Cyrillic-MNIST: a Cyrillic Version of the MNIST Dataset

Bolat Tleubayev, Zhanel Zhexenova, Kenessary Koishybay, Anara Sandygulova
Department of Robotics and Mechatronics, School of Engineering and Digital Sciences
Nazarbayev University, Nur-Sultan, Kazakhstan
{bolat.tleubayev, zhanel.zhexenova, kenessary.koishybay, anara.sandygulova}@nu.edu.kz

Abstract

This paper presents a new handwritten dataset, Cyrillic-MNIST, a Cyrillic version of the MNIST dataset, comprising of 121,234 samples of 42 Cyrillic letters. The performance of Cyrillic-MNIST is evaluated using standard deep learning approaches and is compared to the Extended MNIST (EMNIST) dataset. The dataset is available at https://github.com/bolattleubayev/cmnist

Keywords: dataset, handwriting recognition, optical character recognition, Cyrillic, letters

1. Introduction

Optical Character Recognition (OCR) is the use of computer vision solutions to distinguish printed or handwritten characters in digital images of either scanned documents or photos. Once a text is recognized and converted into American Standard Code for Information Interchange (ASCII), it can be edited or searched as if it was created in a word processor. OCR is a language-specific problem, thus the state of the art in one language does not imply that there exists any working usable solution in another. Many languages have their datasets for the use by the researchers in computer vision to advance the field. For example, the Modified National Institute of Standards and Technology (MNIST) dataset (LeCun et al., 1998) has become a standard benchmark for learning, classification and computer vision systems. An extension of the MNIST, the Extended MNIST (EMNIST) dataset contains digits, uppercase and lowercase handwritten letters of English (Cohen et al., 2017). Other examples of handwritten datasets are CASIA (Liu et al., 2011) for Chinese, CMATERdb (Das et al., 2014) for Bangla and Arabic, and HODA (Khosravi and Kabir, 2007) for Farsi.

Unfortunately, there is no such data for the handwritten Cyrillic script. According to Iliev (2013), there exist 112 languages that use or have used the Cyrillic alphabet. Currently 0.3 billion people, which is 4% of the world's population use the Cyrillic alphabet on a daily basis (WorldStandards, 2020). It is used exclusively for several languages including Russian, Belorussian, Ukrainian, Bulgarian, Kazakh, Kyrgyz, Macedonian, Serbian, Tajik, Turkmen, Uzbek and others. Originally Cyrillic alphabet comprised of 43 letters in addition to modified and combined Greek and Hebrew letters (in the case of the Cyrillic letters for ts, sh, and ch) (Britannica, 2019). The modern Cyrillic alphabets have fewer letters than the original Cyrillic: Russian language has 33 letters, Bulgarian - 30, Serbian - 30, Ukranian - 33. Modern Russian Cyrillic has also been adapted to many non-Slavic languages with an addition of special letters.

OCR has numerous use cases ranging from postcard address recognition (Palumbo et al., 1992) to automatic license plate recognition in modern traffic control systems (Du et al., 2012; Draghici, 1997). OCR has allowed to efficiently exploit the content of low quality historical documents by digitizing them across the globe (Gatos et al., 2004; Kusetogullari et al., 2019). Gordienko et al. (2018) used OCR for the recognition of the historical writings carved in stone on the walls of St. Sophia cathedral in Kyiv (Ukraine).

The work detailed in this paper aims to create the first dataset for the Cyrillic-based languages. The closest dataset is the Cyrillic Oriented MNIST (CoMNIST) (Vial, 2019), which was the first and only attempt to create a dataset for machine learning. However, letters were drawn digitally by using either a computer mouse or a finger on a touch screen by volunteers who were crowd-sourced through social networks. There are around 28,000 278x278 samples of these drawings representing 59 letters (33 Russian of around 15000 samples in total and 26 English letters). Since CoMNIST was collected by drawing letters on a screen rather than handwriting them on a paper, it is inappropriate to be included in our dataset. Being native to Cyrillic script ourselves, we stress the importance of the dataset being handwritten using cursive script in order to address real-world OCR applications. In contrast to CoMNIST, our dataset contains samples handwritten in cursive by the native Cyrillic users that signed consent forms for their data to be publicly available. In addition, Cyrillic-MNIST has a version that shares the same image structure and parameters as the original MNIST dataset allowing for direct compatibility with all existing classifiers and systems. The creation of the first handwritten Cyrillic dataset will change the poor situation with OCR in many Cyrillic-based languages.

MNIST (LeCun et al., 1998) and EMNIST (Cohen et al., 2017) datasets became standard benchmarks for tests of various machine learning algorithms. The MNIST dataset, presented by LeCun et al. (1998), is comprised of the digits first presented in the NIST Special Database 19 (Grother, 1995). The dataset contains 60,000 training and 10,000 test images. The size of images is 28x28 pixels. In total, there are 10 classes for every number. Images are well formatted and orga-

nized, which makes the exploration of various classifiers easier (Deng, 2012).

The Extension of MNIST dataset presents a more complex objective of letters and digits classification. In contrast to MNIST, it contains lowercase and uppercase letters, as well as digits. The division of the EMNIST dataset is done in several ways (Cohen et al., 2017): 1) By_Page (unprocessed binary scans of handwriting forms), 2) By_Author (separate handwritten characters separated by their author), 3) By_Field (characters separated and organized according to the field in the form), 4) By_Class (symbols are separated by 62 classes [0-9], [A-Z], [a-z]), and 5) By_Merge (some lowercase and uppercase classes of letters were merged together due to their similarity comprising a 47-class dataset).

The proposed Cyrillic-MNIST dataset contains 121,234 samples of 42 Cyrillic letters of grayscale (28x28), binary (28x28) and original RGB images of various sizes (Mean = 45x42, Min = 24x20, Max = 141x123, Mode = 41x37, SD = 12x10), with 2000 balanced images per letter. In organizing the dataset, we decided to follow a similar division strategy as in EMNIST (Cohen et al., 2017) and organized the dataset in five different forms to fit different possible purposes of the end users:

1) Russian Balanced: the first type of organization is a balanced dataset of 33 classes of Russian alphabet. This version contains 2,000 samples in each class which adds up to 66,000 samples. The distribution of samples is presented in Figure 1a.

2) Cyrillic Letters Unbalanced: the second type of organization presents unbalanced version of 42 classes of extended Cyrillic alphabet. This version contains all 121, 234 samples. The distribution of samples is presented in Figure 1b.

3) Cyrillic Letters Balanced: the third type of organization is a balanced dataset of 42 classes of extended Cyrillic alphabet. Kazakh alphabet consists of 33 letters of Russian alphabet with additional 9 letters. The dataset contains 2,000 samples in each class which adds up to 84,000 samples in the dataset. The distribution of samples is presented in Figure 1c.

4) Cyrillic Lowercase and Uppercase Unbalanced: another way to organize the dataset was to split every class to lowercase and uppercase letters. An interesting point to note here is that some letters such as і, ь, and ъ do not have an uppercase letter, therefore they are only present in lowercase set. The distribution of samples is presented in Figure 1d.

5) Cyrillic Merged Unbalanced: due to the fact that most of the classes have similar lowercase and uppercase letters, we decided to merge them in one class. Exceptions were 11 letters: А-*а*, Б-*б*, В-*в*, Г-*г*, Ғ-*ғ*, Д-*д*, Е-*е*, Ё-*ё*, З-*з*, Р-*р*, Т-*т*. The distribution of samples is presented in Figure 1e.

2. Methodology

This section describes our process of data collection and raw data handling.

2.1. Data collection

The data collection was approved by Nazarbayev University Research Ethics committee. All participants signed consent forms to agree for the data to be processed and published in an open access format. Participants were mostly students and staff of the university. People were recruited via a social media campaign. Some people sent their handwritten samples by scanning or taking photos of their physically written sheets, whereas others gave their papers for us to digitise them. The dataset was collected from more that 100 adult contributors of different age groups. The resulting dataset contains over 121,234 samples. Contributors were asked to write on blank A4 paper as many letters as they want, in return for a chocolate, tea or coffee. Each contributor spent around 20 minutes writing on empty A4 sheets with a standard pen or pencil. Therefore, contributions by authors are unbalanced, as some participants were willing to write more than others. Participation was completely voluntary.

2.2. Conversion Process

In order to organize raw data into a dataset, we developed an automated data handling pipeline. The process of data conversion involved classical image processing techniques for noise removal and processing of the data.

To convert raw data to be comparable with the state-of-the-art OCR results achieved on MNIST dataset of 28x28 pixel binary images, we had to develop a conversion tool. The tool was developed with the preprocessing done for MNIST and EMNIST in mind (Baldominos et al., 2019). The conversion was done in several steps. Since original images come in different sizes and aspect ratios the Region of Interest is converted to binary image according to a predefined threshold. Afterwards, the image is padded according to its aspect ratio. Finally, it is being resized to 28x28 binary image.

2.2.1. Raw image preprocessing

Firstly, in order to convert the handwritten form to binary images, we used Otsu threshold (Otsu, 1979). Otsu threshold allowed us to better segment letters as it minimized effects of variable lighting conditions and paper quality. Then, in order to remove noise, we applied morphological operations on the resulting binary images. Those operations were done using OpenCV[1] Python library. Finally, in order for the algorithm to consider letters that contain acutes or separate parts (e.g. Ё, Й, Ы, і), we expanded the contours of letters using Sobel operator.

[1] opencv.org/

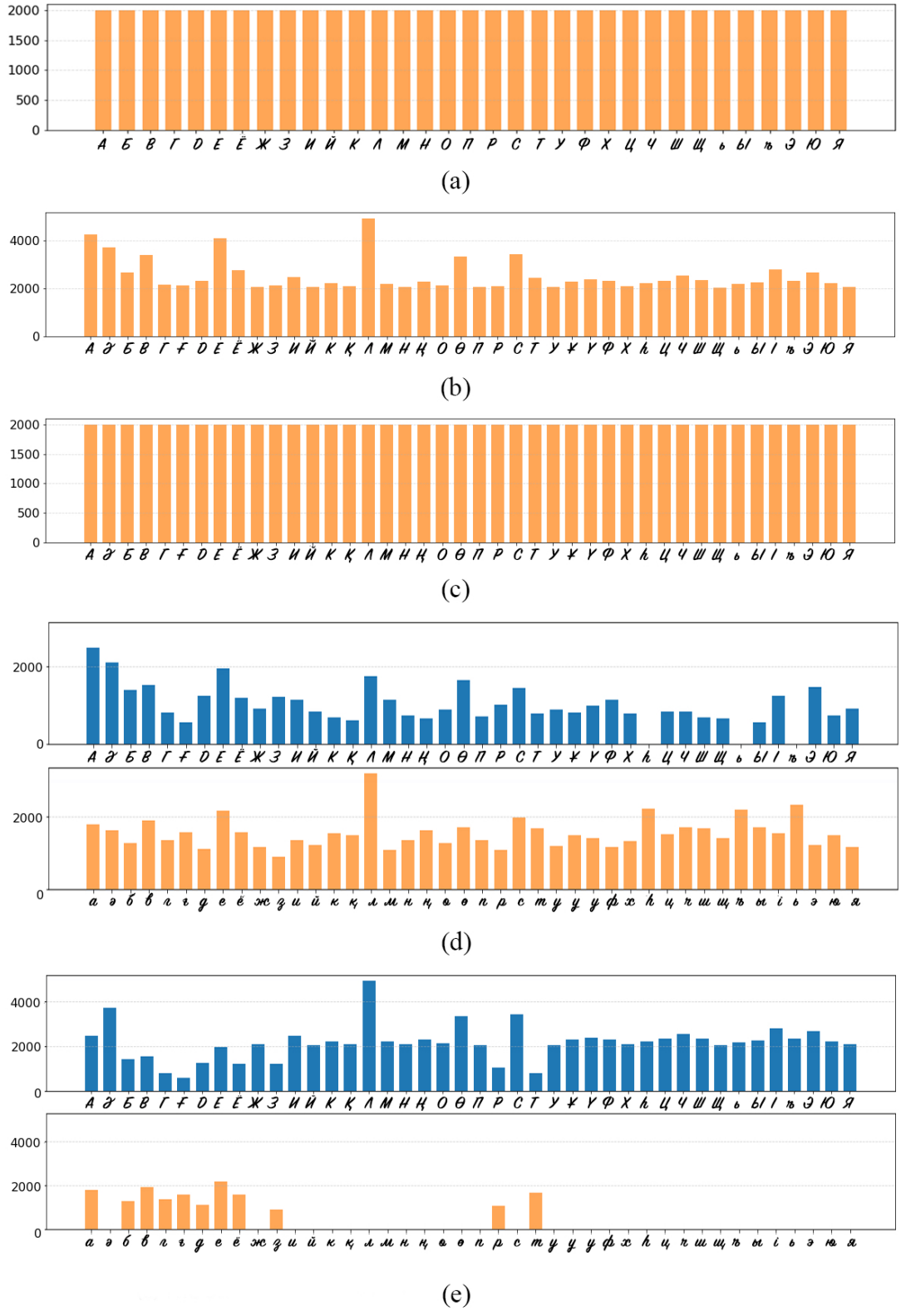

Figure 1: Distribution of samples in the Cyrillic-MNIST dataset: **(a)** Russian Balanced Dataset, **(b)** Unbalanced Letters Dataset, **(c)** Balanced Letters Dataset, **(d)** Unbalanced Cases Dataset, **(e)** Merged Unbalanced Cases Dataset

2.2.2. Contour finding

We chose to locate letters according to their contours for several reasons. Firstly, this method did not need training like any machine learning method. Secondly, letters were of different forms due to individual handwriting styles of contributors, therefore template

matching would not meet our needs either. Thus, contour finding was the best option at the initial stages of data handling.

After converting to binary images, contour finding was performed. This was done using OpenCV function, findContours(), based on the work of Suzuki and Abe "Topological structural analysis of digitized binary images by border following" (Suzuki and others, 1985). Afterwards, contours were analyzed for their length and aspect ratio. Contours shorter than certain value were considered to be noise and discarded. Then, the coordinates of the found contours were stored for further processing.

2.2.3. Cropping

Located letters were then cropped and padded according to their aspect ratio so that the resulting image had a square shape. Then, the square image was resized to 28x28 pixel binary image. This interpolation caused some of the black pixels turning gray. And, finally, the images were binarized again using fixed threshold to eliminate the effect of the interpolation.

3. Experiments

We conducted a series of experiments on three versions of the images (RBG, grayscale, and binary). Table 1 summarizes the results that are grouped by image color space.

3.1. Experiments on RGB data

The results from Table 1 clearly demonstrate that LeNet-like CNN outperforms all the other algorithms. The best accuracy was demonstrated on 53-class data suggesting that visual similarities of the letters are important. Generally, CNN performs better on RGB images suggesting that converting to grayscale or binary versions results in losing some information. In contrast, classical algorithms perform better on binary images rather than grayscale with Random Forest that is competitive with NNs. Thus, we suggest that binary images might be more suitable in case of computational restrictions.

3.2. Comparison with the state-of-the-art OCR results

We selected two standard models (Logistic Regression and Random Forest) and two neural network architectures from the cross-comparison table provided by the MNIST's authors[2]. In particular, we compared 784-500-500-2000 network described in Hinton and Salakhutdinov (2006) and custom CNN similar to LeNet-5 (LeCun et al., 1998). Table 1 provides the obtained results.

[2]yann.lecun.com/exdb/mnist/

3.3. Comparison with EMNIST

In addition, we performed a series of experiments in order to compare the performance of different types of dataset organizations. We evaluated the performance of two machine learning approaches on EMNIST and Cyrillic-MNIST. The first was, Keras[3]-based Neural Network of variable Hidden Layer size and Logistic Regression from Scikit-learn[4] Python libraries. For all dataset organizations we divided samples randomly to 80% training and 20% test sets. Models used for comparison are relatively simple and therefore, do not depict the best performance. Application of more complex, modern classification techniques might increase accuracy.

3.3.1. Neural Network

In this part we have utilized a single hidden layer with variable neuron number, between 1,000 and 10,000. This architecture allowed us to assess the performance of classifier trained on low level features. We have chosen Keras-based Neural Networks due to its simplicity of model creation and computational efficiency.

This evaluation was to compare the performance of NN-based classification on Cyrillic-MNIST Balanced Letters dataset, which contains 42 classes and 84,000 samples, and EMNIST Letters dataset with 26 classes and 145,600 samples.

A single layer NN classifier with variable number of neurons was used to test the performance of CMNIST against EMNIST. NN performed slightly better on Cyrillic-MNIST dataset rather than on EMNIST, with mean accuracy of 92.5% and 91.3% respectively, for several reasons. The first reason may be that Cyrillic-MNIST dataset consists of binary images, whereas EMNIST dataset is comprised of grayscale images. It is easier for the network to work with binary than with grayscale images, as a variance of possible feature values decreases drastically in binary images. Secondly, we had nearly five times less contributors to the dataset than there were in EMNIST (Cohen et al., 2017). This may be the reason of a higher test accuracy of Cyrillic-MNIST, as variability of handwriting styles is much lower than that in EMNIST. This limitation will be addressed by collecting more data reaching the size of the EMNIST dataset.

3.3.2. Linear Classifier

In order to test dataset organizations further we again decided to classify Cyrillic-MNIST Balanced Letters dataset, and EMNIST Letters dataset using linear classifier, namely Logistic Regression. The performance of EMNIST is lower than that of Cyrillic-MNIST dataset, 70.88% and 73.02% respectively, this may be due to higher variance of handwriting styles (i.e. number of authors) and the binary nature of samples of Cyrillic-MNIST.

[3]keras.io

[4]scikit-learn.org/0.20/

Table 1: Experimental Results

2*Image color space	2*Models	42 unbalanced		42 balanced		53 merged		81 upper lower	
		val	test	val	test	val	test	val	test
RGB	LeNet	**97.49**	**97.53**	**96.73**	**96.79**	**97.62**	**97.67**	**92.61**	**92.13**
4*Gray scale	LeNet	**97.19**	**96.81**	**97.06**	**96.69**	**97.51**	**97.24**	**92.42**	**91.84**
	784-500-500-2000-num_cl	96.39	96.16	95.24	95.01	96.55	96.54	91.58	91.19
	Logistic Regression	71.76	71.69	72.27	72.25	73.55	74.12	71.55	71.03
	Random Forest	91.55	91.55	90.19	90.65	91.74	91.48	85.63	85.86
4*Binary	LeNet	**96.25**	**96.19**	**96.75**	**95.76**	**96.71**	**96.7**	**89.07**	**88.57**
	784-500-500-2000-num_cl	96.13	95.97	95.68	95.06	96.2	96.3	88.97	87.83
	Logistic Regression	73.76	73.7	73.21	73.02	75.41	75.35	72.15	72.07
	Random Forest	94.09	94.22	92.88	93.5	94.04	94.18	87.02	87.06

3.4. Comparison of dataset organizations

To compare and evaluate the difference in performance of various organization types of Cyrillic-MNIST dataset, we firstly utilized Logistic Regression and several types of neural networks.

Both Cyrillic-MNIST Unbalanced Letters and Cyrillic-MNIST Balanced Letters datasets perform similarly on linear classifier. Due to the analogous distribution of samples by classes and insignificant difference in number of samples. Whereas, Cyrillic-MNIST Unbalanced Cases and Cyrillic-MNIST Unbalanced Merged Cases have more noticeable difference in performance. This can be explained by the fact that similar in shape classes were merged together, and therefore the classification task became simper. Table 1 displays these results.

Then, we compared Cyrillic-MNIST Unbalanced Letters and Cyrillic-MNIST Balanced Letters datasets. Both datasets are comprised of 42 classes, but Cyrillic-MNIST Unbalanced Letters has 121,234 samples, whereas Cyrillic-MNIST Balanced Letters has 84,000 samples. As it could be seen from Table 1, Cyrillic-MNIST Unbalanced Letters and Cyrillic-MNIST Balanced Letters perform similarly well. The change is relatively small and may be due to the fact that most of the classes in Cyrillic-MNIST Unbalanced Letters have about 2,000 samples, and only several classes have more samples, whereas Cyrillic-MNIST Balanced Letters has 2,000 samples per every class (see Figure 1).

Another two types of dataset organizations that were tested on the classifier are Cyrillic-MNIST Unbalanced Cases and Cyrillic-MNIST Unbalanced Merged Cases. Cyrillic-MNIST Unbalanced Cases consists of 81 classes and 121,234 samples, Cyrillic-MNIST Unbalanced Merged Cases contains the same number of samples, but divided into 53 classes according to similarity in letter appearance. From the Table 1 we can see that Cyrillic-MNIST Unbalanced Merged Cases generally outperforms Cyrillic-MNIST Unbalanced Cases. The reason for that might be that in Cyrillic-MNIST Unbalanced Cases some lowercase and uppercase letters are being confused with each other due to similarity in shape, therefore, test accuracy decreases for Cyrillic-MNIST Unbalanced Cases dataset.

4. Conclusion

This paper presents the first Cyrillic handwritten dataset organized in 5 ways to fit different purposes of the end users. As the aim was to follow the format and structure of the MNIST and EMNIST, Cyrillic-MNIST dataset can easily be merged and utilized for comparison on various machine learning tasks. The dataset was collected from adults, and the way the dataset was collected and handled could be used to other dataset creation efforts. Apart from the obvious benefit of this dataset to 112 languages that have used or use Cyrillic-based alphabets due to off-the-shelf deployment with no or minor data collection needed (e.g. no letters for Bulgarian, +1 letters for Belorussian support, +3 for Ukrainian). In addition, there is a high possibility that our dataset could be used for benchmarking ML algorithms, for example to evaluate how preprocessing steps affect the performance. Since we provide the original RGB images of various sizes, it contains noise due to cropping and real-world computer vision challenges such as different backgrounds, ink types (black pen, blue pen, pencil), image quality and resolutions. This paper also provides the benchmark performance of different classification techniques on the dataset in order to compare different types of organization between each other as well as to compare the Cyrillic-MNIST dataset with EMNIST Letters. With that being said, Cyrillic-MNIST can serve as a standalone dataset as well as an extension of the existing and widely used datasets, which can broaden the variety of applications of MNIST and EMNIST datasets.

5. Acknowledgements

This work was supported by the Nazarbayev University Collaborative Research Program grant, "CoWriting Kazakh: Learning a New Script with a Robot" (award number is 091019CRP2107).

6. Bibliographical References

Baldominos, A., Saez, Y., and Isasi, P. (2019). A survey of handwritten character recognition with mnist and emnist. *Applied Sciences*, 9(15):3169.

Britannica. (2019). Cyrillic alphabet. *The ed. of Encyclopedia Britannica*, May. http://www.britannica.com/topic/Cyrillic-alphabet.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. (2017). Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE.

Das, N., Acharya, K., Sarkar, R., Basu, S., Kundu, M., and Nasipuri, M. (2014). A benchmark image database of isolated bangla handwritten compound characters. *International Journal on Document Analysis and Recognition (IJDAR)*, 17(4):413–431.

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.

Draghici, S. (1997). A neural network based artificial vision system for licence plate recognition. *International Journal of Neural Systems*, 8(01):113–126.

Du, S., Ibrahim, M., Shehata, M., and Badawy, W. (2012). Automatic license plate recognition (alpr): A state-of-the-art review. *IEEE Transactions on circuits and systems for video technology*, 23(2):311–325.

Gatos, B., Pratikakis, I., and Perantonis, S. J. (2004). An adaptive binarization technique for low quality historical documents. In *International Workshop on Document Analysis Systems*, pages 102–113. Springer.

Gordienko, N., Kochura, Y., Taran, V., Peng, G., Gordienko, Y., and Stirenko, S. (2018). Capsule deep neural network for recognition of historical graffiti handwriting. *arXiv preprint arXiv:1809.06693*.

Grother, P. J. (1995). Nist special database 19 handprinted forms and characters database. *National Institute of Standards and Technology*.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

Iliev, I. G. (2013). Short history of the cyrillic alphabet. *International Journal of Russian Studies*, 2(2):1–65.

Khosravi, H. and Kabir, E. (2007). Introducing a very large dataset of handwritten farsi digits and a study on their varieties. *Pattern recognition letters*, 28(10):1133–1141.

Kusetogullari, H., Yavariabdi, A., Cheddad, A., Grahn, H., and Hall, J. (2019). Ardis: a swedish historical handwritten digit dataset. *Neural Computing and Applications*, pages 1–14.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Liu, C.-L., Yin, F., Wang, D.-H., and Wang, Q.-F. (2011). Casia online and offline chinese handwriting databases. In *2011 International Conference on Document Analysis and Recognition*, pages 37–41. IEEE.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66.

Palumbo, P. W., Srihari, S. N., Soh, J., Sridhar, R., and Demjanenko, V. (1992). Postal address block location in real time. *Computer*, 25(7):34–42.

Suzuki, S. et al. (1985). Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46.

Vial, G. (2019). Comnist. http://www.github.com/GregVial/CoMNIST.

WorldStandards. (2020). The world's scripts and alphabets. *World Standards*, May. http://www.worldstandards.eu/alphabets.