# Handwritten Text Classification Based on Convolutional Neural Network

Aldyn Chen*

Portola High School, Irvine, CA, USA

* Corresponding Author Email: 23ChenAldyn@iusd.org

**Abstract.** The convolutional neural network (CNN) is a popular and highly effective deep learning technique for image classification. As the popularity of CNNs grew, the model has become popular in several machine learning problems. This paper utilizes a CNN model and the popular LeNet-5 transfer learned model to classify texts after the words are preprocessed and segmented from an image. The EMNIST database is used to train the models. The paper achieves an 89.36% validation accuracy on the EMNIST Balanced dataset and an 86.64% on the EMNIST By_Class dataset for the CNN model of four convolutional layers and one dense layer. Similarly, the LeNet-5 model obtained a validation accuracy of 85.88% on the EMNIST Balanced dataset and 85.01% accuracy on the EMNIST By_Class dataset. However, despite a higher accuracy in the EMNIST Balanced dataset, the EMNIST By_Class dataset achieves better results in real-world handwritten texts.

**Keywords:** Handwritten Text Classification, Convolutional Neural Network, EMNIST, LeNet-5, Character Segmentation.

## 1. Introduction

As the growth of machine learning continues to rise, more real-world applications such as handwritten recognition grow in popularity and get developed to help people. The advantages of converting an image of text into a usable form, such as a PDF or a Docx file, are boundless: translating texts observed in the real world, the automatic processing of documents or medical prescriptions, and much more. Handwritten recognition (HWR), or handwritten text recognition (HTR), is the ability of a computer to perceive and determine texts after being fed large amounts of data. Handwritten character recognition is one type of optical character recognition (OCR): the identification of text, both printed and digital. The two different types of handwritten character recognition, offline and online character recognition, are categorized based on the types of data they use to identify texts. More specifically, the data for offline character recognition is static, meaning it identifies characters from a fixed image. In comparison, online character recognition is a “dynamic process” [1]. In this paper, an offline character recognition method is implemented.

The classification of single characters and digits has already reached high results using a CNN model as the character classifier. For example, Ahlawat et al. used a CNN model to achieve an accuracy of 99.87% on a highly popular digits database, the MNIST dataset. The dataset contains a total of 70,000 training and testing samples of handwritten numbers in the shape of 28 x 28 pixels [2, 3]. However, several challenges of handwritten recognition are still present: the classification of handwritten numerical and character strings after character segmentation, unconstrained handwriting, connected texts, such as cursive writing, noise in the background, the uniqueness of everyone’s writing style, and the inconsistency of one’s handwriting at different times. All the factors simulate human handwriting and continue to lower accuracies [4]. As a result, the reasons provided are challenging problems in this field. Many models and methods have been used and tested for handwritten recognition: hidden Markov Model, support vector machine, incremental recognition method, part-based method, ensemble method, and neural networks [1, 4]. In this paper, a CNN model is implemented as the character classifier to identify texts. First, the texts getting identified will go through the preprocessing phase of reducing noise, normalizing the values, and segmenting the texts into single characters. After the texts get preprocessed, the single characters will be inputted into the model to be classified. For the data that trains the model, this paper utilizes the EMNIST dataset.

Section 2 discusses image classification and how CNNs get utilized in the field. Section 3 explores the EMNIST dataset, the preprocessing phase, the segmentation phase, and the models that get tested. Section 4 looks into the results of the CNN model and the LeNet-5 model using different datasets of the EMNIST dataset. Finally, the analysis of the results will be presented in section 5.

## 2. Related Work

Handwriting recognition is a category of image recognition. Within image recognition, there are several types: face recognition, pattern recognition, character recognition, and more. The objects are recognized through image classification using machine learning techniques, including the Decision Tree, Random Forest Classifier, and several types of neural networks, such as the multi-layer perceptron and the CNN model. In an image classification algorithm, the following steps get performed: selecting a dataset to train the model, the processes of data preprocessing and feature extraction, selecting an optimal classification method, post-classification processing, and, finally, evaluating the effectiveness of the method. Regarding how effective these machine learning techniques are for image classification; several factors contribute to the overall accuracy. Using different samples of the COREL1000 dataset, Pin W et al. found that SVM performs slightly better than CNN on smaller datasets, but the CNN model achieves significantly higher accuracies on larger datasets. In the study, for the small dataset, the accuracies are 86% and 83%, and for the large dataset, the accuracies are 88% and 98% for SVM and CNN, respectively [5]. Traditional machine learning techniques perform better on smaller datasets, while deep learning techniques achieve higher scores on larger datasets.

Deep learning techniques such as CNNs are commonly used for accurate results for image classification and the categories of image recognition: face recognition, character recognition, and more. Using a SIAMESE CNN model to identify and recognize faces, Weihong W. et al. achieved 91% on the Our Database of Faces dataset and 81% on the Labeled Faces in the Wild database in response to the face landmark detection problem [6]. Some researchers used a CNN model for food classification, classifying food with images for a dietary assessment system. The researchers achieved an accuracy of 92.86% after classifying a food dataset of 16,643 images [7]. Finally, Acharya U. R. and other researchers used a deep 9-layer CNN model to distinguish among five different types of heartbeats using an electrocardiogram (ECG). The researchers achieved 94.03% in distinguishing the heartbeats on the original signals and 93.47% on the preprocessed signals recorded by the ECGs [8].

## 3. Methodology

### 3.1. Data

The handwritten recognition model in this paper is trained under the EMNIST (Extended MNIST) dataset. EMNIST is a dataset of uppercase letters, lowercase letters, and digits in the same format as the MNIST dataset, 28 x 28 pixels. As a result, the data is compatible with models trained with the MNIST dataset, earning its title, Extended MNIST. More specifically, this study uses the EMNIST Balanced and By_Class datasets. The EMNIST Balanced dataset has a total of 131,600 training and testing samples. In comparison, the EMNIST By_Class dataset contains 814,255 training and testing samples [9]. Apart from the size differences between the two datasets, while the EMNIST By_Class dataset has images for all letters in the alphabet, uppercase and lowercase, and numbers, the EMNIST Balanced dataset neglects letters where the uppercase character is similar to the lowercase letter.

### 3.2. Data Preprocessing

Several data normalization methods are available: min-max scaling, z-score normalization, log scaling, and batch normalization. This paper utilizes min-max scaling to the data so the values will be between 0 and 1. The formula is presented below:

$$a = \frac{(b - \min(b))}{(\max(b) - \min(b))}, \tag{1}$$

The value b is the original number and a is the number after min-max normalization is applied. The process of normalization increases the accuracy of machine learning models by reducing the influence that certain factors of the data may have on the overall performance of the model machine learning. When classifying handwriting texts, noise is unavoidable. As a result, to simulate as much noise as possible while maintaining the quality of the letters classified in the testing dataset, this study applied the gaussian blur to each of the samples. Gaussian blur can improve the results gathered from the images in the testing data in the feature extraction phase. Finally, this study applied an encoding technique to the labels. It is necessary to use encoding to remove unintentional biases. This paper uses the one-hot encoding method. In a research paper by Sorensen and others, one hot encoding is one of the best and most popular choices compared to cat boost encoding, categorical encoding, label encoding, ordinal encoding, and mean encoding in multiple machine learning methods [10].

### 3.3. Data Segmentation

Character segmentation is the phase of breaking apart and separating paragraphs of texts into individual characters and symbols. Character segmentation is a crucial step, as the accuracy of the extracted letters will majorly affect the accuracy of the classifier. There are several approaches to character segmentation: the classic approach, which utilizes dissection techniques, recognition-based segmentation, and holistic methods. In this paper, the character segmentation method got implemented through the canny edge detection algorithm from OpenCV, a library for computer vision. The algorithm goes through the following stages: the process of removing noise, extraction of features from the image, identifying the object within overlapping boxes, and a process called hysteresis thresholding. Figure 1 illustrates the results. The characters obtained after applying the canny edge detection method get sorted before they are inputted to get classified by the CNN model [11, 12].

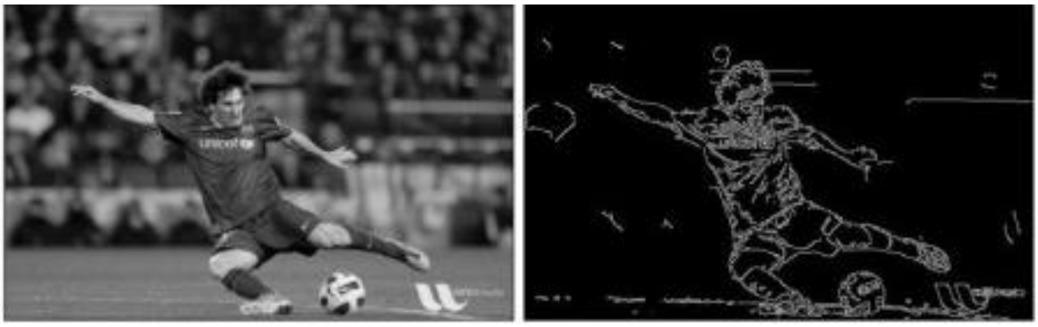


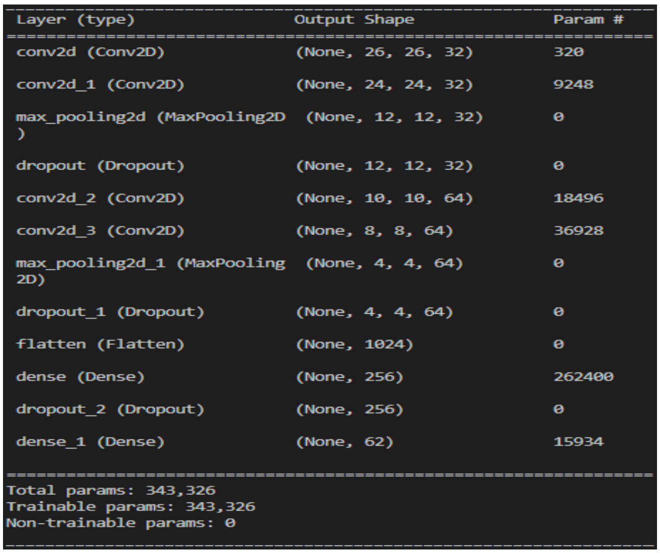
**Figure 1:** Canny Edge Detection in OpenCV [11]

In Figure 1, the left graphic is the original photo of the soccer player, and the right graphic is the results after the canny edge detection method is applied.

### 3.4. Model

#### 3.4.1. Convolutional Neural Network

The CNN is a type of neural network that uses filters to extract desired and distinguishable features from a photo. In recent decades, CNN has risen in popularity and is a highly effective machine learning technique for image recognition and classification. In a study, Hijazi et al. achieved 99.58% using a CNN model on the GTSRB dataset (51,840 images of road signs in Germany), the best and most accurate model to date [13]. A CNN model consists of many building blocks: the convolutional layer, the nonlinearity layer, the fully connected layer, and the pooling layer. The specifics of the CNN model used in this paper are shown in Figure 2.



| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d (Conv2D) | (None, 26, 26, 32) | 320 |
| conv2d_1 (Conv2D) | (None, 24, 24, 32) | 9248 |
| max_pooling2d (MaxPooling2D ) | (None, 12, 12, 32) | 0 |
| dropout (Dropout) | (None, 12, 12, 32) | 0 |
| conv2d_2 (Conv2D) | (None, 10, 10, 64) | 18496 |
| conv2d_3 (Conv2D) | (None, 8, 8, 64) | 36928 |
| max_pooling2d_1 (MaxPooling 2D) | (None, 4, 4, 64) | 0 |
| dropout_1 (Dropout) | (None, 4, 4, 64) | 0 |
| flatten (Flatten) | (None, 1024) | 0 |
| dense (Dense) | (None, 256) | 262400 |
| dropout_2 (Dropout) | (None, 256) | 0 |
| dense_1 (Dense) | (None, 62) | 15934 |

Total params: 343,326
Trainable params: 343,326
Non-trainable params: 0

**Figure 2:** The Layout of the CNN Model

The model contains a total of 4 convolutional layers, 2 pooling layers, 3 dropout layers, and 2 fully connected layers. The activation function for each convolutional layer in the CNN model is called the rectified linear unit.

### 3.4.2. LeNet-5

In 1988, Yann LeCun created the LeNet-5 model, a CNN structure that has a total of eight layers, including the input layer. The purpose behind the model is to classify characters in a digital document, and the model has become popular in both object detection and image classification. The input size of the model is 32 x 32 pixels which doesn't match the samples that are in the format of 28 by 28 pixels in both the EMNIST and MNIST handwritten datasets. The reason is that the potentially characteristic and distinctive features of the testing images can be in the center, where the highest-level feature detectors are situated. Figure 3 illustrates the LeNet-5 CNN architecture.

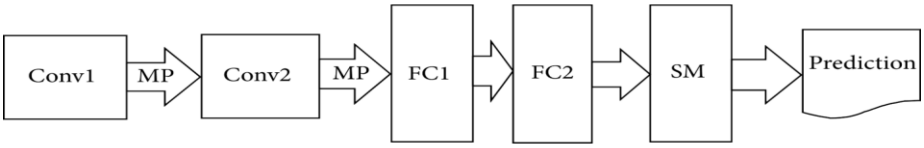



**Figure 3:** Architecture of LeNet-5 [14]

The activation function for each of the fully connected layers ("FC") is the hyperbolic tangent activation function (Tanh). "MP" is the max-pooling layer, "SM" is the output layer that uses the softmax activation function, and "Conv" stands for the convolutional layers. The model contains a total of 3 convolutional layers, 2 pooling layers, and 2 fully connected layers.

## 4. Results

Table 1 shows the validation accuracies of the CNN model used in this paper and the LeNet-5 transfer model on the EMNIST Balanced dataset and the EMNIST By_Class dataset.

**Table 1:** Validation Accuracy of Models

| Dataset | Convolutional Neural Network | LeNet-5 |
|---|---|---|
| EMNIST Balanced Dataset | 89.36% | 85.88% |
| EMNIST By_Class Dataset | 86.64% | 85.01% |

Using 15 epochs, the CNN model achieved 89.36% validation accuracy for the EMNIST Balanced dataset and a slightly lower 86.64% for the EMNIST By_Class dataset. In the case of the LeNet-5 CNN Model, using 10 epochs, the model achieved 85.88% for the EMNIST Balanced dataset and

85.01% for the EMNIST By_Class dataset. For both models, the validation accuracy trained with the EMNIST Balanced dataset is higher than the EMNIST By_Class dataset. In Figure 4, the image [15] got tested on the models.

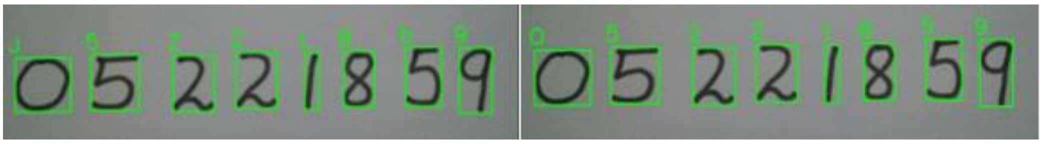


**Figure 4:** CNN Model using EMNIST Balanced Dataset for Digits (left) By_Class Dataset for Digits (right)

On the left of Figure 4, the first prediction for the first character by the CNN model is “J” and the actual answer is the number “0.” The model guessed the second digit correctly, “5.” However, inconsistency is present with the following two “2”s where the model predicts “Z” and “2,” respectively. The digit “1” is also confused with “L,” but the digit “8” is correctly identified. The final two digits, “5” and “9,” are identified as “b” and “9”. From the differences between the actual and predicted values, the CNN model trained under the EMNIST Balanced dataset has troubles where the characters are similar or nearly identical. In comparison, from the right of Figure 4, the model trained using the By_Class dataset from EMNIST correctly identifies all of the digits correctly: “0,” “5,” “2,” “2,” “1,” “8,” “5,” and “9,” respectively. The model under the larger dataset overcame the issue of similar-looking characters. An example is the predictions for the two “2”s in Figure 4.


**Figure 5:** CNN Model using EMNIST Balanced Dataset for Letters (up) and using EMNIST By_Class Dataset for Letters (down)

Moving on to letter classification, at the top of Figure 5, the CNN model using the EMNIST Balanced dataset accurately predicted the first four letters “H,” “E,” “L,” and “L,” but identified the last letter “O” incorrectly as “D.” The error was made on two characters that look highly similar. In comparison, at the bottom of Figure 5, the same model trained using the EMNIST By_Class dataset was able to identify all letters “H,” “E,” L,” “L,” and “O” correctly. Similar to the digit classification, the model overcame the problem of similar-looking characters.

## 5. Conclusions

This paper utilized a CNN model and the LeNet-5 model to classify handwritten characters. The data is preprocessed through min-max normalization, the gaussian blur, and one hot encoding and segmented into single characters through the canny edge detection method from the OpenCV library before they are fed into the models to get classified. The EMNIST By_Class and the EMNIST Balanced datasets trained the two models, and several test images got used to test the results. The EMNIST By_Class dataset contains a significantly higher number of letters and digits for both the training and testing samples. The paper found that despite the validation accuracy of the two models trained under the EMNIST Balanced dataset being higher than their respective models trained using the By_Class dataset from EMNIST, the latter achieved better results in real-world applications. In other words, the model trained with more data achieves more accurate results.

However, several limitations were present throughout the study. First and foremost, due to the lack of resources, the most optimized models may not have gotten trained to test the differences when the models get trained under different datasets. For the same reason, the optimal number of epochs may have able to be used to achieve the best performance. Furthermore, as character segmentation plays a significant and influential role in the handwritten recognition algorithm presented, a more accurate character segmentation algorithm can be used as the current method is highly sensitive to the orientation of the paper in terms of its placement to the camera. If a better character segmentation algorithm is used, higher accuracies may have been able to be obtained. Lastly, more transfer models can get tested if more resources are available, as the time to train the ResNet-50 CNN model took too long to process to be used in this paper.

## References

[1] Preetha, S., Afrid, I. M., &Nishchay, S. K. (2020). Machine Learning for Handwriting Recognition. *International Journal of Computer (IJC)*, *38*(1), 93-101.

[2] Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S., & Yoon, B. (2020). Improved handwritten digit recognition using convolutional neural networks (CNN). *Sensors*, *20*(12), 3344.

[3] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, *29*(6), 141-142.

[4] Saleem, A. S., Helan, R. R. H., Abirami, G., Vivekanandan, S. J., Asha, S., &Nithuja, B. M. (2022). Handwritten Recognition of Character and Number Using Convolutional Neural Network and Support Vector Machine. *International Journal of Modern Developments in Engineering and Science*, *1*(6), 50-53.

[5] Wang, P., Fan, E., & Wang, P. (2021). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters*, *141*, 61-67.

[6] Wang, W., Yang, J., Xiao, J., Li, S., & Zhou, D. (2014, November). Face recognition based on deep learning. In *International Conference on Human Centered Computing* (pp. 812-820). Springer, Cham.

[7] Islam, M. T., Siddique, B. N. K., Rahman, S., &Jabid, T. (2018, October). Image recognition with deep learning. In *2018 International conference on intelligent informatics and biomedical sciences (ICIIBMS)* (Vol. 3, pp. 106-110). IEEE.

[8] Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., Adam, M., Gertych, A., & San Tan, R. (2017). A deep convolutional neural network model to classify heartbeats. *Computers in biology and medicine*, *89*, 389-396.

[9] Cohen, G., Afshar, S., Tapson, J., & Van Schaik, A. (2017, May). EMNIST: Extending MNIST to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)* (pp. 2921-2926). IEEE.

[10] Gnat, S. (2021). Impact of Categorical Variables Encoding on Property Mass Valuation. *Procedia Computer Science*, *192*, 3542-3550.

[11] OpenCV. (2015, December 21). *Canny Edge Detection*. OpenCV. Retrieved August 16, 2022, from https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html.

[12] Rosebrock, A. (2021, December 6). *OCR: Handwriting recognition with opencv, keras, and tensorflow*. PyImageSearch. Retrieved from https://pyimagesearch.com/2020/08/24/ocr-handwriting-recognition-with-opencv-keras-and-tensorflow/.

[13] Hijazi, S., Kumar, R., & Rowen, C. (2015). Using convolutional neural networks for image recognition. *Cadence Design Systems Inc.: San Jose, CA, USA*, *9*.

[14] Zaibi, A., Ladgham, A., &Sakly, A. (2021). A lightweight model for traffic sign classification based on enhanced LeNet-5 network. *Journal of Sensors*, *2021*.

[15] Kukreja, Y. (2018). Medium. photograph, Medium. Retrieved from https://cdn-images-1.medium.com/max/1600/0*ru86ka5SAIxQ917T.