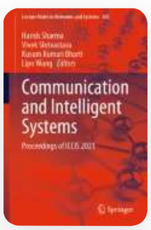
KGAN: A Generative Adversarial Network Augmented Convolution Neural Network Model for Recognizing Kannada Language Digits

| Conference paper | First Online: 19 August 2022

| pp 523–531 | Cite this conference paper



Communication and Intelligent Systems

H. S. Shrisha, V. Anupama, D. Suresha & N. Jagadisha

Part of the book series: Lecture Notes in Networks and Systems ((LNNS,volume 461))

Abstract

Kannada is a south Indian language with a history of two thousand years and spoken by more than sixty million people. Kannada language has its own script for alphabets and digit representations. So there is a need for convolution neural network (CNN) model to recognize

Kannada language scripts. This paper presents a design of a CNN model to recognize Kannada digits. One of the challenges faced while designing a CNN model is data over fitting. Data over fitting is a phenomenon where the trained model arrives at parameter values such that they can classify only the instances provided during training resulting in reduction of accuracy for a new unseen test instance. To overcome this problem, datasets are split into train and test sets. The detriment of this system is lesser number of instances to train the CNN. Increasing the number of training instances is a good approach, but the complexity in data collection is to be answered. In this paper, we explore generative adversarial network (GAN) as an additional data generator and its suitability. Results of analysis on the experiment revealed the following advantages; first, the data augmentation has a positive impact on CNN, next, GAN-generated data meets qualitative requirement as train and test dataset and last, epoch value for training CNN has influence on data under fitting and data over fitting phenomenon.

eBook EUR 213.99

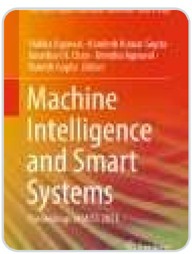
Softcover Book EUR 249.99

Tax calculation will be finalised at checkout

Purchases are for personal use only

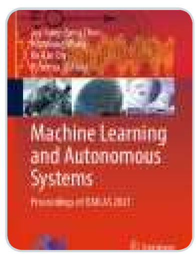
Institutional subscriptions →

Similar content being viewed by others



Augmentation of Handwritten Devanagari Character Dataset Using DCGAN

Chapter | © 2022



Tamil Language Handwritten Document Digitization and Analysis of the Impact of Data...

Chapter | © 2022

Handwritten Devanagari Character Generation Using Deep Convolutional...

Chapter | © 2020

References

1. Ahamed P, Kundu S, Khan T, Bhateja V, Sarkar R, Mollah AF (2020) Handwritten Arabic numerals recognition using convolutional neural network. J Ambient Intell Hum Comput 11(11):5445–5457

Article Google Scholar

2. Akhtar N, Ragavendran U (2020) Interpretation of intelligence in CNN-pooling processes: a methodological survey. Neural Comput Appl 32(3):879–898

Article Google Scholar

3. Alqahtani H, Kavakli-Thorne M, Kumar G (2021) Applications of generative adversarial networks (GANs): an updated review. Arch Comput Methods Eng 28(2):525–552

Article MathSciNet Google Scholar

4. Anupama VG, Kiran A (2022) Extrapolating z-axis data for a 2d image on a single board computer. In: Saraswat M, Roy S, Chowdhury C, Gandomi AH (eds) Proceedings of international conference on data science and applications. Lecture notes in networks and systems, vol 288. https://doi.org/10.1007/978-981-16-5120-5_38

5. Cheng K, Tahir R, Eric LK, Li M (2020) An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset. Multimedia Tools Appl 79(19):13725–13752

Article Google Scholar

6. Jiang X, Lu M, Wang SH (2020) An eight-layer convolutional neural network with stochastic pooling, batch normalization and dropout for fingerspelling recognition of Chinese sign language. Multimedia Tools Appl 79(21):15697–15715

Article Google Scholar

7. Lei F, Liu X, Dai Q, Ling BWK (2019) Shallow convolutional neural network for image classification. SN Appl Sci 2(1):97

Article Google Scholar

8. Madakannu A, Selvaraj A (2020) Digi-net: a deep convolutional neural network for multi-format digit recognition. Neural Comput Appl 32(15):11373–11383

Article Google Scholar

9. Mittal A, Soorya A, Nagrath P, Hemanth DJ (2020) Data augmentation based morphological classification of galaxies using deep convolutional neural network. Earth Sci Inform 13(3):601–617

Article Google Scholar

10. Prabhu VU (2019) Kannada-MNIST: a new handwritten digits dataset for the kannada language. CoRR abs/1908.01242

Google Scholar

11. Wan J, Chen B, Xu B, Liu H, Jin L (2019) Convolutional neural networks for radar HRRP target recognition and rejection. EURASIP J Adv Sign Process 1:5

Google Scholar

12. Xu Y, Goodacre R (2018) On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. J Anal Test 2(3):249–262

Article Google Scholar

13. Yamashita R, Nishio M, Do RKG, Togashi K (2018) Convolutional neural networks: an overview and application in radiology. Insights Imaging 9(4):611–629

Article Google Scholar

14. Zhou H, Ummenhofer B, Brox T (2020) DeepTAM: deep tracking and mapping with convolutional neural networks. Int J Comput Vis 128(3):756–769

Article Google Scholar

Author information

Authors and Affiliations

Canara Engineering College, Visveswaraya Technological University, Benjanapadavu, Bantwal, 574219, India

H. S. Shrisha, V. Anupama & N. Jagadisha

AJ Institute of Engineering and Technology, Visveswaraya Technological University, Mangaluru, 575006, India

D. Suresha

Corresponding author

Correspondence to H. S. Shrisha .

Editor information

Editors and Affiliations

Department of Computer Science and Engineering, Rajasthan Technical University, Kota, India

Harish Sharma

Institutional Area Narela Delhi, National Institute of Technology Delhi, New Delhi, India

Vivek Shrivastava

Design and Manufacturing, Indian Institute of Information Technology, Jabalpur, India

Kusum Kumari Bharti

School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore

Lipo Wang

Rights and permissions

Reprints and permissions

Copyright information

About this paper

Cite this paper

Shrisha, H.S., Anupama, V., Suresha, D., Jagadisha, N. (2022). KGAN: A Generative Adversarial Network Augmented Convolution Neural Network Model for Recognizing Kannada Language Digits. In: Sharma, H., Shrivastava, V., Kumari Bharti, K., Wang, L. (eds) Communication and Intelligent Systems . Lecture Notes in Networks and Systems, vol 461. Springer, Singapore. https://doi.org/10.1007/978-981-19-2130-8_41

.RIS .ENW .BIB

DOI
https://doi.org/10.1007/978-981-19-2130-8_41

Published
19 August 2022

Publisher Name
Springer, Singapore

Print ISBN
978-981-19-2129-2

Online ISBN
978-981-19-2130-8

eBook Packages
Intelligent Technologies and Robotics
Intelligent Technologies and Robotics (R0)

Publish with us

Policies and ethics

KGAN: A Generative Adversarial Network Augmented Convolution Neural Network Model for Recognizing Kannada Language Digits

H. S. Shrisha, V. Anupama, D. Suresha, and N. Jagadisha

Abstract Kannada is a south Indian language with a history of two thousand years and spoken by more than sixty million people. Kannada language has its own script for alphabets and digit representations. So there is a need for convolution neural network (CNN) model to recognize Kannada language scripts. This paper presents a design of a CNN model to recognize Kannada digits. One of the challenges faced while designing a CNN model is data over fitting. Data over fitting is a phenomenon where the trained model arrives at parameter values such that they can classify only the instances provided during training resulting in reduction of accuracy for a new unseen test instance. To overcome this problem, datasets are split into train and test sets. The detriment of this system is lesser number of instances to train the CNN. Increasing the number of training instances is a good approach, but the complexity in data collection is to be answered. In this paper, we explore generative adversarial network (GAN) as an additional data generator and its suitability. Results of analysis on the experiment revealed the following advantages; first, the data augmentation has a positive impact on CNN, next, GAN-generated data meets qualitative requirement as train and test dataset and last, epoch value for training CNN has influence on data under fitting and data over fitting phenomenon.

Keywords Kannada language · Generative adversarial network · Convolution neural network · MNIST dataset

H. S. Shrisha (✉) · V. Anupama · N. Jagadisha
Canara Engineering College, Visveswaraya Technological University, Benjanapadavu, Bantwal 574219, India
e-mail: shree.1259@gmail.com
URL: https://www.canaraengineering.in/

D. Suresha
AJ Institute of Engineering and Technology, Visveswaraya Technological University, Mangaluru 575006, India

H. Sharma et al. (eds.), *Communication and Intelligent Systems*, Lecture Notes in Networks and Systems 461, https://doi.org/10.1007/978-981-19-2130-8_41

1 Introduction

Kannada is a native language of south India, and efforts are being made to preserve and modernize the areas of application of this language. A step toward this goal is publication of Kannada-MNIST [10] dataset for Kannada digits. The said dataset is a collection of sixty thousand images. The dataset symbols provide variations in representation of the kannada digit symbols for training purposes to accommodate different styles of writing. The dimensions of the images are 28×28 and digits are from zero to nine. A sample of Kannada digit representation as in Kannada-MNIST dataset is provided in Fig. 1.

This paper proposes a neural network architecture to classify Kannada digits. There is always a need to enhance image classification accuracy for neural network architecture. Researchers have adopted many ways to achieve this goal including fine tuning training parameters of neural network, number of epochs and learning rates. A situation which is to be avoided in deep learning is data over fitting. Data over fitting decreases the accuracy of predictions for a new unseen input instance. One way to avoid data over fitting is to provide more data for training. Generative adversarial network (GAN) is a good option to generate new training data with diverse coverage of possible new instances. This experiment demonstrates the impact of GAN generated data on classification accuracy in the context of epoch values.

The paper proposes a hybrid system where GAN produces sixty thousand additional images to train a neural network, and its impact on classification accuracy is measured. This paper explores the GAN and subsequent convolution neural network (CNN) network setup, behavior of setup with different epoch values of GAN and CNN, size of additional data set generated by GAN and a discussion on classification accuracy. The contributions of this paper are as follows:

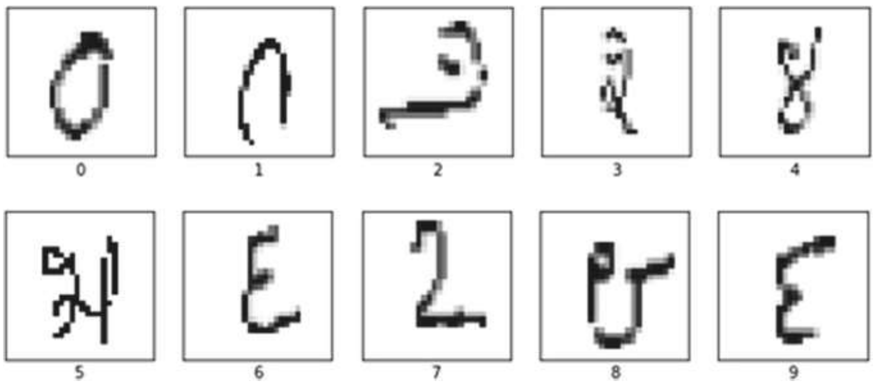


Fig. 1 Sample digits from Kannada-MNIST dataset

- A neural network architecture to recognize and classify Kannada language digits.
- Design of generative adversarial network for data augmentation of Kannada digits samples.
- Analysis of classification accuracy corresponding to epoch values and augmented data.

This paper is organized as follows; Sect. 1 provides introduction about the experiment followed by a brief literature survey on GAN and CNN in Sect. 2. Section 3 provides the description on the experiment including the design of the neural network. Section 4 presents the results of the experiment along with its interpretation. The paper is concluded in Sect. 5.

2 Literature Survey

This section provides an overview of CNN and GAN concepts to accommodate the readers with the basics required to understand the experiment.

2.1 Convolution Neural Network

CNN design involves N dimension convolution layer, pooling strategy, and drop out nodes. 1D convolution is applied for a matrix input with either one row or one column. For example, an image matrix flattened for output dense layer accepts 1D matrix input. 2D convolution is applied on a matrix with row and column. For example, MNIST dataset images have dimensions 28×28. 3D convolution is applied for three dimensional matrices. It can be imagined as a stack of three 2D matrices. For example, RGB images represent three-dimensional matrices. Batch normalization is employed to bind the values in the CNN layers within a range. This increases computational efficiency of the layers [6]. Pooling layer captures the features for image classification [2]. Max pooling is employed in this experiment which selects the maximum value in 2×2 pool window. Pooling keeps the size of the matrix in check after every layer of convolution. Application areas of CNN include radio imaging [13], cartography [14], radar imaging[11] and imaging applications on resource constrained platforms [4].

2.2 Generative Adversarial Network

GAN havs a generator and a discriminator. During training, the generator becomes more efficient in generating new images and discriminator becomes efficient in rejecting the generated image as fake by comparing it with real image [5]. At the end of

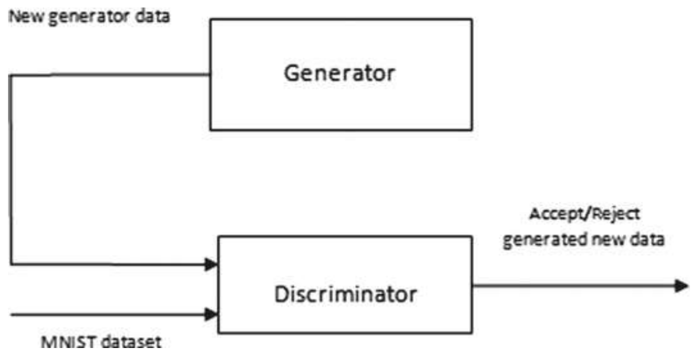


Fig. 2 Overview of a GAN

training, the generator will be capable of producing images which discriminator cannot recognize it as different from real image. An overview of GAN is shown in Fig. 2. GAN can produce additional datasets for training a neural network. Improving the resolution of images, synthesis of images from textual descriptions, medical researches are few applications of GAN [3].

3 Experimental Setup

This experiment explores the idea of using GAN and CNN introduced in Sects. 2.1 and 2.2 for producing additional data, supplementing MNIST data employed for CNN training. Data augmentation is a matured approach for ensemble of CNNs, classification of astronomical bodies [9], etc. Strategy to split the data into train and test set is required. We propose a 50:50 ratio split up of GAN generated data for each class of image. First half earmarked as train set will be merged with primary MNIST dataset. Other half is to be used as test set. Two rationales encouraged us to arrive at this arrangement for dataset;

- Introduction of new data into primary dataset with the aim of achieving better parameter tuning for a wide range of input variations [12].
- GAN-generated test set will provide new instances to the trained CNN.

An overview of experimental setup is given in Fig. 3. Generative adversarial network produces additional dataset of 60,000 images. Newly created dataset is split into train and test set in 50:50 ratios. 30,000 images are combined with Kannada MNIST dataset for training purposes. Remaining 30,000 will be deployed as test set. The accuracy of classification compared to a CNN in combination with variation of epochs in training. NVDIA RTX 2080Ti hardware platform is employed for training the model.

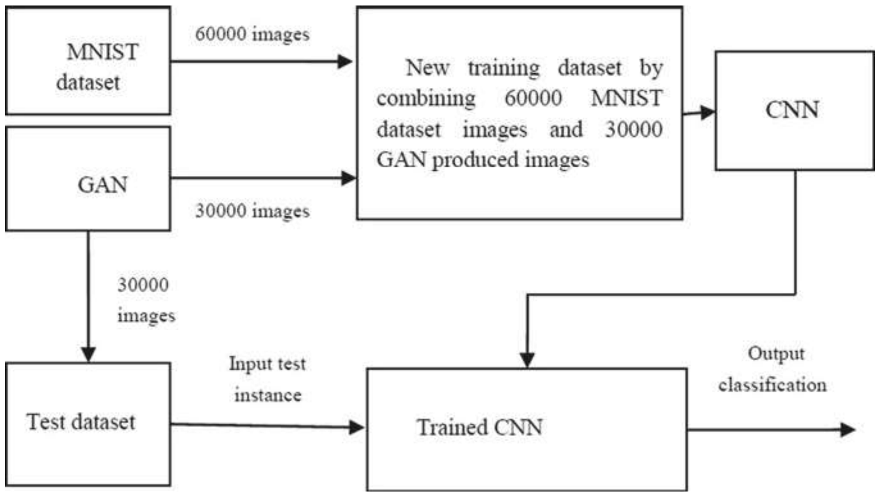


Fig. 3 Experimental setup

3.1 Design of Convolution Neural Network

Design of CNN involves deciding number of layers, number of nodes in each layer, activation function, and techniques for pooling and strategy for learning rate. Adam optimizer with learning rate 0.001 and sparse categorical cross-entropy as log loss is employed. Perceptron uses ReLu activation. Maximum pooling of size 2×2 is used. An overview of these is provided in Table 1.

3.2 Design of Generative Adversarial Network

In GAN, a generator produces images in a bottom-up approach until the discriminator cannot discriminate between real data and generator-produced data. Layers and their output shapes are shown in Table 2.

The discriminator examines the image produced by generator and rejects if comparison between real image and new image does not match. Efficiency of discriminator increases with higher number of epochs limited by computational requirement as a trade-off. Layers of discriminator with their output shapes are provided in Table 3.

Table 1 Parameter values of different layers in CNN

Layer type	Output shape
Conv2D	(None, 28, 28, 32)
Conv2D	(None, 28, 28, 32)
Batch normalization	(None, 28, 28, 32)
MaxPooling2D	(None, 14, 14, 32)
Dropout	(None, 14, 14, 32)
Conv2D	(None, 14, 14, 64)
Conv2D	(None, 14, 14, 64)
Batch normalization	(None, 14, 14, 64)
MaxPooling2D	(None, 7, 7, 64)
Dropout	(None, 7, 7, 64)
Conv2D	(None, 7, 7, 32)
Conv2D	(None, 7, 7, 32)
Batch normalization	(None, 7, 7, 32)
MaxPooling2D	(None, 3, 3, 32)
Dropout	(None, 3, 3, 32)
Flatten	(None, 288)
Dense	(None, 256)
Dropout	(None, 256)
Dense	(None, 10)

Table 2 Generator model of GAN

Layer type	Output shape
Dense	(None, 12544)
Batch normalization	(None, 28, 28, 32)
Batch normalization	(None, 12544)
Leaky ReLu	(None, 12544)
Reshape	(None, 7, 7, 256)
Convolution 2D transpose	(None, 7, 7, 128)
Batch normalization	(None, 7, 7, 128)
Leaky ReLu	(None, 7, 7, 128)
Convolution 2D transpose	(None, 14, 14, 64)
Batch normalization	(None, 14, 14, 64)
Leaky ReLu	(None, 14, 14, 64)
Convolution 2D transpose	(None, 28, 28, 1)

Table 3 Discriminator model of GAN

Layer type	Output shape
Convolution 2D	(None, 14, 14, 64)
Leaky ReLu	(None, 14, 14, 64)
Dropout	(None, 14, 14, 64)
Convolution 2D	(None, 7, 7, 128)
Leaky ReLu	(None, 7, 7, 128)
Dropout	(None, 7, 7, 128)
Flatten	(None, 6272)
Dense	(None, 1)

Table 4 Accuracy versus number of epochs by CNN

Epochs	Accuracy in percent
5000	94.25
10,000	97.40
15,000	97.75
20,000	95.85
25,000	94.75

4 Results and Discussion

CNN presented in Sect. 3.1 is trained and tested on Kannada-MNIST dataset. Performance observed is tabulated in Tables 4 and 5. Table 4 presents accuracy of CNN without GAN data augmentation. Value of epoch should be in between 10,000 to 15,000 for good performance. Above these epoch values, accuracy decreases due to over fitting. Table 5 presents data on the performance of CNN when augmented with dataset produced by GAN. It can be observed that compared to Table 4, CNN when augmented with GAN data for training is resilient to over fitting. We propose that the additional GAN produced data should be at least fifty percent of original dataset to have an impact on CNN training. Below forty percent, the impact is marginal. A graphical representation of increase in accuracy achieved by the proposed GAN augmented training in comparison with standard dataset is provided in Fig. 4.

CNNs designed using different strategies has achieved 93.56 [7], 99.07 [8]. Arabic numeral recognition [1] achieves peak accuracy of 99.75%. Our results show GAN data augmentation makes CNN resilient to data over-fitting indicated by values of accuracy over different epoch values.

Table 5 Accuracy versus number of epochs by CNN trained with augmented data produced by GAN

Epochs	Accuracy in percent
5000	95.35
10,000	98.90
15,000	99.55
20,000	99.55
25,000	99.55

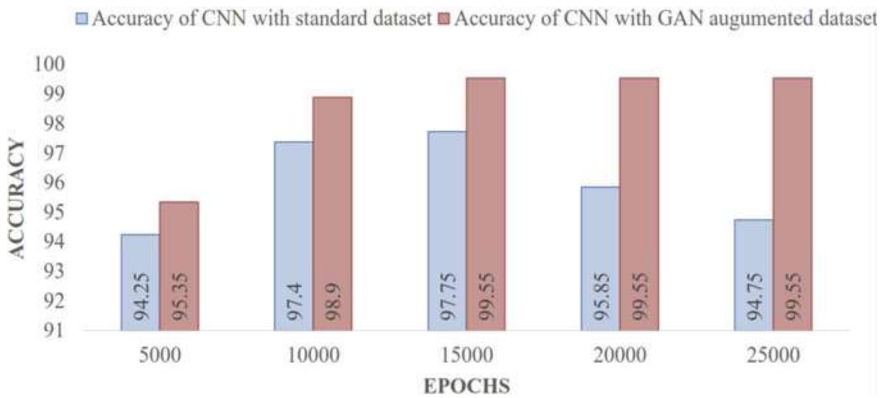


Fig. 4 Epochs versus accuracy for CNN models with standard dataset and GAN augmented dataset

5 Conclusions

CNN can be used for classifying kannada language digits. This paper proposes a CNN design to achieve the same. The experiment explores the possibility of additional data generation using GAN for training and testing purpose. Accuracy of CNN increases with number of training instances available. Data collection and dataset creation require resources. Therefore, there is a need for techniques to generate additional data to train and test the CNN models. A large dataset created by combining data collected through sources and augmentation techniques is an attractive approach to improve performance of CNN. Our experiment split the GAN-generated data into train and test sets in 50:50 ratios. Analysis of results that were presented in Tables 4 and 5 draws the following conclusion:

- The phenomena of data over fitting reduced by 3.1%. This can be attributed to large dataset available for training.
- By examining the result, we can infer that the quality of data generated by GAN is acceptable as train and test data.
- Increasing the value of epoch does not yield increase in quality of generated data after a threshold value. In this experiment, the value is 10,000.

References

1. Ahamed P, Kundu S, Khan T, Bhateja V, Sarkar R, Mollah AF (2020) Handwritten Arabic numerals recognition using convolutional neural network. J Ambient Intell Hum Comput 11(11):5445–5457
2. Akhtar N, Ragavendran U (2020) Interpretation of intelligence in CNN-pooling processes: a methodological survey. Neural Comput Appl 32(3):879–898
3. Alqahtani H, Kavakli-Thorne M, Kumar G (2021) Applications of generative adversarial networks (GANs): an updated review. Arch Comput Methods Eng 28(2):525–552
4. Anupama VG, Kiran A (2022) Extrapolating z-axis data for a 2d image on a single board computer. In: Saraswat M, Roy S, Chowdhury C, Gandomi AH (eds) Proceedings of international conference on data science and applications. Lecture notes in networks and systems, vol 288. https://doi.org/10.1007/978-981-16-5120-5_38
5. Cheng K, Tahir R, Eric LK, Li M (2020) An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset. Multimedia Tools Appl 79(19):13725–13752
6. Jiang X, Lu M, Wang SH (2020) An eight-layer convolutional neural network with stochastic pooling, batch normalization and dropout for fingerspelling recognition of Chinese sign language. Multimedia Tools Appl 79(21):15697–15715
7. Lei F, Liu X, Dai Q, Ling BWK (2019) Shallow convolutional neural network for image classification. SN Appl Sci 2(1):97
8. Madakannu A, Selvaraj A (2020) Digi-net: a deep convolutional neural network for multiformat digit recognition. Neural Comput Appl 32(15):11373–11383
9. Mittal A, Soorya A, Nagrath P, Hemanth DJ (2020) Data augmentation based morphological classification of galaxies using deep convolutional neural network. Earth Sci Inform 13(3):601–617
10. Prabhu VU (2019) Kannada-MNIST: a new handwritten digits dataset for the kannada language. CoRR abs/1908.01242
11. Wan J, Chen B, Xu B, Liu H, Jin L (2019) Convolutional neural networks for radar HRRP target recognition and rejection. EURASIP J Adv Sign Process 1:5
12. Xu Y, Goodacre R (2018) On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. J Anal Test 2(3):249–262
13. Yamashita R, Nishio M, Do RKG, Togashi K (2018) Convolutional neural networks: an overview and application in radiology. Insights Imaging 9(4):611–629
14. Zhou H, Ummenhofer B, Brox T (2020) DeepTAM: deep tracking and mapping with convolutional neural networks. Int J Comput Vis 128(3):756–769