Picture What you Read

Ignazio Gallo[1], Shah Nawaz[1], Alessandro Calefati[1], Riccardo La Grassa[1], and Nicola Landro[1]

[1]Department of Theoretical and Applied Science, University of Insubria, Varese, Italy
{ignazio.gallo,snawaz,a.calefati,rlagrassa}@uninsubria.it

Abstract—**Visualization refers to our ability to create an image in our head based on the text we read or the words we hear. It is one of the many skills that makes reading comprehension possible. Convolutional Neural Networks (CNN) are an excellent tool for recognizing and classifying text documents. In addition, it can generate images conditioned on natural language. In this work, we utilize CNNs capabilities to generate realistic images representative of the text illustrating the semantic concept. We conducted various experiments to highlight the capacity of the proposed model to generate representative images of the text descriptions used as input to the proposed model.**

I. INTRODUCTION

Recent years have seen a surge in multimodal data containing various media types. Typically, users combine text, image, audio or video to sell a product over an e-commence platform or express views on social media. The combination of these media types has been extensively studied to solve various tasks including classification [1], [2], [3], cross-modal retrieval [4] semantic relatedness [5], [6], image captioning [7], [8], multimodal named entity recognition [9], [10] and Visual Question Answering [11], [12]. In addition, multimodal data fueled an increased interest in generating images conditioned on natural language [13], [14]. In recent years, generative models based on conditional Generative Adversarial Network (GAN) have remarkably improved text to image generation task [15], [16]. Furthermore, generative models based on Variational Autoencoders are employed to generate images conditioned on natural language [17], [18]. Generally, image generation from natural language is divided into phases: the first phase learns the distribution from which the images are to be generated while the second phase learns a generator, which in turn produces the image conditioned on a vector from this distribution.

In this work, we are interested in transforming natural language in the form of technical e-commerce product specifications directly into image pixels. For example, image pixels may be generated from the text description such as "Heavy Duty All Purpose Hammer - Forged Carbon Steel Head" as shown in Fig. 1. We assume we are given technical specifications of a set of images available on e-commence platforms, and train the generator block, available inside our model, from the pixel distribution. We propose to use an 'up-convolutional' generative block for this task and show that it is capable of generating realistic e-commerce images. Fig. 3 shows some generated images conditioned on technical product specification along with the original images. Following are the main contributions of our work:

Fig. 1: A Neural Model reading natural language ("Heavy Duty All Purpose Hammer - Forged Carbon Steel Head") can generates a representative image ("Hammer").

- We propose a new loss function to transform a text description into a representative image;
- The proposed model generates images conditioned on technical e-commerce specifications. Moreover, it generates images never seen before;
- An end-to-end convolutional model capable of classifying the text and at the same time generating a representative image of the text. The generated image can be used as text encoding or as a realistic image representing the object described in the input document;
- We propose a model that can also be used to transform a multimodal dataset into a single dataset of images.

II. RELATED WORK

Recently, image generation conditioned on natural language has drawn a lot of attention from the research community. Various approaches have been proposed based on Variational Autoencoders [17], [18], Auto-regressive models [19] and optimization techniques [20]. Similarly, GANs based approaches have noticeably improved image synthesis conditioned on natural language. These approaches consist of a generator and a discriminator that compete in a two player minimax game: the discriminator tries to distinguish real data from generated images, and the generator tries to trick the discriminator. In the proposed model, the generator and discriminator are part of the same model and are linked by a single loss function, in order to generate images within the classification
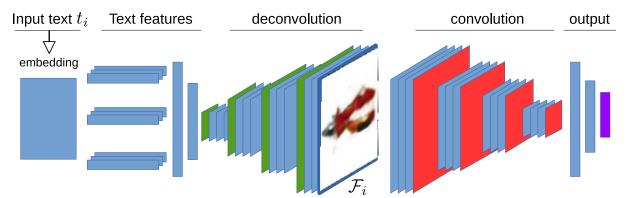
Fig. 2: A schematic representation of the proposed model. The model extracts features from the text document using an embedding layer and three different convolutive filters. Through different deconvolutive layers the textual features are transformed into an image representative of the text. Finally, through convolutive layers, the image and the encoded text are classified.

process. In the following paragraph, we reviewed couple of the ground breaking approaches on image generation conditioned on natural language.

Reed et al. [13] proposed to learn both generator and discriminator conditioned on captions. Zhu et al. [16] proposed a generative method to generate synthesized visual features using the noisy text descriptions about an unseen class. Xu et al. [21] proposed attentional generative network to synthesize fine-grained details at different subregions of the image by paying attentions to the relevant words in the natural language description. Zhang et al. [22] decomposed text to image generation in two stages: the first stage GAN sketches the basic shape and color of the object condition on the natural language, resulting in low resolution image. While the second stage GAN takes first stage results and natural language to generate high-resolution images with photo-realistic details.

Furthermore, various approaches have exploited the capability of 'up-convolutional' network to generate realistic images. Dosovitskiy et al. [23] trained a deconvolutional network with several layers of convolution and upsampling to generate 3D chair renderings given object style, viewpoint and color. In this work we are interested in generating new images through up-sampling but we limit this generative process to a medium resolution.

III. MODEL DESCRIPTION

Our goal is to train a neural network to generate accurate e-commerce images from a low-level and noisy text description. We develop an effective loss function to transform a text document into a representative image and at the same time exploits the information content of the image and the text, to solve a classification problem.

Formally, we assume that we are given a dataset of examples $D = \{t_1, \ldots, t_N\}$ with targets $O = \{(y_1, \mathcal{I}_1), \ldots, (y_N, \mathcal{I}_N)\}$. The inputs t_i are text descriptions describing the objects showed in the images $\mathcal{I}_i$. The targets are tuples consisting of two elements: the class label y_i in one-hot encoding and a RGB image $\mathcal{I}_i$.

Neural networks typically produce class probabilities by using a "softmax" output layer that converts the logit, a_i, computed for each class into a probability p_i, by comparing a_i with the other logits. Softmax function takes an n-dimensional vector of real numbers and transforms it into a vector of real number in range $[0, 1]$ which add upto 1.

$$p_i = \frac{\exp(a_i)}{\sum_j \exp(a_j)} \tag{1}$$

"Cross entropy" indicates the distance between what the model believes the output distribution should be (y_i), and what the original distribution really is. It is defined as

$$\mathcal{L}_0(y, p) = -\sum_i y_i \log(p_i) \tag{2}$$

Cross entropy measure is a widely used alternative of squared error. If we want to minimize the pixel-by-pixel distance between the input image $\mathcal{I}_i$, associated with the text document and a CNN's features layer $\mathcal{F}_i$ that has the same dimensions as the image $\mathcal{I}_i$, then we can apply the following formula

$$\mathcal{L}_1(\mathcal{F}, \mathcal{I}) = \sum_i (\mathcal{F}_i - \mathcal{I}_i)^2 \tag{3}$$

or the following mean version

$$\hat{\mathcal{L}}_1(\mathcal{F}, \mathcal{I}) = \frac{1}{N} \sum_i (\mathcal{F}_i - \mathcal{I}_i)^2 \tag{4}$$

$\mathcal{F}_i$ is the output of the last transposed convolutions – also called fractionally strided convolutions – used to upsampling the text features to attain a feature layer having the same size of the image $\mathcal{I}$.

The final loss function we used in this work is the following

$$\mathcal{L} = \mathcal{L}_0(y, p) + \lambda \mathcal{L}_1(\mathcal{F}, \mathcal{I}) \tag{5}$$

but we also performed experiments replacing the $\mathcal{L}_1$ with the $\hat{\mathcal{L}}_1$.

$$\hat{\mathcal{L}} = \mathcal{L}_0(y, p) + \lambda \hat{\mathcal{L}}_1(\mathcal{F}, \mathcal{I}) \tag{6}$$

Generated	Original	Input Text
		draper expert knipex 27723 side trimmer for electronics for cutting head satin without facet 115 mm
		spax universal screw half round head t star plus 4 cut partial shiny thread galvanized with a2j galvanization
		screws mustad panel vitals bronzed 3x20 mm. conf. 500 pcs universal countersunk flat head screw suitable for screwing ...
		sicutool padlocks 2090p 50 width body 50 mm
		sicutool copper hammers 2718 800 total weight 800 copper hammers
		sicutool nippers for electronics and fine mechanics 557gf lenght total mm 125 wire cutters for electronics and fine mechanics
		sicutool circular saws with teeth shown in hartmetall 4840g 150b type null 150b mm 150 thickness mm 2 8 hole mm 16 teeth nr 20
		valex wrench 20 x 22mm inclined forks of 15 body in chrome vanadium steel with polished finish. size 20x22 mm length 235 mm
		syrom adhesive tape in textile tes special 38 mm x 2 7 m black tightly woven plasticized tape to repair binding edges etc.
		oem 20 pcs sticker number 6 mm 50 black pvc sticker number 6

Fig. 3: Some examples of generated images from test dataset, associated with a correct classification. In some cases, the generated images are slightly different from the respective expected image but the object shown is very similar.

In this last case the contribution of the lambda parameter has a different effect with the same λ values, this because the interval of variability of $\mathcal{L}_1$ and $\hat{\mathcal{L}}_1$ are very different. For example, using the Eq. 6 as loss function, we need much larger λ values to take advantage of the contribution of $\hat{\mathcal{L}}_1$ and obtain in $\mathcal{F}$ a realistic image, representative of the object described in the text t_i.

The λ parameter is important to balance the contribution of $\mathcal{L}_0$ against $\mathcal{L}_1$. Setting $\lambda = 0$ we minimize only $\mathcal{L}_0$ and therefore the feature layer $\mathcal{F}$, representing the input text, will be very different from the image $\mathcal{I}$. In Fig. 5 we have a graphical representation of the image learned by minimizing Eq. 5, using various λ values and it is important to note that when $\lambda = 0$ the model does not generate realistic images.

Learning proceeds by minimizing the loss function $\mathcal{L}$ via Adam optimizer [24]. Since we are using the combination of two loss functions to modify the weights of the entire neural network, we know that within the image $\mathcal{F}$ that we generate, it contains text representations. In fact, in many cases it may happen that the $\mathcal{F}$ image cannot be interpreted as one of the objects belonging to a particular class but the classification is correct.

We experimented with a network for generating images of size 100×100. The structure of the generative network is shown in Fig. 2. Conceptually, the network we propose can be seen as a convolutive classification model for text documents, that incorporates a generative network. The starting point of the proposed model is the classification model of sentences proposed by Kim [25] to transform a text document into a features vector. This is followed by a set of 4 deconvolutive layers that transform textual features into an RGB image that we can then generate. Finally, we used a sequence of 4 convolutive layers that transform the generated image into features for the classification problem.

A. Text features

The input to our model are sequences of words $[w_i, \ldots, w_{|t|}]$ from each input document t, where each word is drawn from a vocabulary V. Words are represented by distributional vectors $\mathbf{w} \in \mathbb{R}^{1 \times d}$ looked up in a word embeddings matrix $\mathbf{W} \in \mathbb{R}^{d \times |V|}$. This matrix is formed by simply concatenating embeddings of all words in V.

For each input text t, we build a matrix $\mathbf{S} \in \mathbb{R}^{d \times |t|}$, where each row i represents a word embedding w_i at the corresponding position i in the document t. To capture and compose features of individual words in a given text from low-level word embeddings into higher level semantic concepts, the neural network applies a series of transformations to the input matrix S using convolution, non-linearity and pooling operations. The convolution operation between S and a filter $\mathbf{F} \in \mathbb{R}^{d \times m}$ of height m results in a vector $\mathbf{c} \in \mathbb{R}^{|t|+m1}$. In our model we used three groups of 128 different kernels in parallel, having dimensions $d \times 3$, $d \times 4$ and $d \times 5$. In this way we obtained three feature maps $\mathbf{c}_i$, having different lengths. Note that the convolution filter is of the same dimensionality

$d = 128$ as the input sentence matrix, so this is like a 1-D convolution operation. To allow the network to learn an appropriate threshold, we also added a bias vector $b \in \mathbb{R}^n$ for each feature map $\mathbf{c}_i$.

Each convolutional layer is followed by the Rectified Linear Unit (ReLU) non-linear activation function, applied element-wise. ReLU speeds up the training process [26], defined as $\max(0, x)$ to ensure that feature maps are always positive. To capture the most important feature – one with the highest value – for each feature map, we apply a max-overtime pooling operation [27] over each feature map. In this way, for each particular filter we take the maximum value as the feature corresponding to this particular filter. The convolutional layer utilizing the activation function and the pooling layer acts as a non-linear feature extractor.

Up to this point we have described the process by which a single feature is extracted from a single filter. The set of these individual features are linked into a single layer and then connected to a subsequent fully-connected layer that has the purpose of connecting the textual features with the next block of deconvolution layers used to transform textual features into an image.

B. Up-sampling

The purpose of the up-sampling block is to transform the features extracted from the text into image format that best represents the description contained in the text.

We use 4 deconvolution layers, each of which doubles the size of the input features. In practice, we start with 512 features maps of size 7×7 and then move to a second layer with 256 features maps of size 13×13, followed by a new layer having 128 features maps o size 25×25, another layer with 64 features maps of size 50×50 and finally, a last layer $\mathcal{F}$ with 3 features maps of size 100×100. The up-sampling blocks consist of the nearest-neighbor up-sampling followed by a 5×5 stride 1D convolution. Batch normalization and ReLU activation are applied after every convolution except the last one where we used a sigmoid to guarantee that the values of features maps $\mathcal{F}$ were all in the range $[0, 1]$. This last layer can be interpreted directly as an image.

C. Classification

The last block is similar to a convolutive neural network that feeds the features $\mathcal{F}$ to 4 successive convolutive layers. Each of these layers produces features maps of size 50×50, 25×25, 13×13 and 7×7 respectively. Starting from the largest layer, we used 2×2 stride and the following number of 5×5 filters: 64, 32, 16 and 8 respectively.

The convolutional layers are followed by two fully connected layers having 1024 and 512 neurons respectively.

For regularization we employ dropout before the output layer with a constraint on l_2-norms of the weight vectors. Dropout prevents co-adaptation of hidden units by randomly dropping out a proportion p of the hidden units during foward backpropagation.
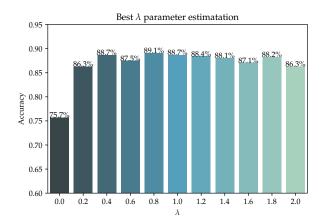

Fig. 4: 11 different executions made on the validation set of the Ferramenta dataset, to estimate the best value for the λ parameter of the proposed loss function in Eq. 5. The best value obtained is for $\lambda = 0.8$.

TABLE I: Information on multi-modal datasets used in this work. A multi-modal dataset consists of an image and accompanying text description. The last column indicates the text description language.

Dataset	#Cls	Train	Test	Lang.
Ferramenta	52	66,141	21,869	IT

The last layer is the output layer that has a number of neurons equal to the number of classes of the problem that we want to learn.

IV. DATASETS

In multimodal dataset, modalities are obtained from multiple input sources. Dataset used in this work consists of images and accompanying text descriptions. We select Ferramenta [3] multimodal dataset that are created from e-commerce website. Table I shows information on this dataset. Ferramenta multi-modal dataset [3] is made up of $88,010$ adverts split in $66,141$ adverts for train set and $21,869$ adverts for test set, belonging to 52 classes. Ferramenta dataset provides a text and a representative image for each commercial advertisement. It is interesting to note that text descriptions in this dataset are in Italian Language.

Another dataset used in our work is the Oxford-102 Flowers dataset [28] containing 8,189 flow images in 102 categories. Each image in this dataset is annotated with 10 descriptions provided by [29]. Because this dataset has class-disjoint training and test sets, with 82 train+val and 20 test classes, we randomly shuffled all the classes and split back into training and test. In this way, all the classes available in the training set are also present in the test set.

V. EXPERIMENTS

The proposed approach transforms text descriptions into a representative image. We use standard CNN hyperparameters.
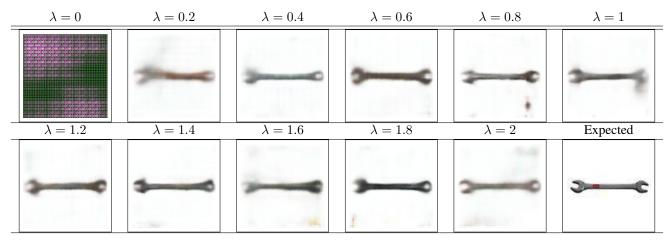
Fig. 5: Some examples of generated images from Ferramenta test dataset. The models trained for each λ by minimizing Eq. 5, were trained for 20 epochs only, to speed up the experiment.

The initial learning rate is set to 0.001 along with Adam as optimizer. In our experiments, accuracy is used to measure classification performance.

The purpose of the first experimental phase is to analyze the generative capacity of our model. We conducted following experiments with this aim in mind: (1) estimate of the best λ parameter for Eq. 5, (2) qualitative analysis of the generated images $\mathcal{F}$, (3) ability of the model to generate new images according to the description given in input.

As a second group of experiments we have analyzed the capacity of the proposed model to generate embedding in image format of the input text. We conducted following experiments with this second aim in mind: (1) estimate of the best lambda parameter to obtain the most significant encoding, (2) extraction of a new dataset of encoded text in image format to compute the classification accuracy using a well-known CNN.

The first experiment concerned the estimation of the best λ value to be used in the proposed loss function described in Eq. 5. To achieve this, we first extracted a validation set from our training set and on this we calculated the classification accuracy to extract the best value to assign to the lambda parameter. Fig. 4 shows the results of all the experiments conducted on the validation set. The accuracy results reported in this figure were obtained by averaging the accuracy values of 5 runs. As can be seen from the figure, the best value we obtained is for $\lambda = 0.8$. To visually analyze the effect of the λ parameter, we also performed a quick test by training 11 different models for 20 epochs using $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2\}$. Then we compared some of the resulting images as shown in Fig. 5. The best defined image is for $\lambda = 0.8$ while for $\lambda = 0$ we have an abstract visual representation since the second part of the loss function described in Eq. 3 has been removed.

As a second experiment, we first trained a model using the entire training set and then visually analyzed generated images with our model on the test set. Fig. 3 shows some examples of generated images beside the images we expected and the text passed as input. As you can see from the figure, many of the images generated are identical to those we expect to find, while some of them represent the same object but arranged differently (see for example the screw and the pliers with the red handle). We observed that some images have no visual meaning and do not represent any of the objects in the training set, even if in many of these cases the classification is correct. This means that the information extracted from the text is still present in the image which is then used by the last block to classify.

In generalization, the proposed model has the ability to generate images that it has never seen in training and this ability is directly correlated with the words we feed in input. In this experiment we tried to mix the tokens of two descriptions belonging to different categories to highlight the capacity. As can be seen from Fig. 8, by taking some tokens from two different descriptions and feeding them to the neural model, in some cases this produces images that are a combination of the objects representing the two descriptions. For example, in the same Fig. 8 you can see an image of an object that is the composition of a screw and a clamp. This is because in the description given as input to the model, the most important tokens of both objects are present.

Using the loss function of Eq. 6, which has a much more restricted range of variability, it is possible to give more emphasis to the encoding of the input text in image format. To find the best λ parameter that produces the best encoding we varied the parameter and for each of its values we computed the classification accuracy on the test set of the Ferramenta and Flowers datasets. Fig. 7 shows the encodings and the accuracy obtained when the parameter changes. On these results we performed the last experiment using the $\lambda = 6$ parameter.

In this latest experiment we extracted two new image datasets using two different models trained on the two datasets. In Fig. 6 you can see some examples of images generated, alongside the original image of the dataset. It can be seen

Fig. 6: Columns (a) and (b) show some examples of images extracted from Flowers and Ferramenta datasets, respectively. The columns to the right of (a) and (b) show the text encodings extracted as features layer $\mathcal{F}$ using $\lambda = 6$ in Eq. 6.
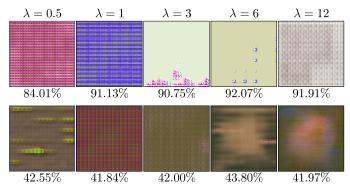

$\lambda = 0.5$	$\lambda = 1$	$\lambda = 3$	$\lambda = 6$	$\lambda = 12$
84.01%	91.13%	90.75%	92.07%	91.91%
42.55%	41.84%	42.00%	43.80%	41.97%

Fig. 7: The top row contains encodings for the same image belonging to the Ferramenta dataset when the model was trained using Eq. 6 with λ parameters showed on the top. The images on bottom row are created using an image of the Flowers dataset. Below each image is the test accuracy obtained with the corresponding λ parameter.

how different source images correspond to a different text encoding. To analyze the information content of these two new datasets we have trained two CNN AlexNet to calculate their classification accuracy. For the Ferramenta dataset we got 93.68% while for the Flowers dataset we got 99.05%. The first result is slightly higher than the one published in [30] while the second result obtained on the Flowers dataset is incredibly high. The reason we got such high accuracy is because our dataset contains 10 different text descriptions associated with the same image. Having divided the training set randomly into training and test, the same images can be found both in the training set and in the test set. Ultimately this means that the new datasets created do not only encode information extracted from the text but also from images.

VI. CONCLUSION

In this work we have proposed a new approach to generate an image that is representative of a noisy text description available in natural language. The approach we proposed uses a new loss function in order to simultaneously minimize the classification error and the distance between the desired image and a features map of the same model. The qualitative results are very interesting but, for the moment, we have ignored the classification performances because this was not our focus of the present work. In the future we want to exploit the same idea to try to improve the classification accuracy that can be obtained with a single convolutive neural model.

Another interesting aspect emerged from this work is that the same approach we proposed can be used to encode in image format both the information contained in the input text and the information extracted from the image associated with the text. This feature is very interesting to be able to incorporate multimodal information into a single image dataset. In this way, multimodal information can be processed directly by a single CNN normally used to process only images.

REFERENCES

[1] J. Arevalo, T. Solorio, M. Montes-y Gómez, and F. A. González, "Gated multimodal units for information fusion," *arXiv preprint arXiv:1702.01992*, 2017.

[2] D. Kiela, E. Grave, A. Joulin, and T. Mikolov, "Efficient large-scale multi-modal classification," *Proceedings of AAAI 2018*, 2018.

[3] I. Gallo, A. Calefati, and S. Nawaz, "Multimodal classification fusion in real-world scenarios," in *Document Analysis and Recognition (ICDAR)*. IEEE, 2017, pp. 36–41.

[4] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5005–5013.

[5] D. Kiela and L. Bottou, "Learning image embeddings using convolutional neural networks for improved multi-modal semantics," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 36–45.

[6] C. W. Leong and R. Mihalcea, "Going beyond text: A hybrid image-text approach for measuring word relatedness." in *IJCNLP*, 2011, pp. 1403–1407.

[7] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128–3137.
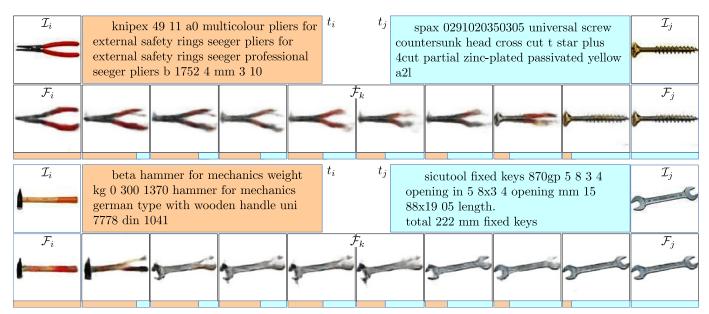
Fig. 8: A set of $\mathcal{F}_k$ images generated by the proposed model. $\mathcal{F}_i$ and $\mathcal{F}_j$ have generated starting from text documents t_i and t_j respectively. All other $\mathcal{F}_k$ images were generated by combining different percentages of tokens extracted simultaneously from t_i and t_j. The two colored rectangles below $\mathcal{F}_k$ images are indicative of the percentages of tokens from t_i and t_j used to generate the $\mathcal{F}_k$ images.

[8] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[9] Q. Zhang, J. Fu, X. Liu, and X. Huang, "Adaptive co-attention network for named entity recognition in tweets," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[10] O. Arshad, I. Gallo, S. Nawaz, and A. Calefati, "Aiding intra-text representations with visual context for multimodal named entity recognition," *arXiv preprint arXiv:1904.01356*, 2019.

[11] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding," *Proceedings of Empirical Methods in Natural Language Processing, EMNLP 2016*, pp. 457–468, 2016.

[12] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *CVPR*, vol. 3, no. 5, 2018, p. 6.

[13] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016.

[14] M. Cha, Y. Gwon, and H. Kung, "Adversarial nets with perceptual losses for text-to-image synthesis," in *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2017, pp. 1–6.

[15] S. Hong, D. Yang, J. Choi, and H. Lee, "Inferring semantic layout for hierarchical text-to-image synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7986–7994.

[16] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal, "A generative adversarial approach for zero-shot learning from noisy texts," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1004–1013.

[17] E. Mansimov, E. Parisotto, J. Ba, and R. Salakhutdinov, "Generating images from captions with attention," in *ICLR*, 2016.

[18] E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov, "Generating images from captions with attention," *arXiv preprint arXiv:1511.02793*, 2015.

[19] S. Reed, A. van den Oord, N. Kalchbrenner, S. G. Colmenarejo, Z. Wang, Y. Chen, D. Belov, and N. de Freitas, "Parallel multiscale autoregressive density estimation," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2912–2921.

[20] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski, "Plug & play generative networks: Conditional iterative generation of images in latent space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4467–4477.

[21] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1316–1324.

[22] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5907–5915.

[23] A. Dosovitskiy, J. T. Springenberg, M. Tatarchenko, and T. Brox, "Learning to generate chairs, tables and cars with convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 692–705, 2016.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP*, 2014, pp. 1746–1751.

[26] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[27] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[28] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 2008, pp. 722–729.

[29] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, pp. 1060–1069.

[30] I. Gallo, A. Calefati, S. Nawaz, and M. K. Janjua, "Image and encoded text fusion for multi-modal classification," in *2018 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Dec 2018, pp. 1–7.