EEL 5840 Fall 2022 Fundamentals of Machine Learning-Final Project

Group-if-and-only-if

Abstract*— *HCR is an important field of research that machine learning targets at. In the report, we solved a Handwritten Mathematical Symbols classification problem by the classification of CNN with the fastai library. We also conduct the other classifiers to compare with CNN which include Naive Bayesian model, FLDA model, Perceptron, LDA model, and SVM model. The results show that CNN hit the highest classify accuracy rate than the other classifiers.

Index Terms*—*HCR, CNN.

I. INTRODUCTION

Optical Character Recognition (OCR) is the ability to recognize images of typed, handwritten, or printed text into the corresponding machine-encoded text. Handwritten character recognition (HCR) is an important field of research in artificial intelligence (AI), computer vision (CV), and pattern recognition (PR) [1]. HDR itself is also one of the most challenging tasks in the domain of OCR. The final project is a type of HCR that is Handwritten Mathematical Symbols Classification.

The machine learning-based approaches mostly focus on feature extraction, dimension reduction, and classification. The feature descriptors can be divided broadly into geometric, statistical, pixel-based, and miscellaneous categories [2]. Feature dimension reduction techniques that reduce the relatively redundant features contain principal component analysis [3], supervised locally linear embedding [4], bidirectional associative memory [5], and genetic algorithm [6]. The most prevalent classification techniques in OCR research can be divided into five broad categories [2,7] : (a) kernel methods including support vector machines [3], kernel Fisher discriminant analysis [3]; (b) statistical methods including Logistic regression [3], linear discriminant analysis [3], decision trees [3], k nearest neighbor [3], hidden Markov model [3], hierarchical Bayesian network [3]; (c) template matching including prototype matching [8], Hough transform [9]; (d) artificial neural networks including multi-layer perceptron [3], Bayesian neural networks [3], recurrent neural network [10], convolutional neural networks [10] and (e) structural pattern such as graphical and grammar-based approaches [3].

ANN such as RNN and CNN are deep learning-based approaches and can create a more abstract representation of the data as the network grows deeper, by which the model can learn significant features that the ML-based methods often fail to grasp [2]. Nowadays, deep neural networks especially convolutional neural networks (CNNs) have been widely applied to image data [3] and achieved classifications that exceed state-of-the-art results specifically for visual stimuli/input [7]. By slight modification of the usual BP algorithm, CNN ensures that the shared-weight constraints are satisfied. Due to the use of local receptive fields and the substantial number of constraints on the weights, the number of independent parameters to be learned from the data is much smaller than if the network was fully connected [3].

The main objective of the final project is to encode ten handwritten math symbols into a computer-interpretable format for creating a digital footprint of the information and then recognizing them in accuracy. Considering CNN's promising characteristics that improve classification accuracy while reducing computational complexity, we employ it to realize the objective of the project.

II. IMPLEMENTATION

A. Data collection and digitization

Math symbols that need to be recognized in the project are encoded in Table 1. The whole class of students contributed to producing the dataset. Then the hard copies got converted into soft-copied images [11]. The digitization, desaturate, and pixel homogenization of the soft copies were processed with the course-provided codes. Finally, each group got the 70% of the overall data (9032 images, 300*300 pixels) for training.

Math Symbol	Label	Integer Encoding
x	x	0
$\sqrt{}$	square root	1
$+$	plus sign	2
-	negative sign	3
$=$	equal	4
$\%$	percent	5
∂	partial	6
$\prod$	product	7
π	pi	8
$\sum$	summation	9

Table 1: Integer encoding for each traffic sign class.

B. Dada preprocessing

1) Mislabeled Data: Our group pretreated the training dataset to correct the mislabeled images. For images that are blank, not recognizable, and not belonging to any symbol category, we relabeled their integer encoding into "-1". Finally, 73 images were reclassified into this category.
2) Data Augment: Before we further transfer the dataset, we applied the process of data augment to rotate, squeeze, and nominate illustrations. But we do not increase the image number.
3) Max-min Normalization: We normalized the data to reduce the effect of illumination differences in images. Considering the maximum and minimum intensity values of pixels in all images, the intensity values are scaled between 0 and 1 in the experiment.
4) Pixel Redundant: To save running time, we reduce the pixel of each image from 300*300 to 100*100 for testing convenience (for other classifiers except CNN).
5) Train and Validation: We split the dataset into 80% and 20% for training and validation.

III. EXPERIMENTS

A. CNN architecture

It is one of the popular approaches that depend on scanning the textual image with a predefined sliding window, from which the required features are extracted and modeled [1]. A convolution operation implies the sliding window (kernel or filter) that shares the same variables (bias and weight vector) was reproduced to form a feature map throughout the entire field of vision. We created a convolutional neural network (CNN) of 43 blocks using the Fastai library. The CNN consists of input, convolution, subsampling, and full-connection layers. The input layer takes the raw pixels of the images, the convolution layer, extracts features from the raw pixels of the image, the max-pooling layer is for the purpose of dimension reduction, and the full connection layer, is for classification.

With the training set, the network was trained by error minimization using back propagation (BP) to evaluate the gradient of the error function. The activation functions applied were ReLU.

B. Other classifiers

In addition to CNN, we also tested out other classifiers by k-fold cross-validation and used confusion matrix and accuracy to monitor their performance. Our tested classifiers include Naive Bayes, FLDA, LDA Feature Extraction, Perceptron, and SVC.

IV. RESULTS

A. Performance of CNN

(1) Training Process

epoch	train_loss	valid_loss	error_rate	time
0	1.341756	0.518686	0.157454	44:27

epoch	train_loss	valid_loss	error_rate	time
0	0.391679	0.235904	0.073702	1:05:36
1	0.232597	0.168626	0.052485	1:05:34
2	0.113626	0.180014	0.044109	1:05:23
3	0.042755	0.149530	0.035176	1:05:21
4	0.017066	0.145328	0.030151	1:05:26

Table 2. Supervision of CNN training process

Confusion matrix

Actual \ Predicted	equal	negative sign	partial	percent	pi	plus sign	product	square root	summation	x
equal	184	0	0	0	0	0	0	0	1	1
negative sign	2	197	0	0	0	1	0	0	0	0
partial	0	0	190	0	0	1	0	0	0	2
percent	0	0	0	157	0	0	0	0	0	0
pi	0	0	2	0	162	0	14	1	0	0
plus sign	0	0	0	0	0	186	0	0	0	2
product	0	0	2	0	11	0	164	2	0	0
square root	0	1	0	0	1	0	1	169	1	1
summation	0	0	1	0	2	0	0	0	169	0
x	0	0	0	0	0	3	1	0	0	159

Table 3. Confusion Matrix for the validation process

(2) Test Process

We use 1/10 of the training data as the test dataset for testing. The overall results of our testing accuracy are 98%. Table 4 is the confusion matrix for the testing process. “0” represent the images not belong to any categories, and 1-10 represents the 0-9 math symbols we are required to recognize.

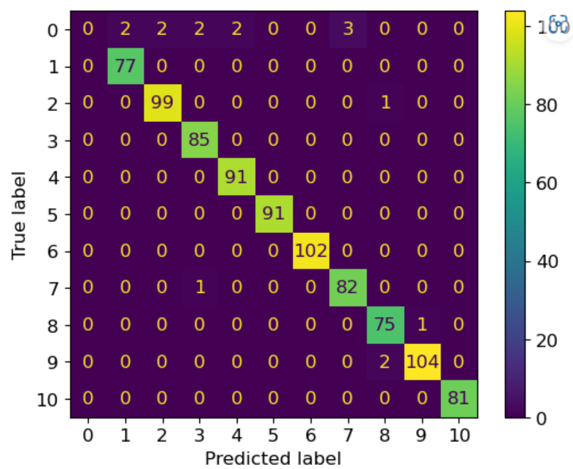

Table 4. Confusion Matrix for the testing process (0 means the label “-1”, 10 means the label “ 9”.)

B. Performance of other classifiers

Besides CNN, we still tested the Naive Bayesian model, FLDA model, Perceptron, LDA model, and SVM model. The performances were described below.

In the Gaussian Naive Bayesian (NB) model, Gaussian NB used the least time for classification and cross-validation but did poorly in both training and testing set and cross-validation, as Naive Bayes assumes that all predictors (or features) are independent, rarely happening in real life. That’s also why the perceptron model could do better. Perceptron took longer time than GNB, But in test and cross-validation, it has a much higher accuracy of ~0.4. SVC takes the longest time among these. Overall, it was a time-efficient but poor performance. FLDA and LDA both did very well in the training set (0.999), but in the testing set, they both gave poor accuracy ~ 0.13 because a linear discriminative classifier cannot find a new linearly separable axis In SVC, we tried three different pairs of the parameter of gamma and c, but it didn't show a significant effect on the test. The accuracy in training was always around 0.999, and 0.10 in testing.

V. CONCLUSIONS

By comparing the performance of CNN and other classifiers, we prove that CNN is the best selection. Apparently, the complexity of a training model determines the performance of a classifier. The advantage of CNN is that it automatically detects important features without human supervision. By using Fastai library which is already trained by millions of images, it helps us to rotate, enlarge, and squeeze the data which significantly improves the recognition accuracy.

REFERENCES

[1] H. M. Balaha, H. A. Ali, and M. Badawy, “Automatic recognition of handwritten Arabic characters: a comprehensive review,” Neural Computing and Applications, vol. 33, pp. 3011–3034, 2021.

[2] A. B. M. A. Rahman, M. B. Hasan, S. Ahmed, T. Ahmed, M. H. Ashmafee, M. R. Kabir, and M. H. Kabir. “Two decades of Bengali handwritten digit recognition: a survey,” IEEE Access. vol. 10, p. 92597, 2022. DOI: 10.1109/ACCESS.2022.3202893.

[3] C. M. Bishop. Pattern Recognition and Machine Learning. Series Editors: M. Jordan. J. Kleinberg, B. Scho¨lkopf. Springer Science + Business Media, LLC. 2006, pp. 561, 326, 191, 217, 181, 654, 124, 607, 327, 246, 277, 359.

[4] Z. J. Li, W. R. Shi, X. Shi, and Z. Zhong, “A supervised manifold learning method,” Comput. Sci. Inf. Syst., vol. 6, no. 2, pp. 205-215, 2009.

[5] F. J. Lopez-Aligue, I. Acevedo-Sotoca, C. G. Orellana, and H. G. Velasco, “Associative memories for handwritten pattern recognition,” in Proc. International Joint Conf. on Neural Networks, 2003, vols. 1-4, pp. 1441-1446, Portland, USA.

[6] V. S. Waghmare, L. R. Ragha, “Application of genetic algorithm for handwritten Kannada characters,” in Proc. 3rd IEEE Int. Advance Computing Conf. (IACC), 2013, pp. 989-994, Ghaziabad, India.

[7] J. Memon, M. Sami, R. A. Khan, and M. Uddin, “Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR),” IEEE Access, vol. 8, p. 142642, 2020, DOI: 10.1109/ACCESS.2020.3012542.

[8] H. Hosseinzadeh, F. Razzazi, “Domain adaptive multiple kernel learning for handwritten digit recognition,” in Proc. 24th Iranian Conf. on Electrical Engineering (ICEE), 2016, pp. pp.1887-1891, Shiraz, Iran.

[9] M. Salam, A. A. Hassan, “Offline isolated Arabic handwriting character recognition system based on SVM,” Int. Arab. J. Inf. Technol., vol. 16, no. 3, pp. 467–472, 2019.

[10] E. Alpaydın, Introduction to Machine Learning, 3rd Ed., The MIT Press Cambridge, Massachusetts, London, England, 2014. pp. 305, 282.

[11] M. Kumar S. R. Jindal, “A study on recognition of pre-segmented handwritten multi-lingual characters,” Arch. Comput. Method E., vol. 27, pp. 577–589, 2020.