



Development of Lampung Script Characters Recognition Model using TensorFlow

**Meizano Ardhi
Muhammad**

Universitas Lampung, Lampung,
Indonesia

Martinus Martinus

Universitas Lampung, Lampung,
Indonesia

Adhi Nurhartanto

Universitas Mitra Indonesia,
Lampung, Indonesia

Yessi Mulyani

Universitas Lampung,
Lampung, Indonesia

Gita Paramita Djausal

Universitas Lampung, Lampung,
Indonesia

Deni Achmad

Universitas Lampung, Lampung,
Indonesia

Sony Ferbangkara

Universitas Lampung, Lampung,
Indonesia

Article Info

Article history:

Received: October 10, 2023

Revised: November 28, 2023

Accepted: December 29, 2023

Keywords:

Lampung script,
CNN,
Machine Learning
Recognition,
TensorFlow,

Abstract

In the face of cultural erosion, particularly the dwindling proficiency in deciphering Lampung characters, this research pioneers an innovative approach to cultural preservation. The Lampung character recognition model was developed using TensorFlow, a robust computer vision and machine learning framework. Convolutional Neural Networks (CNN) are integrated to enhance the image processing capabilities. The research employs the Design Science Research methodology, emphasizing problem identification, solution objectives, design and development, demonstration, evaluation, and communication. The dataset, comprising 3900 instances, is meticulously collected and features diverse Lampung script writing. Through preprocessing and classification, the model undergoes training with an 80:10:10 split for training, validation, and test data. The architecture includes CNN layers with ReLu activation functions, and transfer learning is employed using the MobileNet V2 network model. Demonstrating commendable performance, the model achieves an accuracy spectrum of 0.652 to 0.998. The research not only underscores the viability of the TensorFlow model but also establishes a foundation for future explorations in preserving Lampung cultural heritage. This intersection of advanced machine learning and cultural preservation signifies a promising synergy, ensuring the enduring legacy of Lampung characters amid societal and technological transformations.

To cite this article: M. A. Muhammad, M. Martinus, A. Nurhartanto, Y. Mulyani, M, G. P. Djausal, D. Achmad, S. Ferbangkara, "Development of Lampung Script Characters Recognition Model using TensorFlow," *Int. J. Electron. Commun. Syst.* Vol. 3, No. 2, pp. 97-104, 2023.

INTRODUCTION

In the modern era, the intricate cultural tapestry of Lampung faces the imminent risk of fading into obscurity, particularly concerning the diminishing ability to decipher Lampung characters among its citizens. This erosion is palpable in the Lampung resident's challenges in decoding their traditional script [1], [2].

Indonesia, renowned for its cultural diversity, harbors many characters, with the Lampung character standing out as a distinctive emblem of historical significance. The written

records from the Lampung region scripts in Lampung characters weave narratives of bygone eras through letters, poetry, spells, laws, and counsel [3]. However, a noteworthy statistic from the Central Statistics Agency reveals that 2,333,057 out of 8,109,601 people hail from regions outside Lampung [4].

The confluence of societal shifts propelled by outsiders and technological strides poses a tangible threat to the survival of Lampung's Character, necessitating a collective effort to safeguard this cultural treasure. The

• **Corresponding author:**

Meizano Ardhi Muhammad, Universitas Lampung, Lampung,, INDONESIA. ✉ meizano@eng.unila.ac.id

© 2023 The Author(s). **Open Access.** This article is under the CC BY SA license (<https://creativecommons.org/licenses/by-sa/4.0/>)

application of information technology emerges as a contemporary initiative to facilitate the learning and conservation of Lampung characters [1], [2], [5]. Within the realm of academic exploration, numerous studies have delved into Lampung character recognition, employing diverse methodologies such as CNN detection [5]. The studies provide valuable insights into the challenges and achievements associated with Lampung character recognition, shaping the trajectory of the present research [5], [6]. However, it is noteworthy that comprehensive datasets encompassing Lampung characters, including parent and child characters and their combinations, are yet to be fully developed.

The research at hand adopts a framework approach [7], integrating computer vision and machine learning using TensorFlow to develop a Lampung character recognition model. Computer vision, dedicated to processing visual data like images and videos, converges with machine learning, a subset of artificial intelligence enabling systems to acquire knowledge from data, into TensorFlow [8]. Developed by the Google Brain Team, TensorFlow is a robust machine learning framework [9] that facilitates the training using samples of handwritten Lampung characters.

The integration of TensorFlow introduces a cutting-edge dimension to this endeavor, enhancing the potential for learning and recognizing Lampung characters. TensorFlow can leverage Convolutional Neural Network (CNN), a deep neural network designed for processing structured grid data, such as images [10]. CNNs have demonstrated high efficacy in various computer vision tasks, including image classification, object detection, and image recognition.

In conclusion, this research represents a vital response to the urgent need to preserve the Lampung script. By deploying an innovative pattern recognition model, using TensorFlow, and drawing insights from prior studies, the research addresses the contemporary challenges of cultural neglect amid societal transformations and technological evolution.

METHOD

The research into the Lampung Character Recognition Model employs the Design Science Research (DSR) methodology, with the following section providing a comprehensive

breakdown of the research stages [11]: Problem Identification and Motivation, Define Objectives for a Solution, Design and Development, Demonstration, Evaluation, and Communication, as shown in **Figure 1**.

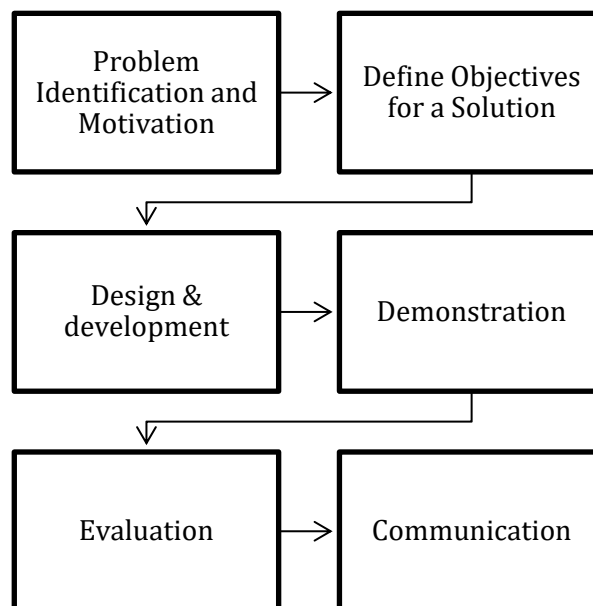


Figure 1. DSRM Process Model

Activity 1: Problem Identification and Motivation

Define the specific research problem: The diminishing ability to read Lampung Characters poses a significant threat to Lampung culture. Lampung residents encounter challenges recognizing Lampung characters [12], particularly when parent characters are combined with child characters.

Activity 2: Define Objectives for a Solution

Infer objectives from the problem definition: The primary objective is to develop a TensorFlow-based Lampung Character Writing Recognition model capable of accurately identifying Lampung characters depicted in images on a computer system [13]. In this study, three characters, namely "Ka," "Sa," and "Ga," were selected from a set of 20 parent characters, along with their combinations with 12 child characters. This approach was chosen to expedite the research process due to time constraints and to conduct tests of the TensorFlow framework and CNN algorithm's feasibility for Lampung character recognition before scaling up to accommodate all characters. Data collection consisted of 3 parent characters (without combinations of child characters), each with 100 instances, and

3 parent characters with 12 combinations of child characters, each with 100 samples, totaling 3900 instances.

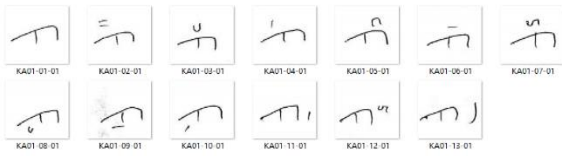


Figure 2. Lampung character "ka" along with 12 combinations of child characters

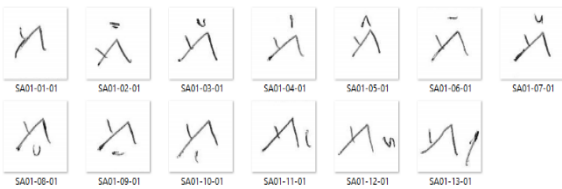


Figure 3. Lampung character "sa" along with 12 combinations of child characters

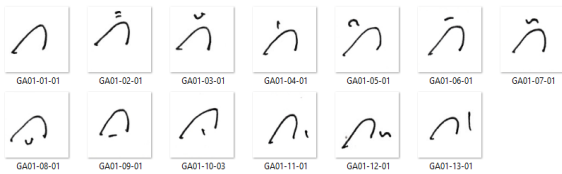


Figure 4. Lampung character "ga" along with 12 combinations of child characters

Activity 3: Design and Development

Create the artifact: Develop the Lampung Character Recognition model using the pattern recognition approach [14], involving dataset gathering and preprocessing phases.

a. Gathering Datasets

Data collection initiates with the recruitment of 100 respondents, aiming to obtain diverse Lampung script writing from various ages and regions, ensuring the uniqueness of the resulting dataset. Lampung characters are written on a dataset sheet with 130 boxes, each sized 60 pixels. The data encompasses handwritten Lampung script, such as Ka, Kan, Ki, Ke, Ke', Kang, Kar, Kau, Ku, Ko, Kai, Kah, K, Ga, Gan, Gi, Ge, Ge', Gang, Gar, Gau, Gu, Go, Gai, Gah, G, Sa, San, Si, Se, Se', Sang, Sar, Sau, Su, So, Sai, Sah, S. Each character is represented by 100 samples, scanned, cropped in each box, and saved in JPG format.

Formulir Dataset Tulisan Aksara Lampung Ka

KA01

Nama : Muhammad Hafiq Yusuf
 Jenis Kelamin : Laki - Laki
 Pekerjaan : Mahasiswa
 Ummat : Islam
 Domisili Asal : Palembang

Aksara	1	2	3	4	5	6	7	8	9	10
ka										
kan										
ki										
ke										
ke'										
kang										
kar										
kau										
ku										
ko										
kai										
kah										
k										
ga										
gan										
gi										
ge										
ge'										
gang										
gar										
gau										
gu										
go										
gai										
gah										
g										
sa										
san										
si										
se										
se'										
sang										
sar										
sau										
su										
so										
sai										
sah										
s										

Figure 5. Lampung Character Dataset sheet

b. Preprocessing

At this stage, the dataset transitions from image format to a matrix or 2-dimensional array, Lampung Character "SAN" as data example shown in Figure. Subsequently, the dataset was divided into training, validation, and test data. Then, feature extraction is performed to obtain unique values from the object.



Figure 6. Lampung Character "SAN" Data

Activity 4: Demonstration

Demonstrate the artifact's use: Employ the Lampung Character Recognition System in experimentation, highlighting its effectiveness in solving instances of the problem utilizing the recognition phases in the pattern recognition approach. Data for training constitutes 80% and 10% for validation.

a. Classification (Training Dataset)

The program backend utilizes the deep learning method with Keras and TensorFlow. Sixteen batches, each comprising 195 samples, are employed for a single iteration of data

training. Ten epochs are applied, resulting in 20 iterations.

The sequential model includes:

1) Convolutional Neural Network (CNN) layers

The CNN layers include two convolution layers and max pooling, utilizing the Rectified Linear Unit (ReLU) activation function, defined as:

$$f(x) = \max(0, x)$$

Where x represents the input to the function, the convolution layers extract attributes from images, and max pooling reduces pixels or image resolution, expediting the training process. The ReLU activation function ensures the maximum value selected from each node calculation result.

2) Dropout Layers

Dropout layers are incorporated to prevent overfitting. Overfitting occurs when a model performs well on training data but falters on test or new data. The dropout mechanism temporarily removes hidden layers randomly during calculation, preventing each neuron's static production of weights.

Activity 5: Evaluation

Observe and measure artifact effectiveness: Compare the solution objectives to actual results obtained through the Lampung Character Recognition model's use, marking the concluding phase of the pattern recognition approach. The results will dictate whether iteration for improvement is necessary or progression based on evaluation outcomes.

Activity 6: Communication

Communicate the research: Share the defined problem, the significance of the artifact, its utility, novelty, design rigor, and effectiveness, tailoring communication to researchers and other audiences while considering disciplinary culture—structure scholarly publications in alignment with the DSR process.

RESULT AND DISCUSSION

Results and Discussion will focus on Demonstration and Evaluation Activities.

Demonstration

a. Dataset Division

The Lampung script recognition system was developed using the Keras library and TensorFlow as the program backend, integrating TensorFlow version 2.0 with the Keras library and utilizing the Python programming language. The collected dataset was divided into training, validation, and test data at an 80:10:10 ratio, resulting in 3120 images for training, 390 for validation, and 390 for testing.

```
#load library untuk train test split
from sklearn.model_selection import train_test_split

[ ] #variabel yang digunakan pada pemisahan data ini
X= df['path']
y= df['tag']

[ ] # split dataset awal menjadi data train dan test
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, random_state=42)

[ ] # kemudian data test dibagi menjadi 2 sehingga menjadi data test dan data validation.
X_test, X_val, y_test, y_val = train_test_split(
    X_test, y_test, test_size=0.5, random_state=42)
```

train size 3120
val size 390
test size 390

Figure 7. Source Code for dividing the dataset

b. Preprocessing

The training process involved 39 classes representing various Lampung characters. Image data augmentation was applied using parameters from the Keras library's Image Data Generator to enhance data accuracy.

```
[60] train_datagen = ImageDataGenerator(rescale=1. / 255,
                                       rotation_range=40,
                                       shear_range=0.2,
                                       zoom_range=0.2,
                                       horizontal_flip=False)

val_datagen = ImageDataGenerator(rescale=1. / 255,
                                 rotation_range=40,
                                 shear_range=0.2,
                                 zoom_range=0.2,
                                 horizontal_flip=False)

test_datagen = ImageDataGenerator(rescale=1. / 255,
                                  rotation_range=40,
                                  shear_range=0.2,
                                  zoom_range=0.2,
                                  horizontal_flip=False)
```

Figure 8. Source Code for image data augmentation

c. Network Architecture

The classification model, structured as a sequential model, featured two convolution layers, two max-pooling layers, and additional components such as neuron input, dropout, and softmax activation for output detection.

```

model = Sequential()
model.add(Conv2D(128, (3, 3), padding='same', input_shape=input_shape))

model.add(Activation('relu'))
model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(num_class))
model.add(Activation('softmax'))
    
```

Figure 9. Source Code for CNN architecture

3) Activation Function

The output layer employed the softmax activation function to extract the maximum value from the calculation results at each node, akin to the ReLu activation function utilized in the hidden layer. ReLu is commonly known as the most straightforward and most widely employed non-linear activation function in constructing network architectures. The ReLu formula is as follows:

$$f(x) = \max(x, 0)$$

The symbol x denotes the input value. The number 0 (zero) functions as a linear unit; if the input is less than zero, the output takes on a zero value. Conversely, if the input value exceeds zero, the output mirrors the input.

d. Model Training

Transfer learning was employed using the pre-existing MobileNet V2 network model to expedite training and enhance accuracy. Transfer learning utilizes pre-existing network models to accelerate and enhance the accuracy of the training process [15].

The model underwent training using 80% of the data for training and 10% for validation, utilizing sixteen batches, each comprising 195 samples in one iteration and ten epochs, resulting in 20 iterations.

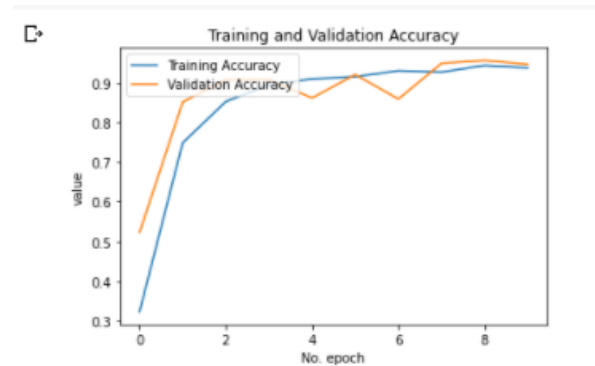
Some of the outcomes from the model training process are depicted in Figure 6. The initial epoch took 1379 seconds to complete, by approximately 23 minutes, with each subsequent iteration requiring 7 seconds. They are attributed to the model's ongoing recognition and computation of initial weights derived from the output data without modification. The time required increases with each iteration, averaging 406 seconds, approximately 7 minutes, per epoch, as the model refines the output weight value from the initial epoch.

```

Epoch 1/10
195/195 [-----] - 1379s 7s/step - loss: 2.3413 - accuracy: 0.3231 - val_loss: 1.7004 - val_accuracy: 0.5231
Epoch 2/10
195/195 [-----] - 404s 2s/step - loss: 0.8359 - accuracy: 0.7484 - val_loss: 0.6023 - val_accuracy: 0.8513
Epoch 3/10
195/195 [-----] - 406s 2s/step - loss: 0.4914 - accuracy: 0.8526 - val_loss: 0.2786 - val_accuracy: 0.9077
Epoch 4/10
195/195 [-----] - 406s 2s/step - loss: 0.3725 - accuracy: 0.8939 - val_loss: 0.3688 - val_accuracy: 0.9077
Epoch 5/10
195/195 [-----] - 403s 2s/step - loss: 0.3348 - accuracy: 0.9096 - val_loss: 0.5073 - val_accuracy: 0.8615
Epoch 6/10
195/195 [-----] - 406s 2s/step - loss: 0.3165 - accuracy: 0.9151 - val_loss: 0.2471 - val_accuracy: 0.9285
Epoch 7/10
195/195 [-----] - 406s 2s/step - loss: 0.2619 - accuracy: 0.9381 - val_loss: 0.6059 - val_accuracy: 0.8590
Epoch 8/10
195/195 [-----] - 404s 2s/step - loss: 0.2611 - accuracy: 0.9266 - val_loss: 0.2259 - val_accuracy: 0.9487
Epoch 9/10
195/195 [-----] - 407s 2s/step - loss: 0.2034 - accuracy: 0.9436 - val_loss: 0.1857 - val_accuracy: 0.9564
Epoch 10/10
195/195 [-----] - 407s 2s/step - loss: 0.2175 - accuracy: 0.9372 - val_loss: 0.2278 - val_accuracy: 0.9462
    
```

Figure 10. Model training results

The accuracy and loss results exhibited improvement over the ten epochs, indicating a well-performing model devoid of signs of overfitting or underfitting [16]. Figure 7 illustrates the accuracy results of the model training, with the accuracy of the training data at 0.323 and the accuracy of the validation data at 0.523 during the first epoch. By the tenth epoch, the accuracy level of the validation data had increased to 0.946 and the training data to 0.943.



```

[44] history.history['accuracy']

[0.32307693362236023,
0.7483974099159241,
0.8525640964508057,
0.8939102292060852,
0.9096153974533081,
0.9150640964508057,
0.9381282167434692,
0.9266025424003601,
0.9435897469520569,
0.9371795058250427]
    
```

Figure 11. Accuracy in model training

In the first epoch, the validation data had a 1.70 value, and the training data had a 2.34 model training loss value (Figure 8). The tenth epoch saw a notable drop in the loss value, with the training data showing a loss value of 0.21 and the validation data showing a loss value of 0.22. According to the accuracy and loss values found, there is neither overfitting nor underfitting [17], [18].

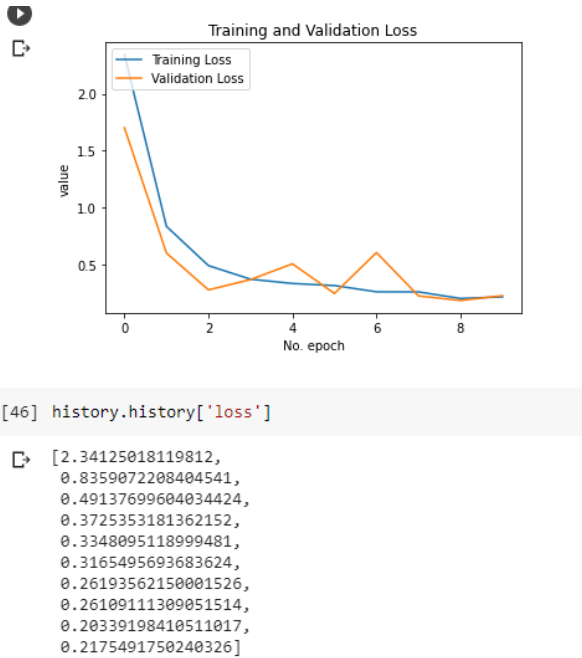


Figure 12. Loss of model training

Evaluation

The model underwent evaluation using a distinct set of test data that remained untouched during the training phase. The results showcased a commendable accuracy of 0.9564 and a minimal loss of 0.1442, underscoring the reliability and robustness of the final model and employing fresh data to gauge the extent of accuracy loss and training effectiveness.

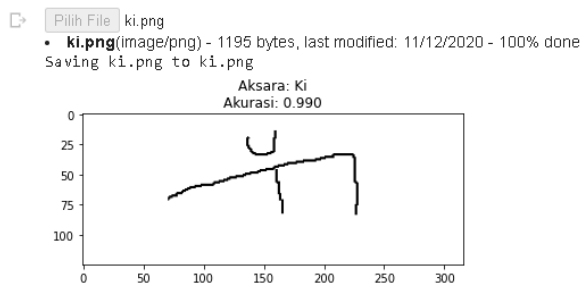


Figure 136. Lampung Character test result

The model evaluation outcomes with test data are illustrated in Figure 9. The accuracy level achieved was 0.9256, accompanied by a loss of 0.1917. Subsequently, as stated before, testing yielded an increased accuracy of 0.9564 and a reduced loss of 0.1442. These findings unequivocally establish the final model as reliable and well-constructed. The trained model is stored in the model.ipynb format, weighing in at 31 MB.

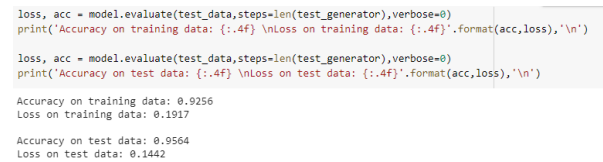


Figure 14. Model evaluation results

Five new data samples underwent testing, resulting in accurate recognition for all characters except "Kai." The system adeptly interprets the intended output, as evidenced by the test results in Table 1. Despite the Kai script exhibiting a 0.652 accuracy, every other sample demonstrated a precision rate exceeding 0.900. This variance might stem from discrepancies between the test and training input data [19], [20], indicating the need for new data to elevate all accuracy levels to 0.900 or above. Nevertheless, aside from this aspect, the constructed model performs admirably well. It's worth noting that while the amount of data for "Kai" was consistent with other characters ("G", "Sa", "San", and "Ko"), the resemblance of the "K" character with the "Kai" character could have contributed to the discrepancy in accuracy observed during testing.

Table 1. Model Test Result

No	Expected Output	System Output	Accuracy	Correct?
1	Character "G"	Character "G"	0.973	TRUE
2	Character "Sa"	Character "Sa"	0.998	TRUE
3	Character "San"	Character "San"	0.907	TRUE
4	Character "Kai"	Character "Kai"	0.652	TRUE
5	Character "Ko"	Character "Ko"	0.993	TRUE

CONCLUSION

This research marks a substantial leap forward in Lampung character recognition. The developed TensorFlow-based handwriting recognition model demonstrates commendable operational smoothness, achieving an accuracy spectrum ranging from 0.652 to 0.998. While celebrating this robust performance, it is crucial to delve into the nuances within this spectrum, understanding variations across different characters and stages of the model. The meticulous generation of a comprehensive dataset encompassing 3900 characters further strengthens our research. This dataset, comprising three main letters and their combinations with twelve sub-letters, is a valuable resource for subsequent studies, contributing significantly to the expanding repository of Lampung script data. While affirming the practicality and reliability of the TensorFlow model, it's essential to acknowledge potential areas for refinement. A more detailed exploration of variations in accuracy across specific characters or scenarios could enhance the precision of our conclusions. Additionally, recognizing any identified limitations or areas for improvement ensures a balanced perspective.

The research outcomes express the viability of the TensorFlow model and establish a solid foundation for future explorations in preserving and revitalizing Lampung cultural heritage. The fusion of advanced machine learning techniques with cultural preservation efforts signifies a promising intersection of technology and tradition, securing the enduring legacy of Lampung characters amidst the evolving landscapes of societal and technological transformations.

REFERENCES

- [1] P. Y. Bagaskara, M. A. Muhammad, M. Mardiana, and M. Komarudin, "Virtual Keyboard Design of Lampung Script Based on Android," *IJECS*, vol. 2, no. 1, pp. 15–22, Jun. 2022, doi: 10.24042/ijeecs.v2i1.11648.
- [2] Mardiana, F. A. Arizal, M. A. Muhammad, Martinus, and G. P. Djausal, "Word Per Minute (WPM) Lampung Script Keyboard," *TEPIAN*, vol. 1, no. 3, pp. 79–83, Aug. 2020, doi: 10.51967/tepi.v1i3.147.
- [3] S. Ferbangkara *et al.*, "Usability of Lampung Heritage Virtual Reality Tour," *JESR*, vol. 4, no. 2, Jan. 2023, doi: 10.23960/jesr.v4i2.107.
- [4] BPS Provinsi Lampung, "Provinsi Lampung dalam Angka 2015," BPS Provinsi Lampung, Bandar Lampung, 1102001.18, 2015.
- [5] N. Sazqiah *et al.*, "Pengenalan Aksara Lampung Menggunakan Metode CNN (Convolutional Neural Network)," *SNIP*, vol. 2, no. 1, Apr. 2022, doi: 10.23960/snip.v2i1.165.
- [6] P. Bintoro and A. Harjoko, "Lampung Script Recognition Using Convolutional Neural Network," *Indonesian J. Comput. Cybern. Syst.*, vol. 16, no. 1, p. 23, Jan. 2022, doi: 10.22146/ijccs.70041.
- [7] Atharva Hase, Viral Jain, Parag Gujarathi, Aditi Bhor, and Swati Bhonde, "Devanagari Character Recognition Using Deep Learning," *IJAR SCT*, pp. 215–220, Jun. 2022, doi: 10.48175/IJAR SCT-4585.
- [8] P. Nayak and S. Chandwani, "Improved Offline Optical Handwritten Character Recognition: A Comprehensive Review using Tensorflow," *International Journal of Engineering Research*, vol. 10, no. 11.
- [9] TensorFlow, "TensorFlow," TensorFlow. Accessed: May 28, 2023. [Online]. Available: <https://www.tensorflow.org/>
- [10] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 1571–1580. doi: 10.1109/CVPRW50498.2020.00203.
- [11] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, Dec. 2007, doi: 10.2753/MIS0742-1222240302.
- [12] N. Mawadda, C. Anwar, and A. Jatmiko, "Implementation of Religious Character Values in Scout Activities at Junior High School South Lampung," *Bulletin of Science Education*, vol. 4, no. 1, pp. 135–146, 2024.

- [13] S. Pattanayak, "Introduction to deep-learning concepts and TensorFlow," in *Pro Deep Learning with TensorFlow 2.0: A Mathematical Approach to Advanced Artificial Intelligence in Python*, Springer, 2023, pp. 109–197.
- [14] F. T. Anggraeny, Y. V. Via, and R. Mumpuni, "Image preprocessing analysis in handwritten Javanese character recognition," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 2, pp. 860–867, 2023.
- [15] Y. Xie *et al.*, "Cross-dataset transfer learning for motor imagery signal classification via multi-task learning and pre-training," *Journal of Neural Engineering*, vol. 20, no. 5, p. 056037, 2023.
- [16] N. Umeorah, P. Mashele, O. Agbaeze, and J. C. Mba, "Barrier options and Greeks: Modeling with neural networks," *Axioms*, vol. 12, no. 4, p. 384, 2023.
- [17] Z. Xu *et al.*, "Agile and accurate CTR prediction model training for massive-scale online advertising systems," presented at the Proceedings of the 2021 international conference on management of data, 2021, pp. 2404–2409.
- [18] R. L. Vallejo *et al.*, "The accuracy of genomic predictions for bacterial cold water disease resistance remains higher than the pedigree-based model one generation after model training in a commercial rainbow trout breeding population," *Aquaculture*, vol. 545, p. 737164, 2021.
- [19] M. C. Frank, M. Braginsky, D. Yurovsky, and V. A. Marchman, *Variability and consistency in early language learning: The Wordbank project*. MIT Press, 2021.
- [20] G. Orrù, M. Monaro, C. Conversano, A. Gemignani, and G. Sartori, "Machine learning in psychometrics and psychological research," *Frontiers in psychology*, vol. 10, p. 2970, 2020.