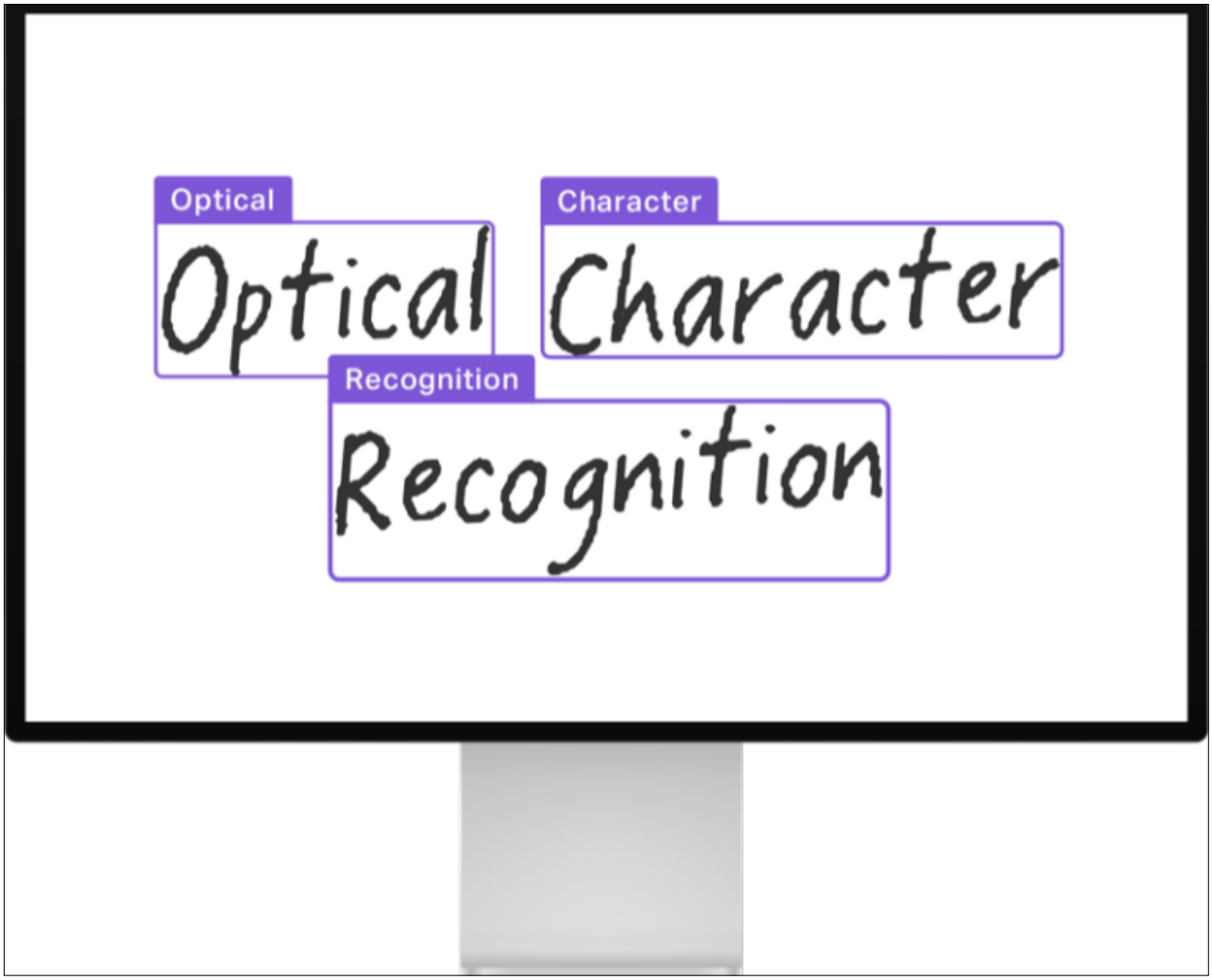


Optical Character Recognition

Prepared for: Eduardo Blanco

Class: CSE 598, Intro to Deep Learning [Fall'21]

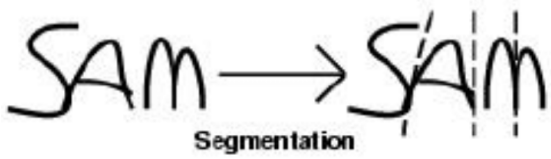
Prepared by: Prakhar Bhartiya

Problem:

The world is rapidly becoming more digital. However, as humans, we still have a large number of papers that are handwritten. Almost all of a library's content may now be stored on a single hard disk. However, transferring handwritten/non-digital material to digital format is difficult. We employ the "Optical Character Recognition(OCR)" approach to solve these issues.

Many computer vision applications need character recognition, such as automatic number plate identification, robot navigation, product search, and image-based translation. The character recognition problem is separated into two categories in the literature: optical character recognition (OCR) and character recognition in natural imagery. EMNIST is the most well-known and biggest dataset for research and benchmarking in optical character recognition.

I'm working on designing an optimum classifier for the EMNIST Letter dataset + EMNIST MNIST dataset as part of this project. It may be utilised in a variety of applications; for example, one of the classifier's uses in OCR implementation is shown below.



Input : Isolated English letter from handwritten document.
Output : Classified class label.

Eg. Input : 
Output: S

Dataset :

EMNIST Dataset

The EMNIST dataset is a set of handwritten character digits derived from the NIST Special Database 19 and converted to a 28x28 pixel image format and dataset structure that directly matches the MNIST Dataset.

Dataset Summary

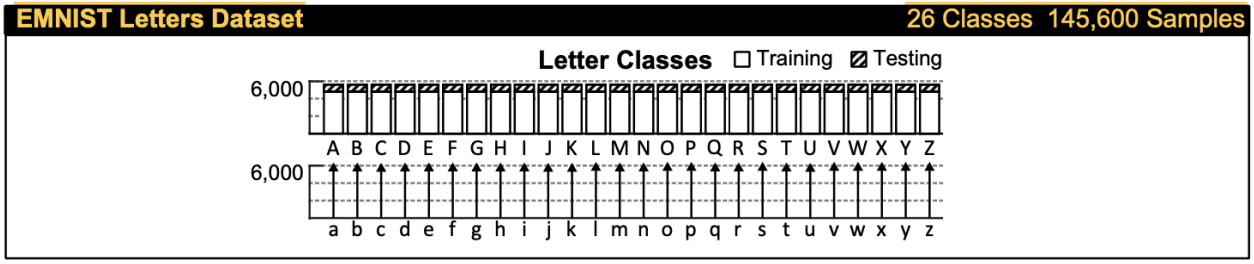
There are six different splits provided in this dataset. A short summary of the dataset is provided below:

- EMNIST ByClass: 814,255 characters. 62 unbalanced classes.
- EMNIST ByMerge: 814,255 characters. 47 unbalanced classes.
- EMNIST Balanced: 131,600 characters. 47 balanced classes.
- EMNIST Letters: 145,600 characters. 26 balanced classes.
- EMNIST Digits: 280,000 characters. 10 balanced classes.
- EMNIST MNIST: 70,000 characters. 10 balanced classes.

I am planning to use the only part of total dataset, EMNIST Letters dataset and EMNIST Digits dataset for training my CNN for classification. And merge them to form a custom larger dataset with 36 classes.

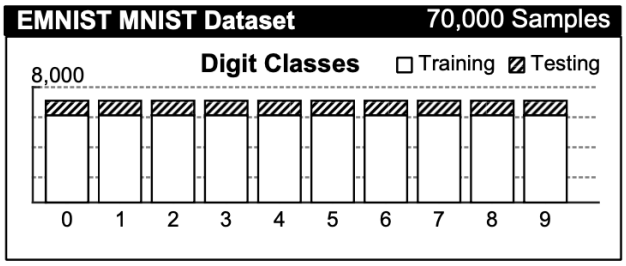
Original EMNIST Letters

26 classes = 26 alphabets in English language. (a/A - z/Z)
It contains both uppercase and lowercase letter in each class folder.
Each class has 5600 images, hence 5600 x 26 = 145600 images.
Each image is 28x28 pixel, in png format.
Dataset consists of handwritten letters from various writers.



Original EMNIST MNIST

10 classes = 10 numbers in English language. (0, 1, 2, ……9)
Each class has 7000 images, hence 7000 x 10 = 70,000 images.
Each image is 28x28 pixel, in png format.
Dataset consists of handwritten digits from various writers.



Abstracted EMNIST MNIST + EMNIST Letters dataset
As we can notice EMNIST Letters has 5600 images per class and EMNIST MNIST contain 7000 images per class. To balance the number of images per class. We will **randomly** select images and downscale the images per class from MNIST. Finally dataset summary looks like :

36 classes = 26 alphabet classes + 10 digits classes
Each class has 5600 images, hence 5600 x 36 = 201,600 images (Total dataset).
Each image is 28x28 pixel, in png format.
Dataset consists of handwritten digits from various writers.

Training dataset : Validation dataset : Testing dataset = 60 : 20 : 20

Resource :

Citation : Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. (2017). EMNIST: an extension of MNIST to handwritten letters. Retrieved from http://arxiv.org/abs/1702.05373
NIST dataset guide : https://s3.amazonaws.com/nist-srd/SD19/sd19_users_guide_edition_2.pdf
NIST dataset : https://www.nist.gov/services-resources/software/public-domain-ocr

Baseline :

Below table is accuracy mentioned on original EMNIST dataset by Linear classifier and OPIUM Classifier.

Resource : Cohen, Gregory & Afshar, Saeed & Tapson, Jonathan & van Schaik, André. (2017). EMNIST: an extension of MNIST to handwritten letters.

TABLE III
SUMMARY OF THE RESULTS FOR THE LINEAR AND OPIUM-BASED CLASSIFIERS ON THE EMNIST DATASET.

	Linear Classifier	OPIUM Classifier
Balanced	50.93%	$78.02\% \pm 0.92\%$
By Merge	50.51%	$72.57\% \pm 1.18\%$
By Class	51.80%	$69.71\% \pm 1.47\%$
Letters	55.78%	$85.15\% \pm 0.12\%$
EMNIST MNIST	85.11%	$96.22\% \pm 0.14\%$
Digits	84.70%	$95.90\% \pm 0.40\%$

Technique	By_Class	By_Merge	Balanced	Letters	Digits
DWT-DCT + SVM [71]	–	–	–	89.51%	97.74%
Linear classifier [67]	51.80%	50.51%	50.93%	55.78%	84.70%
OPIUM [67]	69.71%	72.57%	78.02%	85.15%	95.90%
SVMs (one against all + sigmoid) [86]	–	–	–	–	98.75% *
Multi-layer perceptron [88]	–	–	–	–	98.39% *
Hidden Markov model [91]	–	–	–	90.00% *	98.00% *
Record-to-record travel [89]	–	–	–	93.78% *	96.53% *
PSO + fuzzy ARTMAP NNs [87]	–	–	–	–	96.49% *
Multi-layer perceptron [90]	–	–	–	87.79% *	–
CNN (6 conv + 2 dense) [85]	–	–	**90.59%**	–	**99.79%**
Markov random field CNN [73]	**87.77%**	**90.94%**	90.29%	**95.44%**	99.75%
TextCaps [76]			90.46%	95.36%	**99.79%**
CNN (2 conv + 1 dense) [75]	87.10%	–	–	–	–
Committees of neuroevolved CNNs [64]	–	–	–	95.35%	99.77%
Deep convolutional ELM [78]	–	–	–	–	99.775%
Parallelized CNN [74]	–	–	–	–	99.62%
CNN (flat; 2 conv + 1 dense) [79]	–	–	87.18%	93.63%	99.46%
EDEN [80]	–	–	–	88.30%	99.30%
Committee of 7 CNNs [19]	88.12% *	–	–	92.42% *	99.19% *

Above table is Side-by-side comparison of the results for the EMNIST dataset, including works using similar datasets from NIST Special Database 19. Best results are boldfaced. Results marked with a star (*) indicate that they refer to works using samples from NIST SD 19 which are similar but not equivalent to EMNIST.

Resource : Baldominos, A.; Saez, Y.; Isasi, P. A Survey of Handwritten Character Recognition with MNIST and EMNIST. *Appl. Sci.* **2019**, *9*, 3169. https://doi.org/10.3390/app9153169

Expected Outcome :

Initially I am planning to implement using a simple architecture using Keras library.

Cov2D Layer -> Max Pool Layer -> Flatten -> Dense Layer -> Dense Layer -> Softmax

Depending on accuracy I might plan to increase the layers complexity.

Alternative option to built it from pre existing layer architecture. Like AlexNet, GoogleNet, VGGNet. Below table was mention in "Assessment of Extended MNIST (EMNIST) dataset using Capsule Networks,". But methods are implemented on total dataset.

Resource : R. Anuradha, N. Saranya, M. Priyadharsini and G. D. Kumar, "Assessment of Extended MNIST (EMNIST) dataset using Capsule Networks," 2019 International Conference on Intelligent Sustainable Systems (ICISS), 2019, pp. 263-266, doi: 10.1109/ISS1.2019.8908006.

Models	Accuracy %		
	50%(EMNIST)	75%(EMNIST)	100%(EMNIST)
AlexNet	83.3	91.7	96.9
VGGNet	81.5	94.1	98.3
GoogleNet	88.2	97.7	98.9
CapsNet	95.7	98.9	99.7

Currently I am expecting to achieve **90+ % accuracy in test data** after studying various implementation in research paper. Planning to opt for one the method depending on the accuracy from both the implementation.