EMNIST: extending MNIST to handwritten letters

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik
The MARCS Institute for Brain, Behaviour and Development
Western Sydney University
Penrith, Australia 2751
Email: g.cohen@westernsydney.edu.au

***Abstract*—The MNIST dataset has become a standard benchmark for learning, classification and computer vision systems. Contributing to its widespread adoption are the understandable and intuitive nature of the task, the relatively small size and storage requirements and the accessibility and ease-of-use of the database itself. The MNIST database was derived from a larger dataset known as the NIST Special Database 19 which contains digits, uppercase and lowercase handwritten letters. This paper introduces a variant of the full NIST dataset, which we have called Extended MNIST (EMNIST), which follows the same conversion paradigm used to create the MNIST dataset. The result is a dataset that constitutes a more challenging classification task involving letters and digits, and one that shares the same image structure and parameters as the original MNIST task, allowing for direct compatibility with all existing classifiers and systems. Benchmark results using an online ELM algorithm are presented along with a validation of the conversion process through the comparison of the classification results on NIST digits and the MNIST digits.**

I. Introduction

The importance of good benchmarks and standardized problems in computer vision and learning systems cannot be understated. These provide a fair means to compare, analyze and contrast different learning approaches and techniques, allowing researchers to quickly gain insight into the performance and peculiarities of a method or algorithm. As a single dataset may only cover a specific task, the existence of a good and varied suite of datasets is important to allow a more holistic approach to judging and assessing algorithm performance.

Benchmark datasets generally constitute a classification task, and perhaps the most widely known dataset in the computer vision and neural networks community is the venerable MNIST dataset, which was first introduced in 1998 by LeCun et al. [1].

A good dataset needs to represent a sufficiently challenging problem to make it both useful and to ensure its longevity [2]. This is perhaps where MNIST has suffered in the face of the increasing performance achieved using deep learning and convolutional neural networks. Classification accuracy on MNIST has reached the point at which the accuracy of the labeling can be called into question. Thus, it has become more a means to test and validate a classification system than a meaningful or challenging benchmark.

The accessibility of the MNIST dataset has almost certainly contributed to its widespread use. The entire dataset is relatively small (by comparison to more recent benchmarking dataset), free to access and use, and is encoded and stored in an entirely straightforward manner. The encoding does not make use of complex storage structures, compression, or proprietary data formats. For this reason, it is remarkably easy to access and include the dataset from any platform or through any programming language.

The MNIST database is a subset of a much larger dataset known as the NIST Special Database 19 [3]. This dataset contains both handwritten numerals and letters. It represents a much larger and more extensive classification task, along with the possibility of adding more complex tasks such as semantic interpretations through words interpretation.

The NIST dataset, by contrast to the MNIST, has remained difficult to access and use. Driven by the higher cost and availability of storage when it was collected, the NIST dataset was originally stored in a remarkably efficient and compact manner. Although source code to access the data is provided, it remains challenging to use on modern computing platforms. For this reason, the NIST have recently released a second edition of the NIST dataset. However, the encoding of that dataset remains in the original format from which MNIST was extracted.

The purpose of this paper is to provide a means of converting the digits in the NIST Special Database 19 to a format that is directly compatible with the MNIST digit dataset. This allows the drop-in replacement of the modified NIST datasets without needing to adjust the input layer size or the input range. We have named this resulting database the Extended Modified NIST Dataset (EMNIST).

This paper provides a detailed conversion process for the entire NIST dataset based on the one used to create the MNIST digits. Baseline benchmark accuracies are provided using a three-layer network trained using an Extreme Learning Machine (ELM) classifier, and the conversion process is validated using a comparison of the classification performance on the original MNIST digits and corresponding converted digits from the NIST dataset.

The EMNIST dataset can be accessed and downloaded from https://www.westernsydney.edu.au/bens/home/reproducible_research/emnist.

II. Methodology

This paper introduces the EMNIST dataset and then applies two ELM-based neural networks to the classification tasks made possible with the complete NIST dataset. The classification systems demonstrate the structure and nature of

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

the classification tasks involving both digits and letters, and additionally provides benchmark results for this new dataset.

The purpose of the EMNIST dataset is to function as a drop-in replacement for the MNIST dataset in existing classification systems. As a result, the focus of the methodology is on reproducing the steps used to convert the NIST dataset into the MNIST dataset and apply them to the entire NIST Special Database 19. This conversion process and the modifications implemented to better convert the letters is described in Section II-B.

The classifier used in this work are introduced and described in Section II-C and were applied to three different classification tasks based on combinations of letters, digits and the full classification task using the NIST dataset. In addition, a subset of the EMNIST digits were extracted and used to train a classifier. The results were then compared to an identical network applied to the MNIST digits in order to explore and validate the conversion process used in this work.

The code used to extract the data from the original NIST dataset, the conversion code from 128×128 pixel binary images to the converted gray-scale and the code to perform the training and testing splits are provided as supplementary material to this paper.

A. The MNIST and NIST Dataset

The NIST Special Database 19 [3] contains handwritten digits and characters collected from over 500 writers. The dataset contains both binary scans of the handwriting sample forms, and segmented and labeled numerical digits, uppercase and lowercase letters. The database was published as a complete collection in 1995, and then re-released in a more modern file format in September 2016 [4]. The dataset itself contains, and supersedes, a number of previously released handwriting datasets, such as the Special Database 1, 3 and 7.

The MNIST dataset is derived from a small subset of the numerical digits contained within the NIST Special Databases 1 and 3, and were converted using the method outlined in [1]. The NIST Special Database 19, which represents the final collection of handwritten characters in that series, contains additional handwritten digits and an extensive collection of uppercase and lowercase handwritten letters. The authors and collators of the NIST dataset also suggest that the data contained in Special Database 7 (which is included in Special Database 19) be used exclusively as a testing set as the samples were collected from high school students and pose a more challenging problem.

The NIST dataset was intended to provide multiple optical character recognition tasks and therefore presents the character data in five separate data hierarchies:

- ***By_Page***: The character data was collected through a standard form which the writer was asked to complete. 3699 forms were completed and comprise the source of the dataset, and this hierarchy contains unprocessed binary images of these forms.
- ***By_Author***: This hierarchy contains the segmented character classes organized by writer. It allows for such tasks as writer identification but offers little in the way of classification benefit as each grouping contains digits from multiple classes.
- ***By_Field***: This organization contains the digits and character sorted by the field on the collection field in which they appear. This is primarily useful for segmenting digits as they appear in their own isolated fields.
- ***By_Class***: This represents the most useful organization from a classification perspective as it contains the segmented digits and characters segmented by class. There are 62 classes comprising [0-9], [a-z] and [A-Z]. The data is also split into a suggested training and testing set.
- ***By_Merge***: This data hierarchy addresses an interesting problem in the classification of handwritten digits, which is the similarity between certain uppercase and lowercase letters. Indeed, these effects are often plainly visible when examining the confusion matrix resulting from the full classification task on the *By_Class* dataset. This variant on the dataset merges certain classes, creating a 37-class classification task. The merged classes are C, I, J, K, L, M, O, P, S, U, V, W, X, Y and Z.

The conversion process described in this paper and the provided code is applicable to all hierarchies with the exception of the *By_Page* hierarchy as it contains fundamentally different images. However, the primary focus of this work rests with the *By_Class* and *By_Merge* organizations as they encompass classification tasks that are directly compatible with the standard MNIST dataset classification task. The *By_Author* class represents an interesting opportunity to formulate fundamentally new classification tasks, such as writer identification from handwriting samples, but this is beyond the scope of this work.

B. Conversion Process

The NIST Special Database 19 was originally released in 1995 and made use of an encoding and compression method based on the CCITT Group 4 algorithm and packed into a proprietary file format. Although the release of the database includes code to extract and process the dataset, it remains difficult to compile and run these utilities on modern systems.

A second edition of the dataset was published in September 2016 and contains the same data encoded using the PNG file format. This work makes use of the original dataset and includes code and instructions to extract and convert those files. The post-processing techniques used to create the down-sampled 28×28 pixel images are directly compatible with the second edition of the dataset.

The conversion process converts the 128×128 pixel binary images found in the NIST dataset to 28×28 pixel 8-bit gray-scale images that match the characteristics of the digits in the MNIST dataset. An overview of the conversion process is presented in Figure 1. As described in Section II-A, the NIST dataset contains $814,255$ images in four different hierarchies which affect the labeling and organization of the data. For this work, only the *By_Class* and *By_Merged* hierarchies are used.

The conversion process for both is identical and only the class labels (and number of class labels) changes.

The conversion methodology follows the same overall paradigm as the conversion process used for the MNIST dataset and outlined in [1], but makes use of a different down-sampling method to better handle the variations in shape and size of the characters in the NIST dataset.

In order to convert the dataset, each digit is loaded individually and blurred using a Gaussian filter. A bounding box is fitted to the character in the image and extracted. As the size and shape of the characters and digits vary both from class-to-class and from writer-to-writer, there is a lot of variance in the size of the region of interest. Whereas the original MNIST conversion technique down-sampled the digits to either a 20×20 pixel or a 32×32 pixel frame before placing it into the final 28×28 pixel frame, the technique used in this paper attempts to make use of the maximum amount of space available.

Instead, the extracted region of interest is centered in a square frame with lengths equal to the largest dimension, with the aspect ratio of the extracted region of interest preserved. This square frame is then padded with an empty 2 pixel border to prevent the digits and characters from touching the border. Finally, the image is down-sampled to 28×28 pixels using a bi-cubic interpolation algorithm, resulting in a spectrum of intensities which are then scaled to the 8-bit range.

C. Classification Methodology

The classification results provided in this work are intended to form a benchmark for the datasets provided. As a result, a simple three-layer ELM network, as described in [5], was used to perform the classification. It is expected that deeper and more sophisticated networks will provide higher classification performance.

The networks used in this work were trained using two different methods. The pseudo-inverse required for the ELM cannot be calculated in a single step due to size of the dataset and the networks are instead trained using the Online Pseudo-Inverse Update Method (OPIUM) [6]. This method iteratively calculates the exact pseudo-inverse solution for the output weights and allows the network to handle datasets of any size.

The second training method used is a variant of the OPIUM method in which the computationally intensive calculation of the internal cross-correlation matrix is replaced by an approximation, which greatly improves the speed of the algorithm at the cost of accuracy. This method is referred to as OPIUMLite [6].

The classification networks were tested over a range of hidden layer sizes. A random training order was used and kept constant throughout all the tests. Multiple trials with different random input weights and the same hidden layer size were conducted and the mean classification accuracy and standard deviation reported when appropriate.

III. Dataset Characteristics

The handwriting data used in the Special Database 19 was collected from both Census employees and high-school students. The specifications provided alongside the dataset include a suggestion that the handwritten digits from the student corpus be used as the testing set as they represent a somewhat more difficult task. The structure of the EMNIST dataset separates the NIST dataset into these training and testing sets. These are provided for both the *By_Class* and *By_Merge* data hierarchies.

TABLE I
Breakdown of the number of available training and testing samples in the NIST Special Database 19.

	Type	No. Classes	Training	Testing	Total
By Class	Digits	10	344,307	58,646	402,953
	Uppercase	26	208,363	11,941	220,304
	Lowercase	26	178,998	12,000	190,998
	Total	**62**	**731,668**	**82,587**	**814,255**
By Merge	Digits	10	344,307	58,646	402,953
	Letters	37	387,361	23,941	411,302
	Total	**47**	**731,668**	**82,587**	**814,255**
MNIST [1]	Digits	10	60,000	10,000	70,000

Table I shows the breakdown of the training and testing sets for the various classification tasks in the EMNIST dataset. Both the *By_Class* and *By_Merge* hierarchies contain 814,255 handwritten characters consisting of 731,668 training samples and 82,587 testing samples. It should be noted however, that almost half of these samples are handwritten digits.

The *By_Class* hierarchy contains 52 individual classes covering both the uppercase and lowercase of each letter whereas the *By_Merge* has only 37 letter classes due to the merging of certain letter classes. The digit classes are unaffected by the merged classes and therefore remain the same for both hierarchies. The handwriting collection task involved copying out a section of the U.S. Constitution and resulted in a different number of total uppercase and lowercase letters in the dataset. One of the advantages of the *By_Merge* hierarchy is that it serves to better distribute the number of samples across the 47 classes.

The number of samples per class also varies significantly in the dataset as shown in Table II. The number of handwritten digits per class varies from a low of 36,606 presentations of the digit 5 to a maximum of 44,707 for the digit 1. The table also provides the same statistics on the distribution of digits in the MNIST dataset and shows a similar distribution of samples per class as the full NIST dataset.

Table II also shows the largest and smallest number of letter samples in the dataset and demonstrates the large degree of variability in the number of samples per class. This variability is the result of the collection process. The number of samples per class follows the distribution of letters in English words. This is evident when examining the classes as the letter *e* has 34,508 samples and *t* has 32,623 samples, whereas less frequent letters, such as *k* has only 5807 samples in total.

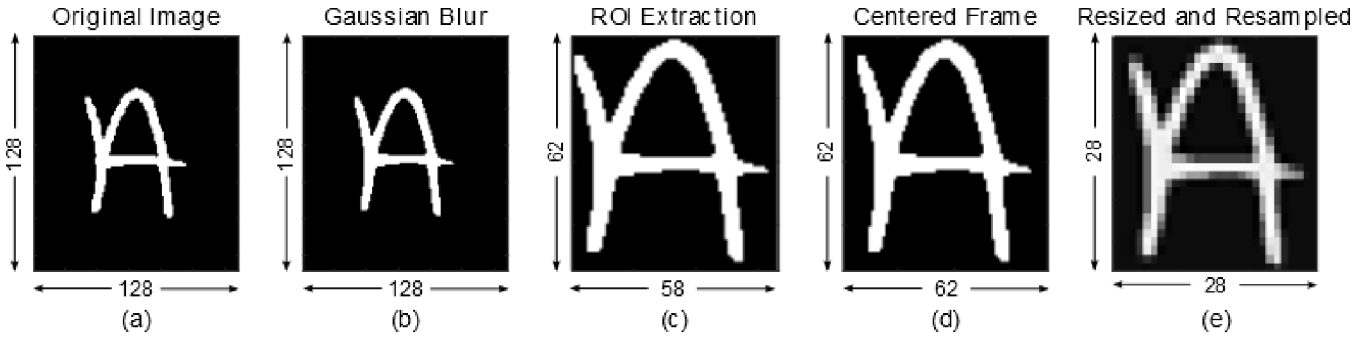

Fig. 1. **Diagram of the conversion process used to convert the NIST dataset.** The original images are stored as 128×128 pixel binary images as shown in (a). A Gaussian filter with $\sigma = 1$ is applied to the image to soften the edges as shown in (b). As the characters do not fill the entire image, the region around the actual digit is extracted (c). The digit is then placed and centered into a square image (d) with their aspect ratio preserved. The region of interest is padded with a 1 pixel border when placed into the square image, matching the clear border around all the digits in the MNIST dataset. Finally, the image is down-sampled to 28×28 pixels using bi-cubic interpolation. The range of intensity values are then scaled to $[0, 255]$, resulting in the 28×28 pixel gray-scale images shown in (e).

TABLE II
SUMMARY OF IMAGES PER CLASS FOR EMNIST AND THE MNIST DATASET

	Min	Max	Mean	Std
Digits	36606	44704	40295	1971
Uppercase Letters	2850	29139	8473	6316
Lowercase Letters	2213	28723	7346	6873
By_Class Letters	5807	34508	15819	8650
By_Merge Letters	2961	32354	11116	7489
MNIST [1]	6313	7877	7000	378.8

IV. RESULTS

The results presented in this work are intended to serve as a benchmark classification accuracy for the EMNIST dataset. These results were achieved using a straightforward three-layer network and are not intended to represent the state-of-the-art classification accuracy. No attempts to optimize either the inputs to the classifier or to tune the classifier itself were performed. Each network was trained using the complete training set and included no input transformations or amended inputs. Multiple trials of each experiment were performed and each network size was tested with a consistent and random training order across all trials of the experiment. Only the random hidden layer weights from the input layer to the hidden layer varied from trials with the same hidden layer size.

Figure 2 presents the results of the full classification task using digits, uppercase, and lower case letters. The same tests were performed on the *By_Class* and *By_Merged* data hierarchies. As expected, the results achieved on the *By_Merged* dataset were consistently higher than those achieved on the *By_Class* dataset, indicating that there are indeed misclassifications between the uppercase and lowercase representations of the same letters.

The highest accuracy was achieved using the network with 10,000 hidden layer neurons and trained with the OPIUM method. It achieved a best-case accuracy of $71.43\% \pm 0.06\%$ on the *By_Class* dataset and $75.68\% \pm 0.05\%$ on the *By_Merge* dataset. It should be noted that although both networks contain the same number of training samples, there are fewer classes in the *By_Merge* dataset. By contrast, the best-case accuracy obtained with the OPIUMLite training method was $62.64\% \pm 1.69\%$ for the *By_Class* dataset and $67.30\% \pm 2.27\%$ on the *By_Merge* dataset.

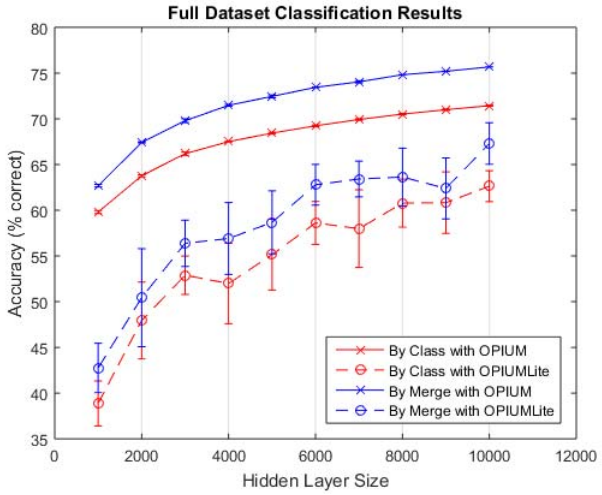

Fig. 2. **Classification results for the *By_Class* and *By_Merge* data hierarchies using the OPIUM and OPIUMLite classifiers**. The above graphs shows the classifier performance for networks trained on the two data hierarchies described in Section II-A using the two different classification methods. As expected, the results of the merged dataset outperform the full 62-class classification task as there are fewer uppercase/lowercase errors. The OPIUM classifier also outperformed the OPIUMLite classifier in every test. 10 trials of each experiment were performed.

It is also clear from the figure that the OPIUM training method outperforms the OPIUM-Lite method when classifying the *By_Merge* dataset for each hidden layer size, but the difference is less pronounced when using the *By_Class* dataset. The OPIUMLite method requires far less computing power and time than the OPIUM method, but this comes at the cost

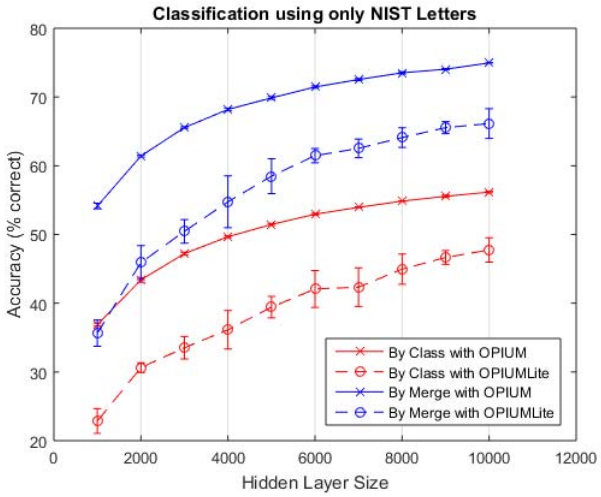

Fig. 3. **Classification results using only the letters in the NIST dataset for the *By_Class* and *By_Merge* data hierarchies.**. The above graph shows the performance of the classifiers trained using only the letter clasees from the two data hierarchies. The *By_Class* hierarchy contains 52 letter classes whilst the *By_Merge* contains only 37. It is clear from the above graph that the *By_Merge* hierarchy outperforms the *By_Class* hierarchy at every hidden layer size .

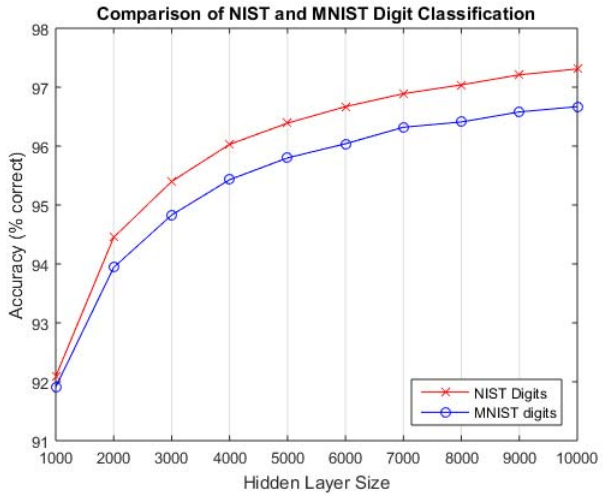

Fig. 4. **Comparison of the classification results for the NIST and MNIST digits**. In order to validate the conversion process, the same 60,000 digits used to create the MNIST dataset were extracted and converted using the process outlined in Section II-B. These two datasets were run through the same ELM classifier with varying hidden layer sizes but identical random layer weights. The results show that the conversion process produces a higher accuracy than the original MNIST, resulting from the differing approach to centering and down-sampling. 10 trials of each experiment were performed.

of classification accuracy.

Figure 3 shows the results achieved when discarding the digits in the NIST dataset and classifying solely on the letters. As with the full classification task, the network performance increased with the number of hidden layer neurons, and the OPIUM classifier reached an accuracy of $56.17\% \pm 0.11\%$ on the *By_Class* dataset and $74.95\% \pm 0.03\%$ on the *By_Merge* dataset. The difference between the performance of the two datasets is far greater in these results, indicating that the misclassifications between uppercase and lowercase letters plays a significant role in determining the performance of these networks. The *By_Class* also performs worse in this test than in the full classification task, due to the lack of the digit classes which represent an easier classification task and serve to bolster the performance of those networks.

It is also clear from the error bars in both Figure 2 and Figure 3 that the OPIUMLite classifier performs less consistently than the OPIUM method. However, only a single training epoch was performed when training the classifiers and it is likely that multiple training epochs would improve the accuracy when using OPIUMLite. The full OPIUM algorithm calculates the exact pseudoinverse, and therefore it is expected that multiple training epochs should not lead to improve performance.

In order to validate the dataset, a subset of the digits from the NIST Dataset were extracted and then converted to a corresponding 28×28 pixel format through the conversion process used to create the EMNIST dataset. A subset of 70,000 digits were selected from the NIST dataset and then classified using ELM networks of varying hidden layer sizes. An identical network was then used to classify the original MNIST digits, with the results presented in Figure 4.

The results of the comparison show that the NIST digits outperformed the MNIST digits for every hidden layer size. This increase in performance is likely due to the slightly different means of converting the images to a 28×28 pixel format. The description of the conversion process in the original paper describing MNIST and the means by which a subset was chosen was followed as closely as possible but a different training and testing split could also explain the difference in achieved accuracy. The results, however, show that the conversion process retains enough of the information for comparable accuracy on the digit classification problem, which implies that a similar amount of information will be retained for the converted letters in the dataset.

V. Conclusions

The NIST dataset represents a logical extension of the classification task presented in the MNIST dataset and the work presented in this paper suggests a consistent means of converting this dataset to a format that is directly compatible with classifiers built to handle the MNIST dataset. Benchmark results show that the classification task is indeed more challenging, and the use of letters in addition to numbers allows for more complex classification tasks involving words and letter frequency predictions. Additionally, the NIST dataset also contains other data hierarchies, such as full page scans of forms and characters organized by writer, which create the potential for deeper and complex classification tasks.

Acknowledgment

The authors would like to thank the organizers and funders of the Capo Caccia and Telluride Cognitive Neuromorphic

Engineering Workshops, at which the initial steps of this work were completed.

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[2] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades," *Frontiers in Neuroscience*, vol. 9, pp. 1–15, nov 2015.

[3] P. Grother, "NIST special database 19 handprinted forms and characters database," National Institute of Standards and Technology, Tech. Rep., 1995.

[4] P. Grother and K. Hanaoka, "NIST special database 19 handprinted forms and characters 2nd Edition," National Institute of Standards and Technology, Tech. Rep., 2016. [Online]. Available: http://www.nist.gov/srd/upload/nistsd19.pdf

[5] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, dec 2006.

[6] A. van Schaik and J. Tapson, "Online and adaptive pseudoinverse solutions for ELM weights," *Neurocomputing*, vol. 149, pp. 233–238, feb 2015.