International Journal of Scientific Research in Computer Science and Engineering **Research Paper**
Vol.9, Issue.5, pp.20-24, October (2021) **E-ISSN:** 2320-7639

Handwritten Dzongkha Alphabet Recognition System using Convolutional Neural Network

Deewas Chamling[1], Yeshi Jamtsho[2*], Yonten Jamtsho[3]

[1,2]College of Science and Technology, Rinchending, Bhutan
[3]Gyalpozhing College of Information Technology, Gyalpozhing, Bhutan

[*]*Corresponding Author: yjamtsho.cst@rub.edu.bt, Tel.: +975-17962261*

Available online at: www.isroset.org

Received: 16/Aug/2021, Accepted: 20/Oct/2021, Online: 31/Oct/2021

***Abstract*—** Pattern recognition is one of the fields in computer vision. With the advancement of deep learning technology, many machine learning algorithms were deployed for classification problems. Optical Character Recognition (OCR) is a method of processing and recognizing a character from a handwritten character or a printed document within a digital image. In this paper, an implementation using Convolutional Neural Network (CNN) was proposed for the classification of Handwritten Dzongkha alphabets. The dataset consists of 30 classes, each representing an alphabet of the Dzongkha language with 500 images in each class amounting to a total of 15000 images. Four layered CNN with a kernel size of 3 produced the optimal result in building the model and achieved an accuracy of 97.22% and a loss of 17.62%. This research is carried out for the first time in Bhutan and the findings from the study will act as the benchmark for future researchers. In the future, more handwritten alphabets need to be collected and trained with pre-trained models to get better accuracy.

***Keywords*—** OCR; Deep Learning; CNN; Pattern recognition; Dzongkha language

I. INTRODUCTION

Optical Character Recognition (OCR) is a method of processing and recognizing a character from a handwritten character or a printed document within a digital image. It is the subfield of Computer Vision (CV) which in turn is a subfield of Artificial Intelligence (AI) that attempts to find patterns to analyze and understand the texts in the image similar to the way human visuals perform. Every person has a unique handwritten style, which implies that handwriting differs from person and person. Handwriting recognition has become a wide and essential subject for studies due to different variations [1].

Dzongkha being the national language of Bhutan represents the identity of the country. Having a huge number of documents that contains priceless information of our history and culture written in Dzongkha, the need for storing and safeguarding them is a top priority and living in a digital era demands a reliable system for processing and storing. While there are many research and applications of handwritten OCR in various languages such as the Devanagari text of India and the Farsi text of Arab, there has been no research conducted towards the Dzongkha alphabet of Bhutan. To address this issue and to promote the Dzongkha language, computerization of the language is a must and OCR is one such solution to overcome this obstacle. Handwritten Dzongkha Alphabets Recognition System (HanDARS) is a proposed system for implementing OCR in Dzongkha. The proposed model utilizes a deep learning algorithm namely Convolutional Neural Network (CNN).

In section II, the related works are discussed followed by the methodology on the development of the handwritten Dzongkha Alphabet recognition system which is elaborated in section III. The experimental results obtained are discussed in section IV and the conclusion with insights to the future work is included in section V.

II. RELATED WORK

There are various works carried out on OCR and on the use of different algorithms for various languages which has vastly promoted the research and development of character recognition. OCR is the process of recognizing a character and converting it into editable texts. There are six major stages in character recognition namely image acquisition, pre-processing, image segmentation, feature extraction, image classification and post-processing carried out in the mentioned sequence [2].

Feature extraction and the classification stage are the crucial stages of OCR where the accuracy of the model is highly dependent upon and many research had been conducted in this area to determine the best model. The authors [3] proposed three features extraction techniques: histogram of projection based on mean distance, histogram of projection based on pixel value, and vertical zero crossings which have been used to improve the rate of recognition achieved an accuracy of 90% for printed Hindi text on the document.

Another study conducted by [4] proposed an image segmentation algorithm for calculating pixel intensities to identify letters in the image and they trained a 5-layer artificial neural network (ANN) on the data. The ANN contains 2500, 2000, 1000 and 800 neurons in the first, second, third and fourth layers respectively. The ANN achieved 93.32% accuracy. Yadav & Jain proposed neural networks to recognize the handwritten character in the offline mode. They have proposed five main stages: image acquisition, pre-processing, segmentation, feature extraction and classification [5].

A recent comparative study was conducted among the traditional classifiers such as the K-Nearest Neighbors algorithm (KNN), ANN and Support Vector Machine (SVM) using all the feature extraction methods. This was conducted on recognizing handwritten Farsi digits. SVM had the highest accuracy of 99.3% whereas the accuracy of KNN was 93.47% and ANN had 97.19%. Since the implementation time is also a huge factor while designing a model, ANN had the lowest implementation time as compared to the other two. However, when all these traditional methods were compared with new methods such as CNN under deep learning, CNN had a lower implementation time than ANN and had a higher accuracy of 99.45% as compared to SVM's 99.3% [6].

III. METHODOLOGY

A. System Overview

This section describes the methodology proposed to develop the model. CNN is chosen for this project for its higher accuracy and lower implementation time than any other method. The proposed system architecture is shown in Figure 1. The alphabet **'ka'** is given as input in the input layer which then undergoes various processes before identifying the character.

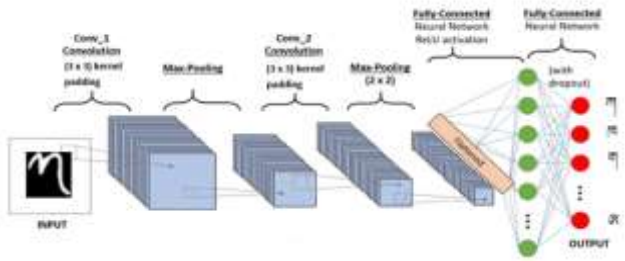

Figure 1. The CNN Architecture

A convolutional layer is the first layer that extracts the features from the image which is given as an input. A kernel is a filter matrix that is used to extract the features from the images. The output received from this layer is known as the Feature map that provides information regarding the images which are then sent to the other layers to extract additional features of the input images.

After the feature map has been extracted, it goes through the pooling layer that aims to reduce its dimensions to reduce the execution time of the machine. Flattening is then done before finally passing onto the fully connected layer. Flattening is converting the data into a 1-dimensional array for inputting it to the next layer. Flattening the output of the convolutional layers is done to create a single feature. The last layer is the fully connected layer. As the name suggests, the neurons which are the basic units of a neural network are connected with each other. This layer will classify the images based on their classes.

B. Model Development

1) Setting up layers and kernel size

As there is no defined rule stating the number of layers to be defined, the study started with a 2-layer neural network skipping the single-layer network as a single-layer network known as perceptron has an input layer directly connected to the output layer and cannot perform operations (such as XOR) required for multiple classes. An experimental setup for this is shown in Table 1.

Table 1. Initial experimental setup for the number of layers and kernel size.

Slot Number	No. of layers	Kernel Size
1.	2-layers	3
		5
2.	3-layers	3
		5
3.	4-layers	3
		5
4.	5-layers	3
		5

2) Hyperparameters – Learning rate

The model required additional hyperparameters for it to be optimally tuned. Therefore, learning rates were initialized. A learning rate is a hyperparameter that controls how fast the model adapts to the given problem. A larger learning rate requires fewer training epoch as significant changes are made to the weights per epoch and a smaller learning rate requires more epochs as only small changes are made to the weights. The range of values to consider for the learning rate is less than 1.0 and greater than 10^-6 [7]. According to [7] the default value of 0.01 works for multi-layer neural networks.

3) Hyperparameters – Dropout

Dropout is a method of ignoring randomly chosen neurons during the training of the model. This reduces overfitting, a scenario where the model is training or learning too well that even the noises present in the images are being identified as features. The dropout value can be given from a range of 0-1 [8]. A dropout of 0.4 to the input layer and a value of 0.2 to the dense layer gave the best results for this model.

4) Hyperparameters – Activation function

The Logistic-Sigmoid, Tangent-Sigmoid or Rectified Linear Unit (ReLU) should be used for the hidden layer depending on the experiments. For more than 2 layers, ReLU works best as it solves the problem of vanishing

gradients [9]. The paper also stated the importance of distinguishing between a regression-based problem and a classification-based problem. For a multiclass classification problem, like this paper, the Softmax function with categorical cross-entropy as its loss function works best. Table 3 shows the type of hyperparameters used with a given value that is best suitable for the model.

Table 2. Hyperparameters defined for the model

Hyperparameters	Value
Learning rate	0.01
Dropout (input layer)	0.4
Dropout (dense layer) Activation function (hidden layer)	0.2 Rectified Linear Unit
Activation function (output layer)	Softmax
Loss function	Categorical cross-entropy

5) Setting up layers and kernel size

Optimizers are techniques or algorithms that are used to adjust the characteristics of the neural network so that it minimizes the losses. For the optimizer, a comparison between Adam [10] and RMSprop, the two most popular optimization algorithm [11], were made based on the testing accuracy and the loss of each optimizer produced on the same dataset with the same hyperparameters. The refined experimental setup after using the optimizer is shown in Table 2.

Table 3. Initial experimental setup for the number of layers and kernel size

Optimizer	Slot Number	No. of layers	Kernel Size
Adam	1.	2-layers	3
			5
	2.	3-layers	3
			5
	3.	4-layers	3
			5
	4.	5-layers	3
			5
RMSProp	1.	2-layers	3
			5
	2.	3-layers	3
			5
	3.	4-layers	3
			5
	4.	5-layers	3
			5

6) Setting up layers and kernel size

A summary of the final model is shown in Figure 2 after it had been fine-tuned with the hyperparameters mentioned in Table 3. It is a graphical representation of the model showing the layer type, the order of the layers and the output shape of each layer.

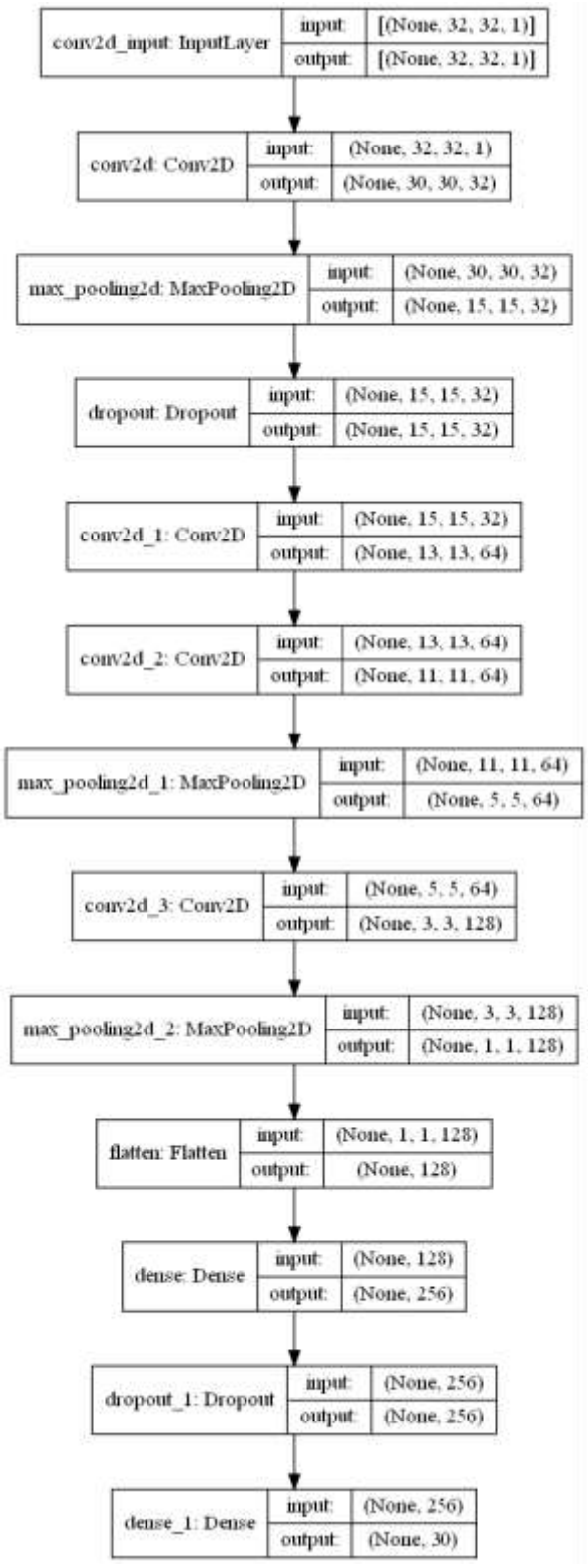

Figure 2. Visual representation of the model

IV. RESULTS AND DISCUSSION

The designed model was tested with various values of layers and hyperparameters which were then compared to find out the most suitable number of layers and the hyperparameters for the model to have the highest accuracy and the lowest loss.

From the various experiments conducted on the above-mentioned hyperparameters, the best result was given by Adam optimizer when used with a 4-layer neural network with a kernel size of 3. The accuracy obtained was 98.59% and its loss was 13.85%. However, this result was only to determine the basic layout of the model and not the finalized version of the model. From the results obtained via a comparison between the two optimizers on different scenarios, a 4-layer neural network was developed with the kernel size of 3 using the Adam optimizer as shown in Table 4.

Table 4. Experimental results

Optimizer	No. of layers	Kernel size	Accuracy	Loss
Adam Optimizer	2-layers	3	94.37%	49.12%
		5	95.77%	35.63%
	3-layers	3	95.77%	39.95%
		5	96.11%	21.56%
	4-layers	3	98.59%	13.85%
		5	97.26%	14.67%
	5-layers	3	96.23%	27.86%
		5	97.18%	15.04%
RMSProp	2-layers	3	90.14%	63.57%
		5	91.55%	58.52%
	3-layers	3	93.48%	35.51%
		5	94.37%	29.21%
	4-layers	3	94.33%	20.93%
		5	95.77%	18.39%
	5-layers	3	92.18%	23.95%
		5	96.47%	30.19%

Combining the hyperparameters obtained from the trial-and-error method and from the literature, the model was trained, and an initial loss of the training and the testing set on the model was graphed as shown in Figure 3.

The testing loss (57.21%) was significantly greater than the training loss (6.16%). To reduce the significant difference between the two sets, the dropout method was used. The loss decreased from 87.59% to 57.21% after using the optimum dropout value as shown in Figure 4.

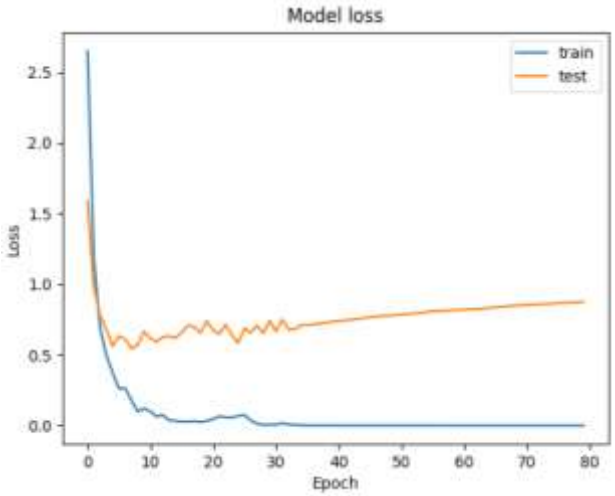

Figure 3. Loss obtained on the training and the testing on the initial model

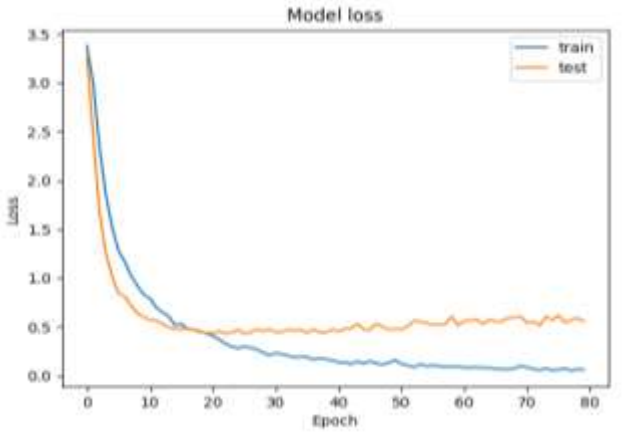

Figure 4. Loss obtained on the training and the testing on the initial model after using the dropout value.

The accuracy obtained was 98.4% on the training images and 91.73% was obtained on the testing set. The accuracy of both the sets is shown in Figure. 5.

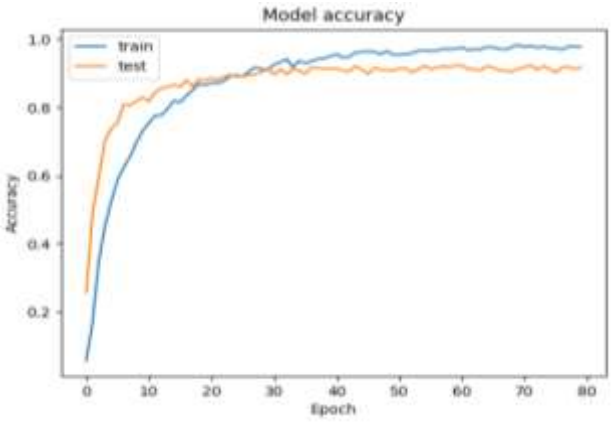

Figure 5. Accuracy obtained on the training and testing set

The saved model was tested on the testing images consisting of 50 images per class and a total of 1500 images and achieved a total accuracy of 97.22% and a loss of 17.62%. To view a detailed summary of the accuracy of every class, the f-score, precision and recall of each class is produced as shown in Table 5.

Table 5. Experimental results

Classes	Precision	Recall	F1-score
character_ 1_ka	0.9735	0.9698	0.9716
character_2_kha	0.9635	0.9612	0.9632
character_3_ga	0.9718	0.9698	0.9708
character_4_nga	0.9737	0.9692	0.9714
character_5_cha	0.9582	0.9486	0.9539
character_6_chha	0.9679	0.9750	0.9714
character_7_ ja	0.9690	0.9648	0.9669
character_8_nya	0.9774	0.9738	0.9756
character_9_ ta	0.9716	0.9688	0.9702
character_10_tha	0.9638	0.9590	0.9614
character_11_da	0.9734	0.9701	0.9717
character_12_na	0.9737	0.9703	0.9720
character_13_pa	0.9714	0.9688	0.9701
character_14_pha	0.9663	0.9613	0.9638
character_15_ba	0.9816	0.9764	0.9790

character_16_ma	0.9788	0.9780	0.9796
character_17_tsa	0.9825	0.9807	0.9816
character_18_tsha	0.9820	0.9778	0.9799
character_19_jsha	0.9737	0.9700	0.9718
character_ 20_wa	0. 9708	0. 9671	0. 9689
character_ 21_zha	0. 9756	0. 9690	0. 9723
character_ 22_za	0. 9757	0. 9719	0. 9737
character_ 23_ha	0. 9856	0. 9840	0. 9848
character_ 24_ya	0. 9763	0. 9743	0. 9753
character_ 25_ra	0. 9872	0. 9856	0. 9864
character_ 26_la	0. 9786	0. 9768	0. 9777
character_ 27_sha	0. 9778	0. 9774	0. 9776
character_ 28_sa	0. 9783	0. 9772	0. 9778
character_ 29_haa	0. 9698	0. 9660	0. 9678
character_ 30_aa	0. 9766	0. 9737	0. 9752

V. CONCLUSION AND FUTURE SCOPE

Handwritten alphabet recognition is one of the important research topics in the field of pattern recognition. Due to variations in the individual handwriting styles, it has become important to do research in the Bhutanese context for the computerization of the national language. This study aims to create a Handwritten Dzongkha Alphabets Recognition System (HanDARS) by using a Convolutional Neural Network. The study proposed 4 convolutional layers with the kernel size of 3x3 to recognize the Bhutanese alphabets. The model recognizes the alphabets with an accuracy of 97.22%.

This model can be integrated along with a web or a mobile application to improve the efficiency of recognizing alphabets. It can be further trained by gathering more sample sizes. Other remaining Dzongkha characters such as the numerical values and special characters can also be collected, processed, and then built on top of this model to increase the proficiency of the model to identify Dzongkha language instead of just Dzongkha alphabets. This research is carried out for the first time in Bhutan and the findings from the research will serve as the benchmark for future researchers.

REFERENCES

[1] Vidushi and M. Agarwal, "Intelligent Handwritten Digit Recognition Based on Multiple Parameters using CNN," *International Journal of Computer Sciences and Engineering*, Vol.**7**, Issue.**5**, pp.**636–641**, **2019**.

[2] M. S. K. Dasari, "Optical Character Recognition of Devanagari Script Using Machine Learning- A Survey," Journal of Xi'an University of Architecture & Technology, Vol. **12**, Issue.**8**, pp.**593–599**, **2020**.

[3] D. Yadav, S. Sanchez-Cuadrado, and J. Morato, "Optical Character Recognition for Hindi Language Using a Neural-network Approach," *J. Inf. Process. Syst.*, Vol.**9**, Issue.**1**, pp. **117–140**, **2013.**

[4] M. Avadesh and N. Goyal, "Optical Character Recognition for Sanskrit Using Convolution Neural Networks," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pp. **447–452, 2018**.

[5] H. Yadav and S. Jain, "Offline Handwritten Character Recognition using Neural Networks," *International Journal of Computer Sciences and Engineering*, Vol.**7**, Issue.**5**, pp.**838–845**, **2019**.

[6] Y. A. Nanehkaran, D. Zhang, S. Salimi, J. Chen, Y. Tian, and N. Al-Nabhan, "Analysis and comparison of machine learning classifiers and deep neural networks techniques for recognition of Farsi handwritten digits," *J. Supercomput.*, Vol.**77**, Issue.**4**, pp.**3193–3222**, **2021**.

[7] Y. Bengio, "Practical Recommendations for Gradient-Based Training of Deep Architectures," in Neural Networks: Tricks of the Trade: Second Edition, G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Berlin, Heidelberg: Springer, pp.**437–478**, **2012**.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, Vol.**15**, Issue.**56**, pp.**1929–1958**, **2014**.

[9] O. K. Oyedotun, A. E. R. Shabayek, D. Aouada, and B. Ottersten, "Going Deeper With Neural Networks Without Skip Connections," in *2020 IEEE International Conference on Image Processing (ICIP)*, pp.**1756–1760, 2020**.

[10] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *ArXiv14126980 Cs*, Jan. 2017, Accessed: Aug. 13, 2021. [Online]. Available: http://arxiv.org/abs/1412.6980

[11] R. Poojary and A. Pai, "Comparative Study of Model Optimization Techniques in Fine-Tuned CNN Models," in *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, pp.**1–4**, **2019.**

AUTHORS PROFILE

Deewas Chamling has always been keen in learning more about the world of Information Technology. He completed his Bachelor's degree of Engineering in Information Technology from the College of Science and Technology located in Phuentsholing, Bhutan in the year 2021. His particular interest in Artificial Intelligence led him to choose AI as his preferred elective in his college and completed his final year project successfully in the same field. He wishes to explore more on the various fields of Artificial Intelligence throughout his career journey from his mentors and colleagues.

Yeshi Jamtsho received M. Eng in Computer Engineering from Naresuan University, Thailand in 2020 and B.Eng in Information Technology from College of Science and Technology (CST), Royal University of Bhutan (RUB), in 2014. He has also received Postgraduate Certificate in Higher Education (PgCHE) from Samtse College of Education, RUB in 2018. Currently, he is working in CST, RUB as an Associate Lecturer since 2014. Deep Learning, machine learning, Artificial Intelligence, Natural Language Processing, Bloackchain, and Computer Vision are his current research interest.

Yonten Jamtsho is currently working as an Associate Lecturer at Gyalpozhing College of Information Technology, Royal University of Bhutan. He did his BSc (Hons) in Computer Science (2015) from Sherubtse College, Royal University of Bhutan and M.E. Computer Engineering (2020) from Naresuan University, Thailand. He has also done Post Graduate Certificate in Higher Education (2021) from Samtse College of Education. During his master study, he did his thesis in the field of image processing and computer visions. He has 6 years of teaching experience and 2 years of research experience. His research interests lies in the field of computer visions, image processing and data science.

