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

Shaik Johny Basha   
Project Guide, Department of CSE  
Lakireddy Bali Reddy College of Engineering (A), MylavaramNTR Dt., Andhra Pradesh, India  
shaikhjanibasha@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 Duggineni Veeraiah   
Project Guide, Department of CSE  
Lakireddy Bali Reddy College of Engineering (A), MylavaramNTR Dt., Andhra Pradesh, India  
veeraiahdvc@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* — Morse code is a method used in telecommunication to encode text characters as standardized sequences of two different electronic pulses usually represented as short pulse(dot) and long pulse(dash). 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 style of strokes. Our work aims to develop an automated Morse code recognition system by training a convolutional neural network (CNN) model using 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 that the proposed approach achieves high accuracy in recognizing Morse code characters in images, making it a promising solution for automated Morse code recognition systems.

Keywords — Morse Code, OCR, AlexNet, Lenet-5, and ResNet-152, Convolutional Neural Network

# Introduction

Morse Code is a method of telecommunication, used to transmit messages to longer distances across telegraph wires via electric pulses represented as short pulse(dot) and long pulse(dash). In the beginning of the 19th century, the messages were delivered either to horses or with ravens. For the first time in the history of mankind a group of scientists invented a telecommunication device called telegraph, which was used to transmit the 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 out of which 26 characters are alphabets and the remaining 10 characters are digits. There is no variation between lowercase and uppercase letters. Each morse code symbol is formed by a sequence of dots and dashes. 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 the letters present in the word are separated by a space which has a duration of three dots, and the words are separated by a space equal to seven 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 |  |

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 [4]. The primary focus of this OCR is to get the information from images, where the image consists 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. 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 [5], Deep Feature Extraction [6], 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 part of Deep Neural Networks that reads the images or pictures as input and helps in image analysis such as feature extraction, layer pooling, etc. [7]. 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 [8].

Diagram

Description automatically generated

1. Generalized view of working of CNN

LeNet-5 was a CNN-based architecture introduced by LeCun et. al. in 1998 [9]. 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 [10]. As per the author of [11], LeNet-5 architecture consists of following functions for classifying MNIST digits: a tanh activation function, two fully connected layers along with a SoftMax classifier. Overview of LeNet-5 architecture has been given in fig. 2, which is taken from [9].

Diagram

Description automatically generated

1. 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 [10]. 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 [12].

A picture containing calendar

Description automatically generated

1. ResNet-152 Architecture

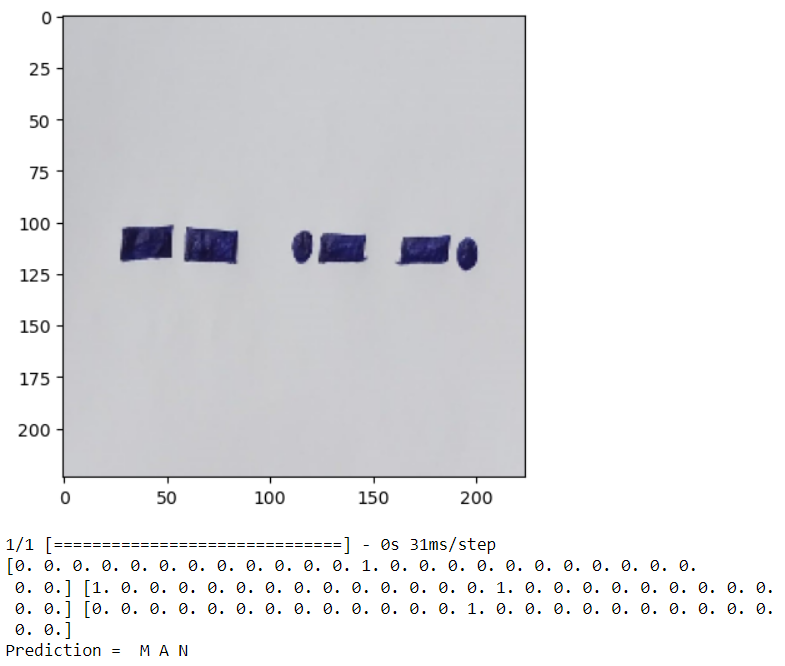
AlexNet was designed with Eight Layers: 5 of them are convolutional layers (3 max-pooling layers and 2 normalization layers) and 3 are fully connected layers (2 fully connected layers and 1 softmax layer). These layers use ReLu activation except on the output layer [13]. The input size of this model is said to be 224\*224\*3, but when the padding happens sometimes, then it will be 227\*227\*3. Comparison of LeNet and AlexNet was given in Fig. 4, which was taken from [14].

Table

Description automatically generated

1. Comaprison of LeNet and AlexNet Layers

Our work aims to find the English meaning to the code in Amharic 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 LBRAMHARIC (created by our team from various age groups of our organization) and concatenated with existing datasets.

****

1. 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.

# Related Work

Balamurugan et al. [10] have introduced the CNN-based method for Character Recognition of Morse Code Language. They have created a dataset of Morse Code Characters. Their work function with a max-pooling of 2\*2. For their dataset and model, they obtained an average accuracy of 81% for characters. But they haven’t performed their work on complex images.

Ren et al. [12] have generated a new dataset with more than 10,000 in Morse Code. In their proposed work, they have used Support Vector Machines (SVM) along with Connectionist Temporal. After that, they tested their model with the printed as well as synthesized datasets and obtained a good accuracy of more than 84% for printed datasets.

LeCun et al. [15] have performed the text recognition for Morse Code Sign Language. In this, they have considered the various types of handwritten images of Morse Code. For their work, they have obtained an accuracy of 91% for all the characters and numerics.

Redmon et al. [16] used various hybrid acoustic modeling units such as phonetics, syllables, and rounded phonetics. They developed a Deep Neural Network based on the above said units and they obtained a good outcome over the GMM (Gaussian Mixture Models). Their results are better with a WER of 11.28% and maximum of 17% for context-dependent and independent models with very limited dataset.

AlMajthoub et al. [19] have used their own dataset with more than 10,000 images. Their work primarily focused on identifying the characters in printed character images. They have done the comparison of various authors work on those type of images. The proposed work of them have achieved an accuracy of 92%, which is more than the other authors. But the problem is they have focused on printed characters, not on handwritten images.

1. A Survey on Handwritten Morse Code Recogniton Systems

|  |  |  |
| --- | --- | --- |
| **Authors** | **Method** | **Accuracy** |
| Balamurugan et al. [10] | Artificial Neural Networks | 81% |
| Ren [12] | Support Vector Machines | 84% |
| LeCun [13] | Artificial Neural Networks | 91% |
| Redmon et al. [14] | CNN-Based Algorithm | 94% |
| Our Method | AlexNet | 96% |

# 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.

Diagram

Description automatically generated

1. Workflow of our Comparison

## LBRMORSE

LBRMORSE is a dataset created by the authors from Lakireddy Bali Reddy College of Engg. We created this dataset by collecting 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 the form using blue or black pen only (one of the filled forms by a member is given in fig. 8).

Graphical user interface, application, PowerPoint

Description automatically generated

1. Data Collection Form Template

**Calendar

Description automatically generated**

1. Sample Form from one of the Member

## Preprocessing of Images

We have collected the samples from 460 members 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 have performed quality check 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.

## 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.

## 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.

## 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 purpose. Epoch has been calculated for every phonetic and syllable till 100 epochs.

## 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).

## 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.

# Results and Its Analysis

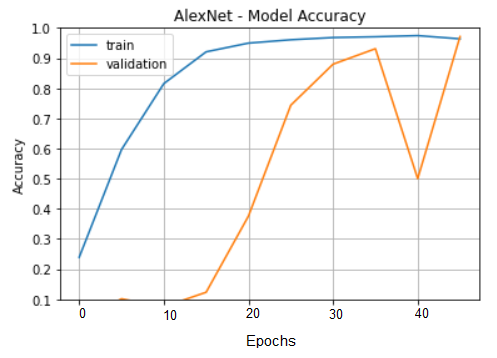
Section IV of our paper deals with the accuracy and prediction results obtained for various frameworks on our dataset. Our work has been performed on Windows 10 Operating System with a RAM of 64 GB and X64 based processor.

## 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.

1. 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 |

****

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

## 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.

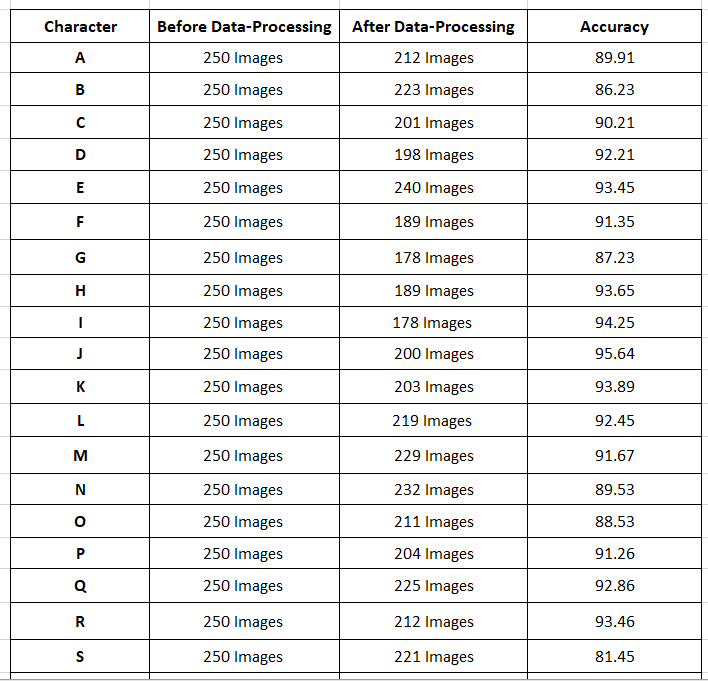
1. Accuracy Obtained for Various Frameworks

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

## 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:

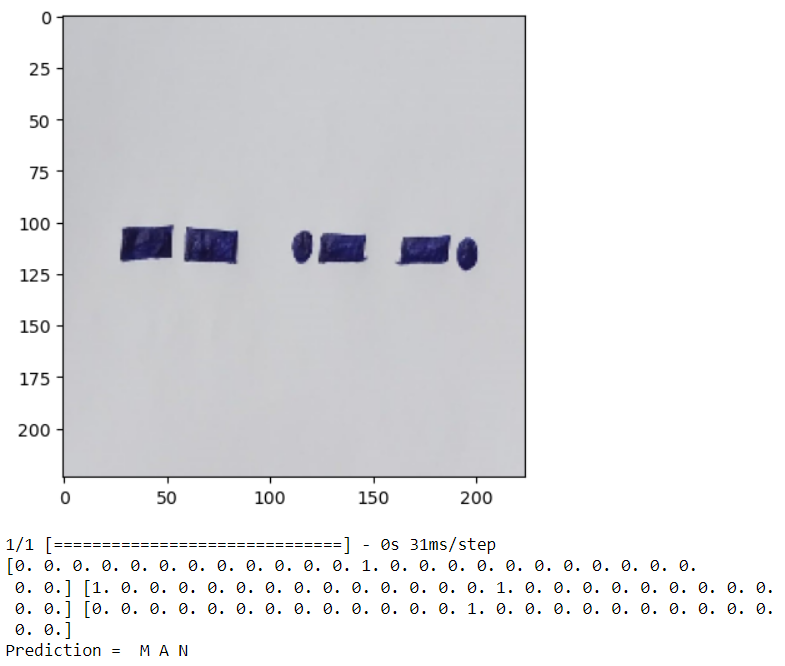
Dynamic Input Accuracy = (Correct Result Obtained)/(Total Images Passed)

****

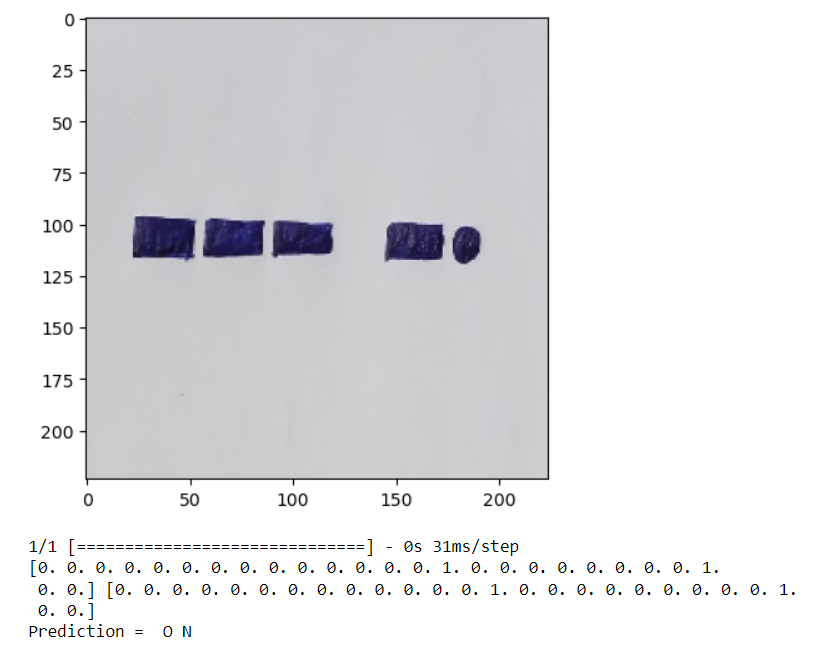
**Table

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. Amharic, <https://en.wikipedia.org/wiki/Amharic>
2. Geʽez script, <https://en.wikipedia.org/wiki/Ge%CA%BDez_script>
3. Optical character recognition, <https://en.wikipedia.org/wiki/Optical_c> haracter\_ recognition
4. Introduction to Convolutional Neural Networks (CNN), <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>
5. Introduction to Convolutional Neural Networks, <https://www.theclickreader.com/introduction-to-convolutional-neural-networks/>
6. 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.
7. Sean Benhur, CNN Architectures from Scratch: From Lenet to ResNet, <https://medium.datadriveninvestor.com/cnn-architectures-from-scratch-c04d66ac20c2>, 15/November/2020
8. 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.
9. AlexNet, <https://en.wikipedia.org/wiki/AlexNet>
10. S. Sudha, S. Mythili and S. S. Balamurugan, "Detection of Text from Morse Code in Images using Convolutional Neural Network," 2021 International Conference on Electronics, Communication, and Aerospace Technology (ICECA), Coimbatore, India, 2021, pp. 1192-1197, doi: 10.1109/ICECA51584.2021.9485803.
11. R. Girshick, J. Donahue, T. Darrell and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 1, pp. 142-158, Jan. 2016, doi: 10.1109/TPAMI.2015.2436995.
12. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
13. Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.
14. J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.
15. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014 arXiv:1409.1556.
16. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going Deeper with Convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.
17. Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 2014, pp. 1701-1708, doi: 10.1109/CVPR.2014.220.
18. C. Yan, Z. He, H. Wang and X. Liu, "Morse code identification based on convolutional neural network," 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, 2019, pp. 1531-1535, doi: 10.1109/CompComm49359.2019.8941357.
19. S. AlMajthoub, H. AlDmour, M. AlKabi and R. Alghazo, "Morse Code Decoder using Deep Learning," 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 2019, pp. 449-454, doi: 10.1109/JEEIT.2019.8717433.
20. A. M. Abd El-Latif, A. M. Ibrahim and H. A. Hefny, "Morse code recognition using neural networks," 2015 8th International Conference on Human System Interaction (HSI), Warsaw, Poland, 2015, pp. 187-192, doi: 10.1109/HSI.2015.7170695.
21. X. Chen, H. Jia, Y. Yang, C. Li and H. Wang, "Morse code recognition based on convolutional neural network with improved structure," 2020 IEEE 3rd International Conference on Electronics Technology (ICET), Chengdu, China, 2020, pp. 1432-1436, doi: 10.1109/ICET48890.2020.00025.
22. G. Li, Y. Li, B. Li, X. Zhang and Q. Li, "Morse code recognition with convolutional neural networks," 2019 IEEE 14th International Conference on Industrial and Information Systems (ICIIS), Ansan, Korea (South), 2019, pp. 475-478, doi: 10.1109/ICIIS47389.2019.9026748.
23. A. Rizk and M. A. Sharaf, "Morse Code Recognition Using Convolutional Neural Network," 2018 International Conference on Innovative Trends in Computer Engineering (ITCE), Aswan, Egypt, 2018, pp. 371-375, doi: 10.1109/ITCE.2018.8354047.
24. N. Sharma and P. Singh, "Morse Code Decoding using Deep Learning," 2019 IEEE 9th International Conference on Advanced Computing (IACC), Tiruchengode, India, 2019, pp. 311-314.