Comparative Analysis of Detection of Text From Morse Code in Handwritten Images using Convolutional Neural Networks

Duggineni Veeraiah
Project Guide, Department of CSE
Lakireddy Bali Reddy College of Engineering (A), Mylavaram
NTR Dt., Andhra Pradesh, India
veeraiahdvc@gmail.com

Shaik Johny Basha
Project Guide, Department of CSE
Lakireddy Bali Reddy College of Engineering (A), Mylavaram
NTR Dt., Andhra Pradesh, India
shaikhjanibasha@gmail.com

Karavadi Raviteja
IV B.Tech Student, Department of CSE
Lakireddy Bali Reddy College of Engineering (A), Mylavaram
NTR Dt., Andhra Pradesh, India
krtproperty72@gmail.com

Mangam Surya Prakash
IV B.Tech Student, Department of CSE
Lakireddy Bali Reddy College of Engineering (A), Mylavaram
NTR Dt., Andhra Pradesh, India
suryaprakash9785@gmail.com

Kasturi Karthik
IV B.Tech Student, Department of CSE
Lakireddy Bali Reddy College of Engineering (A), Mylavaram
NTR Dt., Andhra Pradesh, India
karthikkasturi007@gmail.com

***Abstract* — One of the oldest techniques used in telecommunication for encoding regular characters is Morse Code. Morse Code is categorized into two separate electronic pulses which are dot (aka short pulse) and dash (aka long pulse). Detection of text from images of morse code is a complex process and there is no active research on this area. As these are morse code images, different images have different styles of strokes. Our work aims to develop an Automated Morse code recognition system which is trained by a CNN (convolutional neural network) model with a self-built dataset and involves in collecting and preprocessing images of Morse code characters and creating a labeled dataset for training and testing the CNN model. The dataset creation process includes capturing images of different Morse code characters, augmenting the data to increase the dataset size, and annotating the images to label them correctly. The CNN model is then trained using the created dataset and evaluated for its accuracy in recognizing Morse code characters in images. The results demonstrate comparative analysis of different CNN based frameworks and achieved high accuracy in recognizing Morse code characters in images, making it a promising solution for automated Morse code recognition systems.**

Keywords — Morse Code, Optical Character Recognition, AlexNet, Lenet-5, and ResNet-152, Convolutional Neural Network

I. INTRODUCTION

In the beginning of the 19th century, the messages were delivered either to horses or to ravens. For the first time in the history of mankind a group of scientists invented a telecommunication device called the telegraph, which was used to transmit electric pulses to the other device. They developed an electronic device which could send electric pulses to longer distances within less time. Then, Samuel F.B Morse invented a communication language which can be used with electric pulse signals for communication. These electronic pulses can be represented as long pulse(dash) and short pulse(dot). One short pulse and one long pulse can be represented as A and one long pulse, three short pulses can be represented as B. In this way they have successfully invented a communication language which can be associated with electronic pulses to transmit messages to distinct places.

The morse code consists of a total of 36 characters which is shown in Table I (accessed from [1]) – out of which 26 characters are alphabets and 10 are digits. Lowercase and Uppercase letters don't have any differentiation. Every character is represented with a combination of dash and dots. If we consider the duration of dot as one unit then the duration of dash will be considered as three units. In the basic morse code transmission, every letter in the word is separated by 3 dots and every word is separated by 7 dots.

TABLE I. CHARACTER SET OF MORSE CODE

English Text	Morse Code	English Text	Morse Code
A	• —	S	• • •
B	— • • •	T	—
C	— • — •	U	• • —
D	— • •	V	• • • —
E	•	W	• — —
F	• • — •	X	— • • —
G	— — •	Y	— • — —
H	• • • •	Z	— — • •
I	• •	0	— — — — —
J	• — — —	1	• — — — —
K	— • —	2	• • — — —
L	• — • •	3	• • • — —
M	— —	4	• • • • —
N	— •	5	• • • • •
O	— — —	6	— • • • •
P	• — — •	7	— — • • •
Q	— — • —	8	— — — • •
R	• — •	9	— — — — •

Optical Character Recognition – shortly known as OCR, in present scenario plays a key role in research for finding out the information available in the images by going through different patterns [2] [20]. The primary focus of Optical Character Recognition is to obtain the content from images, where images consist of text that has been a handwritten text, or printed using a printer, text that has been typed, etc. OCR reduces the effort and time taken by the humans to understand the text in a picture without having adequate knowledge of the language.

One of the main challenges to humans is identifying the text of an unknown language which was handwritten or typed or printed. But there may be some chances that a person can understand the text which was typed or printed. But coming to the handwritten, it will be difficult and ambiguous to identify the text as the person-to-person writing is not going to be same [19]. So, to identify the text in an image and convert them into English Language plays a key role. For that, first the character must be extracted from the image in different ways such as Diagonal based feature extraction [3], Deep Feature Extraction [4], etc. Like humans, machines also don't understand the information available in the images when they are not trained to a suitable language. To train a machine with different languages, we can use Convolution Neural Network Based frameworks like VGG-16, InceptionV3, ResNeXTt50, Xception, etc. These frameworks are really helping humans by giving a good prediction rate and high accuracy in results, when they are well trained and tested.

Convolutional Neural Network – shortly known as ConvNet/CNN, which is a fragment of Deep Neural Networks that reads the pictures as an input and provides help in image analysis such as feature extraction, layer pooling, etc. [5]. We know that Matrix Multiplication plays a major role in Neural Networks to perform operations over hidden layers and image analysis. In the same way, this CNN uses a most unique feature called Convolution that produces the new function based on the mathematical operation among two functions. That new function tells the changes that have been made on the images and how it has been modified. A generalized and standard view of how a Convolution Neural Network works has been shown in fig. 1, that has been taken from [6].

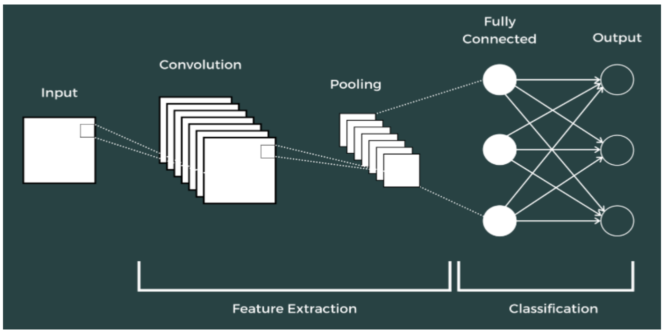

Fig. 1. Generalized view of working of CNN

LeNet-5 was a CNN-based architecture introduced by LeCun et. al. in 1998 [7]. This architecture is made up of 7-Levels which are 2 sets of fully connected as well as sub-sampling layers, and 3 sets of convolutional layers. LeNet-5 architecture is mainly used in classification of handwritten characters either may be on plain paper or bank cheques, or other items which helps for 32*32-pixel grayscale images [8]. As per the author of [9], LeNet-5 architecture consists of following functions for classifying MNIST digits: a tanh activation function, two fully connected layers along with a SoftMax classifier [18]. Overview of LeNet-5 architecture has been given in fig. 2, which is taken from [7].

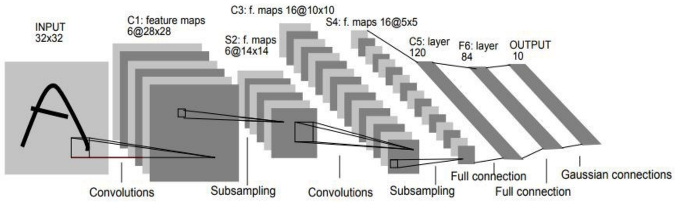

Fig. 2. Overview of LeNet-5 Architecture

ResNet-152, which is the latest version of Residual Neural Network (ResNet), which focuses primarily on the residual learning and skipping the connections [8]. These skip connections feature heavy batch normalization, which are also known as gated units (mostly similar to the elements applied in RNN). If our Neural Network has more than 152 layers, then ResNet-152 is more useful than VGGNet with lower complexity. The architecture of ResNet has been given in fig. 3, which was taken from [10].

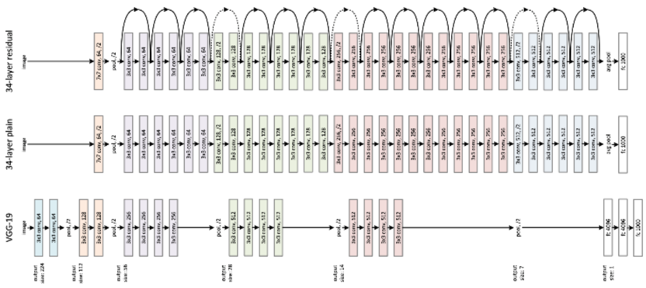

Fig. 3. ResNet-152 Architecture

AlexNet was designed with Eight Layers: 5 of them are convolutional layers (3 max-pooling layers and 2 normalization layers) and 3 are fully connected layers (2 fully connected layers and 1 softmax layer). These layers use ReLu activation except on the output layer [11]. The input size of this model is said to be 224*224*3, but when the padding happens sometimes, then it will be 227*227*3. A comparison of LeNet and AlexNet was given in Fig. 4, which was taken from [12].

Fig. 4. Comaprison of LeNet and AlexNet Layers

Our work aims to find the English meaning to the code in Morse Code which was handwritten as shown in Fig. 5. For that purpose, we have used the existing CNN-Based algorithms named ResNet-152, LeNet-5, and AlexNet on our own dataset named LBRMORSE (created by our team from various age groups of our organization) and concatenated with existing datasets.

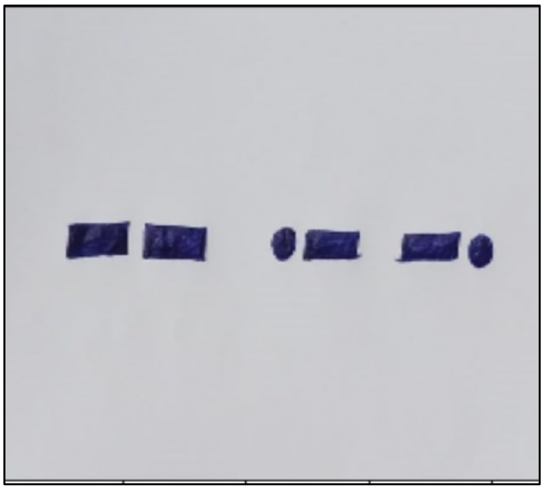

Fig. 5. Sample Word in Morse Code

Our paper has been organized in the following manner: As we discussed the introduction of our research in Section I, Section II deals with the work done by different authors on same area (aka literature survey or related work), Section III goes with our working methodology, Section IV with the results obtained for our work and analysis of different algorithms, and finally the conclusion was given in the ending of paper i.e., Section V.

II. Related Work

Cheng-Hong Yang et al., [13] in their research, Morse code is chosen as an alternative means of conveying messages for those with significant impairments such as muscular dystrophy, cerebral palsy, or hearing loss. For Morse code to be useful as a means of interaction, a consistent typing rate is essential. This limitation severely hampers progress. To accomplish these goals, their suggested system makes use of Support Vector Machines (SVM) and a variable degree variable step size least-mean-square approach. Statistical testing showed that their strategy achieved a 90% recognition rate.

Ching-Hsing Luo et al., [14] in their study, they found that the identification rate of an online Morse code automated recognition system may be enhanced by introducing a novel blend of expert algorithms. The Neural Network (NN) has a recognition rate of 94.4 percent, on average, among three different tests. With the NN trained by a diverse group of specialists, they were able to decipher otherwise intractable time series of Morse codes.

Q. Shanhu et al., [15] in their work, K-Means is used to proactively recognize, classify and decipher Morse code in their study based on matching code table.

W. Li et al., [16] used a fully DL-based NN technology called MorseNet to detect and recognize Morse in a spectrogram. According to their benchmark findings, MorseNet processed 109.5s signals per second when using a GPU, and 83.55s signals per second when without using a GPU.

Y. Yuan et al., [17] used a Deep Learning framework, DeepMorse, was presented as a result of their study; it is used to identify the presence of a Blind Morse Signal in the Wideband Wireless Spectrum. The experimental findings showed that DeepMorse outperformed the modern algorithms on four different datasets when it came to detecting Morse signals. Their experimental findings showed that DeepMorse has got an accuracy of 0.9834 on the 5M dataset, 0.9718 on the 7M dataset, 0.9850 on the 9M dataset, and 0.99 on the 12M dataset.

W. Li et al., [18] used CNN to detect morse code. Their experimental findings showed an accuracy of 99.30% with no frequency drift and 90.23% accuracy with frequency drift.

TABLE II. A Survey on Handwritten Morse Code Recogniton Systems

Authors	Method	Findings / Accuracy Obtained
Cheng-Hong Yang et al., [13]	Support Vector Machines (SVM)	90%
Ching-Hsing Luo et al., [14]	Neural Network (NN)	94.4%
Q. Shanhu et al., [15]	K-Means	Automatic Detection
W. Li et al., [16]	MorseNet	109.5s signals per second (GPU) 83.55s signals per second (without using a GPU)
Y. Yuan et al., [17]	Deep Learning Framework	More than 98%
W. Li et al., [18]	CNN	99.30% with no drift

III. METHODOLOGY

In this section, we are providing the workflow of our work i.e., how we have gathered images, how the impurities were cleaned, how the images were trained and tested, classification of images into folders, applying the CNN based framework on our dataset, and finally testing on dynamic inputs to check whether we are getting correct result or not. The entire process was shown in fig. 6.

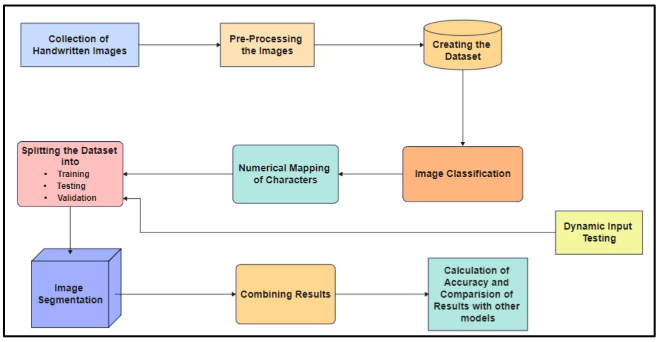

Fig. 6. Workflow of our Comparison

A. LBRMORSE

LBRMORSE – It is a dataset generated by the authors, who are the students of LBRCE (A), Mylavaram. We generated this dataset by gathering the handwritten Morse Code Characters from different age groups ranging between 17 to 55 years i.e., students, teaching and supporting staff of our institute. For creating the dataset, we made a grid of 06*06 i.e., (06 rows of 06 columns each) as shown in fig. 7. Then that was taken out the print and handed over to the members to fill that. To get a better pixel quality, color ratio, smoothing of images, etc. we requested every member to fill in the form using blue or black pen only (one of the filled forms by a member is given in fig. 8).

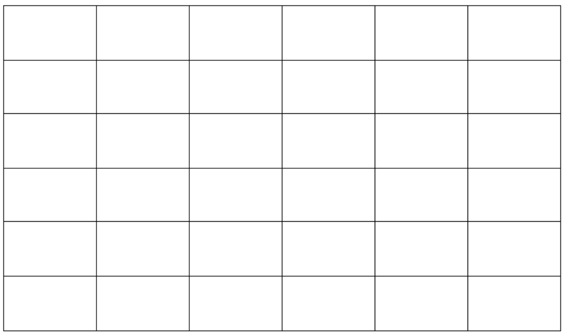

Fig. 7. Data Collection Form Template

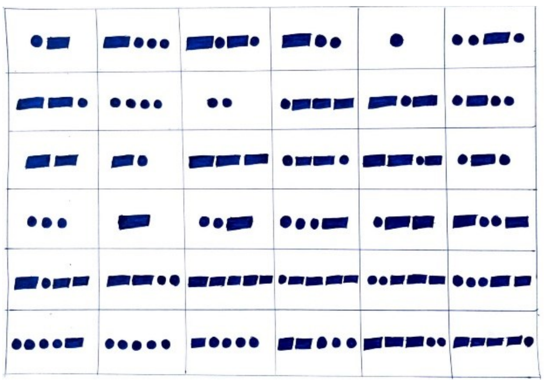

Fig. 8. Sample Form from one of the Member

B. Preprocessing of Images

We have collected samples from 460 members of our organization. Now, we converted the handwritten forms into images by scanning them using a Digital Scanner with a pixel ration of 600 dpi. After that, every image will be divided into 36 small images which represent phonetics and syllables of Amharic Language with a pixel size of 32*32. A total of 36 folders have been created to store the images divided into single images. Then we performed quality checks on individual images to detect any impurities in them or any color issues. Those types of images have been removed from the folders. At the end, the normal handwritten images available in every folder are converted to Grayscale.

C. Mapping

To get a good understanding of images, we have given a symbol name for every folder for all 36 characters. The main intention behind mapping images with symbol name is for detecting the character very easily in the word and mapping the result to output.

D. Training and Testing

Now the complete dataset is ready for usage. Here we are categorizing the dataset into two sections: one is for training data and other is for testing data by considering in the ration of 70:30.

E. Implementation

After the model has trained with training and testing data, now the implementation has been done in Python using different libraries for different activities. Some of the libraries we used are TensorFlow, NumPy, PyPlot in matplotlib, etc. After that, our CNN-Based models have been passed using the Train and Test data. The weights of the samples will be adjusted automatically by CNN which we have used for training purposes. Epoch has been calculated for every phonetic and syllable till 100 epochs.

F. Accuracy Calculation and Analysis

In our work, we have considered the epoch number 100 as the performance metric. Different CNN-based frameworks were trained using our dataset and then we have considered the accuracy obtained from each framework as the Prediction Value (PV).

G. Dynamic Input Testing

At the end, we have passed a random image with Amharic Text (image which was written by a non-participant of dataset creation) and passed through the model to verify whether it is giving the accurate results or not. We passed nearly 250 dynamic images to test the accuracy and got good results.

IV. RESULTS AND ITS ANALYSIS

Section IV of the paper deals with prediction results obtained from various CNN-based frameworks on our own dataset. This task been performed on a 64 GB RAM based computer using the X64 based processor.

A. Model Training at various Epochs:

We trained our models (AlexNet, and LeNet-5) with 5 epochs, 10 epochs, 25 epochs and 50 epochs. From the results, we have observed that we get a better learning rate when the rate of epochs is increasing. In Table III, we have shown the loss and accuracy for various epochs for different models.

TABLE III. ACCURACY COMPARISON AT VARIOUS LEARNING RATES

Epochs	LeNet-5		AlexNet	
	Loss	Accuracy	Loss	Accuracy
5	3.9526	2.31	0.3911	0.6089
10	3.5934	2.59	0.5486	0.4514
15	3.5824	2.94	0.1280	0.872
20	3.5967	2.58	0.0748	0.9252
50	3.5951	2.42	0.0305	0.9695

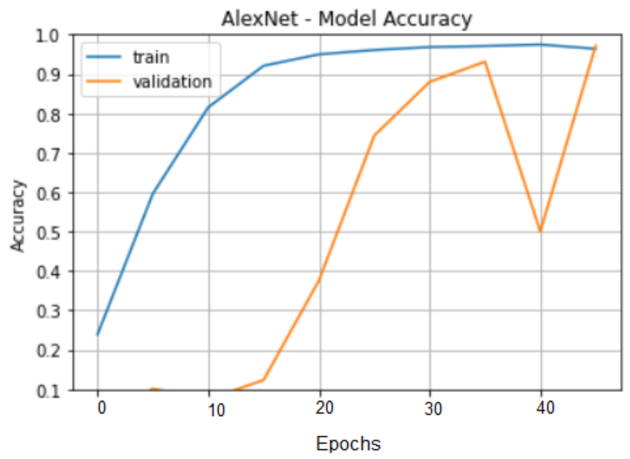

Fig. 9. Training and Validation Accuracy obtained at 50 Epochs for AlexNet

B. Training and Validation Accuracy

Accuracy was calculated for all the models after learning rate. We obtained different accuracies for different models using AlexNet, LeNet-5, ResNet, and VggNet frameworks and the summary of obtained results was shown in Table IV.

TABLE IV. ACCURACY OBTAINED FOR VARIOUS FRAMEWORKS

CNN-Based Framework	Loss	Accuracy
LeNet	3.5951	2.42
AlexNet	0.0305	0.9695
VggNet	2.9874	2.94
ResNet	0.2590	91.25

C. Testing the Model using Dynamic Inputs

We have passed the random inputs (a total of 250) taken from the normal people to our models. For dynamic inputs, we got an accuracy of more than 80%. We have given the sample output obtained from Dynamic Input in fig. 11, fig. 12, fig. 13, fig, 14 and considered the Dynamic Images Accuracy as:

Dynamic Input Accuracy = (Correct Result Obtained)/(Total Images Passed)

Character	Before Data-Processing	After Data-Processing	Accuracy
A	250 Images	212 Images	89.91
B	250 Images	223 Images	86.23
C	250 Images	201 Images	90.21
D	250 Images	198 Images	92.21
E	250 Images	240 Images	93.45
F	250 Images	189 Images	91.35
G	250 Images	178 Images	87.23
H	250 Images	189 Images	93.65
I	250 Images	178 Images	94.25
J	250 Images	200 Images	95.64
K	250 Images	203 Images	93.89
L	250 Images	219 Images	92.45
M	250 Images	229 Images	91.67
N	250 Images	232 Images	89.53
O	250 Images	211 Images	88.53
P	250 Images	204 Images	91.26
Q	250 Images	225 Images	92.86
R	250 Images	212 Images	93.46
S	250 Images	221 Images	81.45
T	250 Images	201 Images	91.67
U	250 Images	189 Images	85.26
V	250 Images	178 Images	86.59
W	250 Images	213 Images	91.86
X	250 Images	222 Images	92.75
Y	250 Images	201 Images	87.85
Z	250 Images	200 Images	92.56
0	250 Images	201 Images	93.76
1	250 Images	221 Images	94.89
2	250 Images	200 Images	95.64
3	250 Images	187 Images	94.52
4	250 Images	178 Images	89.25
5	250 Images	167 Images	88.46
6	250 Images	198 Images	94.78
7	250 Images	200 Images	93.78
8	250 Images	201 Images	92.85
9	250 Images	198 Images	91.56

Fig. 10. Dynamic Inputs Passed to the Model

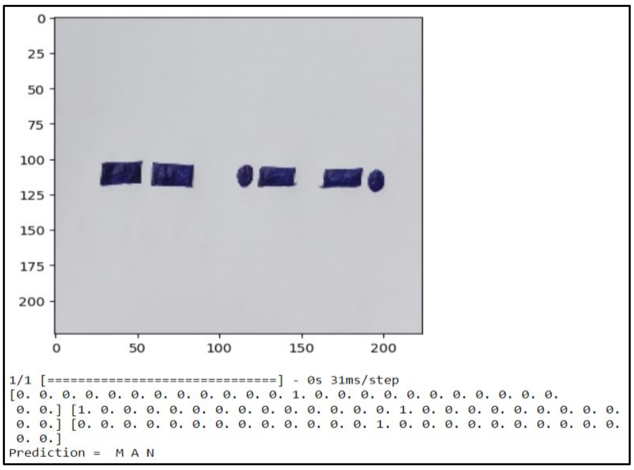

Fig. 11. Sample Result Obtained for Dynamic Input 1

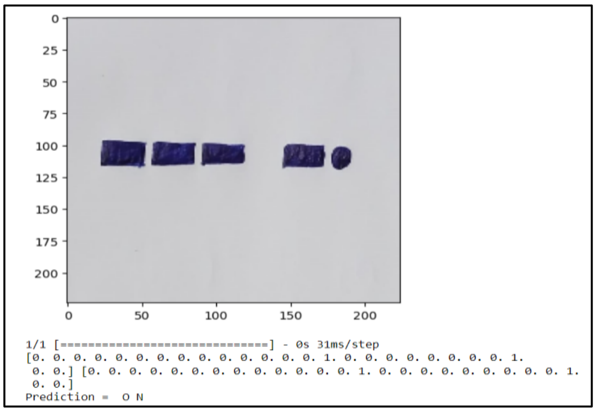

Fig. 12. Dynamic Input 2 Passed to the Model

V. CONCLUSION

The use of Convolutional Neural Networks (CNN) in detection of text from Morse code in images has shown promising results. The process involves converting the images containing Morse code into a format that can be fed into the CNN model for training and testing. The CNN model can successfully learn the patterns in the Morse code images and accurately predict the corresponding text for the individual characters of Morse Code. After training the model with single characters, then apply the image segmentation based on character segmentation and image segmentation. After segmenting the image, apply the image classification on each individual segmented image parts. After classifying each segment combine the results based on words and characters. Finally, the morse code is converted into English Language with better accuracy. However, the accuracy of the model may vary depending on the quality of the images and the complexity of the Morse code patterns. Overall, the use of CNN in text detection from Morse code in images can be a useful tool in various fields, including military and aviation communication, navigation systems, and emergency communication, among others. Further research can be done to improve the accuracy of the model and explore its potential applications in other areas.

REFERENCES

[1] Morse Code, https://en.wikipedia.org/wiki/Morse_code

[2] Optical character recognition, https://en.wikipedia.org/wiki/Optical_character_ recognition

[3] J. Pradeep, E. Srinivasan and S. Himavathi, "Diagonal based feature extraction for handwritten character recognition system using neural network," 2011 3rd International Conference on Electronics Computer Technology, 2011, pp. 364-368, doi: 10.1109/ICECTECH.2011.5941921.

[4] Y. Chen, H. Jiang, C. Li, X. Jia and P. Ghamisi, "Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks," in IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 10, pp. 6232-6251, Oct. 2016, doi: 10.1109/TGRS.2016.2584107.

[5] Introduction to Convolutional Neural Networks (CNN), https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/

[6] Introduction to Convolutional Neural Networks, https://www.theclickreader.com/introduction-to-convolutional-neural-networks/

[7] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient -based learning applied to document recognit ion," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[8] Siddharth Das, CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more…, CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more… | by Siddharth Das | Analytics Vidhya | Medium, 16/November/2017.

[9] Sean Benhur, CNN Architectures from Scratch: From Lenet to ResNet, https://medium.datadriveninvestor.com/cnn-architectures-from-scratch-c04d66ac20c2, 15/November/2020

[10] K. He, X. Zhang, S. Ren, and J. Sun, 'Deep residual learning for image recognition', in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[11] AlexNet: The First CNN to win Image Net, https://www.mygreatlearning.com/blog/alexnet-the-first-cnn-to-win-image-net/, Updated on 31/October/2022.

[12] AlexNet, https://en.wikipedia.org/wiki/AlexNet

[13] Cheng-Hong Yang, Li-Cheng Chin, and Sheng-Chih Hsieh, "Morse code recognition using support vector machines," IEEE EMBS Asian-Pacific Conference on Biomedical Engineering, 2003., Kyoto, Japan, 2003, pp. 220-222, doi: 10.1109/APBME.2003.1302663.

[14] Ching-Hsing Luo and Duu-Tong Fuh, "Online Morse code automatic recognition with neural network system," 2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Istanbul, Turkey, 2001, pp. 684-686 vol.1, doi: 10.1109/IEMBS.2001.1019028.

[15] Q. Shanhu, L. Hongbo and Z. Xu, "Morse Recognition Algorithm Based on K-means," 2019 Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC), Taiyuan, China, 2019, pp. 1-2, doi: 10.1109/CSQRWC.2019.8799149.

[16] W. Li, K. Wang and L. You, "MorseNet: A Unified Neural Network for Morse Detection and Recognition in Spectrogram," in IEEE Access, vol. 8, pp. 161005-161017, 2020, doi: 10.1109/ACCESS.2020.3020860.

[17] Y. Yuan, Z. Sun, Z. Wei and K. Jia, "DeepMorse: A Deep Convolutional Learning Method for Blind Morse Signal Detection in Wideband Wireless Spectrum," in IEEE Access, vol. 7, pp. 80577-80587, 2019, doi: 10.1109/ACCESS.2019.2923084.

[18] W. Li and K. Wang, "Research on Automatic Decoding of Morse Code Based on Deep Learning," 2019 International Conference on Intelligent Computing, Automation and Systems (ICICAS), Chongqing, China, 2019, pp. 488-491, doi: 10.1109/ICICAS48597.2019.00107

[19] S. J. Basha, D. Veeraiah, C. Maanvitha, D. L. Gupta, R. Narayana and M. B. Sahithi, "Comparative Analysis of CNN-Based Frameworks for Handwritten Arabic Numerals Recognition," 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2022, pp. 1382-1388, doi: 10.1109/ICCES54183.2022.9835873.

[20] S. J. Basha, S. R. Madala, K. Vivek, E. S. Kumar and T. Ammannamma, "A Review on Imbalanced Data Classification Techniques," 2022 International Conference on Advanced Computing Technologies and Applications (ICACTA), Coimbatore, India, 2022, pp. 1-6, doi: 10.1109/ICACTA54488.2022.9753392.

[21] S. J. Basha, D. Veeraiah, G. Pavani, S. T. Afreen, P. Rajesh and M. S. Sasank, "A Novel Approach for Optical Character Recognition (OCR) of Handwritten Telugu Alphabets using Convolutional Neural Networks," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2021, pp. 1494-1500, doi: 10.1109/ICESC51422.2021.9532658.