Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

Image Captioning using Deep Learning

Bhamidi Haripriya
ISE Department
BNMIT,Bangalore, India

Syed Haseeb
ISE Department
BNMIT, Bangalore, India

Srushti G M
ISE Department
BNMIT, Bangalore, India

Mrs. Madhura Prakash
Asst. Prof ISE Department
BNMIT Bangalore,India

***Abstract*—In the last few years, deep learning has led to huge success in the field of computer vision and natural language understanding and also in the interplay between them. Among different types of deep learning models, convolutional neural networks have been most extensively studied for the tasks related to visual perception and machine vision. Due to lack of computational resources and training data, it is very hard to the use high-capacity convolutional neural network without overfitting. But recent growth in the availability of annotated data and high-performance GPUs have made it possible to obtain state-of-the-art results using convolutional neural networks.**

Automatically describing the content of an image is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. In this project, a generative model based on a deep recurrent architecture that combines recent advances in computer vision and machine translation is being used. Recurrent architecture is used to generate natural sentences describing an image. The model will be trained to maximize the likelihood of the target description sentence given the training image.

Keywords—Deep Learning, Image captioning, Convolution Neural Network, MSCOCO, Recurrent Nets, Lstm, Resnet.

I. INTRODUCTION

A recent study on Deep Learning shows that it is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Deep Learning (DL) and Neural Network (NN) is currently driving some of the most ingenious inventions in today's century. Their incredible ability to learn from data and environment makes them the first choice of machine learning scientists. Deep Learning and Neural Network lies in the heart of products such as self-driving cars, image recognition software, recommender systems etc. Evidently, being a powerful algorithm, it is highly adaptive to various data types as well.

Image annotation is a process by which a computer system assigns metadata in the form of captioning or keywords to a digital image. It is a Type of multi-class image classification with a very large number of classes. It is used in image retrieval systems to organize and locate images of interest from the database. The goal of image captioning research is to annotate and caption an image which describes the image using a sentence. To train a network to accurately describe an input image by outputting a natural language sentence. The task of describing any image sits on a continuum of difficulty. Some images, such as a picture of a dog, an empty beach, or a bowl of fruit, may be on the easier end of the spectrum. While describing images of complex scenes which require specific contextual understanding and to do this well, not just possibly proves to be a much greater captioning challenge. Providing contextual information to networks has been both a sticking point, and a clear goal for researchers to strive for.

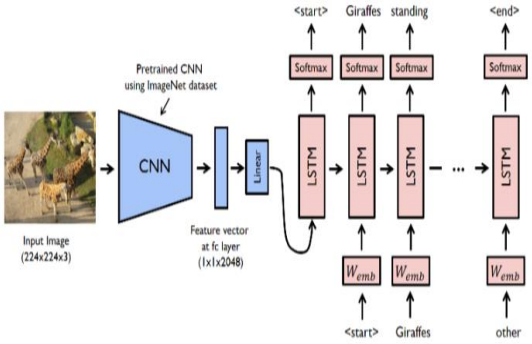


Figure 1. Architectural Design

Our model to caption images are built on multimodal recurrent and convolutional neural networks. A Convolutional Neural Network is used to extract the features from an image which is then along with the captions is fed into an Recurrent Neural Network. The architecture of the image captioning model is shown in figure 1.

Image captioning is interesting because it concerns what we understand and about perception with respect to machines. The problem setting requires both an understanding of what features (or pixel context) represent which objects, and the creation of a semantic construction "grounded" to those objects.

II. RELATED WORK

This paper presents how convolutional neural network based architectures can be used to caption the contents of an image. Captioning here means labelling an image that best explains the image based on the prominent objects present in that image. Deep convolutional neural networks based machine learning solutions are now days dominating for such image annotation problems [1, 2]. Recent researches in [3, 4] has proposed solution that automatically generates human-like description of any image. This problem is of significance in practical applications and moreover it link two artificial intelligence areas i.e. NLP (Natural Language Processing) and Computer Vision. The convolutional network architecture that won the ImageNet Challenge in 2012, have been used for large-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

scale image and video recognition [8]. It uses the large public image repositories, such as ImageNet [11], and high-performance computing systems, such as GPUs or large-scale distributed clusters. In ImageNet LSVRC-2010 contest, 1.2 million high resolution images are classified into the 1000 different classes using Deep Convolutional neural network. The top-1 and top5 error rates achieved using this model are 37.5% and 17.0% respectively and this is substantially better than the state-of-the-art. The neural network architecture which consists of 650,000 neurons with 60 million parameters contains five convolutional layers. Some of these layers were followed by max-pooling layers and three fully-connected layers with a final 1000-way SoftMax layer. With the applications of ConvNets growing rapidly in the computer vision community, a number of attempts have been made to improve the original architecture proposed in [8], in a bid to achieve better accuracy [7]. One of the most recent work in [3] uses an algorithm which learns the semantics very selectively and fuse them into hidden and output states of RNN.

III. MODELS

A. *CONVOLUTIONAL NEURAL NETWORK*

A very deep convolutional neural network model is proficient in extracting visual features from an image in a hierarchical manner, starting from very basic features like edge detectors, and then progressively building more complex features like shape detection. In this section we describe the components of a basic ConvNet model, and show how these different modules are arranged in a network which leads to efficient extraction of visual features.

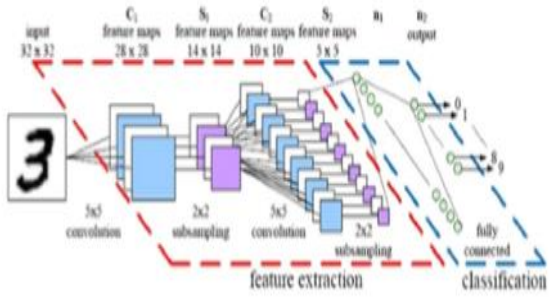


Figure 2. Convolutional neural network architecture

CNNs use a variation of multilayer perceptron designed to require minimal preprocessing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics. A CNN consists of an input and an output layer, as well as multiple hidden layers .The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers. Certain pre-processing steps needed for efficient working of the CNN model are:

a. *Data pre-processing*

The only pre-processing we do here is Mean Subtraction. The mean RGB value is computed on the training set and subtracted from each pixel. Normalization is ignored because normalization only have meaning if different input features have different scales, but they should be of approximately equal importance to the learning algorithm. Mean subtraction is the most common form of pre-processing. It has the geometric interpretation of centering the cloud of data on the origin along every dimension. With images specifically, for convenience it can be common to subtract a single value from all pixels, or to do so separately across the three color channels. By applying Mean Subtraction we get corresponding mean values for RGB [123.682, 116.779 and 103.939]. Same is explained with an example in figure 3.

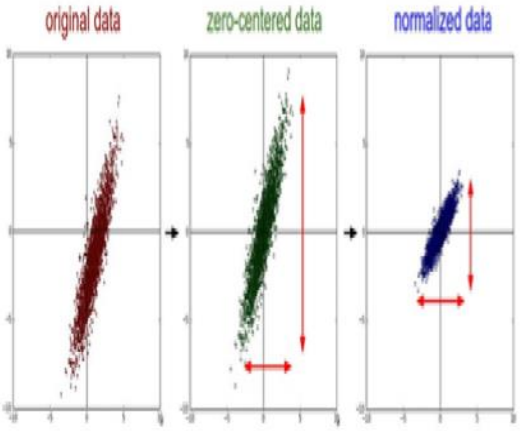


Figure 3. Data pre-processing

b. *Convolutional layer*

Convolutional Neural Networks (ConvNets) are nearly same as regular neural networks. They consist of neurons with weights and biases as learning parameters. A non-linear activation function is applied on inputs of each neuron. The complete network manifests a single differentiable score function from one end pixels of image to class scores at the other. And they have a loss function (e.g. SVM/SoftMax) on the last layer i.e. fully-connected layer. All other steps as in ordinary neural networks also apply in ConvNet in same fashion. Unlike of typical Neural Networks, an explicit consideration in ConvNet architectures is that the images itself are the input to the network. And prominent properties of the images are encoded in the network is the peculiarity of architecture. Images being the input to ConvNet constrain the architecture in a more sensible way. In particular, the neurons in layers of ConvNet are arranged in all three dimensions i.e. width, height and depth which is uncommon to that in trivial neural networks. For example, the input images have the volume dimensions224x224x3.The final output layer dimension is reduced to 1x1x1000, because full input image is transformed into a single vector of class scores (arranged along the depth dimension), moving towards the end of the ConvNet architecture.

c. *Local connectivity*

When dealing with high-dimensional inputs such as images, connecting neurons in two successive volumes fully is very impractical. That is why each neuron is only connected to neurons in a local region of the input volume. The spatial extent of this connectivity is a hyper parameter

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

called the 'receptive field' of the neuron (equivalently this is the filter size). The extent of the connectivity along the depth axis is always equal to the
depth of the input volume. It is important to emphasize again on the asymmetry that how we treat the spatial dimensions i.e. width, height and the depth. The connections are local in space (along width and height), but always full along the entire depth of the input volume.

d. Zero padding

Zero-padding is the process of adding zeroes to the input matrix in symmetric fashion. This kind of modification is most common practice to adjust the size of the input according to our requirement. Same is customarily followed in CNN layers to preserve the input volume dimensions to be used at the output volume. The image is passed through a stack of convolutional (conv.) layers, where we use filters with a very small receptive field of 3x3. This is the smallest size to capture the notion of left/right, up/down, and center. The convolution stride is fixed to 1 pixel; the spatial padding of convolutional layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1 pixel for 3 3 conv. layers. The figure 4 presents the working of 3x3 filter. Rather than using relatively large receptive fields in the first convolutional layers e.g.11x11 with stride 4, or 7x7 with
stride 2, very small 3x3 receptive fields are used throughout the whole net, which are convolved with the input at every pixel (with stride 1). So a stack of two 3x3 convolution layers(without spatial pooling in between) has an effective receptive field of 5x5; three such layers have a 7x7 effective receptive field. The advantage of using a stack of three 3x3 convolutional layers instead of a single 7x7 layer are: • First, incorporating 3 non-linear rectification layers instead of a single one, makes the decision function more discriminative. • Second, there is a decrease in the number of parameters: assuming both the input and the output of a three layer 3x3 convolution stack has C channels, it can be mathematically shown that using filters of 3x3 will eventually lead to smaller number of parameters than using 5x5 or 7x7.

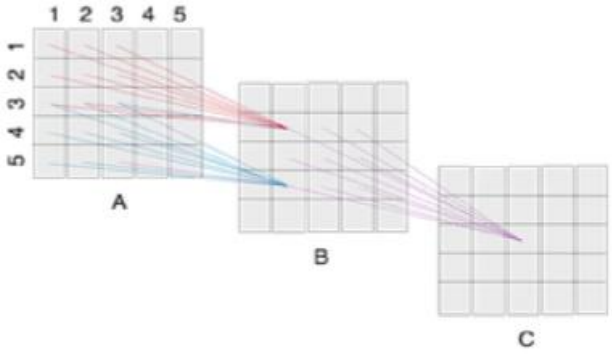


Fig. 4. Working of a 3x3 filter.

e. *Max Polling layer*

Max Pooling layer is very commonly used layer in the Convolutional network architecture. It is used after applying Convolution to the input as it reduces the dimensions of the layer that is given as an input to the layer. As the name suggests the function of the layer is to take input and return the maximum of all the pixels in the range of the filter applied. Mostly small filters are used as using large filters may destroy the image and the effect the classification drastically in an unwanted manner. Max polling example is shown in figure 5.

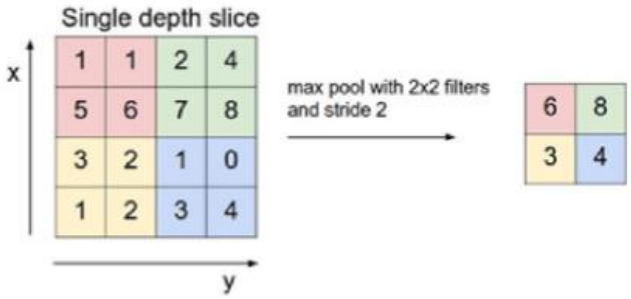


Figure 5. Max pooling operation

f. *Fully connected layers*

In a fully connected layer all the neurons of the input are connected with all the neurons of the output of the layer. In this layer all the neurons have connections with all the activations in the previous layer. They are computationally costly in terms of both memory and time. Activations of the neurons can be computed using matrix multiplication and followed by a bias offset.

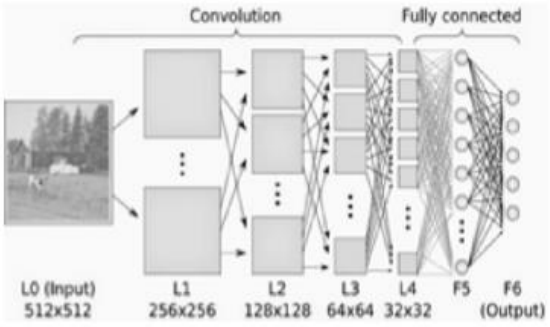


Figure 6. Fully connected layers

g. *ReLu (Rectified linear unit)*

ReLu refers to the Rectifier Unit, the most commonly deployed activation function for producing non-linearity in ConvNets. In the context of artificial neural networks, the rectifier is an activation function defined as f(x)=max(0;x); Where x is the weighted input to a neuron. It has been used in convolutional networks more effectively than the widely used logistic sigmoid, and its more practical counterpart, the hyperbolic tangent. This is mainly because of its sparse activation and efficient gradient propagation (no vanishing or exploding gradient problems).

According to the universal approximation theorem, given enough capacity, we know that a feedforward network with a single layer is sufficient to represent any function. However, the layer might be massive and the network is prone to overfitting the data. Therefore, there is a common trend in the research community that our network architecture needs to go deeper. The authors argue that stacking layers shouldn't degrade the network performance, because we could simply stack identity mappings (layer that doesn't do anything) upon the current network, and the resulting architecture would perform the same. This indicates that the deeper model should not produce a training error higher than its shallower counterparts. They hypothesize that letting the stacked layers

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

fit a residual mapping is easier than letting them directly fit the desired underlaying mapping. And the residual block above explicitly allows it to do precisely.

As a matter of fact, ResNet was not the first to make use of shortcut connections, Highway Network introduced gated shortcut connections. These parameterized gates control how much information is allowed to flow across the shortcut. Similar idea can be found in the Long Term Short Memory (LSTM) cell, in which there is a parameterized forget gate that controls how much information will flow to the next time step. Therefore, ResNet can be thought of as a special case of Highway Network.

However, experiments show that Highway Network performs no better than ResNet, which is kind of strange because the solution space of Highway Network contains ResNet, therefore it should perform at least as good as ResNet. This suggests that it is more important to keep these "gradient highways" clear than to go for larger solution space.

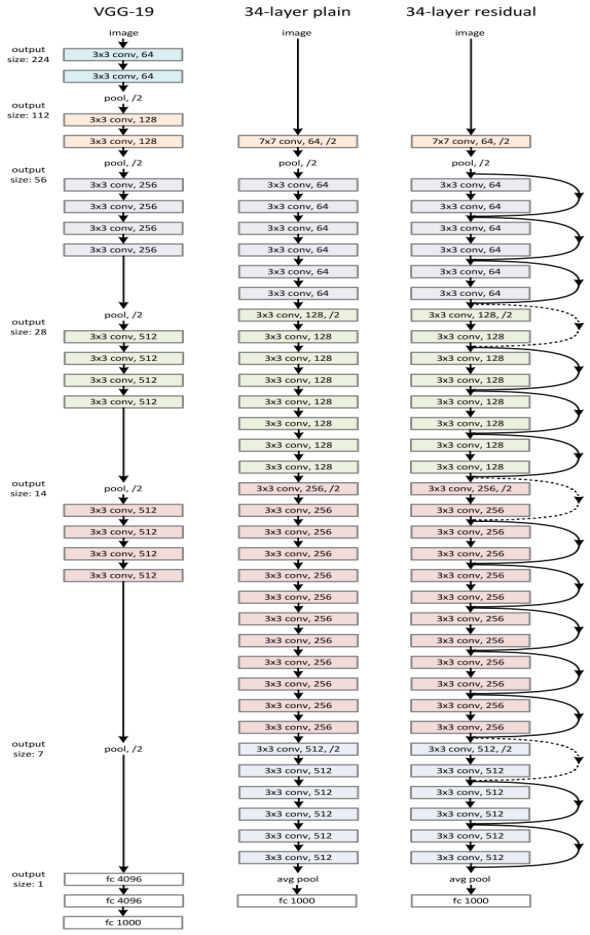


Figure 7. ResNet Architecture

Plain Network. Our plain baselines (Figure 7, middle) are mainly inspired by the philosophy of VGG nets [4] (Figure 7,left). The convolutional layers mostly have 3×3 filters and follow two simple design rules: (i) for the same output feature map size, the layers have the same number of filters; and (ii) if the feature map size is halved, the number of filters is doubled so as to preserve the time complexity per layer. We perform down sampling directly by convolutional layers that have a stride of 2. The network ends with a global average pooling layer and a 1000-way fully-connected layer with softmax. The total number of weighted layers is 34 in Figure 7 (middle).
It is worth noticing that our model has fewer filters and lower complexity than VGG nets [4] (Figure 7, left). Our 34-layer baseline has 3.6 billion FLOPs (multiply-adds), which is only 18% of VGG-19 (19.6 billion FLOPs).

Residual Network. Based on the above plain network, we insert shortcut connections (Figure 7,right) which turn the network into its counterpart residual version. The identity shortcuts can be directly used when the input and output are of the same dimensions (solid line shortcuts in Figure 7). When the dimensions increase , we consider two options:
(A) The shortcut still performs identity mapping, with extra zero entries padded for increasing dimensions. This option introduces no extra parameter; (B) The projection shortcut in Eqn.(2) is used to match dimensions(done by 1×1 convolutions). For both options, when the shortcuts go across feature maps of two sizes, they are performed with a stride of 2.

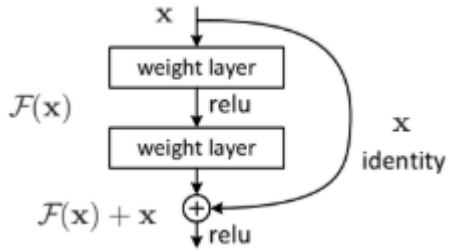


Figure 8. A residual model

In this paper, we address the degradation problem by introducing a deep residual learning framework. Instead of hoping each few stacked layers directly fit a desired underlying mapping, we explicitly let these layers fit a residual mapping. Formally, denoting the desired underlying mapping as H(x), we let the stacked nonlinear layers fit another mapping of F(x) := H(x)−x. The original mapping is recast into F(x)+x. We hypothesize that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping. To the extreme, if an identity mapping were optimal, it would be easier to push the residual to zero than to fit an identity mapping by a stack of nonlinear layers.

The formulation of F(x) +x can be realized by feedforward neural networks with "shortcut connections" (Figure 8). Shortcut connections [2, 3, 4] are those skipping one or more layers. In our case, the shortcut connections simply perform identity mapping, and their outputs are added to the outputs of the stacked layers (Fig. 2). Identity shortcut connections add neither extra parameter nor computational complexity. The entire network can still be trained end-to-end by SGD with backpropagation, and can be easily implemented using common libraries (e.g., Caffe [9]) without modifying the solvers. We present comprehensive experiments on ImageNet[5] to show the degradation problem and evaluate our method. We show that: 1) Our extremely deep residual nets are easy to optimize, but the counterpart "plain" nets (that simply stack layers) exhibit higher training error when the depth increases; 2) Our deep residual nets can easily enjoy

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

accuracy gains from greatly increased depth, producing results substantially better than previous networks.

Similar phenomena are also shown on the CIFAR-10 set, suggesting that the optimization difficulties and the effects of our method are not just akin to a particular dataset. We present successfully trained models on this dataset with over 100 layers, and explore models with over 1000 layers. On the ImageNet classification dataset [5], we obtain excellent results by extremely deep residual nets.

B. RECURRENT NEURAL NETWORK

In this paper, we propose a neural and probabilistic framework to generate descriptions from images. Recent advances in statistical machine translation have shown that, given a powerful sequence model, it is possible to achieve state-of-the-art results by directly maximizing the probability of the correct translation given an input sentence in an "end-to-end" fashion – both for training and inference. These models make use of a recurrent neural network which encodes the variable length input into a fixed dimensional vector, and uses this representation to "decode" it to the desired output sentence. Thus, it is natural to use the same approach where, given an image (instead of an input sentence in the source language), one applies the same principle of "translating" it into its description. Thus, we propose to directly maximize the probability of the correct description given the image by using the following formulation:

$$\theta^{\star} = \arg\max_{\theta} \sum_{(I,S)} \log p(S|I;\theta) \qquad (1)$$

where θ are the parameters of our model, I is an image, and S its correct transcription. Since S represents any sentence, its length is unbounded. Thus, it is common to apply the chain rule to model the joint probability, where N is the length of this particular example as

$$\log p(S|I) = \sum_{t=0}^{N} \log p(S_t|I, S_0, \ldots, S_{t-1}) \qquad (2)$$

where we dropped the dependency on θ for convenience. At training time, (S,I) is a training example pair, and we optimize the sum of the log probabilities as described in (2) over the whole training set using stochastic gradient descent.

It is natural to model with a Recurrent Neural Network (RNN), where the variable number of words we condition upon up to t − 1 is expressed by a fixed length hidden state or memory ht. This memory is updated after seeing a new input xt by using a non-linear function f:

$$h_{t+1} = f(h_t, x_t) \; . \qquad (3)$$

To make the above RNN more concrete two crucial design choices are to be made: what is the exact form of f and how are the images and words fed as inputs xt. For f we use a Long-Short Term Memory (LSTM) net, which has shown state-of-the art performance on sequence tasks such as translation. This model is outlined in the next section. For the representation of images, we use a Convolutional Neural Network (CNN). They have been widely used and studied for image tasks, and are currently state-of-the art for object recognition and detection. Our particular choice of CNN uses a novel approach to batch normalization and yields the current best performance on the ILSVRC 2014 classification competition [12]. Furthermore, they have been shown to generalize to other tasks such as scene classification by means of transfer learning [4]. The words are represented with an embedding model.

C. LSTM-based Sentence Generator

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

The choice of f in (3) is governed by its ability to deal with vanishing and exploding gradients [10], the most common challenge in designing and training RNNs. To address this challenge, a particular form of recurrent nets, called LSTM, was introduced [10] and applied with great success to translation [3, 13] and sequence generation [9]. The core of the LSTM model is a memory cell c encoding knowledge at every time step of what inputs have been observed up to this step (see Figure 2) . The behavior of the cell is controlled by "gates" – layers which are applied multiplicatively and thus can either keep a value from the gated layer if the gate is 1 or zero this value if the gate is 0. In particular, three gates are being used which control whether to forget the current cell value (forget gate f), if it should

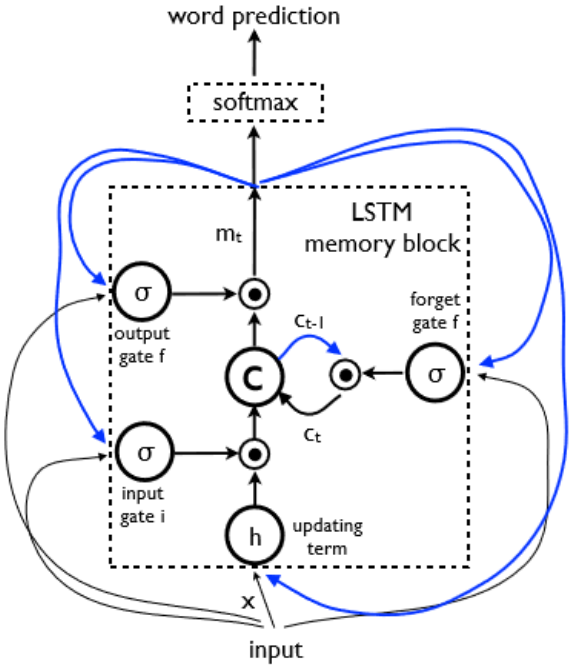


Figure 9. LSTM: the memory block contains a cell c which is controlled by three gates. In blue we show the recurrent connections – the output m at time t − 1 is fed back to the memory at time t via the three gates; the cell value is fed back via the forget gate; the predicted word at time t − 1 is fed back in addition to the memory output m at time t into the SoftMax for word prediction.

read its input (input gate i) and whether to output the new cell value (output gate o). The definition of the gates and cell update and output are as follows:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1}) \quad (4)$$
$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1}) \quad (5)$$
$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1}) \quad (6)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot h(W_{cx}x_t + W_{cm}m_{t-1}) \quad (7)$$
$$m_t = o_t \odot c_t \quad (8)$$
$$p_{t+1} = \text{Softmax}(m_t) \quad (9)$$

where $\odot$ represents the product with a gate value, and the various W matrices are trained parameters. Such multiplicative gates make it possible to train the LSTM robustly as these gates deal well with exploding and vanishing gradients [10]. The nonlinearities are sigmoid σ(·) and hyperbolic tangent h(·). The last equation mt is what is used to feed to a SoftMax, which will produce a probability distribution pt over all words.

IV. METHODOLOGY

The first step involves collecting properly annotated data, large enough so that a complex model trained on it will give satisfactory results. For this purpose, we use MSCOCO dataset, which contains about 1.2 million images of 1000 different categories. The second and the most challenging part include training the model after deciding what its architecture would be. The training phase actually involves two phases, a pre-training phase followed by a fine tuning phase. Among these, the pre-training phase is more challenging and involves much more computational resources than the fine tuning phase because it involves training a model from the scratch, while fine tuning just involves slight modifications to the network's parameters so as to use the network for some different application. We use pre-trained weights of the ResNet model. So, in the training part we train the model to classify an image as belonging to one of the 1000 classes and generate a caption.

A. Training

For any machine learning algorithm to perform the desired task, we've to train the model to do so. By training we mean that we've to obtain the optimal values of the parameters of the model (weights and biases in our case), which will generalize well to perform the required task. The ConvNet training procedure generally follows [8]. Namely, the training is carried out by optimizing the multinomial logistic regression objective using mini-batch gradient descent (based on backpropagation [12]) with momentum. The training was regularized by weight decay and dropout regularization for the first two fully connected layers (dropout ratio set to 0.5). The learning rate was initially set to 102, and then decreased by a factor of 10 when the validation set accuracy stopped improving. The learning was stopped after 370K iterations (74 epochs). The model is trained for the image captioning task.

B. Dataset

COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features: Object segmentation, recognition in context etc.The COCO dataset is an excellent object detection dataset with 80 classes, 80,000 training images,41,000 testing images and 40,000 validation images.

C. Testing

Once the model has been trained for the image classification and captioning task, it can now be used for the object detection phase. The image can contain large objects as well as relatively smaller objects. Our task is to detect all of them, not just the larger objects. For that purpose, we progressively divide the image and classify it as belonging to one of the 1000 classes. We start with the whole image, feed it into the ConvNet architecture. The resulting class mostly represents the most significant (in terms of size) object in the image. Now we divide the image into two equal halves, and feed these two halves into the ConvNet. The idea is that these two halves as separate images will mostly contain a different object as the most prominent object. We repeat this experiment by further dividing these images into 2 halves and classifying them. This step may further detect smaller objects which are prominent in the divided image. Finally, we repeat this step one last time to further divide the image into two halves and classifying them. So, in total we have 15 images of different sizes, formed from the original image. For all the above-mentioned steps, we display top 2 classes per image and threshold the probabilities to threshold. What this means is that we'll only show the second class if its probability exceeds more than threshold i.e. it is a prominent object in the image. Here threshold is given as an input by the user while feeding the image to the Convolutional network. For measuring the accuracy, precision and recall of the model, we define the following terms:

- Correctly matched: True positive (the objects that were detected by the model and were actually present in the image).
- Mistakenly matched: False positive (the objects that were detected by the model but were actually not present in the image).
- Correctly rejected: True negative (the objects that were not detected by the model and were actually not present in the image).
- Mistakenly rejected: False negative (the objects that were not detected by the model but were actually present in the image).

Figure 10. Progressive division and classification of the sample image

For each threshold value, we measure the accuracy, precision and recall of our model on different images and take the average. Below is the plot showing variation of accuracy, precision and recall with different threshold values.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

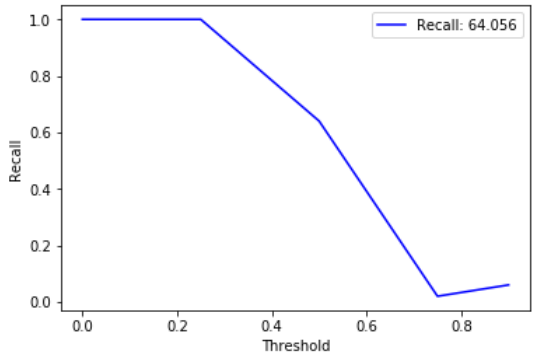


Figure 11. Graph showing the variation of model's recall with threshold values

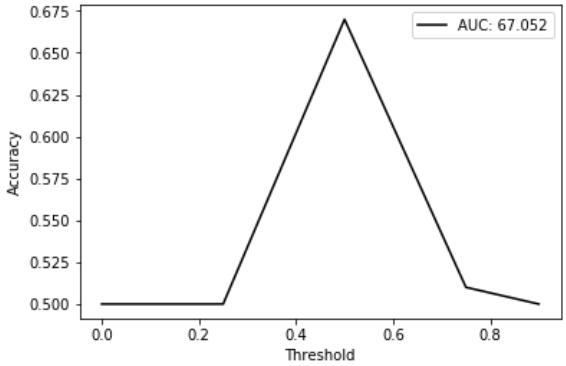


Figure 12. Graph showing the variation of model's accuracy with threshold values

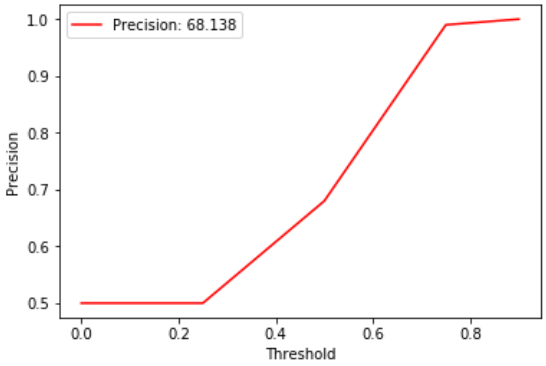


Figure 11. Graph showing the variation of model's precision with threshold values

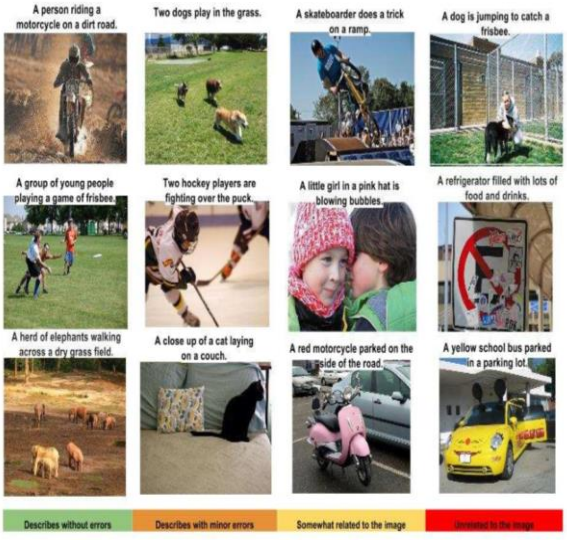
D. Evaluation Metrics

Although it is sometimes not clear whether a description should be deemed successful or not given an image, prior art has proposed several evaluation metrics. The most reliable (but time consuming) is to ask for raters to give a subjective score on the usefulness of each description given the image. In this paper, we used this to reinforce that some of the automatic metrics indeed correlate with this subjective score.

The metrics can be computed automatically assuming one has access to ground truth, i.e. human generated descriptions. The most commonly used metric so far in the image description literature has been the BLEU score [5], which is a form of precision of word n-grams between generated and reference sentences 2. Even though this metric has some obvious drawbacks, it has been shown to correlate well with human evaluations.

V. RESULTS

Many of the challenges that we faced when training our models had to do with overfitting. Indeed, purely supervised approaches require large amounts of data, but the datasets that are of high quality have less than 100000 images. The task of assigning a description is strictly harder than object classification and data driven approaches have only recently become dominant. As a result, we believe that, even with the results we obtained which are quite good, the advantage of our method versus most current human-engineered approaches will only increase in the next few years as training set sizes will grow. Nonetheless, we explored several techniques to deal with overfitting. The most obvious way to not overfit is to initialize the weights of the CNN component of our system to a pretrained model . We did this in all the experiments (similar to [8]), and it did help quite a lot in terms of generalization. Another set of weights that could be sensibly initialized are We, the word embeddings. We tried initializing them from a large news corpus [2], but no significant gains were observed, and we decided to just leave them uninitialized for simplicity. Lastly, we did some model level overfitting-avoiding techniques. We tried dropout [4] and ensemble models, as well as exploring the size (i.e., capacity) of the model by trading off number of hidden units versus depth. Dropout and ensemble gave a few BLEU points improvements, and that is what we report throughout the paper.



VI. CONCLUSION

An end-to-end neural network system that can automatically view an image and generate a reasonable description in plain English. It is based on a convolution neural network that encodes an image into a compact representation, followed by a recurrent neural network that generates a corresponding sentence. The model is trained to maximize the likelihood of the sentence given the image. Experiments on several datasets show the robustness in terms of qualitative results (the generated sentences are very reasonable) and quantitative evaluations, using either ranking metrics or

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

BLEU, a metric used in machine translation to evaluate the quality of generated sentences. It is clear from these experiments that, as the size of the available datasets for image description increases, so will the performance of the proposed model. Furthermore, it will be interesting to see how one can use unsupervised data, both from images alone and text alone, to improve image description approaches.

REFERENCES

[1] Zeiler, Matthew D., and Rob Fergus. ”Visualizing and understanding convolutional networks.” European conference on computer vision. Springer International Publishing, 2014.

[2] Simard, Patrice Y., David Steinkraus, and John C. Platt. ”Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis.” ICDAR. Vol. 3. 2003.

[3] You, Quanzeng, et al. ”Image captioning with semantic attention.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[4] Oriol, Vinyals, et al. ”Show and tell: A neural image caption generator.” arXiv preprint arXiv: 1411.4555, 2014 (2014).

[5] Young, Peter, et al. ”From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions.” Transactions of the Association for Computational Linguistics 2 (2014): 67-78.

[6] Karen Simonyan, Andrew Zisserman, Very Deep Convolutional Networks ForLarge-ScaleImageRecognition,InInternationalConferenceonLearning Representation, 2015.

[7] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. ”Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems. 2012.

[8] Karpathy, Andrej, and Li Fei-fei. ”visual-semantic alignments for generating image descriptions. arXiv preprint arXiv: 1412.2306.” (2014).

[9] Wei, Yunchao, et al. ”CNN: Single-label to multi-label.” arXiv preprint arXiv:1406.5726 (2014).

[10] Deng, Jia, et al. ”Imagenet: A large-scale hierarchical image database.” Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference.

[11] Deng, Jia, et al. ”Imagenet: A large-scale hierarchical image database.” Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference.

[12] LeCun, Yann, et al. ”Backpropagation applied to handwritten zip code recognition.” Neural computation 1.4 (1989): 541-551.

[13] Singh, Dushyant Kumar. ”Recognizing hand gestures for human computer interaction.” Communications and Signal Processing (ICCSP), 2015 International Conference on. IEEE, 2015.