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), MylavaramNTR Dt., Andhra Pradesh, India  
veeraiahdvc@gmail.com

Mangam Surya Prakash   
IV B.Tech Student, Department of CSE  
Lakireddy Bali Reddy College of Engineering (A), MylavaramNTR Dt., Andhra Pradesh, India  
suryaprakash9785@gmail.com Shaik Johny Basha  
Project Guide, Department of CSE  
Lakireddy Bali Reddy College of Engineering (A), MylavaramNTR Dt., Andhra Pradesh, India  
shaikhjanibasha@gmail.com

Kasturi Karthik   
IV B.Tech Student, Department of CSE  
Lakireddy Bali Reddy College of Engineering (A), MylavaramNTR Dt., Andhra Pradesh, India  
karthikkasturi007@gmail.com Karavadi Raviteja   
IV B.Tech Student, Department of CSE  
Lakireddy Bali Reddy College of Engineering (A), MylavaramNTR Dt., Andhra Pradesh, India  
krtproperty72@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, OCR, LeNet-5, and ResNet, AlexNet, and Convolutional Neural Network

# 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.

1. 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 |  |

In today's context, Optical Character Recognition, commonly referred to as OCR, plays a significant role in research by analyzing various patterns to extract information from images [2] [20]. The main objective of OCR is to retrieve textual content from images, which may include handwritten or printed text from a printer, as well as typed text. By employing OCR, the laborious and time-consuming task of comprehending text within an image, without sufficient language expertise, is greatly minimized.

Deciphering text in an unfamiliar language poses a significant challenge for humans, regardless of whether it is printed text, typed data, or handwritten content. While individuals may be able to understand typed or printed text to some extent, handwritten text presents challenge due to the unique writing styles and variations among individuals [19]. Therefore, the ability to identify and convert text from an image into English language assumes a crucial role. To begin with, the characters in the image are extracted using a variety of techniques, including diagonal-based feature extraction [3], deep feature extraction [4], and other relevant methods. Like humans, machines also require training in a specific language to comprehend information contained in images. To train machines with various human understandable languages, frameworks of CNN such as Xception, ResNeXTt50, VGG-16, InceptionV3, and other similar architectures can be utilized. These frameworks can facilitate language training for machines. These frameworks prove invaluable to humans by providing high prediction rates and the results are most accurate when they have been trained properly and tested.

Convolutional Neural Networks (ConvNets/CNNs) are a specialized type of Deep Neural Networks renowned for their ability to process image inputs and perform tasks such as layer pooling and extracting the features [5]. Within Neural Networks, matrix multiplication plays a pivotal role in executing operations within hidden layers and effectively analyzing image data. Convolution, a key feature of CNNs, involves applying mathematical operations between two functions to generate a new function that highlights changes and modifications made to the images. A comprehensive visual representation of the Convolutional Neural Network architecture and its operations can be observed in Figure 1, obtained from [6].

Diagram

Description automatically generated

1. CNN Architecture Generalized View

LeNet-5, was introduced by LeCun in the year 1998 [7] with other researchers, consists of seven levels. The key role of this framework is specifically designed to excel in the grouping of handwritten characters, including those found on normal papers, official cheques/drafts, and various other items. It is particularly effective for processing grayscale images with dimensions of 32x32 pixels [8]. As described by the researchers of [9], the LeNet-5 incorporates various elements for classifying MNIST digits. These components consist of a tanh activation function, two fully connected layers, and a SoftMax classifier [18]. For a visual representation of the LeNet-5 architecture, please refer to Figure 2, sourced from [7].

Diagram

Description automatically generated

1. Gomprehensive Wokring Struture of LeNet-5

ResNet-152, which represents the most recent iteration of the ResNet, is designed with a strong focus on residual learning and skipping the connections [8]. Batch normalization was a key source for skipping the connections and are often referred to as gated units, exhibiting similarities to elements commonly found in Recurrent Neural Networks (RNNs). When dealing with Neural Networks consisting of more than 152 layers, ResNet-152 outperforms VGGNet by providing a more favorable trade-off between performance and complexity. Figure 3, sourced from [10], provides a visual representation of the ResNet architecture.

A picture containing calendar

Description automatically generated

1. ResNet-152 Architecture

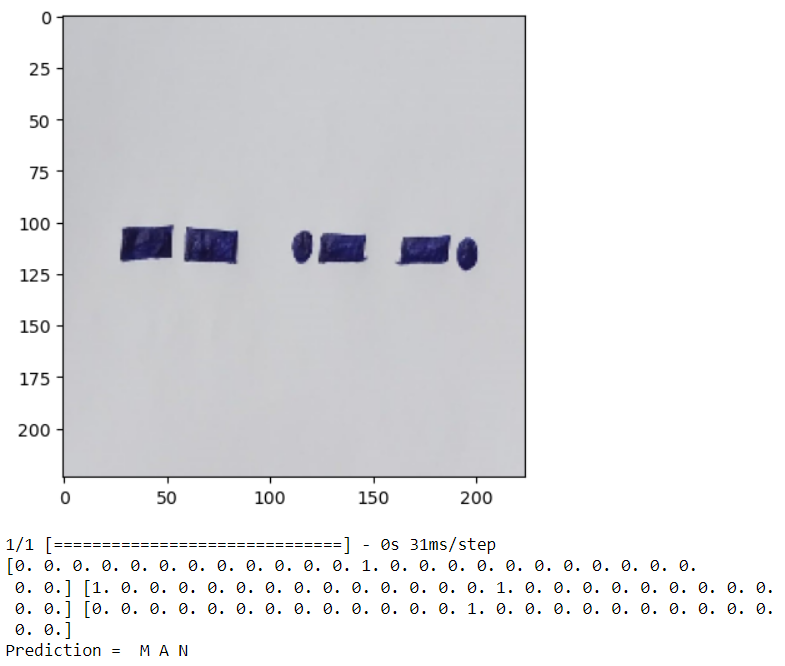
AlexNet is an architecture in the field of neural networks that consists of a total of eight layers. The architecture of these layers includes five convolutional layers, three of which are max-pooling layers, and two normalization layers. Additionally, there are three fully connected layers, consisting of two fully connected layers and one softmax layer. With the exception of the output layer, the rectified linear unit (ReLU) activation function is applied to all the layers [11].The model is specifically designed to handle input sizes of 224x224x3, although occasional padding may increase the dimensions to 227x227x3. Figure 4, sourced from [12], provides a visual comparison between the architectures of LeNet and AlexNet.

Table

Description automatically generated

1. Comaprison of LeNet and AlexNet Layers

Our project aims to translate handwritten Morse Code into English, as depicted in Figure 5. To accomplish this, we utilized CNN-Based algorithms, specifically AlexNet, LeNet-5, and ResNet-152, on our custom dataset called LBRMORSE. This dataset was meticulously curated by our team, which included members from diverse age groups within our organization. Furthermore, we augmented our dataset by incorporating existing datasets to improve the overall training process.

****

1. Sample Word in Morse Code

Our paper is organized as follows: Section I introduces our research topic and provides an overview. In Section II, we conduct a comprehensive literature review, discussing the work and findings of various authors in the same field. This section serves as a comprehensive review of other researchers’ work. Section III outlines our methodology and describes the approach we adopted in our research. In Section IV, a comparison was made among the results obtained and done an analysis of various frameworks used. Finally, in Section V, we conclude the paper by summarizing the findings and drawing our conclusions based on the research conducted.

# 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.

1. 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 |

# Workflow and Its Methodology

In this section, we provide a comprehensive overview of the workflow employed in our work, outlining the specific steps we followed. We describe how we collected images, performed impurity cleaning, trained, and tested the images, organized them into folders based on classification, applied a various framework of CNN to our dataset, and conducted tests on runtime inputs (dynamic) to validate the accuracy of our results. Complete workflow has been visualized in Figure 6.

Diagram

Description automatically generated

1. Representation of Complete Workflow

## LBRMORSE

The authors, who are students at LBRCE (A), Mylavaram, have developed a dataset called LBRMORSE. The dataset consists of handwritten Morse Code characters collected from individuals spanning a wide range of age groups, from 18 to 60 years. The participants include various age groups of learners, employees like faculty and technical staff from institute. To create the dataset, we employed a grid format of 6x6, as illustrated in Figure 7. The grid was printed and distributed to the participants, who were instructed to fill in the Morse Code characters using either a blue or black pen. This approach was implemented to enhance the quality of the pixels, color ratio, and overall smoothing of images. An example of a completed form by one of the participants is displayed in Figure 8.

Graphical user interface, application, PowerPoint

Description automatically generated

1. Template used for Collecting Data

**Calendar

Description automatically generated**

1. Handwritten Morse Code Form from a Participant

## Preparing Images for Prediction

We collected samples from 460 members within our organization. These samples, consisting of handwritten forms, were scanned using a High-Definition Scanner at a resolution of 300 dpi to convert them into images. Subsequently, each image was divided into 36 smaller images, representing numbers and alphabets of the Morse Code, with a pixel size of 32x32. To organize the dataset, we created 36 folders, each corresponding to a specific category, to store the individual images. Quality checks were conducted on each image to identify any impurities or color-related issues. Images found to have such problems were removed from their respective folders. Finally, the remaining handwritten images in each folder were converted to grayscale.

## Character Charting

To facilitate easy identification of characters in words and mapping the results to their corresponding outputs, we have assigned symbol names to each folder representing the 36 characters. This mapping of images with symbol names allows for a better understanding of the images and simplifies the character detection process.

## Training Data and Testing Data

Dataset can be used now and splitted into two sections: training data and testing data. We have divided the dataset in a 70:30 ratio, with 70% of the data reserved for training and the remaining 30% for testing.

## Implementation

After completing the training and testing of the model with our dataset, we proceeded with the implementation in Python. We utilized various libraries for different tasks, including TensorFlow, NumPy, PyPlot in matplotlib, and others. The CNN-based models were then applied to both the training and testing data. During the training process, CNN automatically adjusted the weights of the samples. We trained each phonetic and syllable for up to 100 epochs, enabling comprehensive learning and optimization of the models.

## Calculation of Accuracy and getting Analysis

In our study, we chose to use 100 epochs as a metric to evaluate performance of the various frameworks. We provided training to multiple frameworks with our dataset and measured the accuracy achieved by each framework, which we refer to as the Prediction Value (PV). This allowed us to assess the effectiveness and performance of the frameworks in accurately predicting the desired outcomes.

## Testing the Runtime Inputs

Finally, to validate the accuracy of our model, we conducted a test by inputting a random image containing Handwritten Morse Code. We passed the various runtime images by our trained frameworks to determine whether we are getting accurate results or not. Additionally, we tested the model's performance by evaluating approximately 250 dynamic images and obtained satisfactory results, indicating the effectiveness of our approach.

# Results and Comparison Analysis

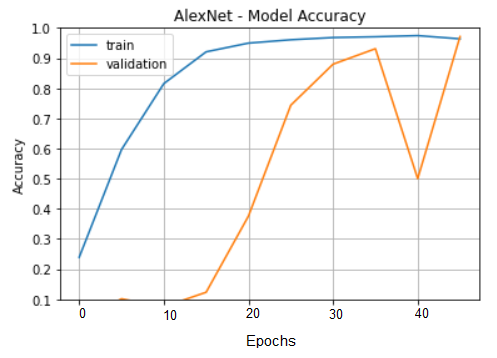
Section IV of the paper centers around the prediction results achieved by employing various CNN-based frameworks on our customized dataset. These computations were executed on a computer system equipped with a 64 GB RAM and an X64-based processor, providing sufficient computational resources for the analysis and evaluation of the frameworks.

## Training the Model at different Epochs:

We conducted training sessions for our models, namely AlexNet and LeNet-5, with different numbers of epochs, including 5, 10, 25, and 50. Based on our findings, we observed that the learning rate improved as the number of epochs increased. Table III presents the loss and accuracy metrics for different models across various epoch settings.

1. Accuracy Comparison for LeNet-5 and AlexNet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |

****

1. Training and Validation Accuracy obtained at 50 Epochs for AlexNet

## Accuracy Calculation

Upon calculating the accuracies based on the learning rate, we observed varying results across different models, such as AlexNet, LeNet-5, ResNet, and VggNet frameworks. The summary of these results, including the accuracies achieved by each model, is presented in Table IV, providing a comprehensive overview of the performance of the individual models.

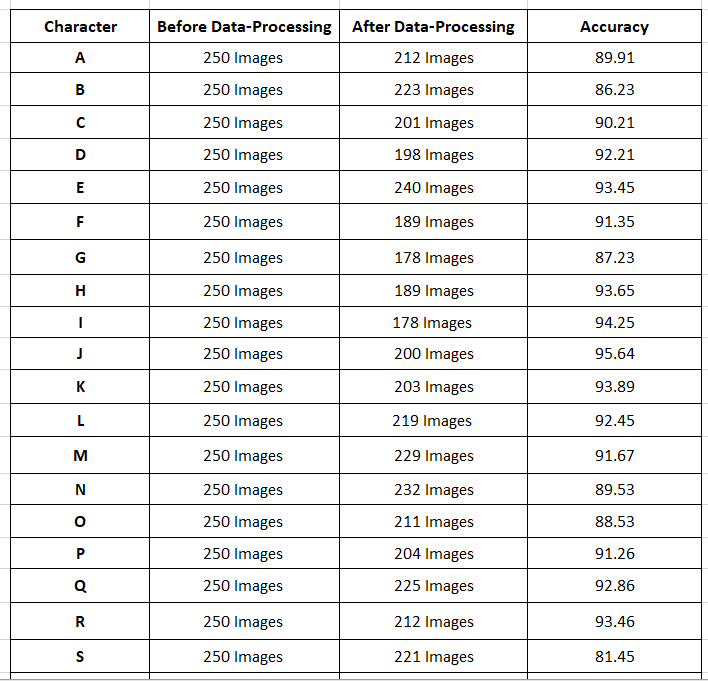
1. Accuracy Obtained for Various Frameworks

|  |  |  |
| --- | --- | --- |
| **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 |

## Model Testing with Runtime Inputs

We subjected our models to random inputs from the public, totaling 250 samples. In the case of dynamic inputs, we achieved accuracy exceeding 80%. Sample outputs obtained from dynamic inputs are displayed in Figure 11, Figure 12, Figure 13, and Figure 14. The accuracy of the dynamic images is represented as follows:

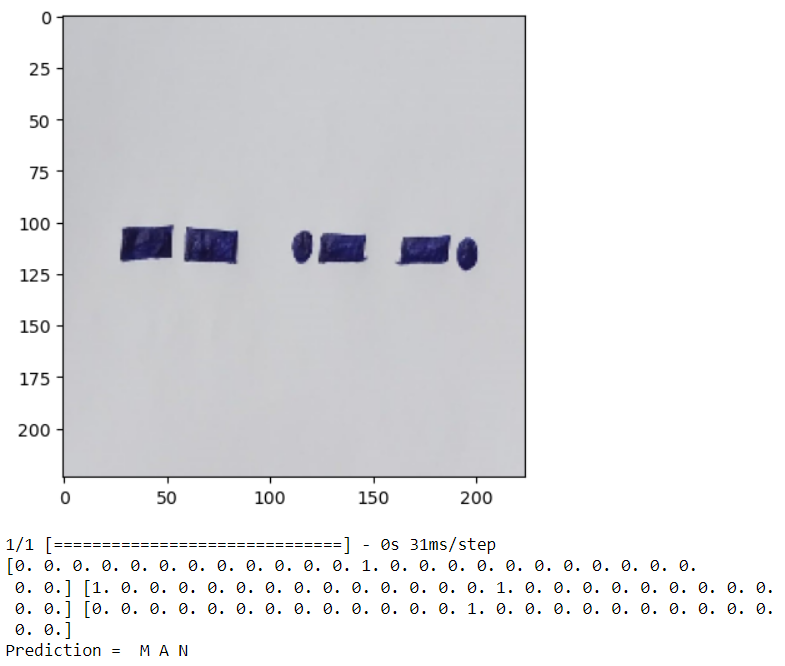
Accuracy of Runtime Input = No. of Images given Correct Output / Total No. of Images Passed

****

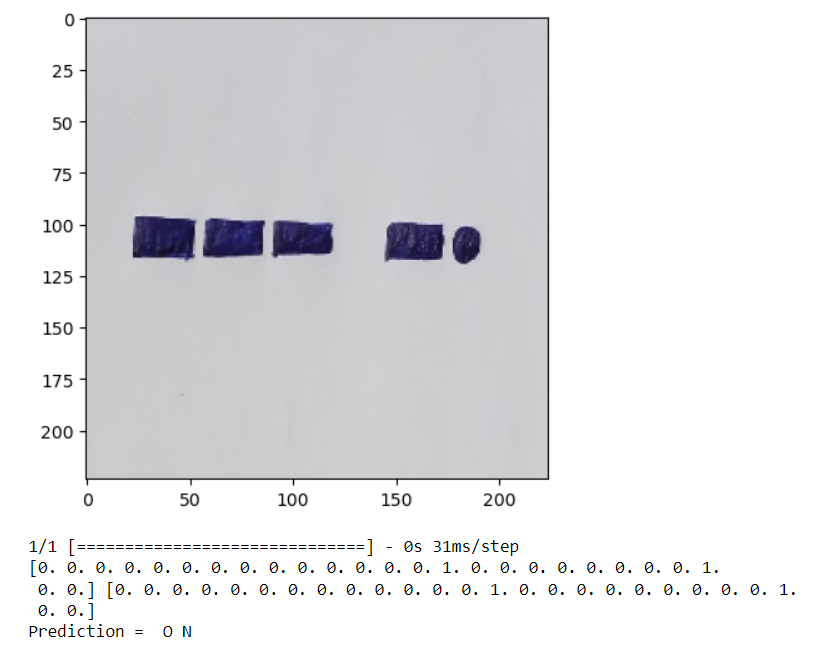
**Table

Description automatically generated**

1. Dynamic Inputs Passed to the Model

****

1. Sample Result Obtained for Dynamic Input 1

****

1. Dynamic Input 2 Passed to the Model

# 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_c> haracter\_ 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](https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5), 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.