CLASSIFICATION OF SEVERELY OCCLUDED IMAGE SEQUENCES VIA CONVOLUTIONAL RECURRENT NEURAL NETWORKS

Jian Zheng, Yifan Wang, Xiaonan Zhang, Xiaohua Li

State University of New York at Binghamton
Department of ECE, Binghamton, NY 13902
{jzheng65, ywang338, xzhan167, xli}@binghamton.edu

ABSTRACT

Classifying severely occluded images is a challenging yet highly-needed task. In this paper, motivated by the fact that human being can exploit context information to assist learning, we apply convolutional recurrent neural network (CRNN) to attack this challenging problem. A CRNN architecture that integrates convolutional neural network (CNN) with long short-term memory (LSTM) is presented. Three new datasets with severely occluded images and context information are created. Extensive experiments are conducted to compare the performance of CRNN against conventional methods and human experimenters. The experiment results show that the CRNN outperforms both conventional methods and most of the human experimenters. This demonstrates that CRNN can effectively learn and exploit the unspecified context information among image sequences, and thus can be an effective approach to resolve the challenging problem of classifying severely occluded images.

Index Terms— Severe image occlusions, image sequence classification, convolutional recurrent neural network

1. INTRODUCTION

Past years have witnessed an explosion of deep learning research [1], represented by various convolutional neural networks (CNNs) for image/vision processing [2] and a variety of recurrent neural networks (RNNs) for sequential data processing [3]. Nevertheless, the majority of results are obtained over high quality data sources. Their performance usually degrades severely over practically distorted data, such as noisy speech signals with room reverberations and noisy images with occlusions. Much more research efforts are still needed to develop more robust deep learning techniques.

Consider the classification of distorted images. Distortions such as blur, noise, occlusion, etc., will degrade classification performance [4]. Severe occlusions, with which major or key areas of the images are blocked, are especially detrimental [5]. Classification of severely occluded images is highly needed in many real applications, e.g., when people block each other in a crowd, when rain/snow stains the cameras of self-driving vehicles, etc. However, the research of this issue has been very limited. The effects of image distortions on deep learning are studied in [6]. The results of [5] indicate that blur, noise, and occlusion will lead to a significant decrease in performance. Authors in [7] propose a deep neural network model for robust image classification, where blur and noise are considered. In [8], re-training and fine-tuning techniques are proposed to alleviate image blur and noise effects. Classification of severely occluded images remains largely an open and challenging task.

When recognizing severely occluded objects, human being tends to exploit a lot of context information to fill the missing parts, such as the relation among objects in the vision field, the common sense about the physical world, and the background or environmental knowledge. This is owing to the exceptional deep learning/reasoning capability of human being [9] [10] that current deep learning algorithms lack. In this paper, we mimic such human learning capability and address the challenging problem of classifying severely occluded images by applying convolutional recurrent neural networks (CRNN) to learn and exploit context information automatically. The proposed CRNN integrates CNN and long short-term memory (LSTM) together so as to use CNN for image spatial representation learning and use LSTM for context information modeling within the visual time series.

Among the various CNN models developed for image classification [11–15], some have been adapted to exploit context information [16]. Algorithms are also designed to make CNNs understand context or common sense more closely as human beings [17] [18]. However, most of such studies are conducted over high-quality image datasets instead of severely occluded images.

The integration of CNN and RNN (or LSTM) has been successfully applied in many other applications. For instance, long-term recurrent convolutional network is proposed for visual recognition and description in [19]. In [20], the authors propose a CRNN model to learn spatial dependencies for better image representation. In visual question answering (VQA) [21], CNN is used for image embedding while LSTM is used for question embedding. Joint CNN and RNN is also used in image captioning [22], video description [23], video classification [24] and action recognition [25] [26]. In contrast, the new CRNN method proposed in this paper focuses on using simpler network configuration to attack the challenging problem of classifying severely occluded image sequences.

In this paper, we first present the general architecture of the proposed CRNN method, in which the LSTM exploits the extracted spatial representations with CNN to learn different patterns among image sequences for further classification. To evaluate its performance, we create three datasets of severely occluded images and conduct extensive experiments. In particular, several human subject experiments are performed to compare the performance of human beings against that of machine learning.

The remainder of this paper is organized as follows. In Section 2, we introduce the CRNN architecture. In section 3 and 4, we present the experiment settings and analyze the experiment results. Conclusions are given in Section 5.

This work is supported by NSF via grant CNS-1443885.

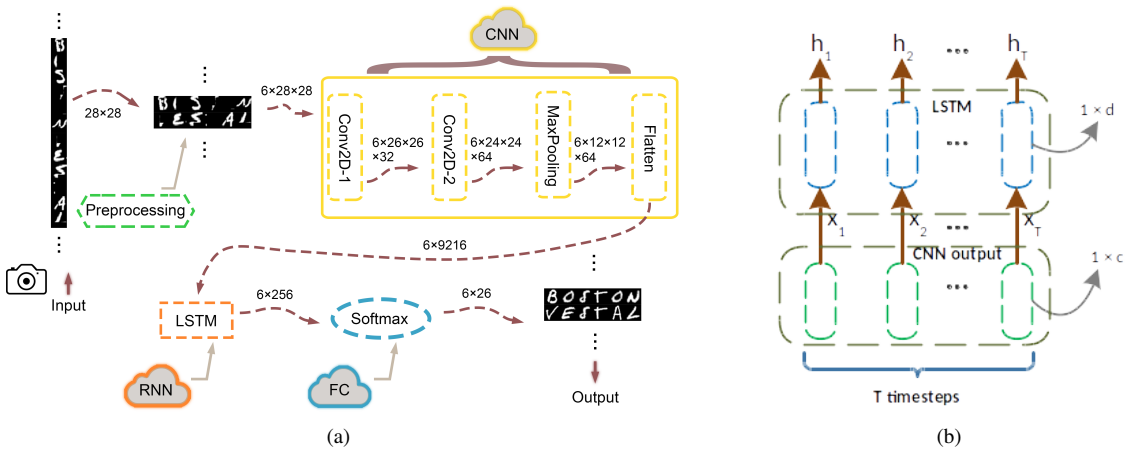

Fig. 1: (a) The general architecture of CRNN based image sequence classifier. (b) The integration of CNN unit with LSTM unit.

2. DEEP CLASSIFICATION ARCHITECTURE

The proposed CRNN classification architecture is illustrated in Fig. 1(a). For better illustration, some occluded image examples created from the EMNIST dataset and the corresponding image dimensions are shown in Fig. 1(a). The CRNN architecture consists of four units: a data preprocessing unit, a CNN unit for spatial feature representation learning, an LSTM unit for visual time series modeling, and a softmax based classifier for image classification.

2.1. Data preprocessing

As shown in Fig. 1(a), firstly, the data preprocessing unit converts input images to image sequences of fixed length T (i.e., T images per sequence). Without loss of generality, assume there are I images to be classified and the size of each image is $W_1 \times H_1 \times D_1$. The input is thus a tensor with dimension $I \times W_1 \times H_1 \times D_1$. To facilitate the LSTM unit to model temporary dependencies among the images in later stages, the tensor is converted into a sequence of tensors of dimension $T \times W_1 \times H_1 \times D_1$, which is denoted as $\mathbf{X}_i = \{\mathbf{x}_{i,1}, \cdots, \mathbf{x}_{i,T}\}$. Each $\mathbf{x}_{i,t}$ is an image. With preprocessing, the I input images are rearranged into L image sequences, i.e., $i = 1, \cdots, L$, where $L = \lceil I/T \rceil$. For instance, if the sequence length is $T = 6$ and the input images have size $28 \times 28 \times 1$, then the dimension of each image sequence will become $6 \times 28 \times 28 \times 1$.

2.2. CNN for image sequences

The CNN unit is then applied on input image sequences to learn the spatial feature representations in each image. The CNN unit consists of two consecutive convolutional layers Conv2D-1 and Conv2D-2, followed by a max-pooling layer with pool size (p, q) for spatial dimensionality reduction.

In the CNN unit, each image $\mathbf{x}_{i,t}$ is converted into a tensor $\mathbf{z}_{i,t}$,

$$\mathbf{z}_{i,t} = f(\mathbf{x}_{i,t}; \{\mathbf{W}_x\}) \tag{1}$$

where $\{\mathbf{W}_x\}$ denotes all the weighting coefficients. Let the kernel size be (k, m) and the number of filters for the two layers be D_2 and D_3, respectively. With the input size $W_1 \times H_1 \times D_1$ for each $\mathbf{x}_{i,t}$, the two successive convolutional layers output two feature maps of sizes $W_2 \times H_2 \times D_2$ and $W_3 \times H_3 \times D_3$, respectively. The max-pooling layer outputs the feature map $\mathbf{z}_{i,t}$ for each input image with size $W_4 \times H_4 \times D_3$. After the max-pooling layer, there is a flatten layer to combine each sequence of T feature maps into a two dimensional matrix $\mathbf{Z}_i = \{\mathbf{z}_{i,1}, \cdots, \mathbf{z}_{i,T}\}$ with dimension $T \times (W_4 \times H_4 \times D_3)$, which is the input to the subsequent LSTM unit.

2.3. LSTM for image sequences

RNNs are neural networks that are popular in modeling context dependencies among sequential data. With the learned image representations through CNN unit, an LSTM unit is built here in order to model the spatial context dependencies among the image sequences. Specifically, a variant of the LSTM architecture is applied here to learn the regular patterns inside the image series.

Given the input time series $\mathbf{Z}_i = \{\mathbf{z}_{i,1}, \cdots, \mathbf{z}_{i,T}\}$ which are the output of the flatten layer, the memory cells in the LSTM layer map the input time series to a representation series $\mathbf{H}_i = \{\mathbf{h}_{i,1}, \cdots, \mathbf{h}_{i,T}\}$,

$$\mathbf{h}_{i,t} = f(\mathbf{h}_{i,t-1}, \mathbf{z}_{i,t}; \{\mathbf{W}_z\}) \tag{2}$$

where $\{\mathbf{W}_z\}$ denotes all the weighting coefficients of the LSTM unit. Fig. 1(b) illustrates how the CNN unit is connected with the LSTM unit. The output of the CNN unit with dimension $T \times c$ is fed into the LSTM unit as input, where c equals to $W_4 \times H_4 \times D_3$. There are d hidden units in LSTM. Therefore, the size of the LSTM unit output $\mathbf{H}_i$ is $T \times d$, and each $\mathbf{h}_{i,t}$ has dimension d.

The LSTM output $\mathbf{H}_i$ is fed into the softmax classifier. The classification output is

$$\hat{\mathbf{y}}_i = f(\mathbf{H}_i; \{\mathbf{W}_h\}) \tag{3}$$

where $\{\mathbf{W}_h\}$ denotes all the weighting coefficients in this stage. The size of $\hat{\mathbf{y}}_i$ depends on the number of classes in the data. For instance, if the input images are generated from the EMNIST dataset, the occluded images are classified into 26 classes. In this case, each $\hat{\mathbf{y}}_i$ has dimension $T \times 26$. The training is conducted via minimizing the reconstruction loss, i.e.,

$$\theta = \arg\min_{\theta} \frac{1}{L} \sum_{i=1}^{L} \mathcal{L}(\mathbf{y}_i, \hat{\mathbf{y}}_i) \tag{4}$$

over the training dataset $(\mathbf{X}_i, \mathbf{y}_i)$, $i = 1, \cdots, L$, where $\theta = \{\mathbf{W}_x, \mathbf{W}_z, \mathbf{W}_h\}$ denotes the weights.

3. EXPERIMENT SETTINGS

3.1. Data preparations

Firstly, we create three occluded image datasets based on MNIST dataset [27], EMNIST dataset [28], and CIFAR-10 dataset [29], respectively. Occlusion effects of various levels and in random positions of an image are designed and generated with MATLAB. Specifically, five different levels of occlusions are created in proportion to the ratio of 1:1:1:1:1, according to different widths of the borders of black rectangles. The wider the borders are, the more the image will be blocked, thus the more severe the occlusions will be. To better illustrate different occlusion levels, Fig. 2 shows some typical occluded images, in which the occlusion level is gradually increased from top to bottom. It can be seen clearly that less occluded can be recognized easily. However, it is getting harder and harder to recognize those occluded images with increasing occlusion levels from top to bottom. For instance, we can hardly identify those images with occlusions of level 4 and level 5 without the aid of any contextual information.

To simulate the severely occluded images that exist in many real situations, we generate occluded image sequences by randomly selecting images of different occlusion levels. Therefore, in a randomly picked image sequence, some images may be slightly occluded, while some others may be severely occluded. Then, in order to incorporate contextual information into image sequences, we arrange the occluded images into image sequences with certain regular patterns. For handwritten digits, image sequences with patterns such as 0-1-2-3-4 and 1-3-5-7-9 are coined and used in experiments. Linear programming (LP) is applied to maximize the number of such image sequences, resulting in 13135 image sequences in total, each with 5 images. For handwritten characters, the images are used to spell American and Canadian city names, such as B-O-S-T-O-N and M-I-L-T-O-N. Similarly, LP is applied to maximize the number of image sequences. As a result, 14611 different 6-letter image sequences are created. For the CIFAR-10 dataset, since CIFAR-10 images are labeled from 0 to 9, image sequences are created based on label patterns similar to the MNIST dataset, which results in 12000 image sequences with 5 images in each image sequence.

In Fig. 2(a), Fig. 2(c), and Fig. 2(e), the image sequences are generated with certain patterns. Therefore, a well-trained human is able to classify most of the occluded image sequences correctly. In contrast, those image sequences in Fig. 2(b), Fig. 2(d), and Fig. 2(f) are created without particular patterns by simply selecting images randomly, making it difficult to classify them without referring to any contextual information.

3.2. Experiment setup

We use 5-image sequences or 6-image sequences in the experiments, i.e., $T = 5$ in the MNIST and CIFAR-10 based experiments and $T = 6$ in the EMNIST based experiments. The other settings for the experiments are the same. In the two convolutional layers, the kernel size (k, m) is (3,3), and the number of filters are $D_2 = 32$ and $D_3 = 64$. Max-pooling with pooling size $(p, q) = (2, 2)$ is deployed. Zero padding and rectified linear unit (ReLU) activation function are applied. Dropout is used in both the CNN unit and the LSTM unit. The output size of the LSTM unit is $d = 256$. A fully connected layer of dimension 10 or 26 is attached to the LSTM layer for image classification with softmax as the activation function.

Our proposed **CRNN** architecture uses a single max-pooling layer. Considering that similar but simpler networks can be realized by applying strides in the convolutional layers, we also experiment with two CRNN variations, i.e., **CRNN-2-S** and **CRNN-4-S**. The CRNN-2-S applies a stride of 2 in the two convolutional layers to replace the max-pooling layer. In CRNN-4-S, the settings of the two convolutional layers remain the same as CRNN, but two extra convolutional layers with a stride of 2 are added after each of the original convolutional layers and the max-pooling layer is removed. The rationale for CRNN-2-S is for network simplicity, while the rationale for CRNN-4-S is for deeper network depth. As comparison, we experiment three CNN models without LSTM: the conventional **CNN** with a max-pooling layer and two similar variations without max-pooling layers, which we call **CNN-2-S** and **CNN-4-S**.

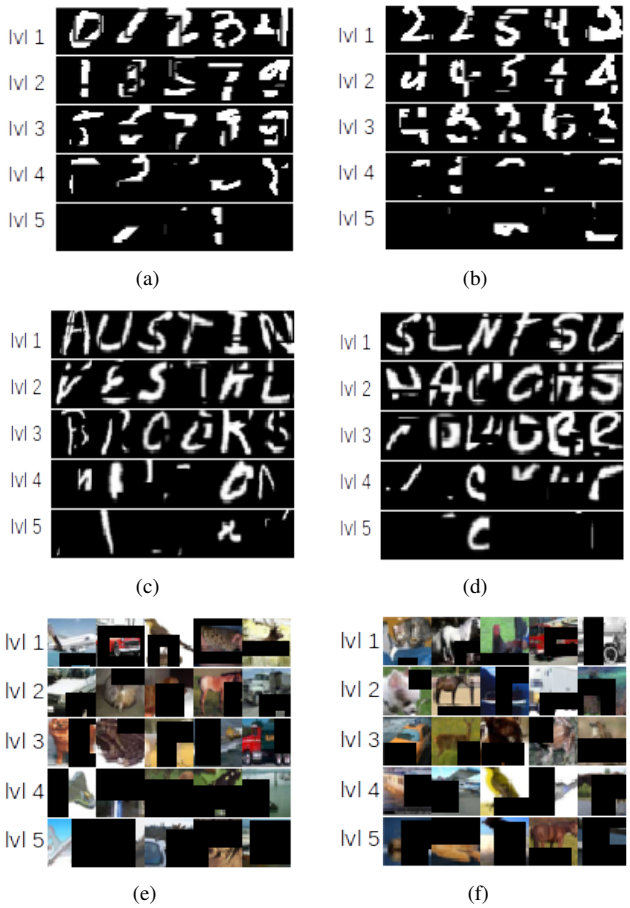

Fig. 2: Examples of occluded image sequences: (a) Digits with regular pattern; (b) Digits without regular pattern; (c) Characters forming city names; (d) Random characters; (e) CIFAR-10 image sequences with regular patterns; (f) CIFAR-10 image sequences without regular patterns.

4. EXPERIMENT RESULTS

Fig. 3 shows three examples of the classification results of the proposed CRNN method, where the classification results are marked on the top of the images. It can be clearly seen that since some images are severely occluded, such as '6', 'O' and 'N', it is hard to classify them correctly without exploiting context information. Our proposed method can successfully learn the regular patterns within the image sequences and use the learned patterns and the less occluded images to infer the severely occluded images.

Table 1: Performances (%) with regular patterns

Dataset	MNIST	EMNIST	CIFAR10
CNN-2-S	86.22	86.01	44.39
CNN-4-S	88.02	87.26	42.12
CNN	89.44	88.90	54.99
CRNN-2-S	98.27	97.95	89.11
CRNN-4-S	98.15	97.90	90.18
CRNN	98.33	98.14	90.36

Table 2: Performances (%) without regular patterns

Dataset	MNIST	EMNIST	CIFAR10
CNN-2-S	86.37	86.10	44.21
CNN-4-S	88.28	87.81	41.40
CNN	89.37	89.22	53.88
CRNN-2-S	86.68	86.66	45.64
CRNN-4-S	88.15	88.16	48.21
CRNN	87.89	87.81	50.62

Table 3: CRNN performance with regular patterns

Seq_len	$T=5$	$T=10$	$T=15$	$T=20$	$T=25$
MNIST	98.33	98.11	98.23	98.10	98.06
CIFAR	90.36	88.29	88.11	88.19	87.93
Seq_len	$T=6$	$T=12$	$T=18$	$T=24$	$T=30$
EMNIST	98.14	97.83	97.76	97.78	97.62

Table 4: CRNN versus human with regular patterns

Dataset	MNIST	EMNIST	CIFAR10
Non-Expert	93.96	92.52	71.11
Expert	100.00	99.26	97.78
CRNN	98.33	98.14	90.36

Table 5: CRNN versus human without regular patterns

Dataset	MNIST	EMNIST	CIFAR10
Non-Expert	75.56	71.79	50.19
Expert	81.00	78.89	71.67
CRNN	87.89	87.81	50.62

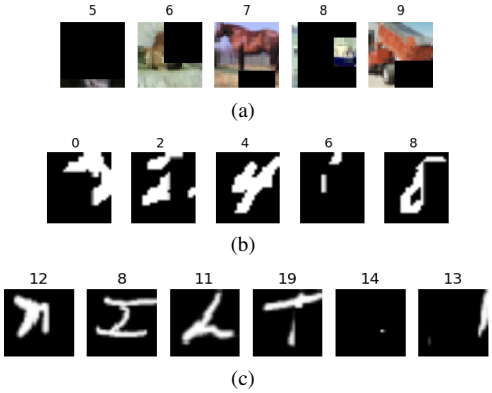

Fig. 3: Three examples of CRNN classification results: (a) Correct classification of the image sequence: 5-6-7-8-9. (b) Correct classification of the digit sequence: 0-2-4-6-8. (c) Correct classification of the city name: M-I-L-T-O-N.

Table 1 and Table 2 show the classification performance of the three CNN methods and the three CRNN methods. It can be seen clearly from Table 1 and Table 2 that occlusions severely degrade the performance of the three CNN methods on each dataset. For example, the classification accuracy for MNIST dataset degrades from the state-of-the-art 99.79% [30] to the level of 86-89%. Similarly, for CIFAR-10 images, the accuracy even drops down to 42.12%. Since the CNN methods cannot exploit context information, the performance is similar for data with or without regular patterns. The three CRNN methods have similar performance as CNN methods with randomized image sequences, but outperform the latter when the image data have regular patterns. The classification accuracy is boosted greatly to 98.33%, 98.14% and 90.36% for the three datasets, respectively. In addition, the proposed CRNN method slightly outperforms the other two CRNN methods that replace max pooling with larger strides in this case.

To study the impact of sequence length T on occluded image sequence classification, extensive experiments with different sequence lengths have been conducted. Note that only image sequences with regular patterns are used. Table 3 shows that the proposed method is fairly robust to sequence lengths. The slight performance degradation might be due to the fact that while increasing lengths of regular patterns, the number of training image sequences is reduced with limited amount of image data.

4.1. Human learning versus machine learning

To compare the performance of the proposed CRNN method with that of human beings when classifying severely occluded images, six human subject experiments are designed and conducted. The first three are conducted by a group of undergraduate students. Only a rather brief explanation of the experiments is provided to them. Without too much training, we call those students **Non-expert**. In contrast, the other three are conducted by some well-trained graduate students, whom we call **Expert**. Each student is asked to classify a set of occluded image sequences, some of which have regular patterns while some are purely random image sequences.

The human experiment results are shown in Table 4 and Table 5, from which it can be seen that human beings can exploit regular patterns even without too much training. Meanwhile, it indicates that the **Non-expert** performance can be much better than the CNNs methods but is inferior to the proposed CRNN method when dealing with patterned image sequences. In contrast, the **Expert** can achieve better performance than the proposed CRNN method in classifying images with regular patterns. Both the CNN and the CRNN methods outperform all the human subjects in classifying images without regular patterns. The fact that the proposed CRNN model outperforms non-experts and has similar performance as the experts verifies its great potential for classifying severely occluded images in practical applications.

5. CONCLUSIONS

We propose to apply convolutional recurrent neural network (CRNN) that integrates convolutional neural network (CNN) with recurrent neural network (RNN) for classifying severely occluded image sequences. Extensive computer and human subject experiments are conducted based on three datasets of severely occluded images we created. Experiment results validate the effectiveness and robustness of proposed CRNN in learning unspecified regular patterns inside the image sequences automatically and making use of such knowledge to facilitate classifying severely occluded images. The proposed CRNN method outperforms both conventional CNNs and non-expert human subjects.

6. REFERENCES

[1] Yoshua Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[2] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[3] LR Medsker and LC Jain, "Recurrent neural networks," *Design and Applications*, 2001.

[4] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[5] Samil Karahan, Merve Kilinc Yildirum, Kadir Kirtac, Ferhat Sukru Rende, Gultekin Butun, and Hazim Kemal Ekenel, "How image degradations affect deep cnn-based face recognition?," in *Biometrics Special Interest Group (BIOSIG), 2016 International Conference of the*. IEEE, 2016, pp. 1–5.

[6] Samuel Dodge and Lina Karam, "Understanding how image quality affects deep neural networks," in *Quality of Multimedia Experience (QoMEX), 2016 Eighth International Conference on*. IEEE, 2016, pp. 1–6.

[7] Samuel Dodge and Lina Karam, "Quality resilient deep neural networks," *arXiv preprint arXiv:1703.08119*, 2017.

[8] Yiren Zhou, Sibo Song, and Ngai-Man Cheung, "On classification of distorted images with deep convolutional neural networks," *arXiv preprint arXiv:1701.01924*, 2017.

[9] J. Friedenberg and G. Silverman, "Cognitive science: An introduction to the science of the mind," *Thousand Oaks, CA: Sage*, 2006.

[10] Stellan Ohlsson, "Deep learning: How the mind overides experience," *Cambridge University Press*.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[12] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[14] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[15] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3642–3649.

[16] Pedro H. O. Pinheiro, Ronan Collobert, and Piotr Dollár, "Learning to segment object candidates," *CoRR*, vol. abs/1506.06204, 2015.

[17] Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun, "Tracking the world state with recurrent entity networks," *CoRR*, vol. abs/1612.03969, 2016.

[18] Wenbin Li, Seyedmajid Azimi, Ales Leonardis, and Mario Fritz, "To fall or not to fall: A visual approach to physical stability prediction," *CoRR*, vol. abs/1604.00066, 2016.

[19] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[20] Zhen Zuo, Bing Shuai, Gang Wang, Xiao Liu, Xingxing Wang, Bing Wang, and Yushi Chen, "Convolutional recurrent neural networks: Learning spatial dependencies for image representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 18–26.

[21] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," *arXiv preprint arXiv:1606.01847*, 2016.

[22] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.

[23] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Kazuhiro Sumi, John R Hershey, and Tim K Marks, "Attention-based multimodal fusion for video description," *arXiv preprint arXiv:1701.03126*, 2017.

[24] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4694–4702.

[25] Basura Fernando, Efstratios Gavves, José Oramas, Amir Ghodrati, and Tinne Tuytelaars, "Rank pooling for action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 773–787, 2017.

[26] Tian Lan, Yuke Zhu, Amir Roshan Zamir, and Silvio Savarese, "Action recognition by hierarchical mid-level action elements," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4552–4560.

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[28] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik, "Emnist: an extension of mnist to handwritten letters," *arXiv preprint arXiv:1702.05373*, 2017.

[29] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," 2009.

[30] Li Wan, Matthew Zeiler, Sixin Zhang, Yann L Cun, and Rob Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 1058–1066.