Descriptive Image Captioning

Ishita Ankit[a], Shreya Jain[b]

[a] *MTH,IIT Kanpur*
[b] *CSE, BITS Pilani, Dubai*

Abstract

Image Captioning aims to fill up the gap between visual and language interpretations of objects scenes and so on. This shall find wide application for instance enabling the robots to interpret the image that they see. The state of the art work in this field was achieved by Dense Captioning[1]. While this model aims to produce small captions for every region proposal, we try to join the broken captions to make a meaningful description for the image. Apart from replicating the results of densecap, we refined the captions by reducing the region proposals and predicted the preposition joining the incomplete captions. The captions and the prepositions were then passed through an encoder-decoder model[3] trained on phrases and sentences to generate meaningful descriptions.

1. Introduction

Image captioning is generation of textual descriptions corresponding to a particular image. Humans can identify objects in the images and produce a grammatically correct sentence associated to it with ease and without any extra effort. Human produced captions are quite apt and have even minute details of the image. It can be attributed to the extremely advanced and complex human vision. It takes only seconds for humans to perceive data while even now it is challenge for computers. The computer sees any image in the form of a 3D array of numbers or what we call pixels. Assuming an image we wish to caption is of the size 300 pixels wide and 300 pixels tall with three channels(red, green, and blue). Thus, a computer would view the image as 270,000 numbers. Range of the numbers is from 0 to 255, where 0 is to define black and 255 to define white.

The research work done evolves around translating these quarter million numbers to a sequence of words composed correctly.In near future, it has been estimated that 85 percent of the data on the internet would be comprised of pixels only. This boom of visual data is majorly due to internet as carrier of data as well as excessive images captured by the sensors these days. This form of data is also referred to as dark matter of the internet as it the most difficult data to observe and understand. To be able to harness any useful information from this huge amounts of visual data, it is necessary that it labelled and captioned for any other application. This can no longer be done through humans. It is only advanced computer vision techniques that can solve the problem. Further, the research work can be implemented to improve image search and tell stories for photo ablum uploads. This work if developed enough could help the visually impaired to perceive the visual data one day.

2. Model Architecture

2.1. Descriptive Image Captioning

A lot of work has been done in the field of image captioning. Many models have been proposed to solve this problem. Approaches to deal with language part has changed significantly over time. Initially, for the language model researchers heavily relied on categories or pre-specified sentences where words could be filled in. Since a couple of years, research works have emphasised on generated language. The kind

Our work focused on associating enriched sentences instead of just one line captions to an images. Our model to solve the problem statement of generating captions for visual data brings together areas of Computer Vision, Natural Language Processing and Deep Neural Networks. For the computer vision side, researchers train their systems on a massive dataset of images, so they learn to identify objects in images. Many models to deal with the vision part have been devised so a pre-trained Convolutional Neural Network is used for this. Language models can then be used to put these words together. This model is built with the goal to associated with images. Our work is majorly paying more focus on words as it could be the key to form a story corresponding the visual data.

2.2. DenseCap - Fully Convolutional Localisation Networks for Dense Image Captioning[1]

The research work in DenseCap: Fully Convolutional Localisation Networks for Dense Image Captioning is done Dr. Fei Fei Li, Dr. Andrej

Karpathy and Dr. Justin Johnson at the Stanford University. The work accomplished to narrow down regions of most interest in the image and describe them. It is computationally efficient and produces best results than any of the previous works done to achieve the solution for the same problem statement.

2.2.1. Convolution Layer

Introduction to CNN - The Convolution Neural Network are powerful deep neural networks made with sole purpose of solving various vision problems. Neural Network would flatten images for feed-forward, leading to loss of spatial structure. CNN keeps all its features intact, which help when learned to help understand the picture better.

The architecture of the CNN can be defined in the following layers-

- Input layer takes the raw pixel values of the image, which would have a width, height and three colour channels (red, green and blue).
- Conv layer is used to calculate dot product of weights of input with a small region they are connected to in the input volume. This may result in voluminous data of height, weight and depth. The height and weight can be changed using stride and padding values. Usually, it is kept same as the input layer. The depth of the result is the number of filters used in the Conv layer.
- Nonlinearity is introduced in the networks as the complexity of these problems cannot be rightly represented linearly. For this functions like Relu, Leaky REly, Tanh, Sigmoid, etc. are used.
- As in most networks, conv layers retain the height and width of the image, it is important to use Pooling layer. It is used to perform downsampling operations to reduce dimensions of height and width. Two kinds of pooling is done - Max and Average to get the desired dimensions.
- Many layers of Convolution, Non-linearity and Pooling are arranged hierarchically to get the best results.
- A Fully Connected Layer is attached in the end which gives the final output. If the layer is used for classification it would give answer as vector with class scores.

The input of this layer is an image, which is preprocessed using a CNN to obtain feature tensor of the image. A pre trained VGG-16[7] was founded by Karen Simonyan and Andrew Zisserman. It is composed of 13 layers of 3x3 convolutions (stride 1 and pad 1) and 5 layers of 2x2 max pooling (stride 2) is used to extract the needed features. The last fully connected layer is removed and the features are used further. Assume we start with an input image of size WxH and depth is 3 (RGB channels). After running it through the convolutional layer, a tensor of features of shape CxWxH where C is the no. of filters used, H=H/16 and W=W/16.

2.2.2. Localisation Layer

It most essentially identifies spatial regions of interest and smoothly extracts a fixed sized representation from each region. Input used here is the tensor of activations of size C x H x W obtained from the previous layer. It internally selects B regions and returns three output tensors :

- Region Coordinates : Matrix Bx4 giving bounding box coordinates for each output region
- Region Scores : Vector of length B with confidence score of each region.
- Region Features :Tensor of shape BxCxXxY giving features for output regions

Each point in Wx H grid of input is projected back into WxH image plane and k anchor boxes of different sizes were centered at this projected point. For each k box, a confidence score and four coordinates are predicted. The input feature map is passed through a 3x3 conv with 256 filters, then ReLU source of non-linearity is introduced, further putting it through 1x1 with 5k filters to get WxHx5k. Box regression is applied with coordinates, width and height of the center used to predict scalars to normalize offsets and log space transforms to output region has center and shape.Box Sampling is used to cut down the way too many region proposals, so necessary to subsample them. In training time, a mini batch of B=256 boxes is selected with at most B/2 positive regions and rest negatives. In test time, B=300 of most confident proposals is used.Bilinear Interpolation is used as the size of each of these vectors needs to be reduced to fixed length.

2.2.3. Recognition Model

In the recognition layer, first, features from each region are flattened into a vector. Then as it is passed through 2 Fully Connected Layers, each with ReLu (source of non-linearity) and regularized using Dropout. Dropout technique is commonly used to avoid over-fitting of models. Each region produces a code of dimension D=4096 that compactly encodes its visual appearance.The codes for all positive regions is collected and put in matrix BxD which passed to the language model.

This layer is also responsible to fine tune the model on the this dataset. In the convolution layer having used a pre trained we acquired a set of features on a different dataset. To get a good output the dataset needs to fine tune on the currently used dataset. Thus, the confidence and position of each proposed region is reevaluated.

2.2.4. Language Model

Introduction to RNN/LSTM

When forming sequence of words, humans do not think of every word from scratch rather follow a continuous chain of thought. SImilar approach is followed by RNN to generate language. These networks are self-looped to introduce the functionality of persistence. During gradient back propagation, in a RNN the gradient is multiplied to its weight matrix as many times as the number of timesteps. Thus, the magnitude of the weights can severely affect the learning process. Low magnitude of weights lead to vanishing gradient where the gradient signal becomes so small that no learning takes place. While very high magnitudes can lead to exploding gradient, in such a case the high gradient signal is at risk to diverge easily. The language model uses a special type of recurrent neural network called the Long Short Term Memory, which outperforms RNN these days. It was introduced in 1997 by Hochreiter amp; Schmidhuber and was popularised later due to the results produced by it in the works that followed its discovery. It is preferred over RNN due to vanishing gradient as well as exploding gradient problems. It has better designed to deal with these problems as it has capability to learn long-term dependencies. The LSTM solves the above problems of RNN as it is designed to have a memory cell - which includes four neural network layers that interact in a unique and significant way at the same time. The first step in the LSTM is to decide which information, we will throw from the cell state. This decision is taken by the forget gate layer. It looks at h(t-1) and x(t) and outputs a number between 0 (completely get rid of this) to 1 (completely keep this).

$$f_t = \sigma(W_t[x_t, h_{t-1}] + b_f) \tag{1}$$

The next step is to decide what new information were going to store in the cell state. This has two parts. First, a sigmoid layer called the input gate layer decides which values well update. Next, a tanh layer creates a vector of new candidate values that could be added to the state. In the next step, well combine these two to create an update to the state.

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i) \tag{2}$$

$$C'_t = tanh(W_c[x_t, h_{t-1}] + b_c) \tag{3}$$

Its now time to update the old cell state, Ct-1 into the new cell state C$_t$.

$$C_t = i_t * C'_t + f_t * C_{t-1} \tag{4}$$

This output will be based on our cell state, but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state were going to output. Then, we put the cell state through tanh and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

$$o_t =_0 [x_t, h_{t-1}] + b_0) \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

LSTM used in Language Model

- Training sequence of tokens s1, s2, ., st feed the LSTM + 2 word vectors $x_{-1}, x_0, x_1, x_2, \ldots, x_t$ where x_{-1}=CNN(I) x_0 is start token.
- LSTM computes a sequence of hidden states ht and output vector yt using recurrence formula ht,yt=f(ht-1,xt)
- Output vector size is V+1 where V is the token vocabulary and one is special END token.
- Targets at time t=0,1,.T-1 are token indices for st+1
- At test time x-1 info is fed to the network. At each step we sample the most likely next token and feed it to the next time step, repeating until END token is reached.

2.3. Combining Geometric, Textual and Image Features to predict prepositions

Even though the DenseCap model achieved state of the art when it came to region wise captioning of the image. It is still quite difficult to still make sense of these captions and the image as a whole. To a third person these snippets would make more sense if linked to each other properly. To extend the done work, we are aiming to fill the gaps between these captions of the overlapping bounding boxes to coorelate them better.

Predicting the most suitable prepositions using combination of geometrical, textual and visual features was implemented to link the bounding boxes. This method was used a previous published research Combining Geometric, Textual and Visual Features for Predicting Prepositions in Image Descriptions. This method may be not enough on its own to define captions completely; it had capability to provide a strong extension to DenseCap. The work predicts prepositions in the following way

In the dataset of Visen[2] made after preprocessing Flickr30, we were given for every overlapping set of bounding boxes, their coordinates and describing each box as landmark and trajectory and the preposition connecting them. From this we made a set of all possible prepositions P, set of landmarks as L and set of trajectors as T. For example, for the phrase on would be the preposition, bicycle the landmark and person the trajectory. The research work used the landmark and trajectory entities present in an image to discover the role of visual features and geometric configurations played in predicting the preposition connecting the two.

The textual features are then used as Word2Vec or OneHot encodings. Then, 11 geometrical features were computed from the given coordinates of the boxes and image size and were then incoorporated in input used for the prediction. The geometrical features derived include ratio of area, aspect ratio, area of each bounding box with respect to enclosing area, intersection over union, euclidean distance between two, area of each bounding box with respect to the image, etc. Visual features are used by using ImageNet for classifying the main object of bounding box if the entity of landmark and trajector is not given.But in case of DenseCap, the parser obtains these entities. A simple multi-class logistic regression was used to all this as input and output a preposition. An overall accuracy of about 67% was achieved by applying on dataset Flickr30.

2.3.1. Predicting Prepositions for overlapping bounding boxes in Densecap[2]

The output obtained from DenseCap contained overlapping bounding boxes per image with brief captions. The first thing required by this method was to only find the significant overlapping bounding boxes in an image.

To apply it to Densecap, the given model had to be modified as we were only interested in overlapping bounding boxes and these boxes were also not completely overlapping in nature. For this changes were made in the overlapping ratio of the selected boxes. The function of overlapping ration is defined as the ratio of area of intersection of the boxes over their area of union of them. This was changed to give ratio area of of intersection of boxes over minimum area of the two boxes. Only when this ratio was less the one then the boxes were picked. To reduce the number of proposals and improve quality, the fixed numbers of most confident proposals were only picked. This model was then completely trained to get the proposals and the captions corresponding to them. These boxes along with caption are stored in a text file. To be able to predict the preposition it was important to pick the relevant word from captions of the boxes. The captions obtained per bounding box had to be passed through Stanford Dependency Parser[6] to obtain the landmark and trajectory.

2.4. Language Model to produce rich language[3]

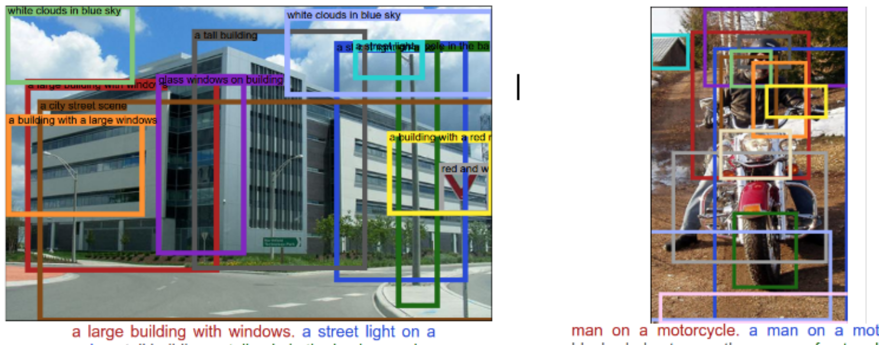
The language model is basically an LSTM encoder-decoder for a sequence-to-sequence model. In the basic model depicted above, every input has to be encoded into a fixed-size state vector, as that is the only thing passed to the decoder. It is used mainly in machine translation problems. We need to train the encoder decoder model to learn the english grammar so that it shall produce meaningful sentences from incomplete captions. We used the Flickr30K dataset containing phrases and sentences to train the model.

For test data, we have a set of captions for every image corresponding to the refined overlapping bounding boxes and the preposition connecting them. Every pair of captions along with predicted prepositions is the input to the LSTM model to generate descriptive sentences.

3. Results

3.1. Densecap Results

The pre-trained model was evaluated on the Visual Genome Dataset[5] to replicate the results as presented in the paper. The self trained model was

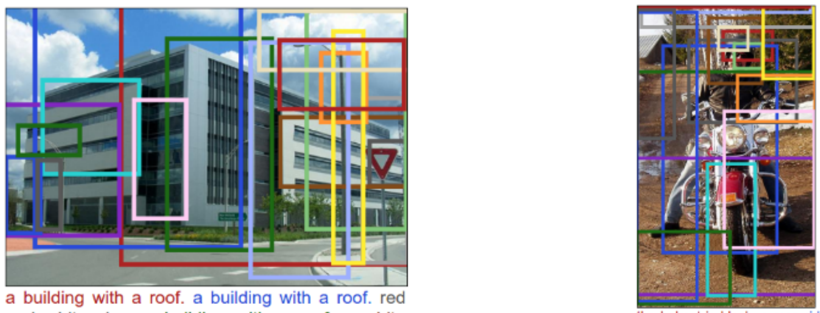


white clouds in blue sky
a tall building
white clouds in blue sky
a large building with windows
glass windows on building
a city street scene
a building with a large windows
a building with a red roof
red and white sign

a large building with windows. a street light on a pole. a tall building. a tall pole in the background. a city street scene. glass windows on building. red and white sign. white clouds in blue sky. white clouds in blue sky. a street light. a building with a large windows. a building with a red roof. a building with a large windows. a stop sign on a pole. a tree in the distance. a tall pole. a small green bush.

man on a motorcycle. a man on a motorcycle. black helmet on the man. front wheel of motorcycle. man wearing black jacket. trees behind the fence. a red motorcycle. man wearing sunglasses. dirt road. a small house in the background. man wearing black jacket. a bike rack. headlight on the front of the motorcycle. dirt on the ground. the bike is black. dirt on the ground. back wheel of motorcycle.

Figure 1: Captions generated using Pre-trained Model



a building with a roof. a building with a roof. red and white sign. a building with a roof. a white building with a red and white sign. a building with a roof. a sign on the street. a tall street light. a tall building. a white building. a tall pole. a tall pole. the sky is cloudy. a window on the building. a tall pole. a tree in the background. a tall green tree. a white building.

the helmet is black. a man riding a motorcycle. a man wearing a black helmet. trees behind the trees. a man wearing a helmet. the motorcycle is black. the tree is green. the man is wearing a hat. the tree is green. the front wheel of a motorcycle. a black and white bench. trees in the background. the helmet is black. a motorcycle parked on the road. the tree is green. a large brown dirt. a large rock. the ground is brown.

Figure 2: Captions generated using Self-trained Model

also evaluated to get the results as mentioned in table 1. The output of the model is shown in Fig.1 and Fig.2.

3.2. Results of Combining Geometric, Textual and Image Features to predict prepositions

The results of densecap was refined to produce captions from bounding boxes with no enclosure(fig3). These captions were then run through a parser to obtain the Landmark and trajector used to predict the preposition. The

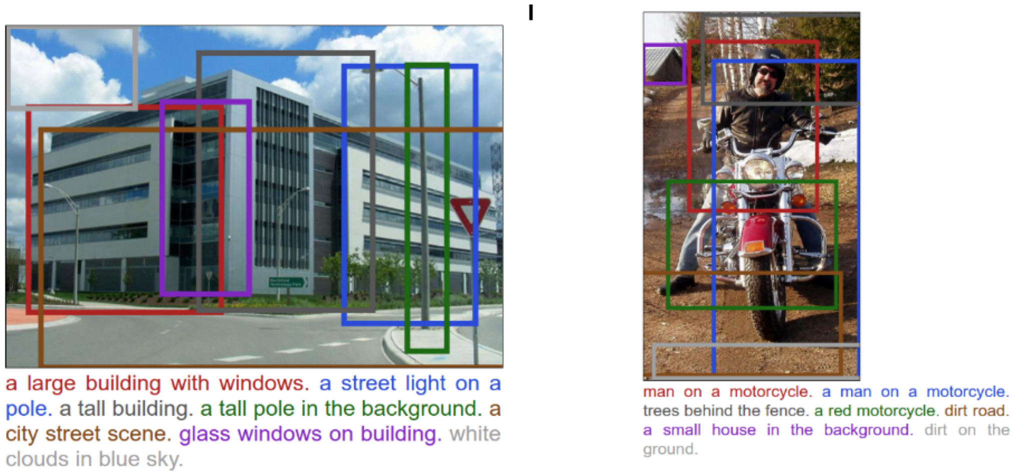

a large building with windows. a street light on a pole. a tall building. a tall pole in the background. a city street scene. glass windows on building. white clouds in blue sky.
man on a motorcycle. a man on a motorcycle. trees behind the fence. a red motorcycle. dirt road. a small house in the background. dirt on the ground.

Figure 3: Refined captions by reducing bounding boxes

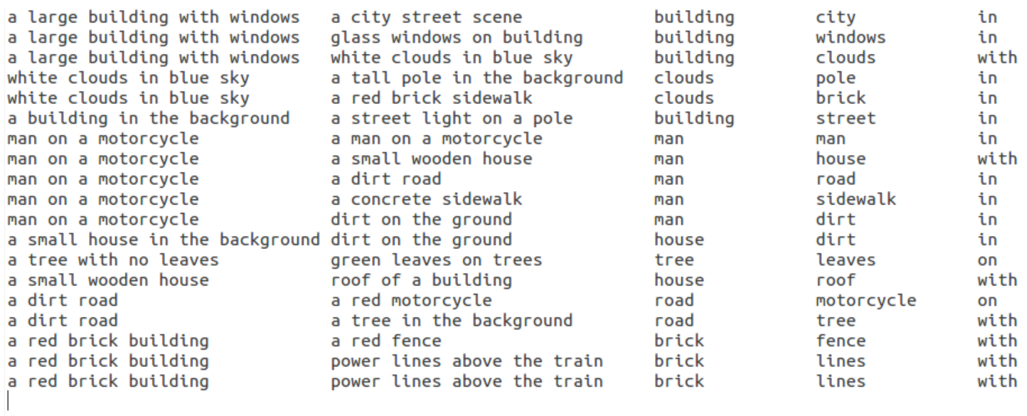

```
a large building with windows   a city street scene           building      city          in
a large building with windows   glass windows on building     building      windows       in
a large building with windows   white clouds in blue sky      building      clouds        with
white clouds in blue sky        a tall pole in the background clouds        pole          in
white clouds in blue sky        a red brick sidewalk          clouds        brick         in
a building in the background    a street light on a pole      building      street        in
man on a motorcycle             a man on a motorcycle         man           man           in
man on a motorcycle             a small wooden house          man           house         with
man on a motorcycle             a dirt road                   man           road          in
man on a motorcycle             a concrete sidewalk           man           sidewalk      in
man on a motorcycle             dirt on the ground            man           dirt          in
a small house in the background dirt on the ground            house         dirt          in
a tree with no leaves           green leaves on trees         tree          leaves        on
a small wooden house            roof of a building            house         roof          with
a dirt road                     a red motorcycle              road          motorcycle    on
a dirt road                     a tree in the background      road          tree          with
a red brick building            a red fence                   brick         fence         with
a red brick building            power lines above the train   brick         lines         with
a red brick building            power lines above the train   brick         lines         with
```

Figure 4: Prepositions predicted for every pair of captions for an image

model for predicting preposition was evaluated on Flickr30k dataset[8] with high level category classes as well as majority head. We obtainied a score of **67 %** against the score of 70 percent reported in the paper. This model was then tested on the refined captions to produce results as shown in fig4.

4. Future Work

The present encoder decoder has been modified from a Machine Translation Model to sentence generation from phrases. The dataset used for now is not apt as the phrases do not contain prepositions. We propose to use a better dataset with phrases and prepositions to generate sentences. Such a dataset is not available online and thus, needs to be made. Further,we can implement stacked LSTMs to produce better results.

5. Acknowledgement

We would like to extend our earnest gratitude to Prof. Vinay Namboodiri for guiding us through out the project. A sincere gratitude to Prof. B.V. Rathish Kumar and Dr.Sujala Shetty for overlooking the project.

6. References

[1] Johnson, Justin and Karpathy, Andrej and Fei-Fei, Li DenseCap: Fully Convolutional Localization Networks for Dense Captioning Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,2016

[2] Ramisa, Arnau and Wang, Josiah and Lu, Ying and Dellandrea, Emmanuel and Moreno-Noguer, Francesc and Gaizauskas, Robert. Combining Geometric, Textual and Visual Features for Predicting Prepositions in Image Descriptions.EMNLP 2015.

[3] Minh-Thang Luong Hieu Pham Christopher D. Manning.Effective Approaches to Attention-based Neural Machine Translation.EMNLP 2015.

[4] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik, Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models, IJCV, 2016

[5] Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li Jia-Li, David Ayman Shamma, Michael Bernstein, Li Fei-Fei.

[6] Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60

[7] Karen Simonyan Andrew Zisserman.Very Deep Convolutional Networks for Large-Scale Image Recognition.Published as a conference paper at ICLR 2015

[8] Peter Young Alice Lai Micah Hodosh Julia Hockenmaier From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions.Transactions of the Association for Computational Linguistics (to appear)