AN EFFICIENT DEEP RESIDUAL-INCEPTION NETWORK FOR MULTIMEDIA CLASSIFICATION

Samira Pouyanfar and Shu-Ching Chen

School of Computing and
Information Sciences
Florida International University
Miami, FL 33199, USA
{spouy001, chens}@cs.fiu.edu

Mei-Ling Shyu

Department of Electrical and
Computer Engineering
University of Miami
Coral Gables, FL 33124, USA
shyu@miami.edu

ABSTRACT

Deep learning has led to many breakthroughs in machine perception and data mining. Although there are many substantial advances of deep learning in the applications of image recognition and natural language processing, very few work has been done in video analysis and semantic event detection. Very deep inception and residual networks have yielded promising results in the 2014 and 2015 ILSVRC challenges, respectively. Now the question is whether these architectures are applicable to and computationally reasonable in a variety of multimedia datasets. To answer this question, an efficient and lightweight deep convolutional network is proposed in this paper. This network is carefully designed to decrease the depth and width of the state-of-the-art networks while maintaining the high-performance. The proposed deep network includes the traditional convolutional architecture in conjunction with residual connections and very light inception modules. Experimental results demonstrate that the proposed network not only accelerates the training procedure, but also improves the performance in different multimedia classification tasks.

Index Terms— Deep learning, Video event detection, Multimedia classification, Residual-Inception, Convolutional neural network

1. INTRODUCTION

Multimedia data has become pervasive in the recent decade with the advent of new technologies, powerful hardware, and larger datasets [1, 2]. Video analysis is one of the most challenging and time-consuming processes in multimedia big data due to its large capacity, multi-modality, and complexity compared to textual or single-modality data. One of the important and challenging topics in video processing is video event detection which extracts meaningful semantic information from the video data [3, 4]. These information can be further used for efficient video searching and retrieval.

Deep learning is an emerging topic but originated from traditional neural networks, which is widely used in Artificial Intelligence (AI) and Machine Learning (ML). The high capability of deep learning in a wide range of applications, especially the visual data, motivates us to integrate it with the application of video and image classification. The recent advancement in image recognition was obtained with deeper and wider networks [5, 6]. One example is the residual architecture which reaches to 152 layers, almost 8 times deeper than GoogLeNet and VGG nets [7]. As the network grows in depth (or sometimes in width [6]), the features can be enriched and more high-level features can be extracted compared to those from early layers.

Now the question is whether deeper networks always generate better performance. In other words, does stacking more layers lead to better learning? In addition, are extremely deep networks computationally reasonable for and applicable to different applications? Based on several experiments reported in [5, 8], network improvements and better learning are not as easy as stacking more layers and a careful design is indispensable. The first problem driven by the depth increase is called "vanishing" gradients [8, 9], in which the network cannot be trained with regards to the gradient based algorithms like back propagations and the convergence is prevented from the early layers. There have been several solutions in the literature to address this problem by using rectified linear activation instead of common activation functions (e.g., sigmoid or tanh) [10] and normalization layers [11]. Another issue is when the training accuracy is saturated and suddenly starts to degrade, which is not due to over-fitting. One solution to this problem is addressed in [5] using Residual Learning.

To address the aforementioned problems, in this paper, we propose a new deep learning architecture based on the traditional Convolutional Neural Networks (CNNs) integrated into two levels of Residual-Inception combination. This architecture is successfully tested on two different multimedia datasets. Specifically, it is applied on a video event detection task containing natural disaster events. The overall Deep

978-1-5090-6067-2/17/$31.00

Residual-Inception network improves the losses compared to the most recent deep learning architectures and also significantly decreases the computational costs.

The rest of the paper is organized as follows. First, the state-of-the art research in deep learning is presented in section 2. Section 3 provides the details of the proposed deep learning network. In section 4, the experimental results on two benchmark multimedia datasets are discussed. Lastly, the paper is concluded in section 5.

2. RELATED WORK

In recent years, deep learning which is originated from conventional artificial neural networks has attracted increasing academic and industry attention. Currently, it has been utilized in a wide range of areas such as image recognition, natural language processing, artificial intelligence, to name a few. To date, several deep architectures are presented in the literature, including Deep Belief Networks (DBNs), Deep Boltzmann Machines (DBMs), Deep auto-encoder, and Convolutional Neural Networks (CNNs) [12]. Among them, CNNs have been commonly used and have shown promising results especially in image recognition and computer vision [6, 13, 14].

AlexNet [13], the first attempt of applying a deep convolutional network on image processing, has made CNNs very popular. The network is initially trained on more than one million images in the 2012 ImageNet Large Scale Visual Recognition (ILSVRC) contest to classify them into the 1000 classes and significantly increased the performance compared to the state-of-the-art approaches.

GoogLeNet [6], a deeper and wider convolutional network is presented in the 2014 ILSVRC by Google. The original network contains 22 layers and utilizes the computing resources inside the network in an efficient way. The GoogLeNet design is also known as "Inception", which tries to introduce more sparsity and more optimal locality into the convolutional layers. By applying GoogLeNet on the ImageNet dataset, the accuracy is improved from 22.6% in the 2013 ILSVRC to 43.9% in the 2014 ILSVRC.

Finally, Microsoft introduced a new deep architecture called "Deep Residual Learning" [5] which beats the human brain in Image Recognition on the ImageNet dataset. The residual network won the ILSVRC and COCO 2015 competitions on different tasks including ImageNet detection and localizations, as well as COCO segmentation and detection. It is extremely deeper than the state-of-the-art architectures and introduces residual mapping into the convolutional layers to decrease the vanishing gradient and degradation problems.

All the aforementioned networks, as well as other well-known deep learning architectures, have led image processing into a new stage where computers can compete with human experts in image classification. It is worth mentioning that ImageNet and other large-scale image datasets have also played a critical role in this advancement. However, very few work has addressed the event detection from videos using deep learning algorithms since working with videos is more challenging and time-consuming. Therefore, in this paper, a deep Residual-Inception network is proposed to detect specific events from videos.

The benefits of combining the Inception and Residual networks are presented in [15], in which the authors utilized residual connection inside the Inception module and designed a new module as Inception-ResNet. However, in our work, a two-level Inception and Residual framework is proposed which improves both accuracy and speed. Therefore, it sequentially applies an inception after a residual module and increases the ratio of convolutions and Residual-Inception blocks gradually. In addition, it only uses a few numbers of Residual and Inception stacks (33 layers) rather than a very deep stack of Inception-ResNet to overcome overfitting in very deep and wide networks such as the Inception and Residual networks.

3. DEEP RESIDUAL-INCEPTION NETWORK

The proposed network contains three main modules, including traditional convolutional neural networks, residual connections for training deep architectures, along with inception modules for retaining computational efficiency.

3.1. Convolutional Neural Network

In general, CNNs are very similar to conventional neural networks with three ideas: 1) local connectivity, 2) shared weights, and 3) spatial sub-sampling [16]. The main advantage of CNNs is that they include fewer parameters and are easier to train compared to fully connected layers. The CNNs architecture basically consists of several convolutional along with subsampling layers and optionally in conjunction with fully connected layers. In the proposed network, we extend the CNNs module to include the following layers:

- The Input layer holds the raw pixels of the image with $m \times m \times r$ (e.g., $255 \times 255 \times 3$), where m is the image height and width, and r is the channel number (e.g., $r = 3$ for RGB images).

- The Convolutional layer includes a number of feature maps where the input data is convoluted with linear filters and then a nonlinear activation function f is applied. Each convolutional layer has k filters with size $n \times n \times q$ which represents the size of the locally connected regions in the image. Here, n is smaller than the dimension of the image, and q is either the same as or smaller than r (the number of channels) and it may vary for each filter. After convolving each filter with the image, k feature maps x^k of size $m - n + 1$ are generated (given in Equation (1)), where W^k and b^k are weights

and bias of the filters, respectively, and x^{k-1} is the input from the previous layer.

$$x^k = f((W^k * x^{k-1}) + b^k). \quad (1)$$

- Batch normalization is used after each convolutional layer in order to further accelerate the training set, as well as to reduce the gradients' dependencies to avoid the risk of overfitting and divergence [11].
- Rectified Linear Unit (ReLu) layer is the most common activation function ($f(x) = max(0, x)$) recently used for the output of convolutional neurons. It increases the nonlinearity and avoids network saturation [10].
- Pooling is a nonlinear downsampling which reduces the size of feature maps and brings more sparseness and robustness to the network. It also reduces the computational overhead and avoids overfitting. Each feature map is subsampled with a $p \times p$ (e.g., $p = 2$ for small images or $p = 5$ for high resolution images) downsampling operation (e.g., max, mean, etc.) as shown in Equation (2), where β represents multiplicative bias and $down(.)$ is a subsampling operation.

$$x^k = f(\beta^k down(x^{k-1}) + b^k). \quad (2)$$

3.2. Residual Module

The residuals are essential for very deep network to avoid the degradation problem. Suppose $H(x)$ is an underlying mapping to be fit by few neighbor layers, where x is the first input. Based on the report in [17], several nonlinear layers are capable of asymptotically approximating complicated functions. Therefore, instead of approximating $H(x)$ using the neighbor layers, a residual function $F(x) := H(x) - x$ will be approximated by these layers. Hence, a residual block (as shown in Figure 1) is defined as follows.

$$y^k = F(x^k, W^k) + x^k, \quad (3)$$

where x^k and y^k are the input and output vectors and F represents the residual mapping and the connection $(F + x)$ is performed by an element-wise addition.

In our view, utilizing residual networks helps the network to learn both weights and depths at the same time. In addition, we ensure the new layer $(N + 1)$ is learning something new by providing the output of the previous layer (N) without any modification to the output of the current layer $(N + 1)$. This technique handles both vanishing gradient and degradation problems in very deep networks.

3.3. Inception Module

The inception module significantly improves the computational efficiency while scaling up the network. This module heavily utilizes NIN [18] in its internal architecture for two reasons: (1) to reduce the input dimension and eliminate the computational bottlenecks and (2) to increase not only the network depth, but also its width to improve the overall performance. In other words, since a bigger size means a larger number of parameters, which causes overfitting in deep networks, leveraging sparsity even inside the convolutions leads to better results. Therefore, the filter-level sparsity blocks are introduced in the inception module. The filter sizes are 1×1, 3×3, and 5×5. All layers along with their output filter banks are combined and concatenated into one output vector. In addition, pooling is added in each inception since it is essential for convolutional networks. To further compress the network and reduce the dimension, 1×1 convolutional layers are added before each expensive convolutions. One sample of this module is used in our network as shown in Figure 2.

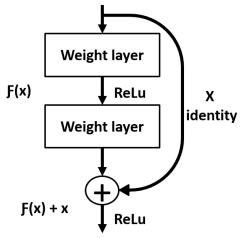


Fig. 1. A residual building block [5]

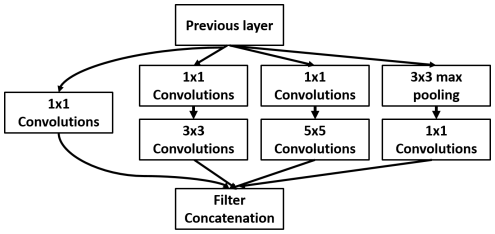


Fig. 2. An Inception building block [6]

3.4. Network Architecture

To handle the issue of overfitting, vanishing gradient, and network saturation problems, we study the combination of residual and inception modules. As the first layers generate low-level abstraction while the higher layers provide more high-level features from the data, the proposed network starts with a light version of each module and the ratio of convolutions and Residual-Inception blocks are gradually increased. The proposed network utilizes a few numbers of Residual-Inception stacks rather than very deep stacks of each single module. It starts with the traditional CNNs along with a lighter version of residual in conjunction with an inception module. Then, an increased dimension of Residual-Inception is added on top of the previous layer. At the end of the last

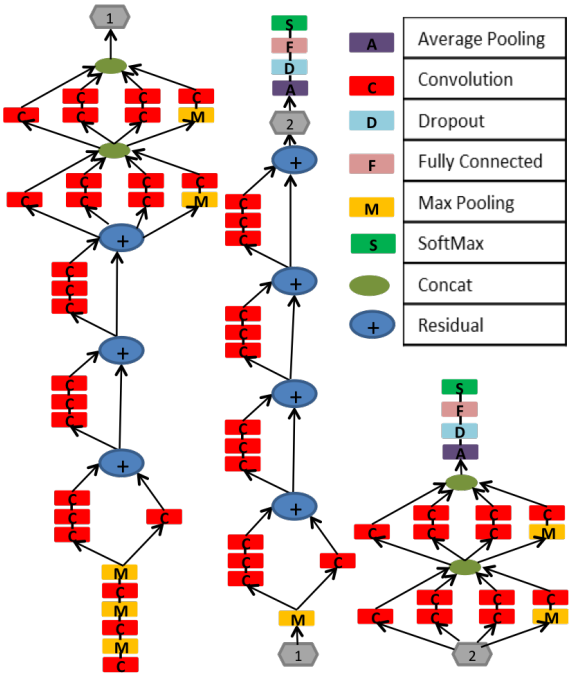


Fig. 3. The proposed deep Residual-Inception network

residual block, an average pooling, a dropout, a fully connected layer, and softmax are added to generate the final classification results. This block is also added to the end of the last inception block groups. In this case, we can evaluate which module generates smaller losses in each training step. Figure 3 depicts a schematic view of the proposed architecture. Table 1 also shows the detailed architecture. In residual blocks, downsampling is performed directly by the convolutional layers using the stride of 2. ReLu is used as an activation function in all convolutions including those inside residual and inception modules. However, it is removed after each element-wise addition operation [19]. We use dropout [20] after average pooling to avoid overfitting. The total number of layers is 33 including 3 CNNs, 21 residual layers, 8 inception layers, and 1 fully connected layer. This network is designed efficiently, which can be run even on devices with limited resources.

4. EXPERIMENTS

4.1. Data Sets

Two datasets are selected to evaluate the proposed network. As mentioned earlier, limited work has been done to handle video event detection using deep learning. Thus, we selected a video dataset containing natural disaster information as described in [21]. This dataset contains almost 7000 video shots from YouTube with skewed distributions and the average P/N ratio is 0.051. It also contains seven disaster classes including

Table 1. Deep Residual-Inception architecture

#	layer	output size	#	layer	output size
1	C	$7 \times 7, 32/2$	8	inca	$28 \times 28 \times 256$
2	M	$3 \times 3/2$	9	incb	$28 \times 28 \times 480$
3	C	$7 \times 7, 64/2$	10	M	$3 \times 3/2$
4	M	$3 \times 3/2$	11	res	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} * 4$
5	C	$3 \times 3, 120/2$	12	inca	$7 \times 7 \times 832$
6	M	$3 \times 3/2$	13	incb	$7 \times 7 \times 1024$
7	res	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} * 3$			
14	A	7×7, avg pool	16	F	$1 \times 1 \times 10$ (8)
15	D	$1 \times 1 \times 1024$	17	S	$1 \times 1 \times 10$ (8)

flood, damage, fire, mud-rock, tornado, lightening, and snow. We divided this dataset into 60% training and 40% testing.

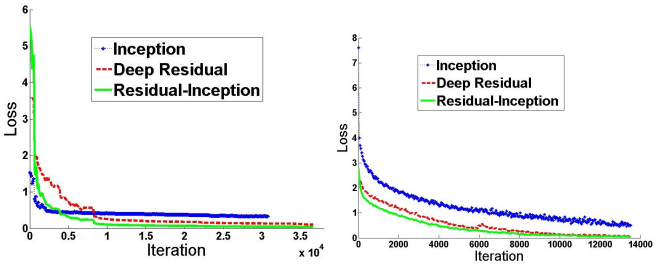
In addition, we conducted more experiments on CIFAR-10, a large public dataset consisting of 60,000 32×32 color images in 10 classes. It is divided into 50k training and 10k testing images. The main focus is to show the functionality of the proposed network on a large dataset compared to well-known deep learning algorithms in different training iterations and times. However, we do not intend to push the state-of-the-art results which also utilized other techniques such as augmentation, ensemble, randomized input order, and sampling methodologies [6]. Therefore, a simple architecture of the proposed network, as well as the ones of the comparison benchmarks are used in these experiments.

4.2. Experiment Setups and Results

For the disaster dataset, the network input is 224×224 images and channel-wise (pixel) mean is used instead of mean image [7]. The learning rate is set to 0.0001 to train the network slowly and avoid overfitting. The input of the network for the CIFAR-10 is 32×32 images with subtracting the mean pixel. We start with a base learning rate of 0.01 and divide it by 10 every 20k iterations. For both datasets, SGD with a momentum of 0.9 and weight decay of 0.0001 is selected to train the model.

Caffe [22] is used as the deep learning framework. Our proposed network is compared with two successful deep learning networks: GoogLeNet (Inception) [6] with 22 layers and Deep Residual with 50 layers (proposed by Microsoft [5]). We used the CPU-based implementation on 6 servers with 64 processors.

Figure 4 depicts the patterns of the learning in the proposed deep Residual-Inception network compared to two selected benchmarks (Inception and Residual network). Specifically, in Figure 4(a), although the proposed network starts with higher losses, it starts to converge after less than 10,000 iterations. The inception network has very low losses at first



(a) Disaster (b) CIFAR10

Fig. 4. Performance comparison on the disaster and CIFAR-10 datasets (Training losses vs iterations)

but it does not show any improvement from early stages, which can be due to over-fitting of this wide and deep network. The residual network shows a similar behavior as our network, but it still has higher losses than the proposed Residual-Inception network in all iterations, due to its very deep architecture. Similar patterns have been shown in Figure 4(b) on CIFAR-10 which includes more data and classes than the disaster dataset. In this figure, the inception network has higher training losses in all iterations; while the Residual-Inception network and the deep residual network have lower losses, respectively. Therefore, based on this experiment, one can conclude that a compact combination of these two benchmarks can converge earlier and produce lower losses. Another experiment has been conducted to show the efficiency of the proposed network, as well as the performance achieved on the testing data. Table 2 shows the specific times each network achieved to the highest performance on our servers with CPU implementation. As can be inferred from this table, the proposed network achieved 0.694 accuracy on the testing data in the disaster dataset in less than 45 hours; while it took more than 150 hours for the residual network and 306 hours for the inception network to achieve 0.688 and 0.653 testing accuracy, respectively. Similarly, our network achieves 0.714 on CIFAR-10 in almost 122 hours. The inception network's performance reaches to 0.696 accuracy in a similar training time; while the residual network achieves only 0.616 in more than 131 hours. As mentioned earlier, all these networks can achieve much higher accuracy using other optimization techniques such as ensemble, automatic learning reduction, and scale augmentation, which are out of the scope of this paper.

Finally, since interesting video event detection is the main purpose of this paper, we have utilized the proposed Deep Residual-Inception network to analyze its behavior on each disaster class. For this purpose, a binary classification is conducted based on the 3-fold cross validation. As this dataset is highly imbalanced, the precision, recall, and F1 values are used as the evaluation metrics instead of accuracy. Table 3 shows the detailed results for each disaster concept. As can be seen from the table, the proposed deep network achieves very

Table 2. Training time and corresponding test accuracy for two datasets

	Disaster		**CIFAR-10**	
Network	Training time (s)	Test accuracy	Training time (s)	Test accuracy
Inception	1,102,665	0.653	442,771	0.696
Residual	546,350	0.688	473,842	0.616
Residual-Inception	161,101	**0.694**	440,476	**0.714**

Table 3. Performance evaluation for different concepts on the disaster dataset

Event	Precision	Recall	F1
Flood	0.920	0.943	0.932
Damage	0.879	0.785	0.829
Fire	0.965	0.940	0.952
Mud-rock	0.971	0.923	0.947
Tornado	0.940	0.897	0.918
Lightening	0.979	0.968	0.973
Snow	0.914	0.798	0.849
Average	0.938	0.893	0.914

promising F1 scores in almost all classes. For example, lightening has the highest F1 score compared to the other classes, which can be due to its discriminative features (e.g., lights) in the corresponding images. Damage and snow classes have the lowest recall and F1 results respectively, because of their complex image texture and color. All in all, the average F1 score of binary classification on the disaster dataset is 0.914, which is higher than the previous work on this dataset.

5. CONCLUSION

This paper presents a new deep learning technique called Deep Residual-inception network, which not only improves the performance but also significantly speeds up the training and convergence processes. In summary, based on the experiments on two different multimedia datasets, the proposed Deep Residual-inception network has shown its superiority and effectiveness while maintaining low computational costs.

Acknowledgements

For Shu-Ching Chen, this research is partially supported by NSF CNS-1461926.

6. REFERENCES

[1] Shu-Ching Chen, Arif Ghafoor, and R. L. Kashyap, *Semantic Models for Multimedia Database Searching and*

Browsing, Springer Science & Business Media, 2000.

[2] Xin Chen, Chengcui Zhang, Shu-Ching Chen, and Min Chen, "A latent semantic indexing based method for solving multiple instance learning problem in region-based image retrieval," in *IEEE International Symposium on Multimedia*, CA, USA, 2005, pp. 37–44.

[3] Shu-Ching Chen, Mei-Ling Shyu, and Chengcui Zhang, "An intelligent framework for spatio-temporal vehicle tracking," in *4th IEEE International Conference on Intelligent Transportation Systems*, CA, USA, 2001, pp. 213–218.

[4] Samira Pouyanfar and Shu-Ching Chen, "Semantic event detection using ensemble deep learning," in *IEEE International Symposium on Multimedia*, CA, USA, 2016, pp. 203–208.

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

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

[7] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[8] Yoshua Bengio, Patrice Simard, and Paolo Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[9] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *13th International Conference on Artificial Intelligence and Statistics*, Sardinia, Italy, 2010, vol. 9, pp. 249–256.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 1026–1034.

[11] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on Machine Learning*, Lille, France, 2015, pp. 448–456.

[12] Jürgen Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, NV, USA, 2012, pp. 1106–1114.

[14] Hsin-Yu Ha, Yimin Yang, Samira Pouyanfar, Haiman Tian, and Shu-Ching Chen, "Correlation-based deep learning for multimedia semantic concept detection," in *16th International Conference on Web Information Systems Engineering*, FL, USA, 2015, Springer, pp. 473–487.

[15] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *AAAI Conference on Artificial Intelligence*, CA, USA, 2017, pp. 4278–4284.

[16] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back, "Face recognition: A convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[17] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio, "On the number of linear regions of deep neural networks," in *Advances in Neural Information Processing Systems*, Quebec, Canada, 2014, pp. 2924–2932.

[18] Min Lin, Qiang Chen, and Shuicheng Yan, "Network in network," *CoRR*, vol. abs/1312.4400, 2013.

[19] Sam Gross and Michael Wilber, "Training and investigating residual nets," `http://torch.ch/blog/2016/02/04/resnets.html`, 2016, retrieved at: 2016-11-20.

[20] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[21] Samira Pouyanfar and Shu-Ching Chen, "Semantic concept detection using weighted discretization multiple correspondence analysis for disaster information management," in *17th IEEE International Conference on Information Reuse and Integration*, PA, USA, 2016, pp. 556–564.

[22] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B. Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM International Conference on Multimedia*, FL, USA, 2014, pp. 675–678.