Convolutional Neural Networks(CNN's) for Handwritten Digit Recognition

David Oluwagbemi
Motion Picture Science
Rochester Institute of Technology
Rochester, New York 14623
Email: ddo7311@rit.edu

***Abstract*—A large incentive for computer vision advances is the development of technology and algorithms that can identify and select objects and points of subject within an image similar to how the human visual system processes stimuli. This application highlights a solution to text identification through the use of machine learning. This paper showcases CNN's.**

1. Introduction

Convolutional Neural networks are an example of machine learning's benefits. By training a model based on a series of images we can use the CNN's for the identification/classification of input images based on how well a given data set represents what we want the model to observe. Our current application is the recognition of handwritten digits, using the MNIST Database of handwritten digits to train our model sampled in Fig.(1).

Figure 1: MNIST Data Set

March 27, 2023

2. Method

2.1. Pre-processing

Before training the model, all images had their pixel values normalized to 1 and converted to floats in order to simplify calculations and improve efficiency. The images were read as (28x28) in size and converted to grayscale to limit the number of channels to 1.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
conv2d_1 (Conv2D)	(None, 24, 24, 64)	18496
max_pooling2d (MaxPooling2D)	(None, 12, 12, 64)	0
dropout (Dropout)	(None, 12, 12, 64)	0
flatten (Flatten)	(None, 9216)	0
dense (Dense)	(None, 256)	2359552
dropout_1 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 10)	2570

Figure 2: Proposed CNN Stucture

2.2. Background

This data set consist of 70,000 (28x28) images, of which 60,000 will be used for training and 10,000 will be used for testing. Splitting the data set prevents over-fitting of the model and provides a pool of images the model has not been seen in order to test the accuracy compared to a new input. CNN's are split into 4 main types of layers; Convolution, Activation, Pooling and Fully-Connected(Dense) Layers. This method also implements Dropout Layers in order to minimize the chance of over fitting.Fully-Connected layers result in placing inputs into classes. Classes are the possible outcomes for our models prediction, this implementation uses 10 classes, the numbers 0-9. The full structure of the proposed CNN is visualized in Fig.3.

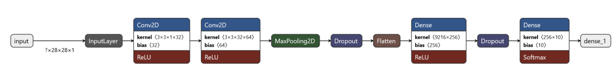

Figure 3: Proposed CNN Visualization

2.3. Convolution Layer

A convolution Layer can develop responses to patterns it recognizes within the image through the use of kernels. These kernels are convolved with the input to produce a

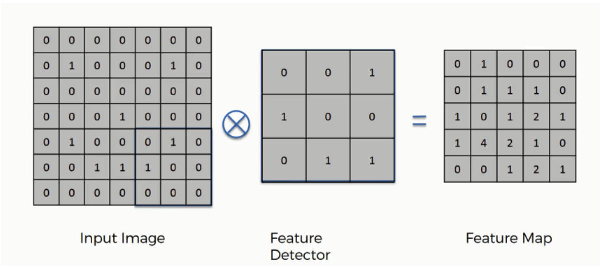

Figure 4: Convolution Layer

feature map that correlates to traits in the image. Kernels can be used to identify properties such as horizontal, vertical, diagonal edges etc. The feature map differs for every unique input and values are passed onto the next layer in the CNN for further processing Fig.4.

2.4. Activation Layer

$$x_i = \max(0, x_i) \tag{1}$$

An activation layers purpose is introducing non-linearity to the model. This layer modifies the data from the previous layer in a non-linear manner in order to add unpredictability to the training and allow the model to produce more complex correlations between the inputs and the classes. An example of this being applied to this use is ReLU(Rectified Linear Unit). ReLU simply clamps all negative values received from the previous layer to zero (1).

2.5. Pooling Layer

A Pooling Layer shrinks the size of the data received from the previous layer. It creates a kernel sized search region that uses various methods of simplifying data. This includes methods such as max pooling and average pooling, the former outputs only the maximum value in the search region and the latter outputs the average of all values in the search region. A stride is the number of pixels the kernel skips when collecting data at each new search region. Stride can add another factor to the sampling of a layers response, reducing the chance of overfitting.

2.6. Fully-Connected Layer

The fully-connected(Dense) Layer is the last layer of the model and typically where classification takes place. Each input image will have a certain weight in each class, numbers 0-9 in the case of this application, after going through the previous processing layers in the CNN. This weight can be seen as a quantifiable certainty in the decision of the models classification. The largest weight is selected as the output for prediction. The digits total 10 classes, 1 for each number, as seen in the final dense layer of our model in Fig.2.

2.7. Flatten and Dropout

Flatten layers rearrange the data from previous layers into a single row, These row values can be thought of as vectors in the direction of each class to be passed to the Dense layers. This allows weights in each class to be summed for classification. Dropout Layers get rid of a specified percentage of data to prevent overfitting.

3. Results

To confirm our model was trained properly, It was evaluated using the 10,000 test images that were split from the initial data set, resulting in an accuracy of 99.11%

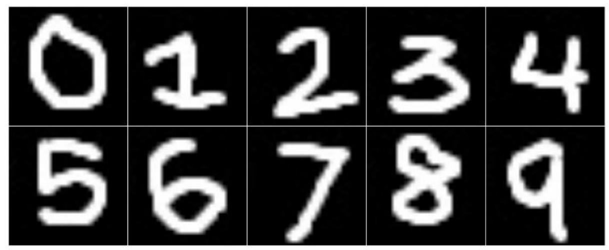

Figure 5: User Handwritten Digits

Using MS Paint, the paint brush tool was used to manually draw the images in Fig.5. These images were ran through the model for prediction, resulting in the following classes.; The filenames corresponding to each number share their index with the respective outputted classes Fig.6.

```
['1.jpg', '0.jpg', '6.jpg', '5.jpg', '3.jpg', '2.jpg', '4.jpg', '7.jpg', '8.jpg', '9.jpg']
[1       0        6        5        3        2        4        7        8        9]
```

Figure 6: Input Images vs Predicted Classes

The classes were predicted with 100% when testing the newly written images. The model was effective in real-use application.

4. Conclusion

The CNN performed with almost 100% accuracy compared to the test data and the newly written user data, justifying its effectiveness in digit identification. This provides an effective solution to the task of recognizing handwritten characters and can be extended to application in letters amongst many other things outside of handwriting.

References

[1] Mayank Jain, Gagandeep Kaur, Muhammad Parvez Quamar and Harshit Gupta, *Handwritten Digit Recognition Using CNN*, 2021 International Conference on Innovative Practices in Technology and Management (ICIPTM) Publication date: February 2021.