Performing Arithmetic Operations using Handwritten Images

Osman Onur KUZUCU
CE Hacettepe University, n21131025@cs.hacettepe.edu.tr

Prof. Dr. Mehmet Önder EFE
CE Hacettepe University, onderefe@hacettepe.edu.tr

Abstract **- In this paper, we worked on performing arithmetic operations based on handwritten images. We used a mnist dataset[1] and math operands dataset[2]. We combined 2 mnist handwritten data and 1 math operands[2]. We used an algorithm which is similar to Lenet-5[3]. We received high accuracy in this project.**

Introduction

In this project, our goal is to perform arithmetic using images. Input images are handwritten images. Project image source for numerical digits is mnist dataset[1]. Project input images combined with one handwritten math operands' images. Network trained similarly to Lenet-5[3]. After the training our input images are separated for network accuracy. Our network achieved ~%98 accuracy.

Generate Data

Our project goal is recognizing the digits and operands from handwritten images. In this perspective, we have to generate handwritten data. Therefore, mnist dataset[1] and math operands dataset[2] are used. Example pictures of the mnist dataset are Figure 1, Figure 2. Example pictures of the math operand is Figure 3. In our project, (+,-,/,*) are used for the train and test network. Size of the mnist dataset is 28*28. Size of the math operands dataset is generally 155*135, but sometimes images have different sizes. Lenet-5 algorithm[3] is performing on 32*32 handwritten images. Therefore, our input images' size must be 32*32 .

In our dataset, two 32*32 images for digits, and one 32*32 image math operand. The sum of the input image must be 32*96(Figure 4). In our project, 100,000 images are generated to train the network. Because of, two digits permutations and one math operand permutation. In the project, 10,000 images are generated to test the network.

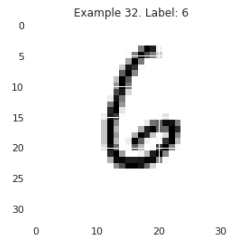

Figure 1

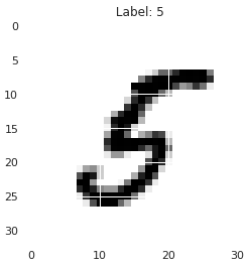

Figure 2

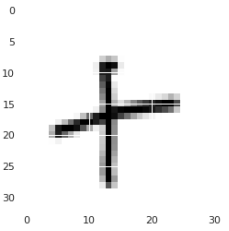

Figure 3

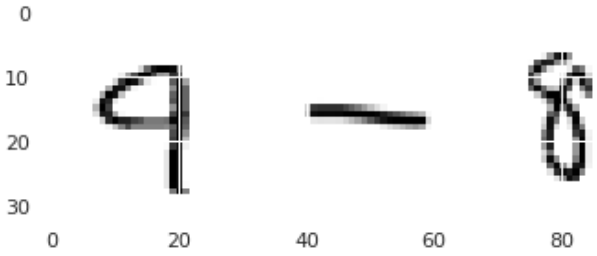

Figure 4

Image Resizing

In the project, 2 image datasets are used. One of the dataset is the mnist dataset[1]. In the mnist dataset[1], images' size is 28*28. In Lenet-5 algorithm, input image size is 32*32. Therefore, image must be resized from 28*28 to 32*32.

The second dataset which is used in the project, is math operand dataset. In this dataset, almost images size are 155*135. However, some images' size are different from 155*135. The math operand dataset images must be resized to 32*32. After the resizing operation, dataset images are white on black(Figure 5). Images must be converted to black on white.(Figure 3) The algorithm is given as Algorithm 1.

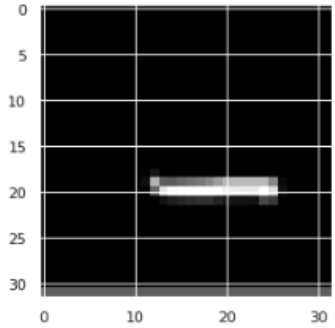

Figure 5

```
def blackToWhite(array):
    i=0
    j=0
    array2=np.zeros([32, 32],dtype=np.uint8)
    while(i<32):
        while(j<32):
          array2[i][j]=abs(255-array[i][j])
          j+=1
        i+=1;
        j=0;
    return array2
```

Algorithm 1

combining images

In the project, input images are 32*96. Generating the 3 images which are arrayed digit, operand and digit.(Figure 4) In this project, images are worked on with Numpy(Array lib in Python). Each array size is 32*32. Input image size is 32*96. Therefore, "hstack" function is used. "hstack" is a function that combines arrays horizontally.

Randomize Data

In this project, mnist dataset[1] and math operand dataset[2] are used. Networks should be trained on random inputs. Because, the accuracy of the network is stable. On the other hand, standard deviation is minimum. Mnist data information about each digit is in Table 1. Math operand data information about each operand is in Table 2. In train data, permutation is used. First digit is selected from {0,1,2,3,4,5,6,7,8,9}. Math operand is selected from {+,/,*,-}. Second digit is selected from {0,1,2,3,4,5,6,7,8,9}. In the project, 100,000 images are generated(Figure 4). Network is trained with input images which are separated by 3. Therefore the network is trained 300,000 images.

Digit	Number of data
0	5923(Train), 980(Test)
1	6742(Train), 1135(Test)
2	5958(Train), 1032(Test)
3	6131(Train), 1010(Test)
4	5842(Train), 982(Test)
5	5421(Train), 982(Test)
6	5918(Train), 958(Test)
7	6265(Train), 1028(Test)
8	5851(Train), 974(Test)
9	5949(Train), 1009(Test)

Table 1

Operand	Number of data
/	544(Train), 78(Test)
+	545(Train), 78(Test)
*	556(Train), 80(Test)
-	549(Train), 80(Test)

Table 2

Images and Labels

In this part, 10 number digits and 4 math operands are used. In the Lenet-5 algorithm, the network's model is "Sequential". Therefore, math operands are not labeled by string. Math operands are labeled with integer value. Images and labels are below(Table 3).

Data	Label
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
/	11
+	14
*	17
-	20

TABLE 3

NETWORK'S MODEL AND LAYERS

In the Lenet-5 algorithm, the network's model is sequential. Network parameters are in Algorithm 2. "Adam" optimizer is used int the network. "Sparse Categorical Cross Entropy " loss function is used in the network. "Accuracy" metric is used in the networks.

```
model = keras.Sequential()
model.add(layers.Conv2D(filters=6,
kernel_size=(3,     3),     activation='relu',
input_shape=(32,32,1)))
model.add(layers.AveragePooling2D())
          model.add(layers.Conv2D(filters=16,
kernel_size=(3, 3), activation='relu'))
model.add(layers.AveragePooling2D())
model.add(layers.Flatten())
model.add(layers.Dense(units=256,
activation='relu'))
model.add(layers.Dense(units=128,
activation='relu'))
model.add(layers.Dense(units=25, activation =
'softmax'))
```

ALGORITHM 2

TRAIN AND TEST THE NETWORK

In the network, the images size is 32*32, 300,000 images(Figure 1) and 300,000 labels are used in network training. 30,000 images(Figure 1) and labels are used to test the network. In the training section, the network's epoch size is 10. In addition, the network's batch size is 128.

NETWORK ACCURACY

After the training section, the accuracy of the network is %99.2 . This accuracy is testing the 30,000 images which sizes are 32*32. However the test images which are 30,000 are not suitable for the project. Therefore three images are combined. After combining the testing images' size is upgraded to32*96, the test image's number is decreased to 10,000. After the combining images, the example image is Figure 5. Testing the network with 10,000 images, the network accuracy is decreased to %97.98. Some test examples are shown in Figure 6, Figure 7, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, Figure 14, Figure 15.

FURTHER RESEARCHES

In the further research, researchers can add other math operands, can increase the digit numbers, can work on decimal numbers.

CONCLUSION

In this project, mnist data and math operand data are used. Images are resized. After the image resizing, images are changed to network stability. Images are generated to train and test the network. After training the network, test images are changed to suitable form. Network is tested with suitable images. The accuracy of the network is %97.98. The accuracy of the network is very high. Therefore, the network is very efficient.

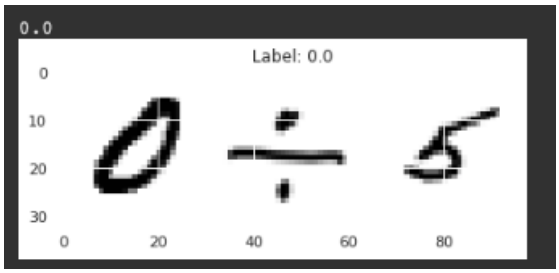

FIGURE 6

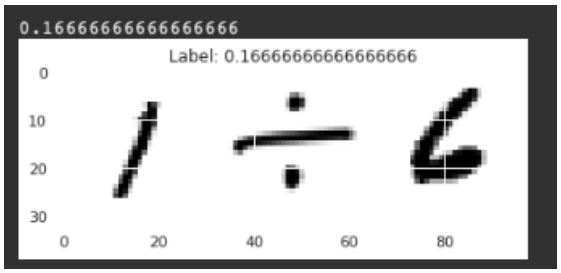

FIGURE 7

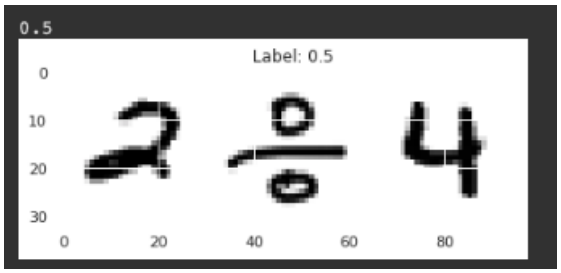

FIGURE 8

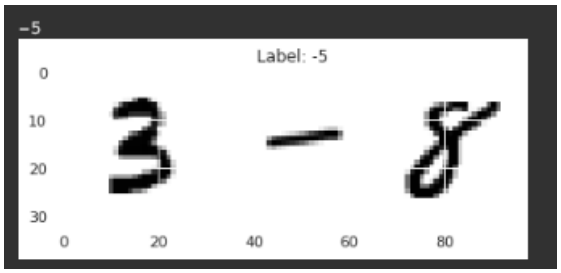

FIGURE 9

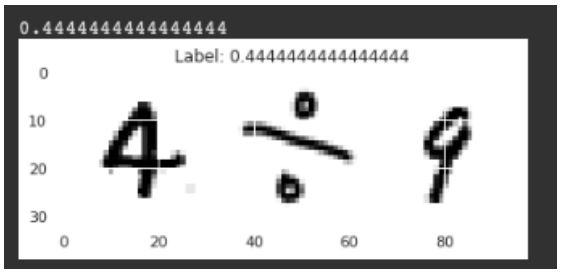

FIGURE 10

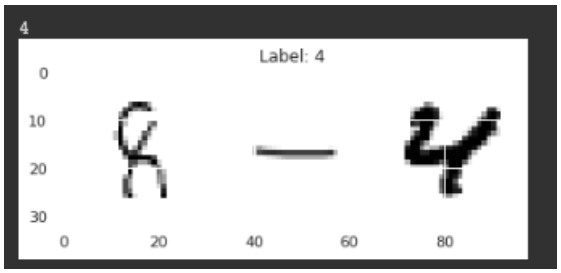

FIGURE 11

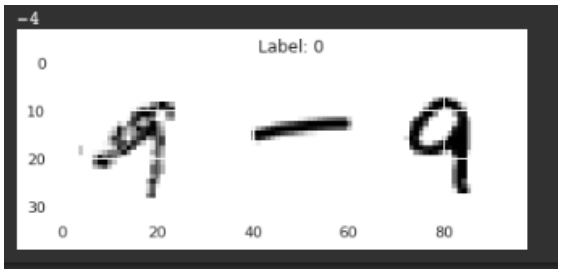

FIGURE 12

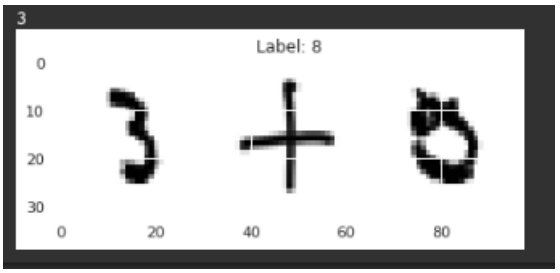

FIGURE 13

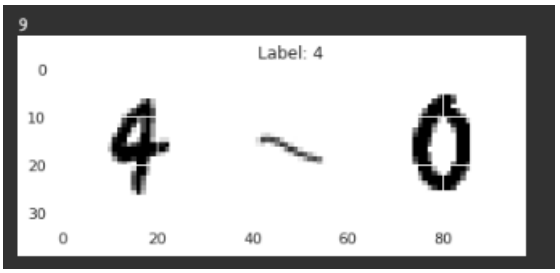

FIGURE 14

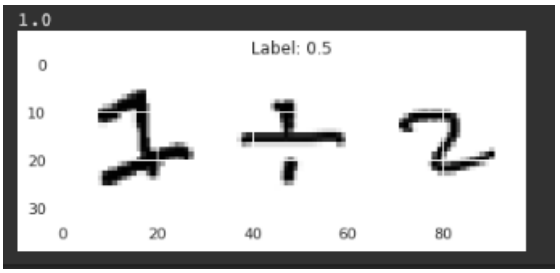

FIGURE 15

REFERENCES

[1] http://yann.lecun.com/exdb/mnist/

[2] https://www.kaggle.com/clarencezhao/handwritten-math-symbol-dataset

[3] http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf

[4] https://www.analyticssteps.com/blogs/lenet-5-tutorial-architecture-features-and-importance

[5] https://towardsdatascience.com/understanding-and-implementing-lenet-5-cnn-architecture-deep-learning-a2d531ebc342

[6] https://medium.com/@mgazar/lenet-5-in-9-lines-of-code-using-keras-ac99294c8086
[7] https://www.pyimagesearch.com/2016/08/01/lenet-convolutional-neural-network-in-python/
[8] https://thecleverprogrammer.com/2021/10/09/lenet-5-architecture-using-python/
[9] https://arxiv.org/pdf/1912.03035.pdf