Visual Word Embedding for Text Classification

Ignazio Gallo[1][0000−0002−7076−8328], Shah Nawaz[1*][10000−0002−7715−4409], Nicola Landro[1][0000−0002−0565−7496], and Riccardo La Grassa[1][0000−0002−4355−0366]

University of Insubria Varese, Italy
{ignazio.gallo,snawaz,nlandro,rlagrassa}@uninsubria.it

Abstract. The question we answer with this paper is: 'can we convert a text document into an image to take advantage of image neural models to classify text documents?' To answer this question we present a novel text classification method that converts a document into an encoded image, using word embedding. The proposed approach computes the Word2Vec word embedding of a text document, quantizes the embedding, and arranges it into a 2D visual representation, as an RGB image. Finally, visual embedding is categorized with state-of-the-art image classification models. We achieved competitive performance on well-known benchmark text classification datasets. In addition, we evaluated our proposed approach in a multimodal setting that allows text and image information in the same feature space.

Keywords: Encoded text · Word Embedding · Multimodal classification.

1 Introduction

Text classification is a common task in Natural Language Processing (NLP). Its goal is to assign a label to a text document from a predefined set of classes. In last decade, Convolutional Neural Networks (CNNs) have remarkably improved performance in image classification [8, 16, 17] and researchers have successfully transferred this success into text classification [1, 20]. Image classification models [8, 16] are adapted to accommodate text [1, 7, 20]. We, therefore, leverage on the recent success in image classification and present a novel text classification approach to cast text documents into a visual domain to categorize text with image classification models. Our approach transforms text documents into encoded images or visual embedding capitalizing on Word2Vec word embedding which convert words into vectors of real numbers [9, 11, 14]. Typically word embedding models are trained on large corpus of text documents to capture semantic relationships among words. Thus these models can produce similar word embeddings for words occurring in similar contexts. We exploit this well-known fundamental property of word embedding models to transform a text document into a sequence of colours (visual embedding), obtaining an encoded image, as shown in Figure 1. Intuitively, semantically related words obtain similar colours

*__Current Affilation:__ Italian Institute of Technology, Genova Italy

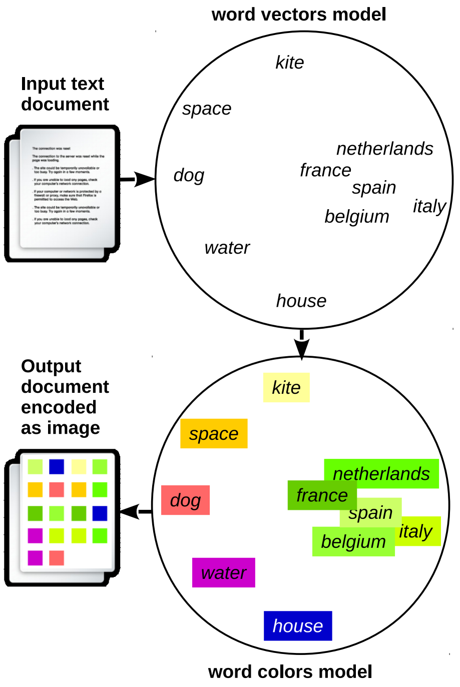

Fig. 1. We exploited a well-known property of word embedding models: semantically correlated words obtain similar numerical representation. It turns out that if we interpret real-valued vectors as a set of colours, it is easy for a visual system to cope with relationships between words of a text document. It can be observed that green coloured words are related to countries, while other words are represented with different colours.

or encodings in the encoded image while uncorrelated words are represented with different colours. Interestingly, these visual embeddings are recognized with state-of-the-art image classification models. In this paper, we present a novel text classification approach to transform word emebedding of text documents into the visual domain. The choice to work with Word2Vec encoding vectors transformed into pixels by splitting them in triplets is guided by two main reasons:

1. we want to exploit existing image classification models to categorize text documents;
2. as a consequence, we want to integrate the text within an image to transform a text-only or images only classification problem, into a multimodal classification problem using a single 2D data [12].

We evaluated the method on several large scale datasets obtaining promising and comparable results. An earlier version of our encoding scheme was published in ICDAR 2017 [2], where we used a different encoding technique that require more space to encode a text document into an image. In this paper, we explore various parameters associated with an encoding scheme. We extensively evaluated the improved encoding scheme on various benchmark datasets for text classification. In addition, we evaluated the proposed approach in a multimodal setting to fuse image and text in the same feature space to perform classification.

2 Related Work

Deep learning methods for text documents involved learning word vector representations through neural language models [11, 14]. These vector representations serve as a foundation in our paper where word vectors are transformed into a sequence of colors or visual embedding. The image classification model is trained and tested on these visual embeddings. Kim [7] proposed a simple shallow neural network with one convolution layer followed by a max pooling layer over time. Final classification is performed with one fully connected layer with dropout. The authors in [20] presented rather deep convolutional neural network for text classification. The network is similar to the convolutional network in computer vision [8]. Similarly, Conneau et al. [1] presented a deep architecture that operates at character level with 29 convolutional layers to learn hierarchical representations of text. The architecture is inspired by recent progress in computer vision [4, 15]. Johnson et al. [6] proposed a simple network architecture by increasing the depth of the network without increasing computation costs. This model performs text region embedding, which generalizes commonly used word embedding. Though Word2vec is one of the state-of-the-art model for text embedding, others approaches such as GloVe, ELMo, and BERT have improved various NLP tasks. The BERT model used a relatively new transformer architecture to compute word embedding and it has been shown to produce state-of-the-art word embedding, achieving excellent performance. Yang et al. [19] proposed the XLNet, a generalized autoregressive pretreatment method that exceeds the limits of BERT thanks to its autoregressive formulation.

In this paper, we leverage on recent success in Computer Vision, but instead of adapting deep neural network to be fed with raw text information, we propose an approach that transforms word embedding into encoded text. Once we have encoded text, we employed state-of-the-art deep neural architectures for text classification.

3 Proposed Approach

In this section, we present our approach to transform Word2Vec word embedding into the visual domain. In addition, we explained the understanding of CNNs with the purposed approach.

3.1 Encoding Scheme

The proposed encoding approach is based on Word2Vec word embedding [11]. We encode a word t_k belonging to a document D_i into an encoded image of size

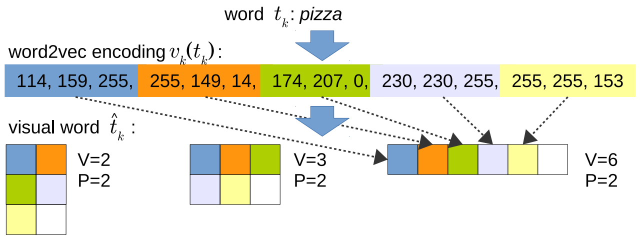

Fig. 2. In this example, the word "*pizza*" is encoded into a visual word $\hat{t}_k$ based on Word2Vec feature vector with length 15. This visual word can be transformed into different shapes, varying the V parameter (in this example $V = 2, 3, 6$ superpixels)

$W \times H$. The approach uses a dictionary $F(t_k, v_k)$ with each word t_k associated with a feature vector $v_k(t_k)$ obtained from a trained version of Word2Vec word embedding model. Given a word t_k, we obtained a visual word $\hat{t}_k$ having width V that contains a subset of a feature vector, called superpixels (see example in Fig. 2). A superpixel is a square area of size $P \times P$ pixels with a uniform color that represents a sequence of contiguos features $(v_{k,j}, v_{k,j+1}, v_{k,j+2})$ extracted as a sub-vector of v_k. We normalize each component $v_{k,j}$ to assume values in the interval $[0 \dots 255]$ with respect to k, then we interpret triplets from feature vector v_k as RGB sequence. For this very reason, we use feature vector with a length multiple of 3. Our goal is to have a visual encoding that can be generic to allow the use of existing CNN models; for example, the AlexNet has an 11×11 kernel in the input layer, which makes it very difficult to interpret visual words with 1×1 superpixels ($P = 1$).

The blank space s around each visual word $\hat{t}_k$ plays an important role in the encoding approach. We found out that the parameter s is directly related to the shape of a visual word. For example, if $V = 16$ pixels then s must also have a value close to 16 pixels to let the network understand where a word ends and another begins.

3.2 Encoding Scheme with CNN

It is well understood that a CNN can learn to detect edges from image pixels in the first layer, then use the edges to detect trivial shapes in the next layer, and then use these shapes to infer more complex shapes and objects in higher layers [10]. Similarly, a CNN trained on our proposed visual embedding may extract features from various convolutional layer (see example in Fig. 3). We observed that the first convolutional layer recognizes some specific features of visual words associated with single or multiple superpixels. The remaining CNN layers aggregate these simple activations to create increasingly complex relationships between words or parts of a sentence in a text document. Fig. 3 also highlights how the different convolutional layers of a CNN activate different areas corresponding to single words (layers closest to the input) or sets of words distributed over a 2-D space (layers closest to the output). This is a typical behavior of deep

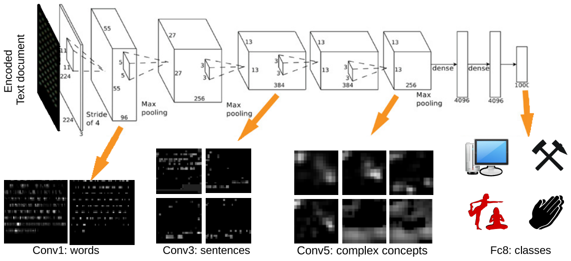

Fig. 3. Starting from an encoded text document, the resulting image is classified by a CNN model normally employed in image classification. The first convolutional layers look some particular features of visual words while the remaining convolutional layers can recognize sentences and increasingly complex concepts.

models that work on images, while 1-D models that work on text usually limit themselves to activating only words or word sequences.

To numerically illustrate this concept, we use the receptive field of a CNN. The receptive field r is defined as the region in the input space that a particular CNN feature is looking at. For a convolution layer of a CNN, the size r of its receptive field can be computed by the following formula:

$$r_{out} = r_{in} + (k - 1) \cdot j_{in} \tag{1}$$

where k is the convolution kernel size and j is the distance between two consecutive features. Using the formula in Eq. 1 we can compute the size of the receptive field of each convolution layer. For example, the five receptive field of an AlexNet, showed in Fig. 4, have the following sizes: *conv1* 11 × 11, *conv2* 51 × 51, *conv3* 99 × 99, *conv4* 131 × 131 and *con5* 163 × 163. This means that the *conv1* of an AlexNet, recognizes a small subset of features represented by

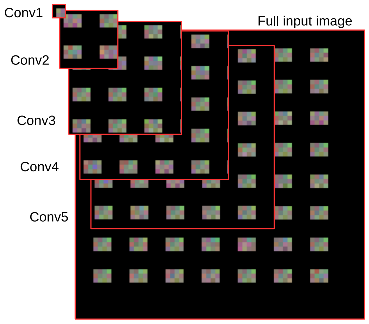

Fig. 4. The receptive fields of the five convolution layers of an AlexNet. Each receptive field is cut from a 256 × 256 image to analyze the quantity of visual words that each *conv* layer is able to analyze on each pixel of its feature map.

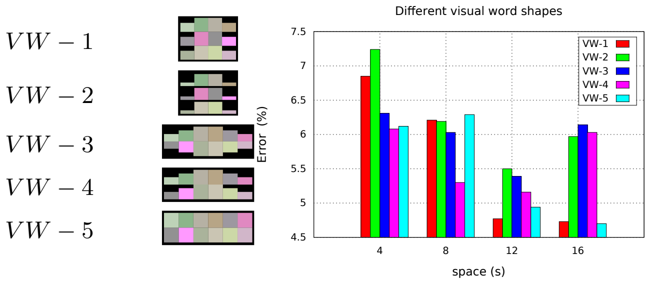

Fig. 5. On the left, five different designs for visual words (VW) represented by 36 Word2Vec features, over the 20news-bydate dataset. The width V of these words is 4 for the first two on the top and 6 for the rest. The first four visual words consist of super pixels with different shapes to form particular visual words. On the right, a comparison over these different shapes of visual words.

superpixels, while the *conv2* can recognize a visual word (depending on the configuration used for the encoding), up to the *con5* layer where a particular feature can simultaneously analyze all the visual words available in the input image.

4 Dataset

Zhang *et al.* [20] introduced several large-scale datasets which covers several text classification tasks such as *sentiment analysis, topic classification* or *news categorization*. In these datasets, the number of training samples varies from several thousand to millions, which is considered ideal for deep learning-based methods. In addition, we used 20news-bydate dataset to test various parameters associated with the encoding approach.

5 Experiments

The aim of these experiments is twofold: (i) evaluate configuration parameters associated with the encoding approach; (ii) compare the proposed approach with other deep learning methods. (iii) to validate the proposed approach on a real-world application scenario. In experiments, percentage error is used to measure the classification performance. The encoding approach mentioned in Section 3.1 produces encoded image that are used to train and test a CNN. We used AlexNet [8] and Googlenet [17] architectures as base models from scratch. We used a publicly available Word2Vec word embedding with default configuration parameters as in [11] to train word vectors on all datasets. Normally, Word2Vec is trained on a large corpus and used in different contexts. However, we trained this model with the same training set for each dataset.

Table 1. Comparison between CNNs trained with different configurations on our proposed approach. The width V (in superpixels) of visual words is fixed while the Word2Vec encoding vector size and space s (in pixel) varies. H is the height of visual word obtained.

s	V	H	w2v feat.	error (%)
4	4	1	12	7.63
8	4	1	12	5.93
12	4	1	12	**4.45**
16	4	1	12	4.83
4	4	2	24	6.94
8	4	2	24	5.60
12	4	2	24	5.15
16	4	2	24	**4.75**
4	4	3	36	6.72
8	4	3	36	5.30
12	4	3	36	**4.40**
16	4	3	36	4.77

5.1 Parameters setting

We used 20news-bydate dataset to perform a series of experiments with various settings to find out the best configuration for the encoding scheme. In the first experiment, we changed the space s among visual words and Word2Vec feature length to identify relationships between these parameters. We obtained a lower percentage error with higher values of s parameter and a higher number of Word2Vec features as shown in Table 1. We observed that the length of feature vector $v_k(t_k)$ depends on the nature of the dataset. For example in Fig. 6, a text document composed of a large number of words cannot be encoded completely using a high number of Word2Vec features, because each visual word occupies more space in the encoded image. Moreover, we found out that error does not decrease linearly with the increase of Word2Vec features, as shown in Table 3.

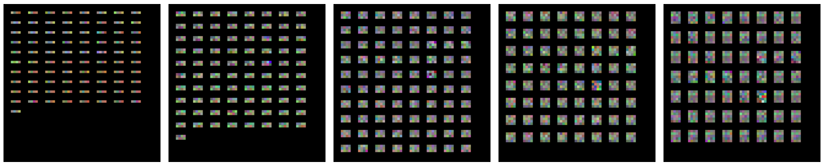

Fig. 6. Five encoded images obtained using different Word2Vec features length and using the same document belonging to the 20news-bydate dataset. All the images are encoded using space $s = 12$, superpixel size 4×4, image size $= 256\times256$ and visual word width $V = 16$. The two leftmost images contain all words in the document encoded with 12 and 24 Word2Vec features respectively, while 3 rightmost encoded images with 36, 48 and 60 features length cannot encode entire documents.

Table 2. Comparison of different parameters over the 20news-bydate dataset. In the leftmost table we changed the size of the encoded image from 100 × 100 to 500 × 500 and the crop size is also changed by multiplying the image size with a constant i.e. 1.13. Here *sp* stands for superpixel, *w2v* is for number of Word2Vec features, *Mw* stands for Max number of visual words that an image can contain and #*w* is the number of text documents in the test set having a greater number of words than *Mw*. We fixed the remaining non-specified parameters as follow: $s = 12$, $V = 4$, $sp = 4$, image size= 256.

image size	**crop**	**error**
500 × 500	443	**8.63**
400x400	354	9.30
300x300	266	10.12
200x200	177	10.46
100x100	88	15.70

sp	**error**
5x5	8.96
4x4	**8.87**
3x3	10.27
2x2	10.82
1x1	10.89

stride	**error**
5	8.7
4	8.87
3	8.33
2	**7.78**
1	12.5

w2v	**Mw**	**#w**	**error**
12	180	50%	9.32
24	140	64%	8.87
36	120	71%	**7.20**
48	100	79%	8.21
60	90	83%	20.66

We tested various shapes for visual words before selecting the best one, as shown in Fig. 5 (on the left). We showed that the rectangular shaped visual words obtained higher performance as highlighted in Fig. 5 (on the right). Moreover, space s between visual words plays an important role in the classification, in fact using a high value for the s parameter, the convolutional layer can effectively distinguish among visual words, also demonstrated from the results in Table 1. The first level of a CNN (*conv1*) specializes convolution filters in the recognition of a single superpixel as shown in Fig. 3. Hence, it is important to distinguish between superpixels of different visual words by increasing the parameter s.

These experiments led us to the conclusion that we have a trade-off between the number of Word2Vec features to encode each word and the number of words that can be represented in an image. Increasing the number of Word2Vec features increases the space required in the encoded image to represent a single word. Moreover, this aspect affects the maximum number of words that may be encoded in an image. The choice of this parameter must be done considering the nature of the dataset, whether it is characterized by short or long text documents. For our experiments, we used a value of 36 for Word2Vec features, considering results presented in Table 3.

5.2 Data augmentation

We encode the text document in an image to exploit the power of CNNs typically used in image classification. Usually, CNNs use "*crop*" data augmentation technique to obtain robust models in image classification. This process has been used in our experiments and we showed that increasing the number of training samples by using the *crop* parameter, results are improved. During the training phase, 10 random 227 × 227 crops are extracted from a 256 × 256 image (or proportional crop for different image size, as reported in the leftmost Table 3) and then fed to the network. During the testing phase, we extracted a 227 × 227 patch from the center of the image. It is important to note that thanks to space s introduced around the encoded words, the encoding of a text document in the

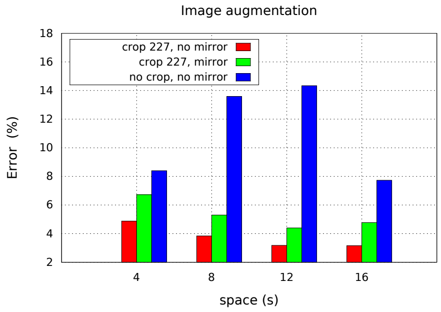

Fig. 7. Classification error using data augmentation: (*mirror* and *crop*) over the 20news-bydate test set.

Table 3. Comparison of different parameters over the 20news-bydate dataset. Here *sp* stands for superpixel, *w2v* is for number of Word2Vec features, *Mw* stands for Max number of visual words that an image can contain and *#w* is the number of text documents in the test set having a greater number of words than *Mw.* We fixed the remaining non-specified parameters as follow: $s = 12$, $V = 4$, $sp = 4$, image size= 256.

sp	error
5x5	8.96
4x4	**8.87**
3x3	10.27
2x2	10.82
1x1	10.89

stride	error
5	8.7
4	8.87
3	8.33
2	**7.78**
1	12.5

w2v	Mw	#w	error
12	180	50%	9.32
24	140	64%	8.87
36	120	71%	**7.20**
48	100	79%	8.21
60	90	83%	20.66

image is not changed by cropping. So, cropping is equivalent to producing many images with the same encoding but with a shifted position.

The "*stride*" parameter is very primary in decreasing the complexity of the network, however, this value must not be bigger than the superpixel size, because larger values can skip too many pixels, which leads to information lost during the convolution, invalidating results.

We showed that the *mirror* data augmentation technique, successfully used in image classification, is not recommended here because it changes the semantics of the encoded words and can deteriorate the classification performance. Results are presented in Fig. 7.

5.3 Comparison with other state-of-the-art text classification methods

We compared our approach with several state-of-the-art methods. Zhang *et al.* [20] presented a detailed analysis of traditional and deep learning methods. From their papers, we selected the best results and reported them in Table 4. In addition, we also compared our results with Conneau *et al.* [1] and Xiao*et al.* [18] . We obtained comparable results on all the datasets used: DBPedia, Yahoo Answers!, Amazon Polarity, AGnews, Amazon Full and Yelp Full. However, we obtained a higher error on Sogou dataset due to the translation process explained in the paper [20]. It is interesting to note that the papers [1, 20] propose text adapted variants of convolutional neural networks [4, 8] developed for computer vision. Therefore, we obtain similar results to these papers. However, there is a clear performance gain compared to the hybrid of convolutional and recurrent networks [18].

Table 4. Testing error of our encoding approach on 8 datasets with Alexnet and GoogleNet. The best results are shown in bold. XLNet is a very recent approach based on BERT.

Model	**AG**	**Sogou**	**DBP.**	**Yelp P.**	**Yelp F.**	**Yah. A.**	**Amz. F.**	**Amz. P.**
Xiao *et al.*	8.64	4.83	1.43	5.51	38.18	28.26	40.77	5.87
Zhang *et al.*	7.64	2.81	1.31	4.36	37.95	28.80	40.43	4.93
Conneau *et al.*	8.67	3.18	1.29	4.28	35.28	26.57	37.00	4.28
Johnson and Zhang	6.87	**1.84**	0.88	2.64	30.58	**23.90**	34.81	3.32
Our encoding scheme + AlexNet	9.19	8.02	1.36	11.55	49.00	25.00	43.75	3.12
Our encoding scheme + GoogleNet	7.98	6.12	1.07	9.55	43.55	24.10	40.35	3.01
XLNet Yang *et al.*	**4.45**	-	**0.60**	**1.37**	**27.05**	-	**31.67**	**2.11**

Table 5. Percentage errors on 20news-bydate dataset with three different CNNs.

CNN architecture	error
Encoding scheme + AlexNet	4.10
Encoding scheme + GoogleNet	3.81
Encoding scheme + ResNet	2.95

5.4 Comparison with state-of-the-art CNNs

As expected, in Table 4 we performed better using GoogleNet, compared to results obtained using the same configuration on a less powerful model like AlexNet. We, therefore, conclude that recent state-of-the-art network architectures, such as InceptionResNet or Residual Network would further improve the performance of our proposed approach. To work successfully with large datasets and powerful models, a high-end hardware and large training time are required, thus we conducted experiments only on 20news-bydate dataset with three network architectures: AlexNet, GoogleNet and ResNet. Results are shown in Table 5. We performed better with ResNet which represents one of the most powerful network architecture.

6 Multimodal Application

We use two multimodal datasets to demonstrate that our proposed visual embedding brings significant benefits to fuse encoded text with the corresponding image information [3]. The first dataset named Ferramenta [3] consists of 88, 010 image and text pairs split in 66, 141 and 21, 869 for train and for test sets respectively, belonging to 52 classes. We used another publicly available dataset called Amazon Product Data [5]. We randomly selected 10, 000 image and text pairs belonging to 22 classes. Finally, we randomly selected 10, 000 image and text pairs of each class dividing into train and test sets with 7, 500 and 2, 500 samples respectively.

We want to compare the classification of advertisement made in different ways: using only the encoded text description, using only the image of the advertisement and the fused combination. An example is shown in Fig. 8. The model trained on images only for Amazon Product Data, we obtained the following first two predictions: 77.42% Baby and 11.16% "Home and Kitchen" on this example. While the model trained on the multimodal Amazon Product Data, we obtained the following first two predictions: 100% Baby and 0% "Patio Lawn and Garden" for the same example. This indicate that our visual embedding improves classification performance compare to text or image only. Table 6 shows that the combination of text and image into a single image, outperforms best result obtained using only a single modality on Ferramenta and Amazon Product Data. It also demonstrate that the combination of text and image into a single image, outperforms best result obtained using only a single modality on both datasets.

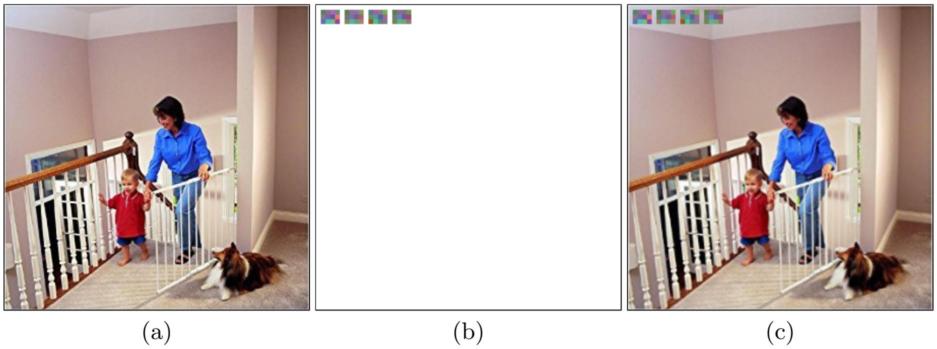

Fig. 8. An example of multimodal fusion from the Amazon dataset belonging to the class "Baby". (a) shows the original image, (b) is a blank image with the encoded text only and (c) shows the image with the superimposition of the encoded text in the upper part. The text in this example contains only the following 4 words "Kidco Safeway white G2000". The size of all images is 256 × 256.

7 Conclusion

In this paper, we presented a new approach to classify text documents by transforming the word encoding obtained with Word2Vec into RGB images that main-

Table 6. Percentage error between proposed approach and single sources.

Dataset	Image	Text	**Fused Image**
Ferramenta	7.64	12.1	**5.16**
Amazon Product Data	53.9	35.9	**27.3**

tain the same semantic information contained in the original text document. The main objectives achieved are (1) the possibility of exploiting CNN models for classifying images directly without any modification, obtaining comparative results; (2) have a tool to integrate semantics of the text directly into the representative image of the text to solve a multimodal problem using a single CNN [13]. Furthermore, we presented a detailed study of various parameters associated with the coding scheme and obtained comparable results on various datasets. As shown in the section dedicated to the experiments, the results clearly show that we can further improve the text classification results by using newer and more powerful deep neural models.

References

1. Conneau, A., Schwenk, H., Barrault, L., Lecun, Y.: Very deep convolutional networks for text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. vol. 1, pp. 1107–1116 (2017)
2. Gallo, I., Nawaz, S., Calefati, A.: Semantic text encoding for text classification using convolutional neural networks. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 05, pp. 16–21 (Nov 2017). https://doi.org/10.1109/ICDAR.2017.323
3. Gallo, I., Calefati, A., Nawaz, S.: Multimodal classification fusion in real-world scenarios. In: Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on. vol. 5, pp. 36–41. IEEE (2017)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
5. He, R., McAuley, J.: Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In: proceedings of the 25th international conference on world wide web. pp. 507–517. International World Wide Web Conferences Steering Committee (2016)
6. Johnson, R., Zhang, T.: Deep pyramid convolutional neural networks for text categorization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. pp. 562–570 (2017)
7. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751. Association for Computational Linguistics (October 2014)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

9. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning. pp. 1188–1196 (2014)
10. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5188–5196 (2015)
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, pp. 3111–3119. NIPS'13 (2013)
12. Nawaz, S., Calefati, A., Janjua, M.K., Anwaar, M.U., Gallo, I.: Learning fused representations for large-scale multimodal classification. IEEE Sensors Letters **3**(1), 1–4 (2018)
13. Nawaz, S., Kamran Janjua, M., Gallo, I., Mahmood, A., Calefati, A., Shafait, F.: Do cross modal systems leverage semantic relationships? In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019)
14. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
16. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
17. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
18. Xiao, Y., Cho, K.: Efficient character-level document classification by combining convolution and recurrent layers. arXiv preprint arXiv:1602.00367 (2016)
19. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. In: Advances in neural information processing systems. pp. 5753–5763 (2019)
20. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in neural information processing systems. pp. 649–657 (2015)