

Handwritten Japanese Kanji Character Recognition using different Pruning Algorithm

Shashank Pandkar¹, Kartik Sabane², Sachin Rathod³, Prafullchandra Bansode⁴ and Shalaka Deore⁵

¹⁻⁵Department of Computer Engineering, MES's Wadia College of Engineering, Pune

Email: pandkaeshashank@gmail.com, sr3630768@gmail.com, kartiksabane@gmail.com, prafulbansode1643@gmail.com, shalaka.deore@mescoepune.org

Abstract—With the advent of image recognition using deep learning, various applications of it are put forth. Optical Character Recognition as well as Handwritten Character Recognition is at the forefront of it. For this reason, optimized deep learning models are required to increase the accuracy of recognition. One such model is Capsule network. Instead of traditional CNNs which causes invariance, capsule networks provide equivariance. In terms of OCR, spatial arrangement of the feature is more important than features themselves. But capsule networks are computationally heavy. Hence, by applying pruning to a CapsNet model, we can reduce the computational costs and time complexity.

Index Terms— CNN, Capsule Network, Pruning, OCR, AARCN, Japanese Kanji.

I. INTRODUCTION

Japanese is the national language of Japan, where it is spoken by about 124 million people, making it the sixth most spoken language worldwide. Three different character sets are used in Japanese writing: hiragana, katakana, and kanji. Japanese writing uses logo graphic letters called kanji. Together with hiragana and katakana, these three scripts are used to write the Japanese language. "Han characters" is the literal meaning of the Japanese term "kanji." Chinese characters, which were brought to Japan more than a millennium ago, are the source of Kanji characters. They have been modified, nevertheless, to conform to Japanese terminology, pronunciation, and syntax. The characters of kanji can be very intricate.

There are thousands of various characters that are often used, and some characters have many strokes. Just like with Chinese characters in general, there is no definitive count of kanji characters. About fifty thousand characters make up the comprehensive Dai Kan-Wa Jiten available in Japan. About 85,000 characters make up the Zhonghua Zihai, which was released in China in 1994. However, most of these characters are uncommon varieties or archaic forms, and few are used regularly in any nation.

Recent advances in deep learning have led to the development of capsule networks (CapsNet), which are especially useful for applications involving organized and hierarchical input, such as character recognition. Create the network architecture for the capsule. Many layers of capsules make up a typical CapsNet design. Convolutional layers are an effective way to extract significant information from the unprocessed photos as the first layer. Apply the first layer of the capsule. Each capsule may identify unique characteristics of kanji characters and represent a portion of the character. Use the previously processed dataset to train the CapsNet. Compared to traditional convolutional neural networks, capsule networks could need bigger datasets and longer training periods.

II. MOTIVATION

Kanji recognition models are valuable for language learners. They can assist in language education, helping students learn the characters and their meanings. Kanji recognition can be integrated into translation services,



Figure 1. Comparison between Kanji, "Big" (right), "Dog" (middle) and "Fat" (left)

allowing users to translate Kanji text into other languages or vice versa. The major problems that deep learning model face when working on Japanese Kanji characters is that many of the characters share same features which mislead the model as shown in Figure 1.

Using capsule networks for Kanji recognition can offer several advantages and potentially address some of the challenges associated with recognizing Kanji characters [1]. Capsule networks are inherently more robust to variations in pose, orientation, and deformation. The primary concept behind the introduction of capsules was the development of a neural network with inverted graphics capabilities [2].

Pruning removes unnecessary parameters from the network. Capsule networks, like other neural network architectures, can become quite large as they learn to represent hierarchical structures in data [3]. Pruning can significantly reduce the model size, making it more memory-efficient and easier to deploy on resource-constrained devices. Pruning can help prevent overfitting, especially in cases where the network has too many parameters for the available training data.

III. LITERATURE SURVEY

An overview of current developments in routing techniques and capsule architecture is presented in paper [2]. Furthermore, it discovers that while altering routing protocols or architectural designs has received a lot of attention recently, other, more intricate details—specifically, squash function—have received less attention. The significance of capsule networks as a potential remedy for Convolutional Neural Networks' (CNNs') generalization problems is highlighted in the paper. It does not, however, go into certain restrictions. It would be crucial to include specific case studies or examples that highlight the situations in which CNNs fall short of generalizing and how capsules have solved these problems to increase the robustness of the research.

The importance of capsules and the expanding corpus of studies on them are acknowledged in the study. Another method for encapsulating spatial relationships in data is through capsule networks, and the summary of current developments in capsule architecture and routing techniques is a useful addition. Nam Tuan Ly and Masaki Nakagawa [4] explained the feature extractor employs a CNN to extract relevant features from the input images. self-attention-based encoder, which helps the model understand relationships within the sequence of extracted features. CTC decoder is responsible for generating the label sequence from the encoder's output. The advantage of this approach is the self-attention mechanism in the encoder helps the model capture complex dependencies in handwritten text, making it robust to various writing styles. Some limitations like model architecture being relatively complex, which could make it computationally expensive. AACRN's performance heavily depends on the quality and quantity of training data. Gathering a large, diverse dataset of handwritten Japanese text lines may be challenging. Solution is model optimization techniques like quantization and pruning can be applied. Augmenting the dataset by introducing variations in writing styles and other characteristics can improve model generalization.

Hao Ren et. al. [9] presented Evaluating Generalization Ability of Convolutional Neural Networks and Capsule Networks for Image Classification via Top-2 Classification. The Full Connectivity (FC) method, which the Capsule Network uses among all capsules, has numerous parameters. The ParameterSharing (PS) mechanism between capsules is introduced in this study in an effort to reduce the number of parameters. While the research emphasizes the benefits of using Capsule Networks (CapsNet) to mitigate Convolutional Neural Networks' (CNNs') vulnerability to image perturbations, it

skips over the specific drawbacks of using CapsNet. A thorough explanation of the possible disadvantages and difficulties associated with applying CapsNet to image classification is required. The research highlights the potential advantages of Capsule Networks over CNNs, particularly in terms of their superior capacity to characterize object properties. It is a significant advance to introduce the Top-2 classification challenge as a quantitative way to assess how well CNNs and CapsNet generalize.

Lobna et. al. in [13] explained model to detect COVID-19 with unbalanced datasets, this paper suggests using Capsule Neural Networks (CapsNet) in conjunction with a pre-trained Convolutional Neural Network (VGG16). The Gaussian optimization method has been used to optimize the parameters of this model, which employs SMOTE (Synthetic Minority Over-Sampling Technique) to minimize the likelihood of producing outlier data. The four primary phases of the suggested COVID-19 detection model are pre-trained, classification and optimization, evaluation, and data preparation. This study shows that adding a capsule network increases accuracy; nevertheless, there are some drawbacks as well, including unstable training, greater complexity, and inefficiency when handling spatial hierarchies.

In paper [14] discusses a modified version of capsule networks known as residue capsule networks. The primary aim of this paper is to reduce the size of capsule networks to be comparable to today's CNN models by minimizing the number of parameters required for training. RCNX utilizes 3-staged feature detectors where the output of each primary capsule is relayed to the next feature detector. Each detector network is linked to its own DigitCaps, resembling an ensemble model. The class predictions from all the DigitCaps are then fed into a fully connected network ending with a sigmoid activation function. During testing, RCNX achieved an accuracy of 89.31%, and the evaluation model size is 1.58 million parameters. One limitation of the model is its susceptibility to overfitting, making it prone to issues like Gradient Disappearance and Explosion. Additionally, it faces challenges related to Difficulty in Hyperparameter Tuning.

Morteza-Mousa-Pasand "Convolution Neural Network Pruning using Filter Attenuation" 6th June 2020 [6]. The Advantage of these approach is filter attenuation is that it allows for gradual pruning. Filters are not abruptly removed from the network, which makes performance drop during pruning. produces better results compared to conventional pruning methods, such as L1-Norm and standard deviation (STD) criteria. Some limitations like Implementing filter attenuation add complexity to the pruning process. Setting appropriate attenuation and pruning thresholds can be challenging. While fine-tuning can help recover accuracy, it can also be time-consuming and might require several iterations to achieve optimal results. The Solution is developing efficient algorithms and data structures to manage attenuated filters and facilitate quick recovery is essential. This can help reduce the computational overhead of filter attenuation. Explore optimization techniques that aim to reduce the number of iterations required for achieving optimal accuracy after pruning.

S. Rakshith, Jayesh Rajkumar, Vachhani, Sourabh Vasant Gothe, Rishabh Khurana. "Self-Similarity Matrix-Based CNN Filter Pruning." Samsung R&D Institute, Bangalore, India. 3 Nov 2022 [12]. In large CNN models, the major problem faced is the huge computational cost and time complexity required. To reduce these costs, quantization and pruning are the two methods used. Out of these two, pruning is more relevant as it allows the least amount of damage to the model's accuracy. In this paper, the method is based on similarity matrix pruning, specifically the self-similarity matrix. After CNN processing, the feature sequence of the sample is extracted. These n-features are then used to construct an $n \times n$ self-similarity matrix. The ranking of any two features is based on a function that returns a low value if two features are similar and a high value if they are dissimilar. This method not only removes the feature maps but also eliminates redundancies found in each feature map. However, in this method, the model can become sensitive to noise. Additionally, the ambiguous nature of the similarity threshold, which decides if two feature vectors are similar or dissimilar, poses a challenge. The primary goal of pruning is to reduce the computational size of the model while retaining its accuracy.

Arjun Narukkanchira Anilkumar, Mohamed El-Sharkawy. 'RCNX: Residual Capsule NeXt.' 2021 [14]. This paper discusses a modified version of capsule networks known as residue capsule networks. The primary aim of this paper is to reduce the size of capsule networks to be comparable to today's CNN models by minimizing the number of parameters required for training. RCNX utilizes 3-staged feature detectors where the output of each primary capsule is relayed to the next feature detector. Each detector network is linked to its own DigitCaps, resembling an ensemble model. The class predictions from all the DigitCaps are then fed into a fully connected network ending with a sigmoid activation function. During testing, RCNX achieved an accuracy of 89.31%, and the evaluation model size is 1.58 million parameters. One limitation of the model is its susceptibility to overfitting, making it prone to issues like Gradient Disappearance and Explosion. Additionally, it faces challenges related to Difficulty in Hyperparameter Tuning.

Zi Wang Chengcheng Li Xiangyang Wang "Convolutional Neural Network Pruning with Structural Redundancy Reduction" The University of Tennessee, Knoxville, TN, USA & Sun Yat-sen University, Guangzhou, China 8 Apr 2021 [15]. To quantify redundancy, compute the l-covering number and quotient space size, and create a graph for each convolutional layer in a CNN. Once redundancy has been determined, remove unnecessary filters from layers. Channel pruning eliminates all of a network's filters, negating the requirement for specialized hardware. The method focuses on calculating the amount of redundancy in each layer and removing certain filters from the layer(s) that are the most redundant. For this assessment, we create an undirected graph for every layer in which the vertices correspond to individual filters and the edges show the separations between filter weights. We measure the amount of redundancy in this graph using two metrics: the quotient space size and the l-covering number. These metrics serve as measurements of the redundancy within each layer. The results achieved are an increased accuracy of ResNet56 model by 0.37% and reduced FLOPs by 44.1%. Thus proving, Structural Redundancy Reduction pruning can improve accuracy and reduce FLOPs.

Main Contributions of our study are summarized below:

1. Created a new dataset of handwritten Japanese Kanji characters of 100 samples for 100 different characters, altogether total of 10000 labeled images which is publicly available.
2. Presented a CapsNet variant model to recognize Japanese Kanji handwritten characters. The model is designed to recognize equivariant patterns present in Japanese Kanjis and a self-attention encoder which further improves the accuracy.
3. Multiple researcher papers have discussed using CapsNet for OCR models, but none have tried it for Japanese Character. Also, by adding the encoder in CapsNet, we can improve the precision and add another layer for more accurate results. The HJKCR-CapsNet model is trained on Kmnist dataset as well as custom dataset of 100 characters. The model is successful in achieving higher accuracy than base CapsNet model.
4. Proposed HJKCR achieves good recognition rate with low minimum loss as compared with other approaches found in literature survey.

The paper is structured as follows: Section 4 discusses Proposed Methodology. The conclusion of the paper is given in Section 5.

IV. PROPOSED METHODOLOGY

In this paper, we discuss methods to improve the existing OCR deep learning model for Japanese Kanji Character recognition.

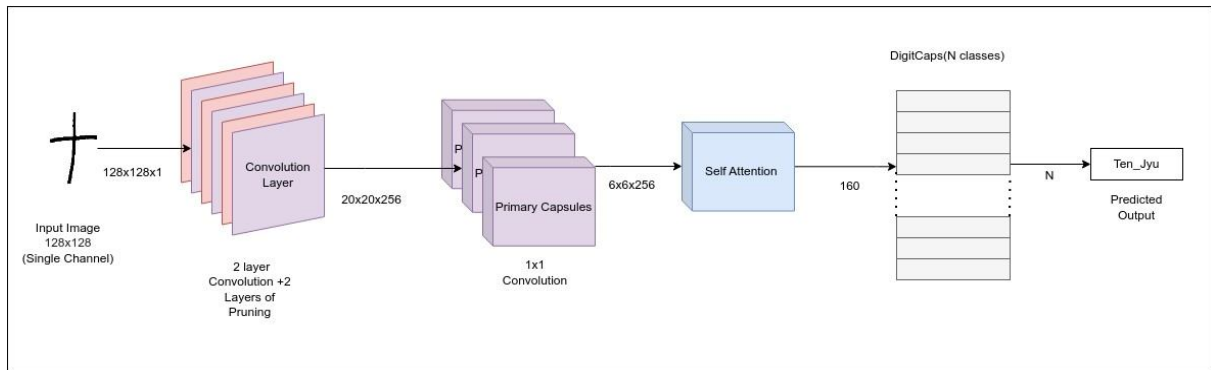


Figure 2. HJKCR Model Architecture

A. Convolution layer and feature extraction

The Model Architecture is as shown in the Figure 2. The model consists of 2 convolution layers, followed by these convolution layers is an attention-based encoder which encodes the images. Followed by the Capsule layers.

For control experiment with standard dataset of kmnist, 2 convolution layers are used without maxpooling. Input images are 28*28*1. We discovered that 6-7 convolution layers are most optimal. Input layer takes in an image of shape 128*128*1. After the final convolution layer is a self-attention layer [4]. This layer helps capture the salient features of the image and focus more on the important regions of the image.

B. Capsule-layer

This layer consists of primary capsules which are convolution layers. The output of Primary capsule layer is calculated using the equation 1. Followed by Digit capsules, where multiple neurons work together to detect features.

$$u_i = w_i * x \quad (1)$$

Where, u_i is the pre-squashed output of capsule I, W_i is the weight matrix of the capsule I and x is the input from the previous layer. Followed by Digit capsules, where multiple neurons work together to detect features. Here each digit capsule is intended to express and identify higher-level and more complicated properties, especially in the context of tasks involving the classification of objects or the recognition of numbers [16, 17]. In the same layer, dynamic routing is implemented using a "routing-by-agreement" algorithm which is based on the equation 2 and equation 3. Equation 2 shows the softmax function for c_{ij} :

$$c_{ij} = \text{softmax}(b_{ij}) \quad (2)$$

Equation 3 represents the calculation of s_j :

$$s_j = \sum_i c_{ij} \otimes u_i \quad (3)$$

Where c_{ij} is coupling coefficient between capsule i and capsule j , b_{ij} is the log prior probability that capsule i should be coupled to capsule j and s_j is the pre-squashed output of capsule j .

In an iterative update process, each primary capsule "votes" for the DigitCaps it believes should be activated, and the DigitCaps capsules "agree" on the final routing weights based on these votes. After that, this is sent to the squash() function, which compresses the output vector between 0 and 1 [18, 19]. The squash() function works on equation 4, Where v_j is the squashed output of capsule j and s_j is the pre-squashed output of capsule j .

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \quad (4)$$

C. Dynamic Routing("Routing by Agreement")

In dynamic neural networks, neurons can iteratively change their connections based on the agreement or disagreement in their predictions, in contrast to traditional neural networks where information flow is determined by static weights. Through this iterative process, capsules can focus on relevant details while disregarding noise, resulting in a more comprehensive understanding of the input. In general, dynamic routing makes it easier for a neural network to create representations that are more interpretable and context-aware, which enhances the network's capacity to handle complicated, hierarchical input. The proposed network cannot use the traditional multi-classification loss function since many classifications are allowed to exist at the same time and the probability value of the capsule network's output cannot be normalized [5]. The margin loss that works well in CapsNet is given in following equation 5.

$$L_k = T_k \max(0, m^+ - \|v_k\|^2) + \lambda(1 - T_k) \max(0, \|v_k\| - m^-) \quad (5)$$

Where L_k is the loss for capsule k , T_k is the binary indicator, m^+ and m^- are constants and λ is a weighting parameter.

D. Parasite Pruning

Every CNN layer has a second layer added by the parasitic method, which enables continual updates to pruned filters and gentle pruning [5]. This mechanism is divided into two stages: the pruning stage takes place in odd epochs while the recovery stage happens in even epochs. Following the pruning procedure, accuracy and optimization issues are addressed during the retraining of the original CNN to preserve its performance. Network efficiency can be maximized while maintaining functionality with the help of filter pruning.

E. Magnitude Pruning

Magnitude-based pruning is a method for making neural networks smaller and less complex [3]. Iteratively, it finds and eliminates less significant weights, usually those that fall below a predetermined threshold. Networks that have been pruned are adjusted to restore lost performance. This approach, nevertheless, runs the danger of

removing potentially significant weights too soon. The "Lottery Ticket Hypothesis," a related idea, postulates that a trained network may have a smaller subnetwork with appropriate weights that can equal the accuracy of the original network. By keeping a mask for selecting which weights to keep, dynamic network surgery improves the pruning procedure. In general, magnitude-based pruning maintains performance while optimizing network efficiency.

F. Similarity-based Pruning

Similarity-Based Pruning evaluates how similar neural network filters or weights are to one another. They use the inner product as a measure for similarity. Based on inner product values, it groups similar filters together [3,12]. The mean filter then replaces the group of comparable filters. It is discovered that this strategy retains accuracy better than the tiny filters trimming method, once an ideal value for the threshold for similarity is established.

V. CONCLUSION

In conclusion, this model can improve the existing OCR models by reducing the time complexity as well as space complexity of the model. It also deals with the invariance problem that occurs in traditional CNN models by using capsule networks. By applying pruning to the model, it also reduces the increased computation required by capsule networks while still retaining the same accuracy.

For the future, HJKCR model can be trained multiple scripts such as Chinese, Korean, Devanagari, etc. Alongside different languages, HJKCR model can be scaled to identify and translate multiple characters or lines of characters. By applying pruning to complex models, we can also release efficient and cost-effective models for other OCR tasks.

ACKNOWLEDGMENT

The authors would like to extend their heartfelt gratitude to **Dr.S.P. Deore** for their invaluable guidance and support throughout the writing process of this paper. Their expertise and mentorship have been instrumental in shaping our ideas and refining our work.

We also express our sincere appreciation to **Dr.(Mrs.)N.F.Shaikh** for providing us with relevant assistance and resources that greatly contributed to the completion of this research.

In end our special thanks to **Dr. Mrs. M. P. Dale** for providing various resources such as laboratory with all needed software platforms, continuous Internet Connection, for Our Project.

Thank you for your unwavering support and encouragement.

REFERENCES

- [1] L. Shen, B. Chen, J. Wei, H. Xu, S.-K. Tang, and S. Mirri, "The challenges of recognizing offline handwritten Chinese: A technical review," *Applied Sciences*, vol. 13, no. 6, p. 3500, 2023.
- [2] N. Harilal and R. Patil, "Effectiveness of the recent advances in capsule networks," *arXiv preprint arXiv:2210.05834*, 2022.
- [3] S. Vadera and S. Ameen, "Methods for pruning deep neural networks," *IEEE Access*, vol. 10, pp. 63280–63300, 2022.
- [4] N. T. Ly, C. T. Nguyen, and M. Nakagawa, "Attention augmented convolutional recurrent 10 network for handwritten Japanese text recognition," in *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 163–168, IEEE, 2020.
- [5] X. Han, L. Xue, Y. Xu, and K. Huang, "A parasitic mechanism-based filter pruning method for deep convolutional neural networks," in *2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, vol. 1, pp. 45–48, IEEE, 2020.
- [6] M. Mousa-Pasandi, M. Hajabdollahi, N. Karimi, S. Samavi, and S. Shirani, "Convolutional neural network pruning using filter attenuation," in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 2905–2909, IEEE, 2020.
- [7] A. S. Gaikwad and M. El-Sharkawy, "Pruning convolution neural network (SqueezeNet) using Taylor expansion-based criterion," in *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 1–5, IEEE, 2018.
- [8] M. Han and H. Zhang, "Offline handwritten Chinese character recognition based on improved capsule network," in *2022 4th International Academic Exchange Conference on Science and Technology Innovation (IAECST)*, pp. 1019–1024, IEEE, 2022.
- [9] H. Ren, J. Su, and H. Lu, "Evaluating generalization ability of convolutional neural networks and capsule networks for image classification via top-2 classification," *arXiv preprint arXiv:1901.10112*, 2019.

- [10] L. Xiaolin, R. C. Panicker, B. Cardiff, and D. John, "Multistage pruning of CNN based ecg classifiers for edge devices," in 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 1965–1968, IEEE, 2021.
- [11] R. Harang and H. Sanders, "Magnificent minified models," arXiv preprint arXiv:2306.10177, 2023.
- [12] S. Rakshith, J. R. Vachhani, S. V. Gothe, and R. Khurana, "Self-similarity matrix based CNN filter pruning," in International Conference on Computer Vision and Image Processing, pp. 73–83, Springer, 2022.
- [13] L. M. AbouEl-Magd, A. Darwish, V. Snasel, and A. E. Hassanien, "A pre-trained convolutional neural network with optimized capsule networks for chest x-rays covid-19 diagnosis," Cluster Computing, vol. 26, no. 2, pp. 1389–1403, 2023.
- [14] A. Narukkanchira Anilkumar, RCNX: Residual Capsule Next. PhD thesis, 2021.
- [15] Z. Wang, C. Li, and X. Wang, "Convolutional neural network pruning with structural redundancy reduction," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 14913–14922, 2021.
- [16] A. N. Anilkumar and M. El-Sharkawy, "Rcnx: residual capsule next," in 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), pp. 1–6, IEEE, 2021.
- [17] S. Srinivasulu, J. R. Raj, S. Gowri, J. Jabez, et al., "A novel method for brain tumor classification using capsule network," in 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), pp. 107–110, IEEE, 2022.
- [18] J. Xu, "Comparison of convolutional neural network and capsule network in image recognition case study of micro biopsy images of breast tumor," in 2022 2nd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI), pp. 674–682, IEEE, 2022.
- [19] M. Kumar, J. Ramasamy, M. P. Kumar, and B. Harsha, "An improved segmentation method for brain cancer using capsule neural networks.," ICTACT Journal on Image & Video Processing, vol. 13, no. 4, 2023. 11
- [20] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," Advances in neural information processing systems, vol. 30, 2017.
- [21] H. Shah and V. Manjula, "A performance analysis of deep convolutional neural networks using Kuzushiji character recognition," in 2020 International Conference on Decision Aid Sciences and Application (DASA), pp. 1068–1071, IEEE, 2020.