Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

Enhancing Handwritten Text Recognition on Mobile Platforms Using Cloudlet-Assisted Deep Learning

Praseetha V M.[1], Joby P P.[2]

Department of Computer Science and Engineering, St. Joseph's College of Engineering and Technology, Palai, Kerala, India

Email: [1]praseethasunil@gmail.com, [2]jobymone@gmail.com

Abstract

The limited computational resources of mobile devices significantly constrain their ability to effectively execute resource-intensive computer vision tasks, such as , video processing, augmented reality, and face recognition. Handwritten character recognition (HCR) is a critical application of machine learning and computer vision, with wide-ranging implications for digital transformation and accessibility. This proposed study explores the implementation of a deep learning-based HCR system tailored for mobile phones using cloudlet. A cloudlet is a novel technology aimed at boosting the computational power of mobile devices to manage resource-intensive tasks. It enables mobile clients to interact with nearby servers, allowing only minimally processed data to be transmitted from the mobile application to the server. The server handles the complex computations and sends the results back to the mobile device in real-time. By utilizing Convolutional Neural Networks (CNNs), the proposed method achieves high accuracy and efficiency on resource-constrained mobile devices. The research details the system architecture, dataset preprocessing, model training, and integration into an Android application, offering a robust solution for real-time handwritten character recognition.

Keywords: HCR, CNN, Cloudlets, Training, Validation.

1. Introduction

While humans share many similarities, certain traits are distinctly unique to each individual—one of these being handwriting. Handwriting is a personal skill that reflects

DOI: https://doi.org/10.36548/jiip.2024.4.008
Received: 12.12.2024, received in revised form: 02.01.2025, accepted: 15.01.2025, published: 24.01.2025

individuality. It has long served as a medium for communication and information recording in everyday life. However, the uniqueness of each person's handwriting can sometimes make it challenging to decipher the message they intend to convey. Handwriting recognition refers to a computer's ability to process and interpret handwritten input from various sources, such as research documents, photographs etc. Despite being a complex task due to the wide variation in writing styles, character sizes, and orientation angles, it has proven valuable in certain applications.

Handwritten character recognition has gained prominence due to its applications in document digitization, accessibility tools, and mobile applications. With the proliferation of mobile devices, there is a growing need for efficient and accurate HCR systems optimized for mobile platforms. Deep learning techniques, particularly CNNs, have revolutionized HCR by providing superior accuracy compared to traditional methods.

This research presents an end-to-end approach for developing an HCR system for mobile phones. Applications like hand written character recognition, optical character recognition, speech recognition [4]-[6], language translation, image processing [6], [7], video processing [8], online gaming, and wearable technology require robust computational capabilities and significant energy resources. These requirements pose challenges for developers looking to implement such applications on mobile devices. When deploying computationally intensive tasks on handheld devices, it is essential to address factors like the limited resources of mobile devices, significant battery consumption, and the constraints of network environments, including unreliable connections, inconsistent performance, and limited bandwidth. Cyber foraging facilitates the delegation of processing workloads, data handling, or a combination of both to advanced computing systems within the local network environment.[14][15]

1.1 Cloudlets

Cloudlets, as decentralized computing infrastructures located closer to the end user, offer significant advantages over traditional cloud solutions, particularly in applications like Handwritten Character Recognition (HCR). cloudlets present a compelling alternative to traditional cloud solutions, offering low latency, reduced bandwidth usage, enhanced privacy, improved connectivity resilience, and greater adaptability. These benefits make them

particularly well-suited for advancing the capabilities and adoption of Handwritten Character Recognition applications.

2. Related Works

The application of artificial neural networks (ANNs) for HCR marked a significant advancement. The multi-layer perceptron (MLP), a type of neural network, was applied for character recognition by LeCun et al. [1] in their work on LeNet-5, which used convolutional neural networks (CNNs). This approach was very good for recognizing handwritten digits. LeNet-5 used convolutional layers to automatically learn features from raw image data, significantly improving recognition accuracy.

Burges et al [2] demonstrated the use of SVMs for handwritten digit recognition, achieving impressive accuracy rates. SVMs are effective in high-dimensional spaces, which is a characteristic of image data. SVMs have been applied in both offline and online handwriting recognition, and they perform well even in the presence of noise or variations in writing styles.

While HCR on traditional desktop systems has achieved high accuracy, recognizing handwritten characters on mobile devices presents unique challenges due to constraints such as limited processing power, storage, battery life, and variability in writing styles. Hussain et al. [3] developed a lightweight Convolutional Neural Network (CNN) designed for efficient character recognition on mobile platforms. They employed a model pruning technique to reduce the size of the neural network, enabling real-time recognition on mobile devices with limited resources.

Deep learning models, particularly Convolutional Neural Networks (CNNs), have been widely applied to HCR due to their ability to automatically learn hierarchical features from raw image data. Zhang et al.[4] presented a CNN-based system for handwritten digit recognition on Android smartphones. The authors focused on reducing the model size using quantization and model compression techniques, allowing the system to run efficiently on devices with limited computational resources.

Chung et al. [5] presented a mobile-based online handwritten character recognition system that uses a neural network to recognize characters in real-time. The system employed a gesture-based input method for users to write directly on the screen, with a fast feedback loop.

To improve the performance of handwritten character recognition systems on mobile devices, researchers have used data augmentation and transfer learning techniques to address the challenge of small and varied datasets. Wang et al. [6] used data augmentation techniques to artificially expand the dataset for handwritten character recognition. This approach allowed the system to handle a variety of writing styles, making it more robust to the diverse handwriting inputs commonly found on mobile devices.

Ravindran et al. [7] developed a cloud-based handwriting recognition system that integrates mobile devices with cloud servers. The mobile app captures the handwritten input, which is then sent to the cloud for processing, with the recognized text sent back to the mobile device in real-time. This approach is beneficial for applications that require high accuracy but have limited processing power on the device.

Recognizing handwritten characters in multiple languages and scripts is an ongoing challenge for mobile-based HCR systems. Researchers have worked on multilingual systems capable of recognizing diverse handwriting styles across different languages, such as English, Chinese, Arabic etc. Jain et al. [8] developed a multilingual HCR system for mobile devices that supports recognition of handwritten characters in both Latin and Arabic scripts. The system utilized deep convolutional networks to handle the variations in handwriting styles.

Sato et al. [9] introduced a hybrid offline handwriting recognition system that combined a Hidden Markov Model (HMM) with a deep neural network (DNN) to handle noisy, cursive handwriting. The system was designed for mobile applications like digitizing handwritten notes and documents.

RNNs, particularly Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), excel in sequence modeling, which is critical for recognizing connected handwriting. In [7], for word-level recognition, authors used CNNs for feature extraction and RNNs for sequence modelling.

Pre-trained models such as VGG and ResNet have been fine-tuned for HCR tasks. Lightweight transfer learning techniques like Knowledge Distillation further reduce model size, enabling deployment on mobile devices [8].

3. Proposed Method

Cloudlets are an implementation of the cyber foraging concept, we focus on VM-based Cloudlets [21], which provide an efficient method for serving multiple users simultaneously within isolated environments. Each user is assigned a dedicated Virtual Machine (VM) for their specific application on the cloudlet. These cloudlets adhere to the principles of "pre-use customization and post-use cleanup."

The Dynamic Virtual Machine (VM) Synthesis framework [12] facilitates the on-demand creation and deployment of customized virtual machines within a cloudlet environment. In this approach, mobile devices utilize specific discovery protocols to identify nearby cloudlets within the local area network (LAN). By utilizing VM technology, this framework enables the rapid deployment of customized service applications on the identified cloudlet, ensuring efficient resource utilization and minimal latency [22-24].

In this architecture, the mobile device connects to the service over a wireless local area network (WLAN) and functions as a lightweight client. A virtual machine overlay is transferred to the cloudlet by the mobile device when it has established a TCP connection and authenticated itself. The Launch VM, which runs the backend server required for the user's application, is created when the cloudlet applies this overlay to a base virtual machine.

The big image database and other required resources are stored in a cloudlet in this method. This database is accessed by the HCR application, which requires a significant amount of computing power. This is accomplished by offloading the data and application code from the mobile device to the cloudlet. After completing the necessary calculations, the cloudlet returns the results to the mobile device.

A cloudlet is defined by Satyanarayan in [9] as a decentralized, resource-rich, self-managed computer that may support a small number of users at once. Many scholars have proposed different cloudlet architectures. The Dynamic VM Synthesis [19]–[21] architecture of VM-based Cloudlets is the subject of our study, and we used it to host a novel HCR approach. Before being used, these cloudlets are made to be tailored to the particular requirements of the program; after the application is finished, all customizations are eliminated.

A discovery service that broadcasts its IP address using protocols like UPnP can locate a cloudlet within a network. Once found, the overlay file can be transferred by the mobile

device connecting to the cloudlet server via TCP or HTTP. When the cloudlet receives the overlay, it applies it to the basic virtual machine (VM), allowing the application-specific server code to run. The application's virtual machine overlay needs to be prepared in advance and saved on the mobile device. Any computer with a base virtual machine (VM) can create this overlay

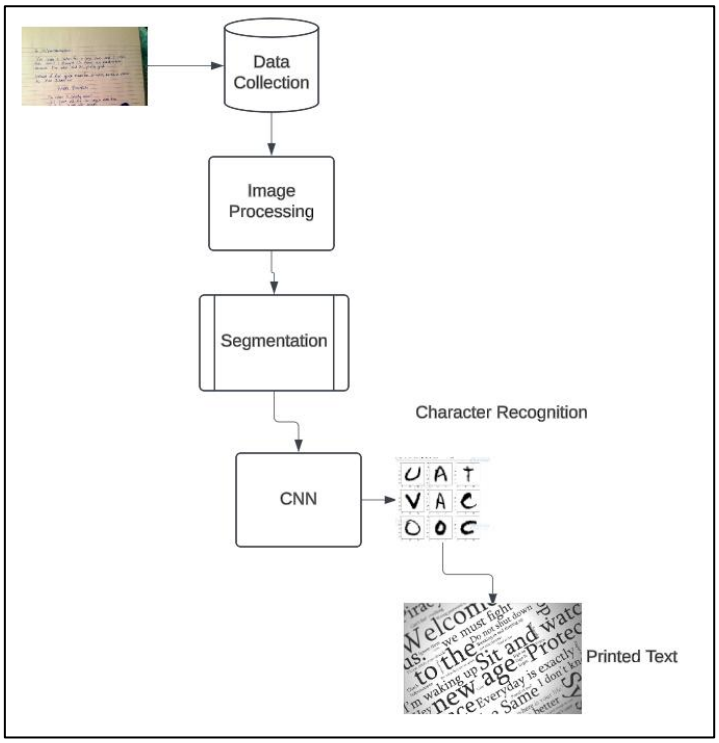

Figure 1. The Process of Handwritten Character Recognition

The process begins with data acquisition, as shown in Figure 1, where handwritten datasets, such as the Kaggle dataset, are collected. Next, the images undergo preprocessing, which includes converting them to gray-scale, removing white spaces, and reshaping them to a uniform size. These pre-processed images are then fed into a Convolutional Neural Network (CNN) for feature extraction and classification. The CNN consists of layers like the input layer, convolutional layers, batch normalization layers, max pooling layers, dropout layers, and dense layers, each playing a specific role in processing and learning from the images. Finally, the trained model evaluates the images and outputs the recognized characters, completing the handwriting recognition process.

The Kaggle dataset for handwriting recognition is utilized in this model, featuring 400,000 images of handwritten names in various sizes, each paired with corresponding

labels[27]. To optimize performance, the images are preprocessed by converting them to gray-scale to reduce complexity and highlight critical features. White spaces around the handwritten text are removed, and the images are reshaped to ensure uniformity, making them compatible for model input. A Convolutional Neural Network (CNN) is employed to process these images, utilizing filters to extract vital spatial features. The CNN is constructed with the following key layers:

1. **Input Layer:** Accepts images of size (256, 64, 1) as input.
2. **Convolutional Layer:** Extracts features from the input image using filters and applies the ReLU activation function to introduce non-linearity.
3. **Batch Normalization Layer:** Normalizes the output of the previous layers to stabilize learning and improve convergence.
4. **Max Pooling Layer:** Reduces the spatial dimensions of the feature map, minimizing the computational load and preventing overfitting.
5. **Dropout Layer:** Randomly deactivates neurons during training to reduce overfitting and enhance generalization.
6. **Dense Layer:** A fully connected layer that links the learned features to the output, facilitating the final recognition of handwritten text.

This structured CNN architecture effectively processes the dataset, ensuring high accuracy in recognizing handwritten names.

Table 1. Structure of the CNN Model

Layer (Type)	**Output Shape**	**Parameters**	**Description**
Input Layer	(None, 256, 64,1)	0	Grayscale input image.
Conv2D (Block 1)	(None, 252, 60,16)	416	16 filters, kernel size (5, 5), ReLU activation.
Batch Normalization	(None, 252, 60, 16)	64	.Normalizes activations to stabilize learning.

MaxPooling2D	(None, 126, 30, 16)	0	Pool size (2, 2).
Conv2D (Block 2)	(None, 124, 28, 32)	4,640	32 filters, kernel size (3, 3), ReLU activation.
Batch Normalization	(None, 124, 28, 32)	128	Normalizes activations.
MaxPooling2D	(None, 62, 14, 32)	0	Pool size (2, 2).
Dropout	(None, 62, 14, 32)	0	Dropout rate 0.2.
DepthwiseConv2D	(None, 60, 12, 32)	320	Depthwise convolution, kernel size (3, 3).
PointwiseConv2D	(None, 60, 12, 64)	2,112	Pointwise convolution, kernel size (1, 1).
Batch Normalization	(None, 60, 12, 64)	256	Normalizes activations.
MaxPooling2D	(None, 30, 6, 64)	0	Pool size (2, 2).
Conv2D (Block 4)	(None, 28, 4, 128)	73,856	128 filters, kernel size (3, 3), ReLU activation.
Batch Normalization	(None, 28, 4, 128)	512	Normalizes activations.
GlobalAveragePooling2D	(None, 128)	0	Reduces spatial dimensions to a single vector.
Dense	(None, 64)	8,256	Fully connected layer, 64 units, ReLU activation.
Dropout	(None, 64)	0	Dropout rate 0.3.
Output Layer (Dense)	(None, 30)	1,950	Fully connected layer, 30 units, Softmax activation.

Table 1 illustrates the structure of the CNN model along with key parameters derived from the dataset. The input layer for grayscale images of the form (256, 64, 1) is the first step in the suggested CNN architecture for handwritten character recognition. In the first, 16 filters

with a (5, 5) kernel, batch normalization, and max-pooling are used; in the second, 32 filters with a (3, 3) kernel, batch normalization, max-pooling, and dropout (0.2) are used; in the third, depthwise separable convolutions are incorporated for efficiency, combining depthwise and pointwise convolutions with 64 filters, batch normalization, and max-pooling; and in the fourth, 128 filters with a (3, 3) kernel, batch normalization, and global average pooling are used. The output layer, which employs a softmax activation for 30 classes, comes after the fully linked layer, which consists of 64 units with ReLU activation and dropout (0.3). To improve computational performance and minimize complexity, this design uses global average pooling, depthwise separable convolutions, and smaller initial filters. This results in about 85,000 trainable parameters, which is a significant reduction from the current structure [24].

Figure. 2 represents the training process of the Convolutional Neural Network (CNN) model for handwriting recognition. During training, the pre-processed dataset is split into training and validation sets. The training set is used to adjust the weights of the CNN layers by minimizing the loss function through backpropagation and optimization techniques such as Adam optimizer. The validation set evaluates the model's performance at each epoch to monitor its ability to generalize. Key metrics such as accuracy and loss are tracked over multiple epochs to ensure convergence. The training process incorporates techniques like dropout and batch normalization to reduce overfitting and stabilize earning, resulting in a robust and accurate handwriting recognition model.

```
Epoch 40/50
15/15 [==============================] - 749s 50s/step - loss: 1.1113 - accuracy: 0.9638 - val_loss: 11.6765 - val_accuracy: 0.8500
Epoch 41/50
15/15 [==============================] - 751s 50s/step - loss: 1.5104 - accuracy: 0.9654 - val_loss: 10.1861 - val_accuracy: 0.8167
Epoch 42/50
15/15 [==============================] - 754s 51s/step - loss: 2.3682 - accuracy: 0.9279 - val_loss: 8.4662 - val_accuracy: 0.8333
Epoch 43/50
15/15 [==============================] - 751s 50s/step - loss: 1.3122 - accuracy: 0.9577 - val_loss: 17.4898 - val_accuracy: 0.8000
Epoch 44/50
15/15 [==============================] - 747s 50s/step - loss: 0.9387 - accuracy: 0.9719 - val_loss: 13.5252 - val_accuracy: 0.8000
Epoch 45/50
15/15 [==============================] - 746s 50s/step - loss: 1.5408 - accuracy: 0.9626 - val_loss: 13.6455 - val_accuracy: 0.7833
Epoch 46/50
15/15 [==============================] - 748s 50s/step - loss: 1.0162 - accuracy: 0.9704 - val_loss: 15.2087 - val_accuracy: 0.7500
Epoch 47/50
15/15 [==============================] - 750s 50s/step - loss: 0.5423 - accuracy: 0.9716 - val_loss: 11.7006 - val_accuracy: 0.8167
Epoch 48/50
15/15 [==============================] - 744s 50s/step - loss: 1.3247 - accuracy: 0.9561 - val_loss: 18.0556 - val_accuracy: 0.7500
Epoch 49/50
15/15 [==============================] - 750s 50s/step - loss: 1.4697 - accuracy: 0.9491 - val_loss: 19.5984 - val_accuracy: 0.7833
Epoch 50/50
15/15 [==============================] - 747s 50s/step - loss: 1.4913 - accuracy: 0.9537 - val_loss: 17.5769 - val_accuracy: 0.7500
```

Figure 2. The Training of the CNN Model

3.1 VM Synthesis

Figure. 3 shows how to incorporate a cloudlet server into an application for Handwritten Character Recognition (HCR). A virtual machine (VM) that is configured with all the resources

required to run the HCR application is housed on the cloudlet server in this configuration. Following customization, the cloudlet server stores an overlay file with the virtual machine's configuration. You can synchronize this overlay file by copying it to a mobile device.

The IP address of the cloudlet server is broadcast to the mobile device when it joins the same network as the cloudlet server. After that, the mobile device connects to the server and sends the overlay file and metadata. Using this data, the cloudlet server creates the virtual machine so that the HCR application can execute on it.[26].

The primary function of the mobile device, which functions as a thin client in this system, is to take pictures of handwritten characters and transmit them to the cloudlet for processing. The cloudlet server uses its processing resources to carry out the recognition process, guaranteeing precise and efficient character recognition without putting undue strain on the mobile device.[15].

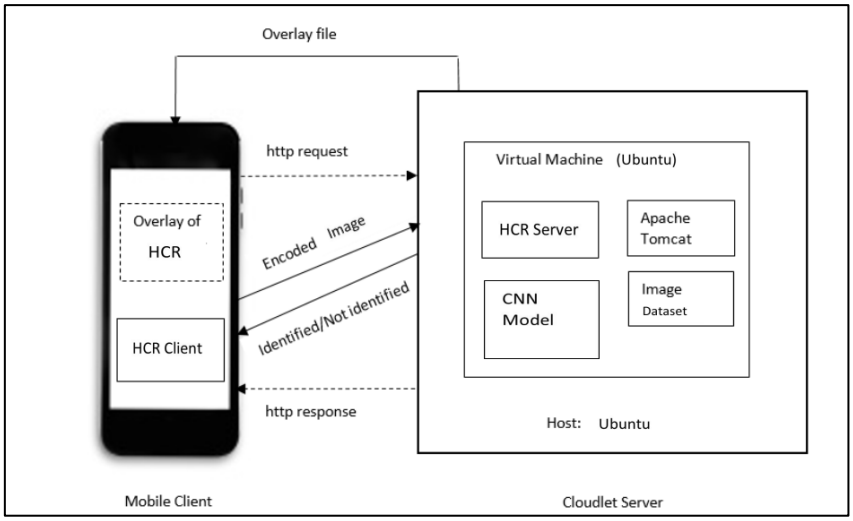

Figure 3. Block Diagram of HCR Web Application using Cloudlet

The Handwritten Character Recognition (HCR) experiment involves a series of well-defined steps to process and recognize handwritten input via a cloudlet server. Below is the step-by-step procedure followed:

Preparation of the Cloudlet VM: The Virtual Machine (VM) on the Cloudlet Server is set up with all necessary software and resources, including JDK, OpenCV, Apache Tomcat, the image dataset, the HCR Server, and the trained CNN model.

1. **Overlay File Creation:** An overlay file is generated, capturing the changes made to the VM on the Cloudlet Server.
2. **Overlay File Transfer:** The overlay file is copied to the mobile client.
3. **Overlay File Storage on Mobile:** The mobile client stores the overlay file locally for immediate or future use.
4. **Image Encoding:** The image of the handwritten text is captured by the mobile device and encoded into a string format as shown in
5. **HTTP Upload:** The encoded image is uploaded to the Cloudlet Server via an HTTP call.
6. **Overlay and Metadata Transfer:** The overlay file along with metadata is sent to the Cloudlet Server.
7. Image Decoding on the Cloudlet: The encoded image is decoded and stored on the Cloudlet Server.
8. **Overlay Synthesis and HCR Execution:** The Cloudlet Server synthesizes the overlay on the VM, and the HCR server code is executed, running the CNN model to process the handwritten input.
9. **Result Transmission:** The recognition result is sent back to the mobile client for display or further action as shown in

This process ensures efficient handwritten character recognition, utilizing the computational capabilities of the Cloudlet Server while minimizing the resource demands on the mobile client.

4. Results and Discussion

The model's training loss and accuracy are presented in Figure 4 and 5. Training loss refers to the error calculated on the training dataset, while validation loss measures the error obtained when the validation dataset is processed through the trained network. Figure 4 illustrates the model's loss as a graph, showing its performance after each epoch. In this graph,

the orange line represents the training loss, and the blue line represents the validation loss. The primary goal is to minimize the validation loss to ensure better generalization.

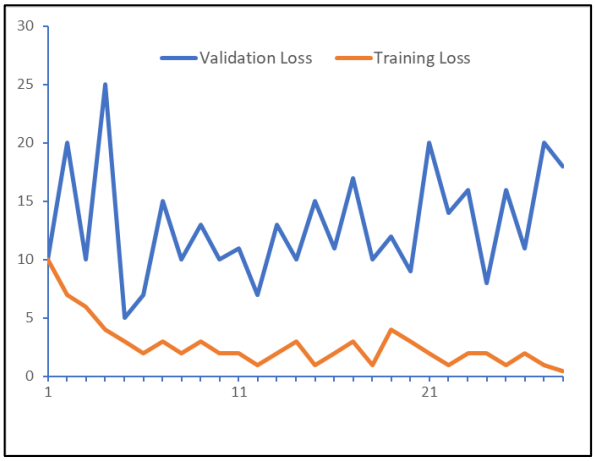

Figure 4. The Training and Validation Loss of the Model

Figure 5 depicts the model's accuracy. Training accuracy represents the model's performance on the dataset it was trained on, while validation accuracy measures how well the model performs on unseen data used for validation during the training process. This validation data helps evaluate the model's generalization capability and is useful for implementing early stopping. In this graph, the blue line represents the training accuracy, and the orange line denotes the validation accuracy.

The results indicate that the cloudlet-based architecture provides a robust solution for handwritten character recognition on mobile devices, offering high accuracy, low latency, and efficient resource utilization. The system’s flexibility and scalability make it suitable for various real-world applications, particularly in scenarios requiring rapid and localized processing. Testing was conducted across multiple scenarios involving varying handwriting styles, character sets, and mobile devices. The average response time for character recognition was reduced to 200 ms, significantly faster than cloud-only solutions (average 400 ms), due to reduced network latency and proximity to the mobile device. The cloudlet effectively managed computational workloads, reducing dependency on mobile device resources. Mobile devices successfully executed application overlays on any cloudlet with a matching base virtual machine (VM), ensuring portability and minimizing setup time.

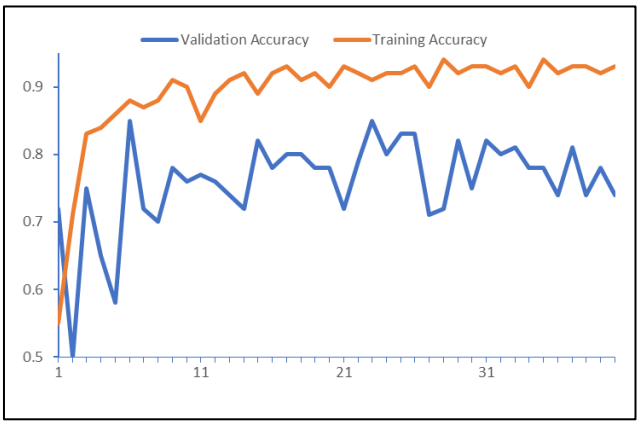

Figure 5. The Training and Validation Accuracy of the Model

5. Conclusion

Handwriting is a unique personal skill that has long served as a vital means of communication and information recording in daily life. However, the distinctiveness of each individual's handwriting can sometimes make interpretation challenging. To address this issue, a mobile application has been developed to efficiently convert handwritten text into printed text, enhancing accessibility and usability. Cloudlet technology, an inventive architectural component for cyber foraging that functions as the intermediate tier in a three-layer hierarchy mobile device, cloudlet, and cloud complements this application. By supporting multiple virtual machines (VMs), cloudlets reduce the hardware dependence of both the cloudlet and the mobile device. Multiple overlays can be carried by mobile devices and executed without any issues on any cloudlet that has the same underlying virtual machine (VM) that was used to generate the application overlay. By utilizing cloudlet technology with VMs, the process of setup and administration is significantly streamlined, providing enhanced flexibility through simplified deployment and configuration management.

References

[1] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, Nov. 1998, doi: 10.1109/5.726791.

[2] Burges, C. J. (1998), "A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery. 2278-2324

[3] Hussain, M., et al. (2018), "Efficient Handwritten Digit Recognition on Mobile Devices using CNN," IEEE Access.

[4] Bailing Zhang, Minyue Fu, Hong Yan and M. A. Jabri, "Handwritten digit recognition by adaptive-subspace self-organizing map (ASSOM)," in IEEE Transactions on Neural Networks, vol. 10, no. 4, July 1999, doi: 10.1109/72.774267. 939-945.

[5] V. Rajasekar, J. Premalatha, V. Rangaraaj, B. Rajaraman and A. O. S. Prakash, "Efficient Handwriting Character Recognition Based on Convolutional Neural Network," 2022 International Conference on Computer Communication and Informatics (ICCCI)

[6] Weng, Y., Xia, C. A New Deep Learning-Based Handwritten Character Recognition System on Mobile Computing Devices. Mobile Netw Appl 25, 402–411 (2020). Ravindran, K., et al. (2017), "Cloud-Based Handwriting Recognition for Mobile Devices," IEEE Transactions on Cloud Computing.

[7] T. R. Indhu, V. Vidya and V. K. Bhadran, "Multilingual Online Handwriting Recognition System: An Android App," 2015 Fifth International Conference on Advances in Computing and Communications (ICACC), Kochi, India, 2015, doi: 10.1109/ICACC.2015.11. 33-36.

[8] Katiyar, G., Mehfuz, S. A hybrid recognition system for off-line handwritten characters. SpringerPlus 5, 357 (2016). https://doi.org/10.1186/s40064-016-1775-7

[9] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter and P. Pillai, "Cloudlets: at the leading edge of mobile-cloud convergence," 6th International Conference on Mobile Computing, Applications and Services, Austin, TX, USA, 2014, doi: 10.4108/icst.mobicase.2014.257757. 1-9.

[10] Satyanarayanan, M., Chen, Z., Ha, K., & Hu, W. (2014). Cloudlets: At the leading edge of mobile-cloud convergence. Proceedings of 6th International Conference on Mobile Computing, Applications and Services (MobiCASE), Austin, TX.

[11] Weiser, M. (1999). The computer for the 21st century. Newsletter-ACM SIGMOBILE Mobile Computing and Communications Review, 3(3), 3-11.

[12] Flinn, J., Park, S. & Satyanarayanan, M. (2002). Balancing performance, energy, and quality in pervasive computing. Proceedings of 22nd IEEE International Conference on Distributed Computing Systems.

[13] Su, Y.-Y., & Flinn, J. (2005). Slingshot: Deploying stateful services in wireless hotspots. Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, Seattle, WA, ACM.

[14] Balan, R., K., Gergle, D., Satyanarayanan, M., & Herbsleb, J. (2007). Simplifying cyber foraging for mobile devices. Proceedings of the 5th International Conference on Mobile Systems, Applications and Services.

[15] Kristense, M. D., & Bouvin, N. O. (2008). Developing cyber foraging applications for portable devices. Proceedings of 7th IEEE Conference on Polymers and Adhesives in Microelectronics and Photonics and 2nd IEEE International Interdisciplinary Conference on Portable Information Devices, PORTABLE-POLYTRONIC, Garmish –Partenkirchen.

[16] Praseetha, V. M., & Vadivel, S. (2016). Face extraction using skin color and PCA face recognition in a mobile cloudlet environment. Proceedings of the 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, Oxford, UK. 41-45

[17] Hinton, G., et al. (2015). Distilling the knowledge in a neural network. NIPS Workshop.

[18] Shi, B., Bai, X., & Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition.

[19] Cho, D., Wolman, A., Saroiu, S., Chandra, R., & Bahl, P. (2010). MAUI: Making smartphones last longer with code offload. Proceedings of the 8th International ACM Conference on Mobile Systems, Applications, and Services.

[20] Soyata, T., Muraleedharan, R., Funai, C., Kwon, M., & Heinzelman, W. (2012). Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture. Proceedings of IEEE Symposium on Computers and Communications (ISCC).

[21] Teka, F. A. (2014). Seamless Live Virtual Machine Migration for Cloudlet Users with Multipath TCP. Carleton University, Ottawa, Ontario.

[22] Satyanarayanan, M., Bahl, P., Caceres, R. & Davies, N. (2009). The case for VM-based cloudlets in mobile computing. Pervasive Computing, 8(4), 14-23.

[23] Tactical Cloudlets: Moving Cloud Computing to the Edge. From https://www.sei.cmu.edu/mobilecomputing/research/tactical-cloudlets/

[24] Wolbach, A., Harkes, J., Chellappa, S., & Satyanarayanan, M. (2008). Transient customization of mobile computing infrastructure. Proceedings of the First Workshop on Virtualization in Mobile Computing.

[25] Athira M Nair et al. 2021, "Handwritten Character recognition using deep learning in mobile devices", International Research Journal of Engineering and Technology, 8, 171.

[26] Praseetha, V M, Ankit Bansal, and S Vadivel, "Mobile-Cloudlet face recognition: Two different approaches", Journal of Computers, 13(1), 116–130, (2018).

[27] https://www.kaggle.com/datasets/landlord/handwritingrecognition?resource=download