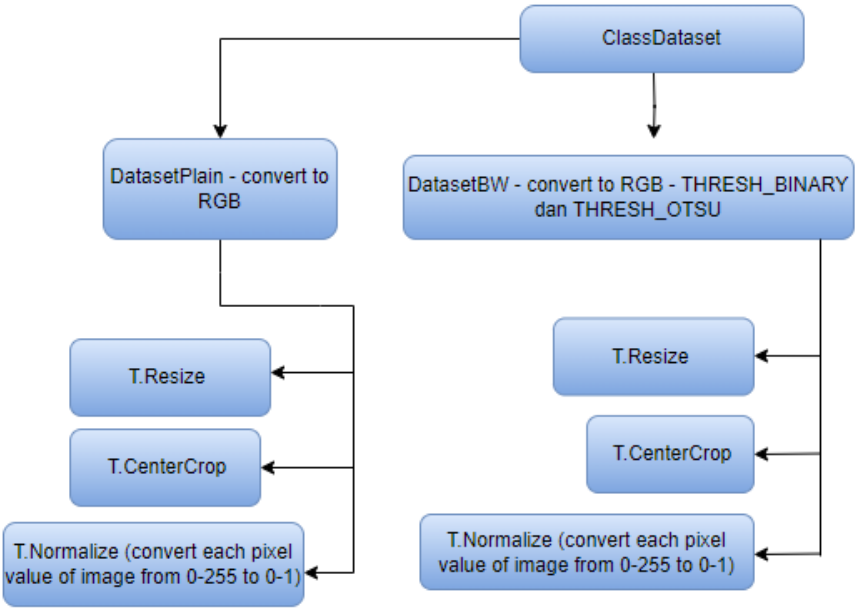

Figure 2. Class dataset

Table 2. Experiment scenarios

File Training Name	Type of settings
VGG16	1. T.Resize((227,227)) 2. Without BW operation
MobileNetV1	1. Focus on T.CenterCrop(110) transform 2. Use BW operation
MobileNet V1 update	1. Focus on T.CenterCrop(110) transform 2. Use BW operation 3. Imbalance dataset 4. Testing mode does not use imbalance
Simple CNN	1. T.Resize((28,28)) 2. Use BW operation
Simple CNN update	1. T.Resize((28,28)) 2. Use BW Operation 3. Imbalance dataset 4. Testing mode does not use imbalance

2.3. VGG16 Model, MobileV1Net Model, Simple CNN Model

These models will be imported into other files as libraries. The architecture of the VGG16, MobileNetV1, and Simple CNN models can be described in Figure 3, Figure 4, and Figure 5.

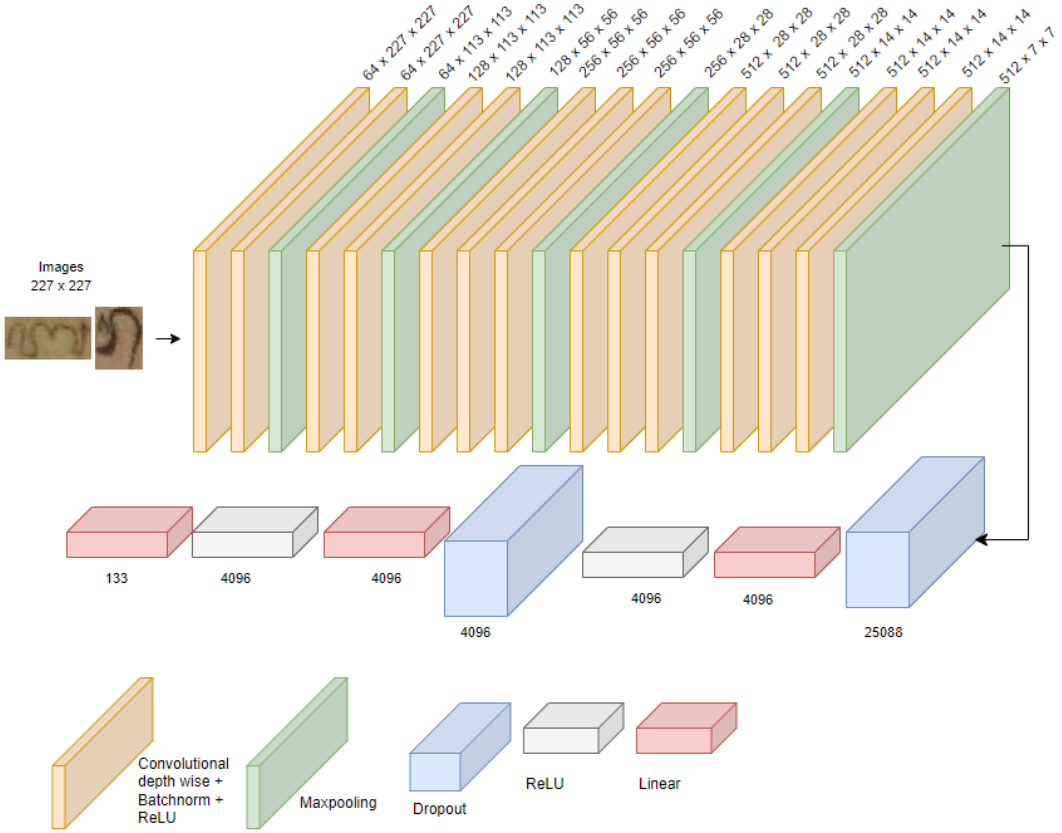

Figure 3. VGG16 Model Architecture

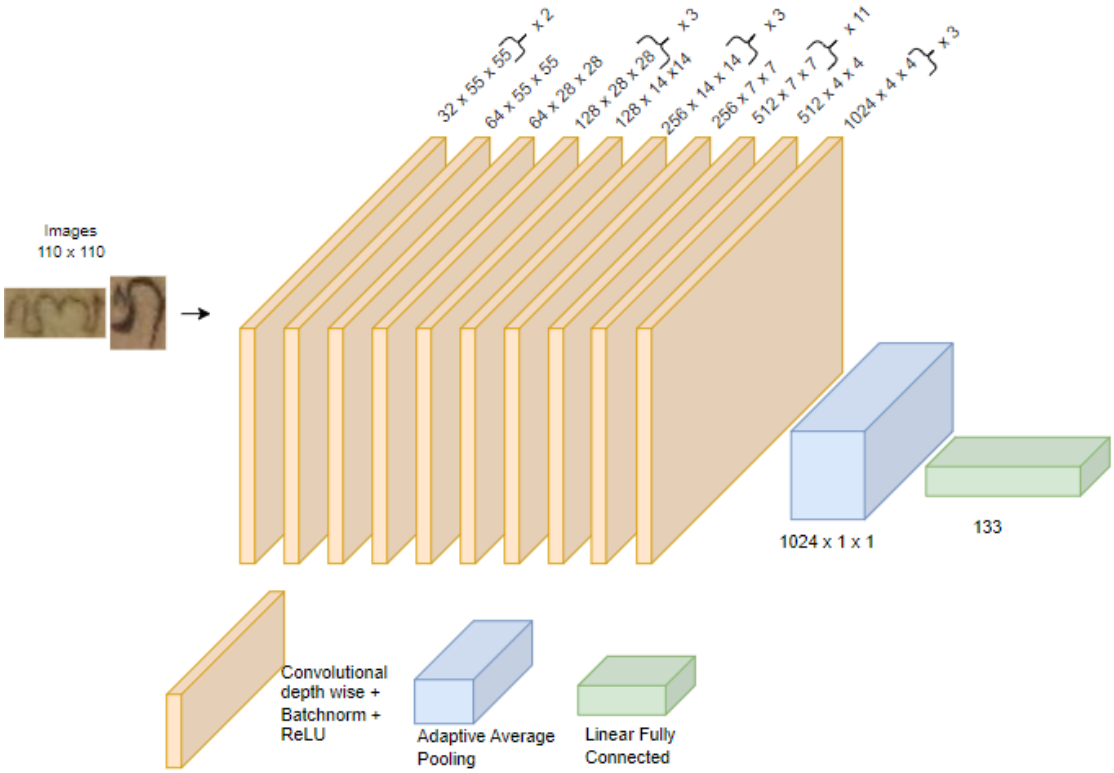

Figure 4. MobileNetV1 Model Architecture

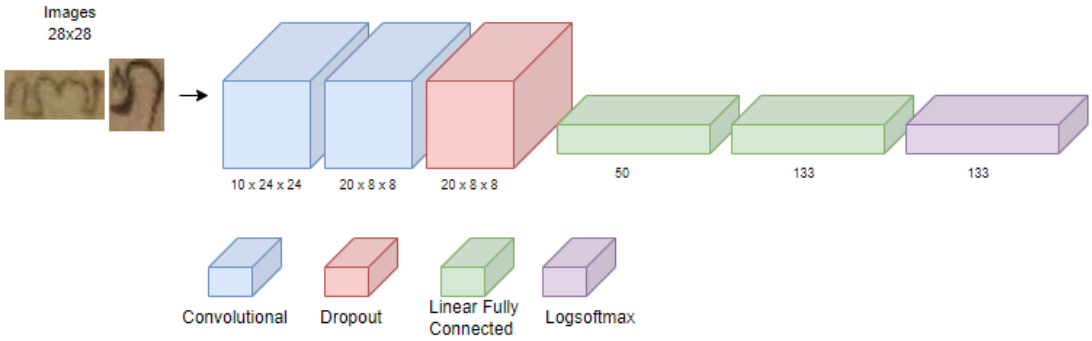

Figure 5. Simple CNN Model Architecture

The three models, namely VGG16, MobileV1Net, and Simple CNN, are not connected but are trained per each model. Using the dataset described in Figure 2 and applying the dataset settings in Table 2, the three models were developed into five different training files. The five files include VGG16, MobileNetV1, MobileNetV1 update, Simple CNN, and Simple CNN update. Furthermore, the author applies scenarios and parameters for the five files in the training process, as in Table 3.

2.4. Training Model

The stages of the training process: first, preparing the class dataset, augmenting the dataset, and determining the number of batches per 1 training. Second, set the model, choosing whether to use the previous training session: set the folder to save the training, set the model and loss function, and choose to use the previous training results. Third, determine the iteration and stop settings, whether to stop using a particular epoch or stop using the stop file, and save the weights and loss.

Table 3. Scenarios and parameter settings

File Training Name	Type of settings	Learning rate (lr) / momentum	Loss	Optimizer
VGG16	1. T.Resize((227,227)) 2. Without BW operation	0.005/0.9	CrossEntropyLoss (Linear)	SGD
MobileNetV1	1. Focus on T.CenterCrop(110) transform 2. Use BW operation	0.01/0.9	CrossEntropyLoss (Linear)	SGD
MobileNet V1 update	1. Focus on T.CenterCrop(110) trasnform 2. Use BW operation 3. Imbalance dataset 4. Testing mode does not use imbalance	0.01/0.9	CrossEntropyLoss (Linear)	SGD
Simple CNN	1. T.Resize((28,28)) 2. Use BW operation	0.001/0.5	NLLLoss (logsoftmax)	SGD
Simple CNN update	1. T.Resize((28,28)) 2. Use BW Operation 3. Imbalance dataset 4. Testing mode does not use imbalance	0.001/0.5	NLLLoss (logsoftmax)	SGD

2.5. Training Result Evaluation

Based on the training results of the three models, training accuracy, test accuracy, validation accuracy and loss calculations are carried out to evaluate the training results at this stage.

3. Result and Discussion

The distribution of the Isolated Character Recognition of Balinese Script in the Palm Leaf Manuscript Images dataset, namely in Challenge-3-ForTrain.zip, is indeed uneven, according to research [6]. Some characters are only sometimes found in the palm lead manuscript collection. Therefore, the experiments in this study try to use the SMOTE (Synthetic Minority Oversampling Technique), which is data augmentation for minority data classes or oversampling the dataset. The author applies an Imbalanced Dataset Sampler to MobileNetV1 imbalance training and Simple CNN imbalance model training, except that the evaluation process of these two models has not applied SMOTE evaluation. SMOTE is popularly used for resampling datasets to obtain another dataset with the same or at least a similar number of instances of each class, thus reducing the bias of majority classes and providing correct classifier support for minority classes [18]. The SMOTE oversampling technique also contributes to good performance when used for prediction in various machine learning models[19].

Table 4. The results of five models training

File Training Name	Type of settings	Last epoch	Last loss
VGG16	1. T.Resize((227,227)) 2. Without BW operation	6	0.02
MobileNetV1	1. Focus on T.CenterCrop(110) transform 2. Use BW operation	25	0.00
MobileNet V1 update	1. Focus on T.CenterCrop(110) trasnform 2. Use BW operation 3. Imbalance dataset 4. Testing mode does not use imbalance	38	0.001
Simple CNN	1. T.Resize((28,28)) 2. Use BW operation	500	1.506
Simple CNN update	1. T.Resize((28,28)) 2. Use BW Operation 3. Imbalance dataset 4. Testing mode does not use imbalance	500	1,.605

Training from scratch for these five training models uses 10,540 data records with a quotation division of 90% as training data, 5% as test data, and 5% as validation data. The VGG16 model uses DatasetPlain by resizing the input image to 227 x 227 pixels. Other parameters include a learning rate of 0.005, momentum of 0.9, Loss function CrossEntropy Loss, and SGD optimizer. This VGG16 model produces a training accuracy of 91.01%, a testing accuracy of 85.3%, and a validation accuracy of 83.93%. The last epoch is six epochs with a loss of 0.02. The total number of parameters in this VGG16 model is 134,813,893 params.

The MobileNetV1 model was developed into two files, MobileNetV1 and MobileNetV1 update. Both use BW datasets with transform center crops of 110 x 110 pixels, a learning rate of 0.01, momentum of 0.9, and a loss function cross-entropy loss and SGD optimizer. What distinguishes these two files is the use of an Imbalanced dataset sampler. MobileNetV1 update uses an Imbalanced dataset (SMOTE) while MobileNetV1 does not. MobileNetV1 has training, testing and validation accuracy of 85.24%, 72.82% and 74.53% respectively. MobileNetV1 update produces a training accuracy of 86.0%, testing accuracy of 69.57%, and validation accuracy of

71.97%. The training accuracy of MobileNetV1 update has increased when compared to MobileNetV1. However, testing and validation accuracy decreased. The parameters trained on the MobileNetV1 model are a total of 3,341301 parameters.

The simple CNN model is also implemented in two different model files: the Simple CNN and SimpleCNN updates. The parameters used in both model files are image size resized to 28 x 28 pixels, using DatasetBW, the learning rate of 0.001, and momentum of 0.5. The loss function uses Negative Log Likelihood Loss (NLLLoss) and SGD optimizer. As in the MobileNetV1 model, the difference between these two files is using Dataset Imbalance (SMOTE). Simple CNN does not use the imbalanced sampler dataset, but the Simple CNN update applies the imbalance sampler to its training process. Simple CNN produces a training accuracy of 83.49%, a testing accuracy of 77.44%, and a validation accuracy of 77.78%. The Simple CNN update model obtained training, testing, and validation accuracy sequentially of 79.72%, 72.48%, and 71.11%. In this model, with the use of imbalanced datasets, there is a decrease in accuracy in all three processes: training, testing, and validation. The number of trainable params in the Simple CNN model shows 28,613 params.

Based on the training results presented in Table 4, table 5, Figure 6a to Figure 6f, it is found that the VGG16 recognition model has the highest accuracy value in the order of training, testing, and validation of 91.01%, 85.3%, and 83.93%. The Simple CNN update model produces the lowest training accuracy value of 79.72%. The lowest testing accuracy value is on the MobileNetV1 model at 69.57%, and the lowest validation accuracy value is on the Simple CNN update model at 71.11%. The VGG16 model also has the lowest epoch of 6 epochs. The highest number of epochs in Simple CNN and Simple CNN update is 500. The lowest loss is obtained in the MobileNetV1 model of 0.000. Our findings are based on the performance of VGG and MobileNet architectures in Sutramiani's research [13], namely the first best VGG19 followed by MobileNetV2 in the comparison of 5 CNN pre-trained models. CNN model training from scratch without feature extraction, as in Kesiman's research[11] recognition rate, reached 84.3086%, slightly higher than in this study, whose training accuracy reached 83.49%. Feature extraction on the Kesiman CNN model is done with various combinations of digital image processing.

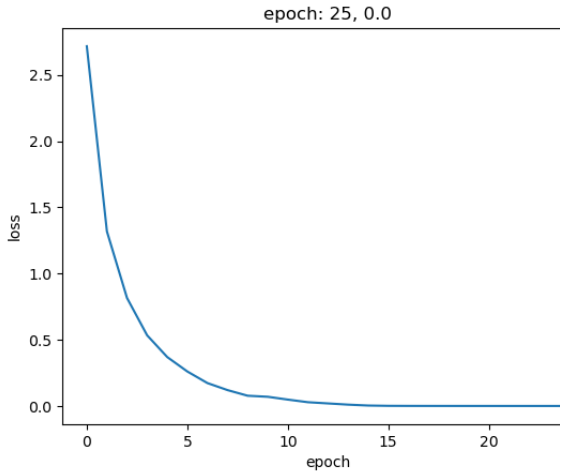

Figure 6a. MobileNetV1

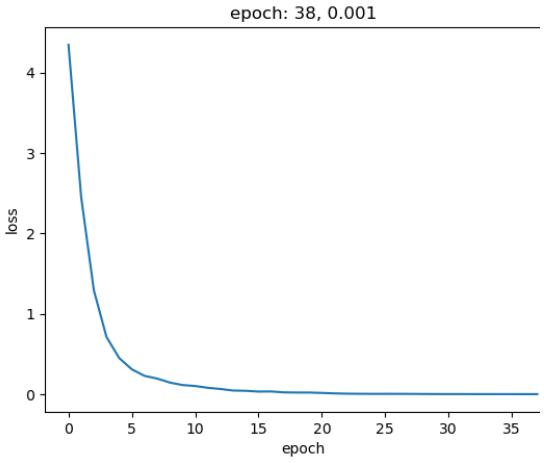

Figure 6b. MobileNetV1 Imbalance

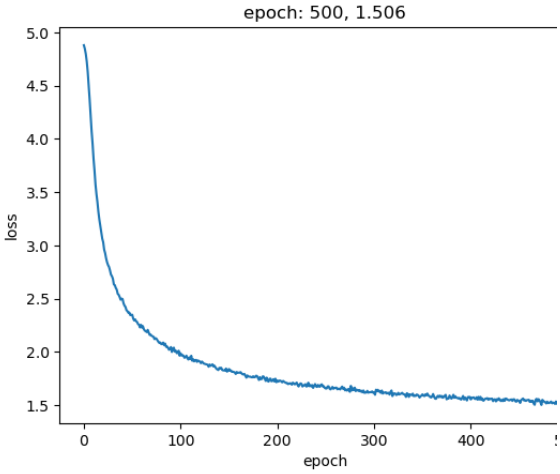

Figure 6c. Simple CNN

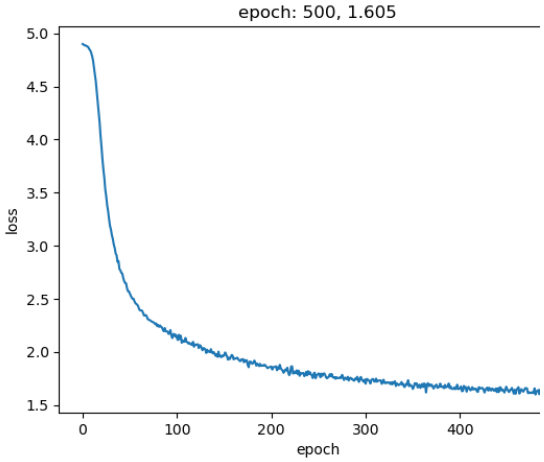

Figure 6d. Simple CNN Imbalance

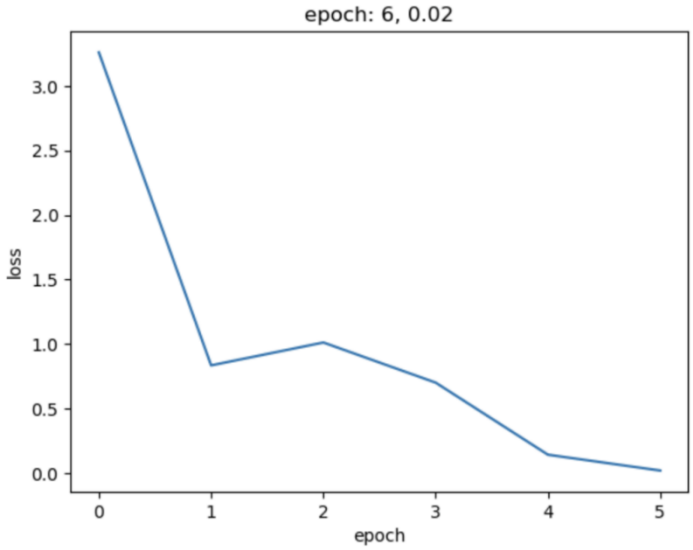

Figure 6e. VGG16

Table 5. Training, testing and validation accuracy of five training models

File Training Name	Train Accuracy	Test Accuracy	Validation Accuracy
VGG16	91.01%	85.3%	83.93%
MobileNetV1	85,24%	72.82%	74.53%
MobileNet V1 update	86.0%	69.57%	71.97%
Simple CNN	83.49%	77.44%	77.78%
Simple CNN update	79,72%	72.48%	71.11%

Using imbalanced datasets also affects the accuracy of training, testing, and validation results. MobileNetV1 models that use imbalanced sampler datasets with more epochs produce higher accuracy than MobileNetV1 training models without imbalanced datasets. The Simple CNN model that uses the imbalance sampler dataset at the same number of epochs produces lower accuracy than the Simple CNN training model without imbalance datasets. The number of trainable parameters also determines the performance of the model. The more trainable parameters, the more convergent a model is. As stated in the study[20], the parameters determine the level of accuracy and convergence speed.Table 6 shows the order of the largest parameters to the least

from VGG16, MobileNetV1, and, finally, Simple CNN. Thus, the order of least to most epoch also follows, namely VGG16, MobileNetV1, and finally, Simple CNN.

Table 6. Scenarios and parameter settings

Model	Number of params	Number of trainable params
MobileNetV1	3.341.301	3.341.301
Simple CNN	28.613	28.613
VGG16	134.813.893	134.813.893

4. Conclusion

This paper offers three recognition models by applying training from scratch using CNN models and their variations. The CNN variation models used are VGG16 and MobileNetV1. 3 models were developed into five different training files: the VGG16 model, the Simple CNN model, the Simple CNN update model, the MobileNetV1 model, and the MobileNetV1 update model. Some of the findings obtained that the VGG16 model has the highest accuracy value, with a training accuracy of 91.01%, testing accuracy of 85.3%, and validation accuracy of 83.93%. The lowest training accuracy value is on the Simple CNN update model of 79.72%. The lowest testing accuracy value is on the MobileNetV1 model at 69.57%, and the lowest validation accuracy value is on the Simple CNN update model at 71.11%. The VGG16 model also has the lowest epoch of 6 epochs. The highest number of epochs in Simple CNN and Simple CNN update is 500. The lowest loss is obtained in the MobileNetV1 model of 0.000. Based on these findings, the VGG16 model has the best performance.

Our findings also indicate that the greater the number of parameters used in the training process, the better the model's performance.The limitation of this research is the use of different device specifications in several models, so it cannot compare the time required during the training process. Future works of this research are how to process the recognition results into a sequence of characters to form a complete word or sentence**.**

References

[1] I. W. G. Y. Tenaya, “Prasasti Raja Jayasakti di Desa Bebandem Karangasen,” *Sudamala*, vol. 05, pp. 18–31, 2019.

[2] G. A. Sambodo, Y. K. Suprapto, and E. M. Yuniarno, “Application of Photogrammetry Techniques in Reconstructing the Carving on Stone Inscriptions,” *Berk. Arkeol.*, vol. 40, no. 2, pp. 309–328, 2020, doi: 10.30883/jba.v40i2.597.

[3] I. N. Sunarya, I. G. M. Suarbhawa, and I. W. Sumerata, “Penelitian Prasasti Kintamani,” Denpasar, 2015.

[4] H. Prihatmoko, “Kajian Epigrafis Prasasti Babahan,” *Forum Arkeol.*, vol. 29, no. 3, p. 117, 2017, doi: 10.24832/fa.v29i3.100.

[5] I. W. A. S. Darma, “Implementation of Zoning and K-Nearest Neighbor in Character Recognition of Wrésastra Script,” *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 10, no. 1, p. 9, 2019, doi: 10.24843/lkjiti.2019.v10.i01.p02.

[6] G. Indrawan, A. Asroni, L. Joni Erawati Dewi, I. G. A. Gunadi, and I. K. Paramarta, “Balinese Script Recognition Using Tesseract Mobile Framework,” *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 13, no. 3, p. 160, 2022, doi: 10.24843/lkjiti.2022.v13.i03.p03.

[7] E. Paulus, S. Hadi, M. Suryani, I. Suryana, and Y. D. Simanjuntak, “Evaluating Ancient Sundanese Glyph Recognition Using Convolutional Neural Network,” *J. Phys. Conf. Ser.*, vol. 1235, no. 1, 2019, doi: 10.1088/1742-6596/1235/1/012063.

[8] A. A. Hidayat, K. Purwandari, T. W. Cenggoro, and B. Pardamean, “A Convolutional Neural Network-based Ancient Sundanese Character Classifier with Data Augmentation,” *Procedia Comput. Sci.*, vol. 179, no. 2020, pp. 195–201, 2021, doi: 10.1016/j.procs.2020.12.025.

[9] B. J. Bipin Nair, K. V. Aadith Raj, M. Kedar, S. P. Vaishak, and E. V. Sreejil, “Ancient Epic Manuscript Binarization and Classification Using False Color Spectralization and VGG-16 Model,” *Procedia Comput. Sci.*, vol. 218, pp. 631–643, 2022, doi: 10.1016/j.procs.2023.01.045.

[10] J. Ravi, “Handwritten alphabet classification in Tamil language using convolution neural network,” *Int. J. Cogn. Comput. Eng.*, vol. 5, no. April 2023, pp. 132–139, 2024, doi: 10.1016/j.ijcce.2024.03.001.

[11] M. W. A. Kesiman, S. Prum, J. C. Burie, and J. M. Ogier, “Study on feature extraction methods for character recognition of Balinese script on palm leaf manuscript images,” *Proc. - Int. Conf. Pattern Recognit.*, vol. 0, pp. 4017–4022, 2016, doi: 10.1109/ICPR.2016.7900262.

[12] M. Avadesh and N. Goyal, “Optical character recognition for sanskrit using convolution neural networks,” *Proc. - 13th IAPR Int. Work. Doc. Anal. Syst. DAS 2018*, pp. 447–452, 2018, doi: 10.1109/DAS.2018.50.

[13] N. P. Sutramiani, N. Suciati, and D. Siahaan, “MAT-AGCA: Multi Augmentation Technique on small dataset for Balinese character recognition using Convolutional Neural Network,” *ICT Express*, vol. 7, no. 4, pp. 521–529, 2021, doi: 10.1016/j.icte.2021.04.005.

[14] I. P. B. G. Prasetyo Raharja, I. M. Suwija Putra, and T. Le, “Kekarangan Balinese Carving Classification Using Gabor Convolutional Neural Network,” *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 13, no. 1, p. 1, 2022, doi: 10.24843/lkjiti.2022.v13.i01.p01.

[15] T. Septianto, E. Setyati, and J. Santoso, “Model CNN LeNet dalam Rekognisi Angka Tahun pada Prasasti Peninggalan Kerajaan Majapahit,” *J. Teknol. dan Sist. Komput.*, vol. 6, no. 3, pp. 106–109, 2018, doi: 10.14710/jtsiskom.6.3.2018.106-109.

[16] O. Sudana, I. W. Gunaya, and I. K. G. D. Putra, “Handwriting identification using deep convolutional neural network method,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 4, pp. 1934–1941, 2020, doi: 10.12928/TELKOMNIKA.V18I4.14864.

[17] M. W. A. Kesiman *et al.*, “AMADI _ LontarSet : The First Handwritten Balinese Palm Leaf Manuscripts Dataset,” in *15th International Conference on Frontiers in Handwriting Recognition*, 2016, pp. 168–173, doi: 10.1109/ICFHR.2016.39.

[18] M. Juez-Gil, Á. Arnaiz-González, J. J. Rodríguez, C. López-Nozal, and C. García-Osorio, “Approx-SMOTE: Fast SMOTE for Big Data on Apache Spark,” *Neurocomputing*, vol. 464, pp. 432–437, 2021, doi: 10.1016/j.neucom.2021.08.086.

[19] Narayanan and Jayashree, “Implementation of Efficient Machine Learning Techniques for Prediction of Cardiac Disease using SMOTE,” *Procedia Comput. Sci.*, vol. 233, no. 2023, pp. 558–569, 2024, doi: 10.1016/j.procs.2024.03.245.

[20] Y. Lu *et al.*, “Influence of the parameters of the convolutional neural network model in predicting the effective compressive modulus of porous structure,” *Front. Bioeng. Biotechnol.*, vol. 10, no. September, pp. 1–11, 2022, doi: 10.3389/fbioe.2022.985688.