Generating Synthetic Training Data using Neural Style Transfer for Historical Map Text Recognition

UM-CS-2019-004

Pia Bideau
College of Information and Computer Sciences
pbideau@cs.umass.edu

Jerod Weinman
Grinnell College
jerod@acm.org

Erik Learned-Miller
College of Information and Computer Sciences
elm@cs.umass.edu

Abstract

In this work, we address the problem of recognizing text in historical maps. While there has been good progress recently in both general optical character recognition (OCR) and scene text recognition, the area of text recognition for historical documents remains relatively unexplored. Text recognition on historical maps presents a number of challenges: text appears in dramatically different sizes, text can be stretched over large regions with significant intervening spaces, and the text baseline is often not horizontal - it can be vertical or even curved. The spatial distortion of text and other distractors such as rivers, streets, and geographical boundaries can make this text recognition particularly challenging. CNNs have shown great success in the area of scene text recognition. In this report, we apply a CNN architecture proposed by Jaderberg et al. [4] to the task of text recognition in historical maps. To increase the amount of training data for this problem, we present a new approach for producing abundant training data using using neural style transfer *[2]. Using this automatically generated map text data set, which contains many of the challenges inherent to map text recognition, we fine-tune a pretrained CNN and show improved performance on a set of historical maps.*

1. Introduction

Written text is a way to communicate, to spread information. Humans write books, articles, and notes to keep and share information with each other. In these cases, text is the only source of information, but text also can be combined with graphical elements. Documents in which text interacts with graphical elements include technical manuals, paintings, and maps. In this work we address the problem of text recognition on historical maps. In general the purpose of text in maps is to "label" and "name" graphical elements like rivers, cities, countries, buildings and other geographical entities. The text in historical maps is not strictly horizontal, often appearing in a wide variety of orientations and making it difficult to apply standard text reading techniques. Often it is aligned with a graphical element that is described by the text. Text may be aligned, for example, with a river or a street and written vertically or with a curved baseline. In addition to the high variability in orientation and size, map text is often crossed by the map's graphical elements. Making matters worse, those graphical features often use the same style as the word, which makes it difficult to distinguish between the *background* and the *text*.

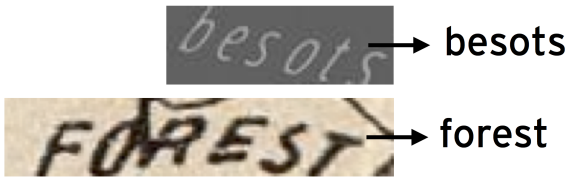

Figure 1. **Text recognition on historical maps.** The top word image "besots" is part of the Synthetic Word Dataset by Jaderberg et al. [4, 3]. The bottom word image "FOREST" comes from the historical map dataset [13]. Text recognition on maps comes with additional challenges such as rivers, streets, and geographic boundaries crossing the text. These distracting map features are often drawn in the same style as the text, as shown in this figure, which makes text recognition a very challenging task.

To our knowledge the difficulties that come with maps have been not specifically addressed in existing text recognition approaches. In particular, existing word recognition data sets focus on other problems such as scene text recognition, handwriting recognition, and OCR. In this work we focus on cases in which text interacts with graphical ele-

ments, text is overlaid by graphical elements, and corrupted text. The following contributions are:

- We propose a method to automatically generate a training data set for text recognition showing typical difficulties that might appear on maps. This new data set is generated using neural-style transfer [2]. We apply a style image to two content images (map background and text) separately. Having two images with similar style allows us to model the difficult cases in which text has to be recognized in a graphical environment that is visually hard to distinguish from text. This method allows the creation of an unlimited amount of synthetic training data without using real images, whose availability is limited.
- We fine-tune an existing CNN for text recognition that is pretrained on synthetic scene text data [4]. With a modestly sized training set, we achieve a small improvement on the map text dataset [13].

2. Related Work

Traditional text recognition has shown excellent performance on classical text documents. While text recognition on handwritten documents and machine written documents are well-studied problems with relatively mature solutions, scene text recognition remains being a challenging problem [1, 11, 7, 8]. [9, 12, 5, 10, 6] are CNN based approaches to scene text recognition. [9, 5, 6] use an LSTM network that is able to incorporate sequence based information - the knowledge of predicting a particular character when some character was previously observed. In this work we focus on a text recognition approach by Jaderberg et al. [3] using a CNN that predicts a character sequence of fixed length.

3. Neural-Style Transfer for Data Augmentation

An image can be separated into its style and its content. For example Figure 2 shows at the bottom row an artificially generated portrait of Brad Pitt. The content of this image is "Portrait of Brad Pitt". However "Portrait of Brad Pitt" alone doesn't specify the style - a portrait could be a drawing, a sketch a photograph etc. Neural-style transfer algorithms are able to extract a style from a *style image* and apply its style to a *content image*.

In the following section we will describe the idea of using neural style transfer to automatically generated a very realistic looking text recognition data set, that can be used for general text recognition tasks and addresses especially the problem of corrupted text due to a similar looking background. For example text on maps are often corrupted by rivers or streets crossing the text. Furthermore text, rivers, streets often show the same style.

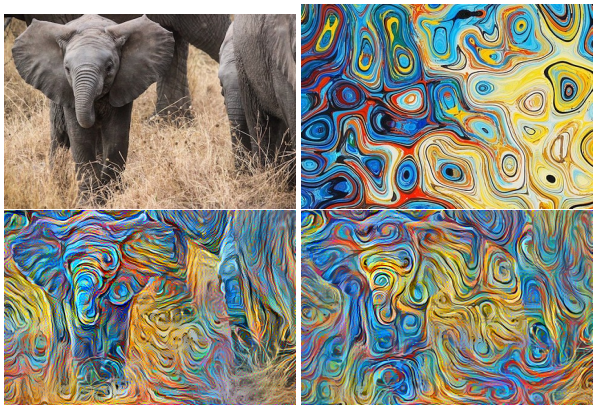

Figure 2. **Neural-Style Transfer: Content/Style Trade-off** This image shows a style image and a content image (top row). The style can be ported to the content image, which shows a baby elephant. The images at the bottom show the generated images with transfered style and different style weight. The style weight is increasing from the left to the right.

3.1. Generating a new Synthetic Text Recognition Dataset

We generated a new text recognition data set that addresses the challenges of text recognition that appear on maps and sometimes other technical manuals.

- corruption of text due to other graphical elements that are close to text or overlie the text.
- Other graphical elements that interact with the text show similar style as the text.

We created 4 different *style images* that show the style of "map". To generate those style images we took extracts of 4 different maps of the historical map dataset by Weinman et al. [13] and manually removed the text. Examples are shown in Figure 3 on the left. Besides this style image we used two content images: (1) background and (2) text. The content image for background can be seen in Figure 3 (left lower image). As text content we used the subset of the synthetic scene text dataset [4] which show a high variety of projective distortions of text written in different fonts. However this data set [4] is generated by natural image blending, thus the background varies but the background style and background font do not interact with each other directly due to their different style. The corruption of text due to the interaction of text and background that show similar style however is critical for text recognition on historical maps.

Given three input images one style image (out of the 4 map style images), a background content image and a text content image the procedure of automatically generating map text can be describes in tree steps:

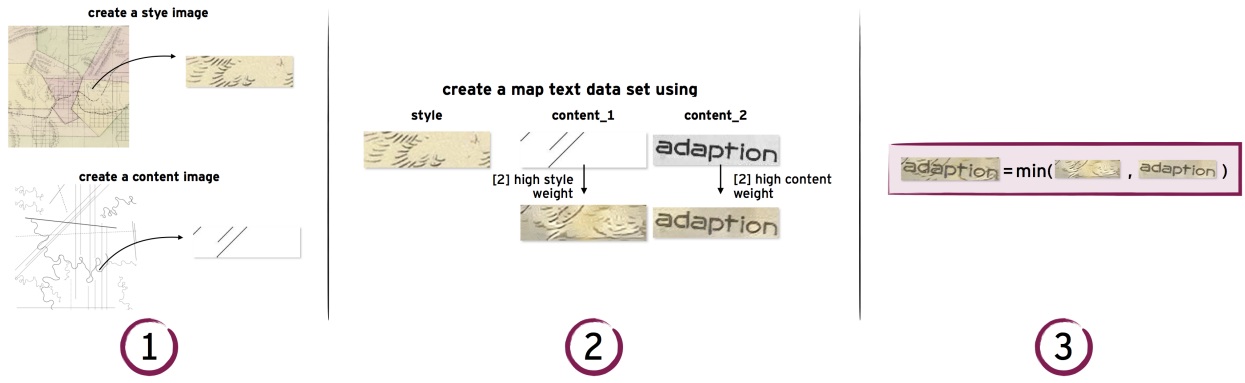

Figure 3. **Generating a synthetic dataset for text recognition on maps using neural style transfer.**

1. Take a random crop of the style image and the background content image

2. Apply neural-style transfer as described in [2] [1]. For both content images (text and background) the same style image generated in step 1 is applied. The background image is created with a content weight of 20 and style weight of 1000. The text image is created with a content weight of 100 and style weight of 200. It is important to keep a high content weight on the text since other wise its characters get significantly corrupted and the deformation is unrealistic. Please note that it is important that the content image for text shows dark text on a lighter background - not the other way around. The given map style, that is dark elements on lighter background mainly transform darker elements.

3. Take the minimum at each pixel. This will always choose the darker color thus the text as well as the graphical elements of background are kept.

The availability of real training data is often limited. With this procedure it is possible to generate an enormous amount of synthetic training data. We generated a total amount of 32800 map text images for training.

4. Character Sequence Encoding

The proposed network architecture by Jaderberg et al. [3], which is entirely character based, comes with high flexibility for word recognition. The network predicts a character sequence of length 23. For each character a classifier is trained separately. Thus 23 different classifiers are trained, one classifier for the first character, one classifier for the second character and so forth. Each character predicts a single $c_i \in \mathcal{C} = \{1, 2, ..., 36\}$. We have 36 different classes: 26 characters (case insensitive), 10 numbers and on non character class. So given an image that is showing the word "Hello" the network's predicted output would be a "h e l l o $*$". Here $*$ describes the non character class. Each character is predicted as $c_i = \arg\max_{c_i \in \mathcal{C} \cup \{\phi\}} P(c_i|\phi)$ - the probability of that character at position i given a set of learned features ϕ.

Training was done using the softmax loss. For fine tuning we used our generated synthetic maps data set. The new data set comprises 32800 map word images.

5. Experiments

In this section we discuss our results for historical map text recognition. Our work is based on the CNN of Jaderberg et al., which was originally trained on a synthetic scene text dataset [4]. We show results of two different experiments. First, we show results for the fine-tuned network using the new synthetic map data set (see Section 2). Second, we show results a network that was fine-tuned using the same subset of the synthetic word dataset as for the first experiment, but with map-like distractors only. In this case we did not apply style transfer either to the map content image (3: content_1) or the text content image (3: content_2), before generating the overlay image of both.

Text recognition in which distracting clutter has a significantly different style than the word being recognized may be an easier problem than recognizing text that has the same style as its distractors, we hypothesized. For example typeset word overlaid with a hand-painted green line may be easier to read than a hand-painted green word with the same overlaid line. This problem can be observed on historical maps. Text recognition on historical maps where the graphical elements directly interact with text of the same style is challenging, since it is hard to distinguish between the relevant text and other graphical elements, due to similarities in style.

From the original synthetic text data set of 9 million im-

[1] https://github.com/jcjohnson/neural-style

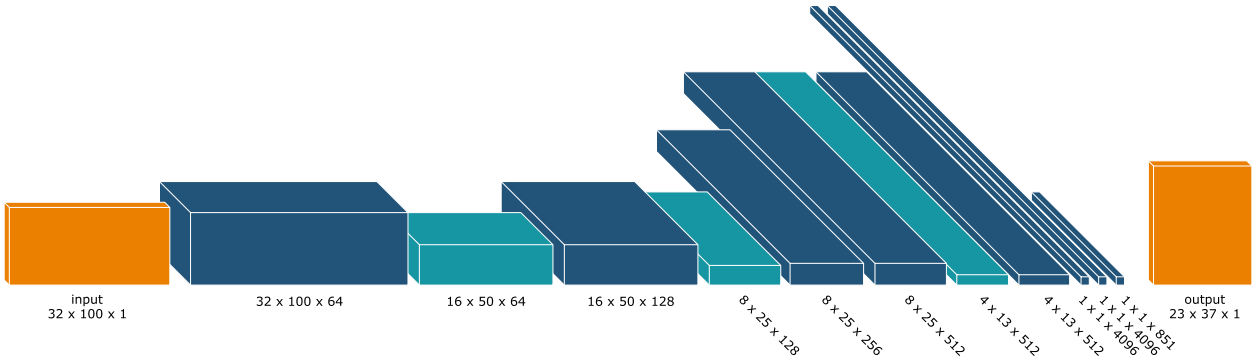

Figure 4. **CNN architecture** of Jaderberg et al. [3] with 23 independent classifiers as output. Each classifier predicts a character at each position of the word.

ages, we took a subset of 32,800 images that we modified for fine-tuning. For our main experiment (distractors+style transfer) 32,800 generated map word images were used with the style transfer as described in Section 2. 10% of this data was kept for validation. In the fine-tuning, we used a batch size of 450. The learning rate was set to 0.00005. We also generated 32,800 map word images with distractors only and without style transfer. To avoid overfitting we added a dropout layer. The dropout layer was added before the last classification layer. We used a dropout rate of 0.5.

Figures 5 and 6 show the objective and the error function for (1) data set with map distractors only and (2) final data set with map distractor + style transfer. The error can be interpreted as the number of wrong predicted characters in a word. Figure 5 shows the error function during fine-tuning with our new synthetic map text dataset with style transfer. The error is significantly decreasing during the first few epochs until we reach a minimum error of 1.6 after 80 epochs.

Figure 5 shows the error function during fine-tuning with word images with text distractors only and without style transfer. The error also decreases, but the minimum error that is reached after about 60 epochs is slightly higher than in Figure 5.

6. Evaluation

For final evaluation of text recognition on historical maps we used the 80th epoch of the fine-tuned CNN. As the test set we used the map dataset of Weinman [13]. 31 annotated maps are provided. 4 of those 31 maps were used as style images to produce the word images in map style. Thus we excluded these 4 maps from our testing data set. The remaining 27 maps included 11,463 words in total for evaluation. We evaluated the word accuracy w_{acc}, which describes the percentage of completely correct predicted words, as well as the character accuracy $char_{acc}$. There are various types of faults that can be made like inserting a charac-

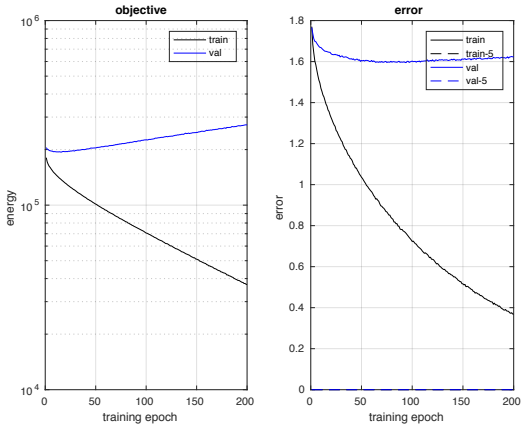

Figure 5. **Objective and error function during fine-tuning** using our generated synthetic map dataset *(distractors + style transfer)*

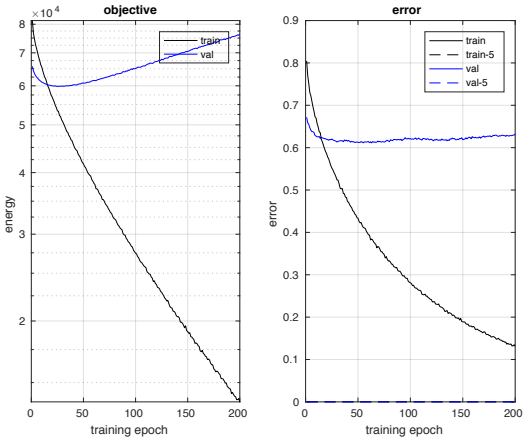

Figure 6. **Objective and error function during fine-tuning** using our generated synthetic map data set but without neural style transfer (*distractors*)

Figure 7. **Empirical Results** showing the text recognition results based on the fine-tuned CNN using synthetic map training data. Results for map D0042-1070007.

ter, removing a character or replacing a character. Due to the networks architecture, the network predicts a character sequence of fixed length 23. To compute the character accuracy $char_{acc}$ we remove all characters predicted as "non character" and compare each character separately. The amount of correctly predicted characters is divided by the total number of characters given in the ground truth. For example if the correct word is "hello" and the predicted word is "hilo" we predicted two characters correctly. Thus the character accuracy is $char_{acc} = 2/5 = 0.4$.

Previous results using the original text recognition network of Jaderberg at al. [3] show a character accuracy of about 88%. However the word accuracy is much lower at 66%. This is not surprising since a single wrong character produces a completely wrong recognized word, the character accuracy might still be high if the word is long.

After fine-tuning with our new map data set *distractors + style transfer* word accuracy is improved by more than 2% and the character accuracy is improved by 1%. Just based on these results one cannot tell whether the improvement is mainly due to text *distractors* that have been added or due to the *distractors + style transfer*. To analyze how much impact each of these changes has, we fine-tuned the network using word images where just map distractors have been added. These were the same map distractors as for the new map data set *distractors + style transfer* just without transferring the map style. The character accuracy of *distractors* goes down by about 0.2% compared to *distractors + style transfer*. This shows that the majority of the improvement is due to added distractors. There is just a small positive impact of the style transfer. We have to keep in mind that the network works with gray scale images. Thus the similar color style that was produced by neural style transfer had almost no impact, but the training could profit from the similarity due to texture and shape that was produced by style transfer.

Figure 7 shows empirical results for text recognition on the map dataset [13]. There are still several wrongly predicted characters due to background clutter. Note in the 5th row of Figure 7 the word "Big" was recognized as "bug". This might be due to the CNN architecture that contains 23 independent character classifiers. Here the classifier trained on recognizing the second character predicted a "u" instead of an "i". If there is no sequence knowledge a "u" can be seen between "i" and "g". A recurrent neural network might have the ability to recognize those words better since it is able to incorporate the knowledge of the character sequence.

		Jaderberg [3]		(1) distractors		(2) distractors + style transfer	
map	number of words	word acc	char acc	word acc	char acc	word acc	char acc
D0006-0285025	87	0.4368	**0.8571**	**0.4943**	0.8525	0.4253	0.8416
D0042-1070001	258	0.4302	0.8065	0.4690	**0.8261**	**0.4845**	0.8226
D0042-1070002	251	0.2510	0.7255	0.3187	0.7498	**0.3625**	**0.7604**
D0042-1070004	373	0.1233	0.5679	0.1314	0.6112	**0.1609**	**0.6448**
D0042-1070005	216	0.3981	0.8145	**0.4537**	0.8325	0.4444	**0.8358**
D0042-1070006	460	0.3848	0.7980	0.4087	0.8151	**0.4130**	**0.8177**
D0042-1070007	322	0.3416	0.7833	**0.3851**	0.8015	**0.3851**	**0.8095**
D0042-1070009	301	0.1462	0.6305	**0.2126**	0.6686	0.2060	**0.6865**
D0042-1070010	218	0.3532	0.8245	**0.3991**	**0.8409**	0.3945	**0.8409**
D0042-1070012	214	0.3879	0.8232	**0.4579**	**0.8423**	0.4299	0.8246
D0042-1070013	128	0.1719	0.6995	**0.2734**	0.7557	0.2500	**0.7568**
D0042-1070015	415	0.2916	0.6613	0.3060	**0.6940**	**0.3108**	0.6932
D0079-0019007	135	0.2222	0.6316	0.2074	0.6111	**0.2741**	**0.6643**
D0089-5235001	367	0.6022	0.8643	**0.6294**	**0.8835**	0.6158	0.8699
D0090-5235001	391	0.7059	0.9261	0.7263	0.9340	**0.7340**	**0.9391**
D0117-5755018	864	0.8646	**0.9706**	0.**8808**	0.9687	0.8773	0.9650
D0117-5755024	1024	0.8525	0.9623	0.8828	0.9669	**0.8857**	**0.9670**
D0117-5755025	759	0.8656	0.9632	**0.8867**	**0.9669**	0.8788	0.9652
D0117-5755033	764	0.8743	0.9745	**0.8874**	**0.9753**	0.8796	**0.9753**
D0117-5755035	432	0.8194	0.9655	**0.8472**	**0.9663**	0.8333	0.9659
D0117-5755036	592	0.8733	0.9714	0.8818	0.9701	**0.8885**	**0.9706**
D5005-5028052	632	0.7579	0.9283	0.7753	0.9337	**0.7816**	**0.9370**
D5005-5028054	320	0.7438	0.9186	**0.7719**	0.9267	0.7594	**0.9312**
D5005-5028097	391	0.7596	0.9403	0.7647	0.9398	**0.7801**	**0.9477**
D5005-5028100	488	0.7971	0.9325	**0.8074**	0.9348	**0.8074**	**0.9409**
D5005-5028102	197	**0.8376**	0.9332	**0.8376**	**0.9341**	0.8223	**0.9341**
D5005-5028149	864	**0.7720**	**0.9446**	0.7697	0.9442	0.7708	0.9434
all maps	11463	0.6592	0.8877	0.6823	0.8965	**0.6829**	**0.8988**

Table 1. **Word accuracy and character accuracy** shown for the original CNN trained on the synthetic word dataset and results after (1) fine-tuning using the 60th epoch of the network that was fine-tuned using word images with *distractors* (2) fine-tuning using the 80th epoch of fine-tuned network with images with *distractors + neural style transfer*

7. Conclusion

We presented a promising new technique for data augmentation, that allows the creation of large amounts of training data matched to the style of a particular problem. This is especially important in the case of text recognition of historical documents, since training data is limited in this domain. Style transfer allows one to imitate the style of historical documents, and thus we can produce data with "historical" style synthetically. This may be relevant not only for text recognition on historical maps, but other application areas as well.

The CNN used in this work has 23 *independent* classifiers for characters. This allows unconstrained word prediction, but it may be less robust due to the high amount of flexibility. A dictionary based approach might be considered for future work as well as a recurrent neural network, that might incorporate sequence based knowledge.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No.:1526350.

References

[1] J. L. Feild and E. G. Learned-Miller. Improving open-vocabulary scene text recognition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 604–608. IEEE, 2013.

[2] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

[3] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *arXiv preprint arXiv:1412.1842*, 2014.

[4] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014.

[5] C.-Y. Lee and S. Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. *arXiv preprint arXiv:1603.03101*, 2016.

[6] H. Li and C. Shen. Reading car license plates using deep convolutional neural networks and lstms. *CoRR*, abs/1601.05610, 2016.

[7] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *Asian Conference on Computer Vision*, pages 770–783. Springer, 2010.

[8] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3538–3545. IEEE, 2012.

[9] B. Shi, X. Bai, and C. Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *CoRR*, abs/1507.05717, 2015.

[10] R. Smith, C. Gu, D.-S. Lee, H. Hu, R. Unnikrishnan, J. Ibarz, S. Arnoud, and S. Lin. *End-to-End Interpretation of the French Street Name Signs Dataset*, pages 411–426. Springer International Publishing, Cham, 2016.

[11] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, pages 1457–1464. IEEE, 2011.

[12] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3304–3308. IEEE, 2012.

[13] J. Weinman. Toponym recognition in historical maps by gazetteer alignment. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1044–1048. IEEE, 2013.