Deep visual embedding for image classification.

SALEH, A., ABDEL-NASSER, M., SARKER, M.M.K., SINGH, V.K., ABDULWAHAB, S., SAFFARI, N., GARCIA, M.A. and PUIG, D.

2018

Deep Visual Embedding for Image Classification

Adel Saleh[1], Mohamed Abdel-Nasser[2], Md. Mostafa Kamal Sarker[1], Vivek Kumar Singh[1], Saddam Abdulwahab[1], Nasibeh Saffari[1], Miguel Angel Garcia[3] and Domenec Puig[1]

[1]Dept. of Computer Engineering and Mathematics, Rovira i Virgili University, 43007 Tarragona, Spain
[2]Dept. of Electrical Engineering, Aswan University, 81542 Aswan, Egypt
[3]Dept. of Electronic and Communications Technology, Autonomous University of Madrid, Madrid, Spain

***Abstract*—This paper proposes a new visual embedding method for image classification. It goes further in the analogy with textual data and allows us to read visual sentences in a certain order as in the case of text. The proposed method considers the spatial relations between visual words. It uses a very popular text analysis method called 'word2vec'. In this method, we learn visual dictionaries based on filters of convolution layers of the convolutional neural network (CNN), which is used to capture the visual context of images. We employee visual embedding to convert words to real vectors. We evaluate many designs of dictionary building methods. To assess the performance of the proposed method, we used CIFAR10 and MNIST datasets. The experimental results show that the proposed visual embedding method outperforms the performance of several image classification methods. Experiments also show that our method can improve image classification regardless the structure of the CNN.**

***Index Terms*—Deep learning; Embedding; Image classification.**

I. Introduction

Several methods have been proposed for classification, clustering and indexing text documents in the literature. Those methods are fully grown and very effective to deal with huge numbers of classes. Textual data contains words and thus it can be processed using information about words. It is always possible to learn a word representation upon the existence of a given word in different documents. This idea can be also applied to images. Indeed, any image can be considered as a document containing a set of patches. However, this analogy would not stand further because text words are discrete values while local image patches are represented using high-dimensional and real-valued descriptors. To obtain discrete visual words, we should find a way to project image patches or patch descriptors to a single integer number per patch or descriptor. Therefore, local patches or descriptor vectors can be represented in terms of the region (i.e. discrete word) that they belong to. In text prepossessing field, vector representations of word (word embedding or *word2vec*) has been recently proposed to capture the context-based relationships between words. Those representations achieve the state-of-the-art results in different text processing tasks. Indeed, the main obstacle to apply those methods to images is the above mentioned nature of image patches.

In this paper, we build a bridge between images and text documents, which is applicable to images in total analogy with textual word embedding. In an end-to-end fashion, the way of visual embedding is done by the same manner as in natural language processing (NLP). The proposed method uses a very popular text analysis method called 'word2vec'. In this method, we learn visual dictionaries based on filters of convolution layers of the convolutional neural network (CNN), which is used to capture the visual context of images. We employee visual embedding to convert words to real vectors. In addition, we assess different designs of dictionary building methods.

The rest of the paper is organized as follows. Section II presents the related works. Section II discusses the word embedding. Section III explains the proposed method. The experimental results are provided in section IV. The conclusions and the future work are provided in section V.

II. Related works

Word embedding is basically real value vectors that capture semantic similarity. Such vectors are used as representations of words for NLP tasks, such as sentiment analysis, text classification and document clustering. These representations have been learned using neural networks [1], [2] and then used as to initialize a multi-layer network model [1], [2]. Like these approaches, we learn word embedding from image online and fine-tune them to predict visual class of image. Xu et al. [5] and Lazaridou et al. [6] use visual appearance to improve the word2vec representation by predicting real image representations from *word2vec*. While they focus on capturing the appearance, we focus on capturing the visual words inside the image. Another groups of researchers use visual and textual attributes to learn distributional models of the meaning of words [7], [8]. In turn, our set of visual words are learned in the training step of CNN. Other researchers use word embedding inside a larger models for complex tasks, such as image captioning [9], [10] and image retrieval [9]. These work are multi-modal (i.e. textual-visual). In turn, we only do visual embedding.

Bags-of-visual words (BOVW) have been proposed by Sivic et al. in [21]. BOVW becomes very popular due to its efficiency and high performance. Other works, such as probabilistic latent semantic analysis (pLSA) [23] and latent Dirichlet allocation (LDA) [24] use unsupervised classification. They compute latent concepts in images using the co-occurrences of visual words. These techniques obtain best results when the number of categories is known. Random forests classifier

have been used in some works, e.g., [24], however, the state-of-the-art results were obtained with support vector machines (SVM) classifier. The authors of [25] combine local matching of the features and specific kernels based on the χ^2 to get the better results. Most BOVW methods build their visual words in the clustering step, in turn, we learn our dictionary from the data using CNN or we build them using local binary pattern (LBP) or Gabor filters.

Vision and NLP: Recently, different problems at the intersection of NLP and vision are considered to be interesting to researchers [19], [20]. Significant breakthroughs have been achieved in several tasks, such as visual question answering [10]–[12], image captioning [13]–[15], aligning text and vision [16], [17] and video description [18]. Unlike these works, our approach is generic, i.e., it can be also used for multiple tasks.

III. Word embedding

In NLP, word embedding is a set of language modeling algorithms where words and phrases are mapped to vectors of real numbers. Mathematically, embedding is a word projection from a space with one dimension (sparse vector) to a continuous vector space with much lower dimension. The authors of [29], [33] showed that mapping can be done using neural networks. In literature, other word embedding models were proposed, such as probabilistic models [29], dimensionality reductions based models [31], word context based [31]. Experiments showed that word and phrase embeddings are able to boost the performance in NLP tasks (for example, syntactic parsing and sentiment analysis) when they used as the underlying input representation. Taking into account this intuition, and supposing that patterns from a given (or learned) dictionary are occurring in the images, any image can be converted to an image of visual words instead of image containing colors. In this work, we follow [33] to do the embedding step. Below, we discuss some options to build the dictionaries.

Simple pixel dictionary: It is clear that the use of values of image pixels are the simplest dictionary that can be built. It consists of gray-scale values of the pixels. The main disadvantages of this approach are:

- It does not capture any information about the context. It only describes a single pixel.
- It is sensitive to the noise.

Local binary pattern (LBP): The LBP operator was proposed by Ojala et al. [26]. Indeed, LBP descriptor is one of the most successful descriptors due to its outstanding advantages, such robustness to illumination changes, low complexity in terms of computation and implementation. LBP characterizes the structure of a local image patch. The main idea of LBP is to convert differences between a given pixel value of the central point and those of its neighbors. The value of the binary pattern is used to label the given pixel. The responses of LBP is calculated as follows.

$$LBP = \sum_{p=0}^{P} s(g_p - g_c)2^p \qquad (1)$$

where P is number of all neighbor pixels, g_c is the center pixel value and g_p is neighbor pixels. As shown in Eq.1, for any pixel in the neighbor, if the center pixel's value is greater than the neighbour's value, $s = 0$ is set to the neighbor pixel; otherwise, $s = 1$ is set to the neighbor pixel. This gives an 8-digit binary number and it is usually converted to a decimal value.

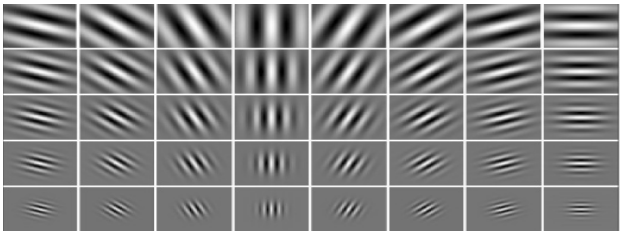

Fig. 1: Gabor filters

Gabor filters: Fig. 1 sows a set of Gabor filters with various scales and rotations. Jones and Palmer explained in [26] that the real part of the complex Gabor function is a good simulation to the receptive field weight functions that found in simple cells in mammals striate cortex. In other words, image analysis using Gabor filters is similar to the human visual system. The impulse response of Gabor filters is defined analytically as a Gaussian function multiplied by a sinusoidal wave or a plane wave in case of 2D Gabor filters. As shown in Eq. 2, the filters have a real and an imaginary components representing orthogonal directions. Components can be formed into a complex number or used individually. The complex form of Gabor filters can be given as follows:

$$g(x, y, \lambda, \theta, \psi, \omega, \gamma) = exp(-\frac{x^{'2} + \gamma^2 y^{'2}}{2\omega^2})exp(i(2\pi\frac{x^{'}}{\lambda} + \psi)) \qquad (2)$$

And the real part can be given as following:

$$g(x, y, \lambda, \theta, \psi, \omega, \gamma) = exp(-\frac{x^{'2} + \gamma^2 y^{'2}}{2\omega^2})cos(2\pi\frac{x^{'}}{\lambda} + \psi) \qquad (3)$$

where $x^{'} = xcos\theta + ysin\theta$ and $y^{'} = -xsin\theta + ycos\theta$. Convolving an image I with a bank of Gabor filters with N filters will produce N image. Stack the resulting images horizontally as shown in Fig. 2 and for every x, y in output

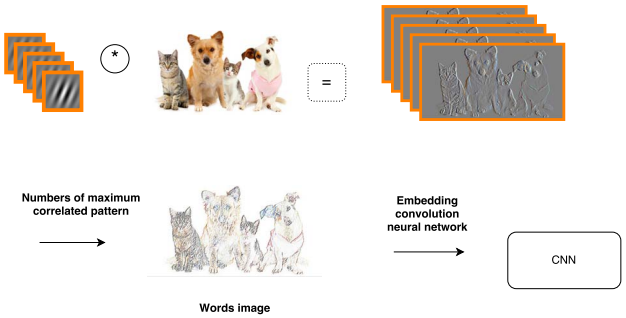

Fig. 2: Embedding procedure for analytically given dictionary

image write the number of the filter with highest response in the same coordinates. It always possible to find the word in a given location using a similar policy (see Fig. 2).

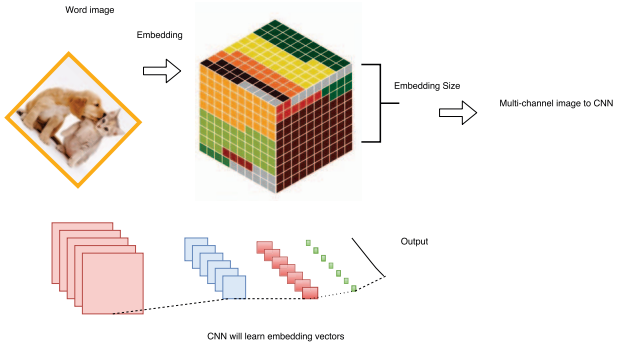

Fig. 3: CNN architecture for a given dictionary

IV. Proposed Method

In this section, we show how to compute the visual words and find them inside the images. The proposed method is independent on dictionary building step as long as the resulting dictionary has finite numbers of visual words. Each patch in the image will be converted to the number closest visual word in the used dictionary. This step produces 'words image' (see Fig. 3). It will be passed to CNN which captures the spatial context of words. After that, every word in the word image will be embedded to real values vector with embedding size d, resulting an image with d channels. Then, d-channel image will pass through layer of the CNN and the process will continue until we get a stable embedding vector.

As shown in Fig. 4, in the proposed method we use filters learned by CNN to build dictionary. The first layer of the CNN is a convolution layer with N filters. Afters passing throw this layer, we will get N feature maps. Choosing the number of filters with highest responses, and then pass the resulting responses to the next layers. It is possible to learn filters (build a dictionary). The size of the dictionary equals the number of filters of first convolutional layer. In this paper, we adopt learned filters to build dictionary. We also believe that Gabor dictionary is a special case of learned filters.

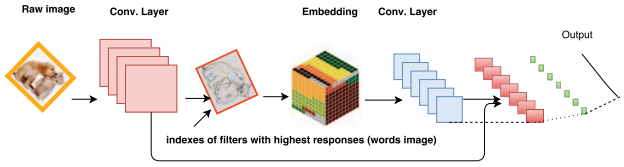

Fig. 4: Proposed method

V. Experimental Results an Discussion

A. Datasets

To assess the performance of the proposed method, we use two state-of-the-art image classification datasets: MNIST and CIFAR-10.

The MNIST dataset contains 70000 of 28x28 gray-scale images. The images contain hand-written digits of 0 to 9. In the dataset, there are 60000 training images and 10000 testing images.

The CIFAR-10 dataset contains 60000 RGB images of 32x32 pixels of 10-classes. The dataset is split into 50000 training images and 10000 testing images.

B. Model Setup

In our proposed model, the first layer of CNN is a convolution layer. This layer consists of 128 filters (the size of dictionary in our model is 128 words). The output of the first layer is an image with 128 channels. To convert the multi-channel image to word image, we take the index of the channel with highest response. The intuition behind that is that word (filter) with highest response is the closest to the patch in given location. Thus, we have a single-channel image which contains indexes of closest words. The next embedding layer will convert each single pixel of the word image to a vector of real-values. In our experiments, embedding size of 16 is used. Thus, each word image will be converted to an image of 16 channels containing real values. The next layer is a convolutional layer, in which the number of filters is set to the visual dictionary size. Afterwards, the outputs of the first and second convolution layers are concatenated to learn visual words, which are the filters of the first convolutional layer.

C. Results

Table I shows that the performance of CNN can be improved by adding embedding in bottom layers of the model. Our custom CNN model gives an error of 1.0 with MNIST dataset without embedding and 0.5 after embedding which is quite good improvement. Fig. 5 shows the change of accuracy across training and testing epochs of CNN while Fig. 6 presents the confustion matrix of the proposed method with MNIST dataset. Applying the same model to CIFAR10 dataset gives an

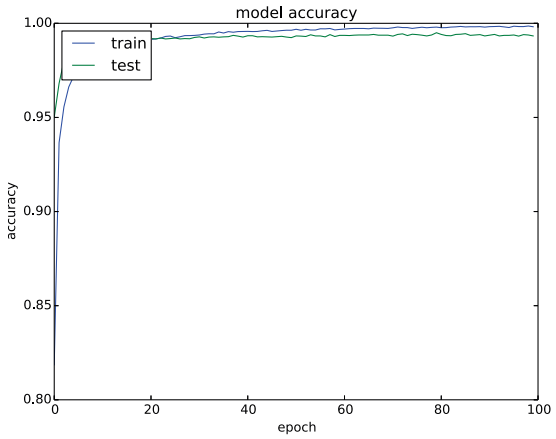

Fig. 5: The accuracy of the proposed method with MNIST dataset

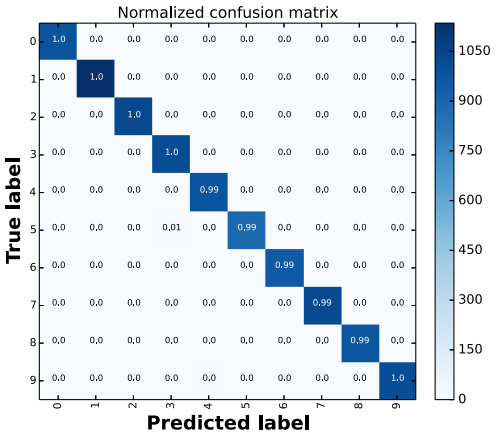

Fig. 6: The confusion matrix of the proposed method with MNIST dataset

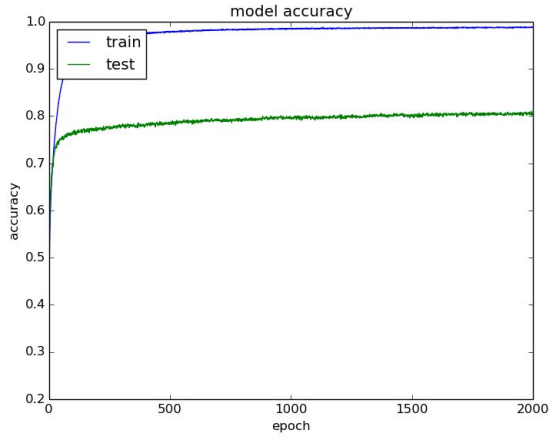

Fig. 7: The accuracy of the proposed method with CIFAR10 dataset

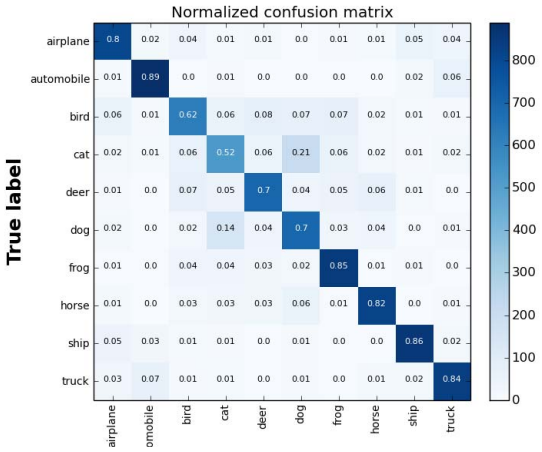

Fig. 8: The confusion matrix of the proposed method with CIFAR10 dataset

accuracy of 0.79 without embedding and 0.81 after embedding on the bottom of CNN. Fig. 7 demonstrates the change of accuracy across training and testing epochs of CNN while Fig. 8 presents the confustion matrix of the proposed method with CIFAR10 dataset. We avoid putting embedding on the bottom of state-of-the-art models, such as VGG16 and ResNet in order to show the contribution of the visual embedding itself. Table II demonstrates that learned filters as dictionary give much better performance compared to LBP dictionary.

TABLE I: Performance of a custom convolution neural network before and after embedding application

Method	MNIST (error)	CIFAR10 (accuracy)
Before	1.0	79.01
After	**0.5**	**81.01**

TABLE II: Performance of a LBP Dictionary and Learned Filters

Method	MNIST (error)	CIFAR10 (accuracy)
Learned Filters	**0.5**	**81.01**
LBP dictionary	0.7	61.01

Table III shows that the proposed method gives a promising performance with MNIST and CIFAR10 datasets. Our method shows quite good performance on the MNIST (an error of 0.5) and promising behavior on CIFAR10 (an accuracy of 0.81). The degradation of behavior in CIFAR10 is due to two factors: the difficulty of classification problem on this dataset and the simplicity of our model. Table III also shows that the proposed method is on par with several recently published image classification methods.

TABLE III: Comparing the proposed method with related methods

Method	MNIST (error)	CIFAR10 (accuracy)
Proposed method	0.5	81.01
Generalizing pooling [34]	0.31	92.38
ResNet	-	94.07
Deep Fried Convnets [35]	0.71	-
Adversarial Examples [36]	0.78	-
PCANet Examples [37]	0.62	78.67

ACKNOWLEDGEMENT

This paper is partially funded by project DPI2016-77415-R.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented a CNN model to embed visual words in images. We have treated images as textual document, build visual words and embed them to capture the spacial context surrounding them. The learned embedding performed better than the same CNN model with original images. The main goal of our proposed method is to push a framework that shows how to build a bridge between embedding in text data and how to adapt it for visual data and wide range of generic image and video tasks. In the future work, we will apply image embedding on the bottom

of state-of-the-art CNN models, such as VGG16 and ResNet. In addition, we will apply it to recognize actions in videos.

REFERENCES

[1] R.Collobert, J.Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning (2008, July).160-167.

[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. In Advances in Neural Informa- tion Processing Systems, (2013), pages 3111–3119.

[3] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model. Journal of machine learning research, (2003),1137–1155.

[4] T.Mikolov, J. Kopecky, L. Burget, O. Glembek,. Neural network based language models for highly inflective languages. In Acoustics, Speech and Signal Processing, (2009). 4725-4728.

[5] R. Xu, J. Lu, C. Xiong, Z. Yang, J. J.Corso, Improving word representations via global visual context. (2014) In NIPS Workshop on Learning Semantics.

[6] A. Lazaridou, N. T.Pham, M. Baroni, . Combining language and vision with a multimodal skip-gram model. (2015), arXiv preprint arXiv:1501.02598.

[7] C. Silberer, V. Ferrari, , M. Lapata, . Models of Semantic Representation with Visual Attributes. (2013) (pp. 572-582).

[8] C. Silberer, M. Lapata. Learning Grounded Meaning Representations with Autoencoders. (2014). (721-732).

[9] R. Kiros, R. Salakhutdinov, R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models.(2014) arXiv preprint arXiv:1411.2539.

[10] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recogni- tion,(2015) pages 3156–3164, .

[11] S. Antol, A. Agrawal, J. Lu, M. Mitchell, , D. Batra, C. Lawrence Zitnick, , D. Parikh, D. Vqa: Visual question answering. In Proceedings of the IEEE International Conference on Computer Vision (2015), 2425-243).

[12] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, W. Xu. Are you talking to a machine? dataset and methods for multilingual image question. In Advances in Neural Information Processing Systems. (2015). 2296-2304.

[13] X. Chen, , C. Lawrence Zitnick Mind's eye: A recurrent visual representation for image caption generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015), 2422-2431.

[14] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell. Long-term Recurrent Convolutional Networks for Visual Recognition and Description.(2014) arXiv. org.

[15] M. Hodosh, P. Young, J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. Journal of Artificial Intelligence Research, (2013), 853-899.

[16] A. Karpathy, L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3128-3137.

[17] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural lan- guage models. (2014),11-13.

[18] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition 2625–2634.

[19] S. Kottur, R. Vedantam, J. M. Moura, D. Parikh.Visual word2vec (visw2v): Learning visually grounded word embeddings using abstract scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.(2016).4985-4994.

[20] P. Tirilly, V. Claveau, P. Gros. Language modeling for bag-of-visual words image categorization. In Proceedings of the 2008 international conference on Content-based image and video retrieval.(2008, July). ACM.

[21] J. Sivic, A. Zisserman. Video google: A text retrieval approach to object matching in videos.In iccv (Vol. 2, No. 1470, pp. 1470-1477).(2003, October).

[22] A. Bosch, A. Zisserman, X. Muñoz. Scene classification via pLSA. Computer Vision–ECCV. (2006). 517-530.

[23] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In Proceedings of ICCV, (2005).

[24] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In ECCV: Workshop on Statistical Learning in Computer Vision, Prague, Czech Republic.(2004)

[25] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In Proceedings of ICCV.(2007).

[26] T. Ojala, M. Pietikäinen, and T. Maenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7. Jul. (2002). 971–987

[27] J. P. Jones, L. A. Palmer. An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. Journal of neurophysiology.(1987).1233-1258

[28] I. Fogel, D. Sagi. Cybern. (1989) 61-103

[29] T. Mikolov,I. Sutskever,K. Chen, G. S Corrado, J. Dean.Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems. (2013). (3111-3119).

[30] A. Globerson, G. Chechik, F. Pereira,N. Tishby. Euclidean embedding of co-occurrence data. Journal of Machine Learning Research, 8(Oct), (2007). 2265-2295.

[31] R. Lebret, R. Collobert.Word emdeddings through hellinger pca. (2013). arXiv preprint. arXiv preprint arXiv:1312.5542.

[32] O. Levy, Y. Goldberg, I. Ramat-Gan . Linguistic Regularities in Sparse and Explicit Word Representations. In CoNLL (2014).(171-180).

[33] J. Pennington, R. Socher, C.D. Manning. (2014, October). Glove: Global Vectors for Word Representation. In EMNLP (Vol. 14. (2014, October). 1532-1543.

[34] C. Y. Lee, Gallagher,Z. Tu Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In International conference on artificial intelligence and statistics.(2016).

[35] Yang Z., M. Moczulski,M. Denil, N. de Freitas, A. Smola, L. Song, Z. Wang. Deep fried convnets. In Proceedings of the IEEE International Conference on Computer Vision.(2015), 1476-1483

[36] I. Goodfellow, J. Shlens, C. Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572. (2014).

[37] T. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, Y. Ma. (2015). Pcanet: A simple deep learning baseline for image classification?. IEEE Transactions on Image Processing, 24(12), 5017-5032.