Image Captioning Using Flickr 8K Dataset

Omar Gaballah 201801697
Ahmed AbdelSalam 201801597
Ibrahim Hamada 201800739
Sohaila Zaki 201800998

Contents:

Problem Definition

01

Problem Definition:

- Generating meaningful sentences in the form of human-written text.

- Image captioning has plenty of applications such that using image captions to create an application to help people who have low or no eyesight.

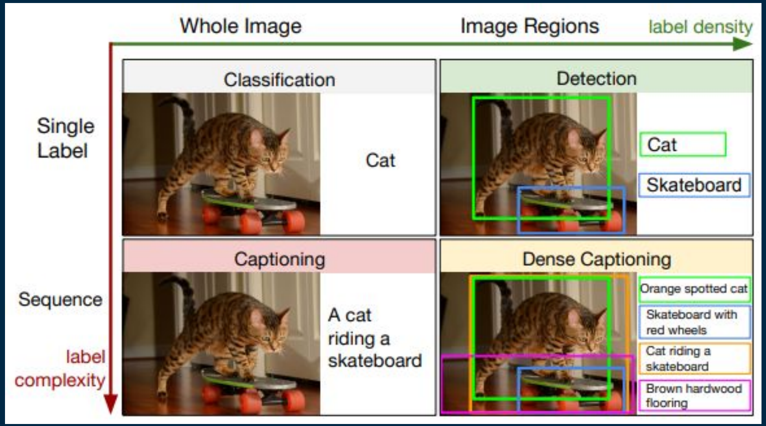


Figure 1. Image Captioning

Dataset

02

Flickr30K:

- It has been widely used in the field of sentence-based image description.
- The dataset consists of 31,783 images that capture people engaged in everyday events and activities.
- Each image is annotated using 5 different sentences by human annotators.
- Accordingly, the dataset contains around 158,000 captions for 31,000 images.

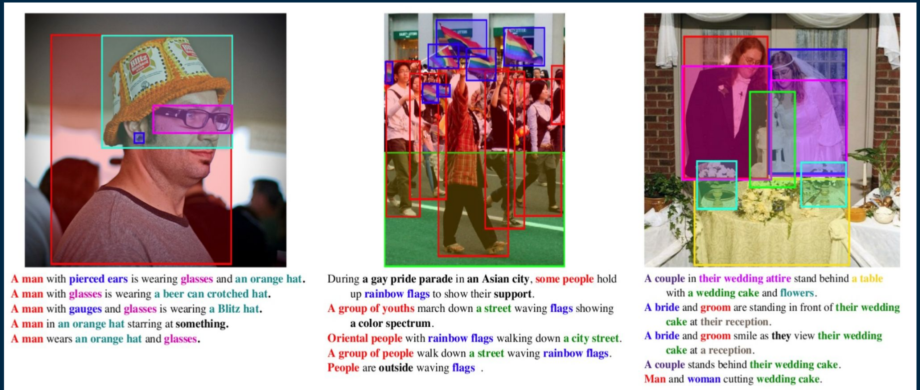


Figure 2. Some Dataset Examples

Image Model 03

ResNet50:

- Our chosen architecture to get image features and represent images.

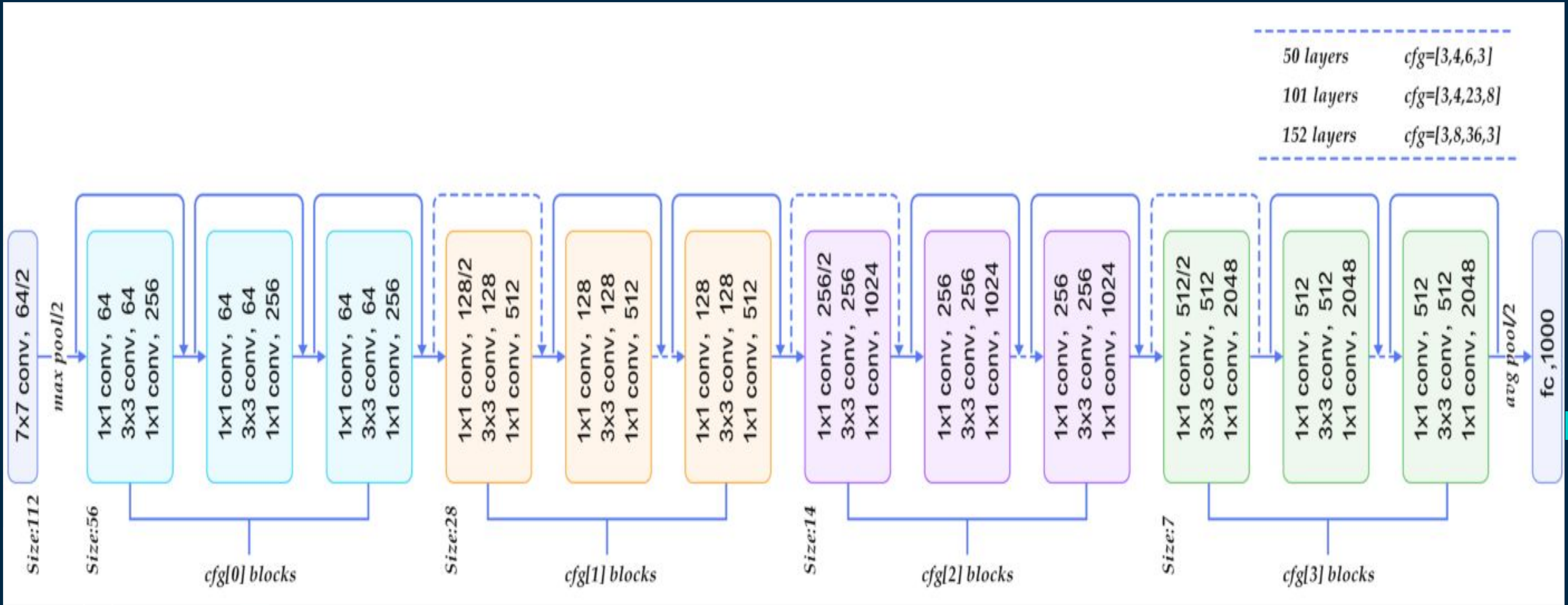


Figure 3. ResNet50 Architecture

Why ResNet50?

- (Canziani et al.) compared between computational complexity and accuracy.
- ResNet50 offers one of the best trade-off between complexity and accuracy

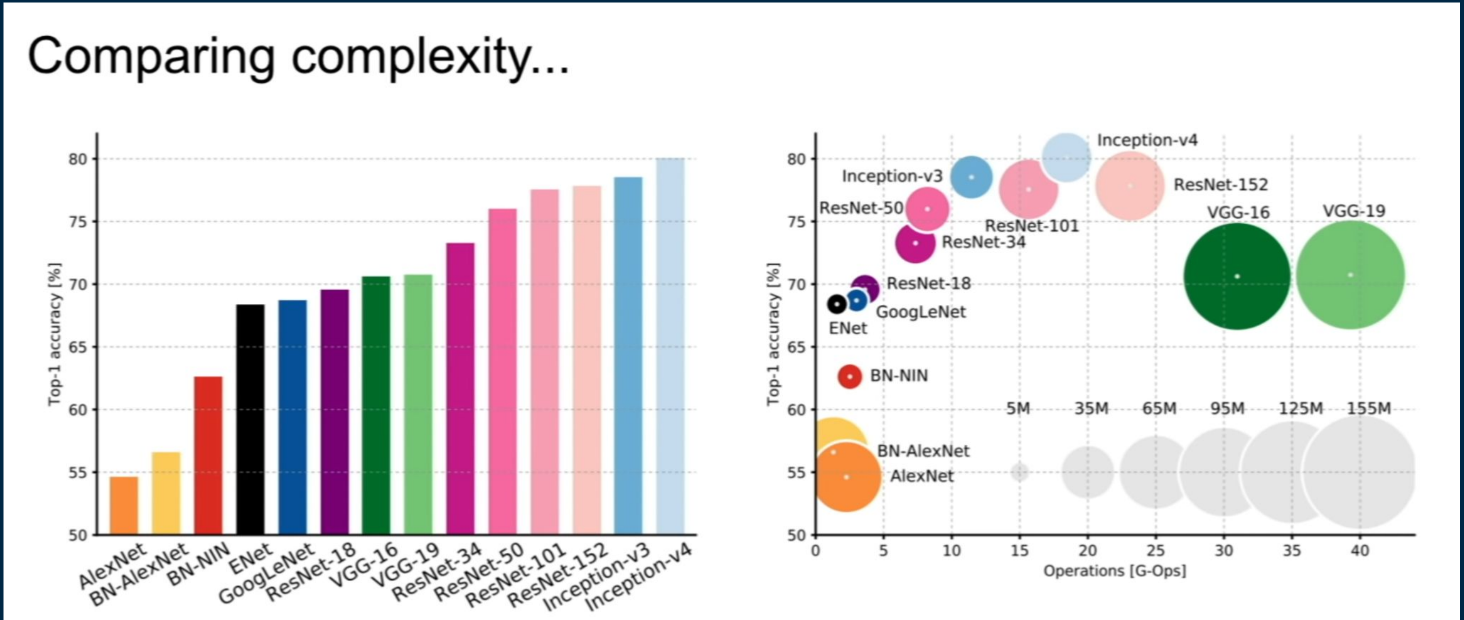


Figure 4. Comparison between different Deep Neural Network Models

Implementation:

- Image Resizing to 224*224
- Fed the resized images to ResNet50 Model
- ResNet50 Model's Last Layer is dropped
- The second last layer will be considered as our output layer from the image model.

```
conv5_block3_out (Activation)   (None, 7, 7, 2048)   0          conv5_block3_add[0][0]
__________________________________________________________________________________________________
avg_pool (GlobalAveragePooling2 (None, 2048)         0          conv5_block3_out[0][0]
==================================================================================================
Total params: 23,587,712
Trainable params: 0
Non-trainable params: 23,587,712
```

Language Model

04

Language model: Agenda

1. Preprocessing
2. Tokenizer
3. Extracting semantic features

Preprocessing

1. Lowercasing all the words
2. Adding start and end tokens in every caption
3. Determine maximum length of tokens in a single caption

Tokenization

1. Use keras Tokenizers
2. Determine the vocab size
3. Create Dictionary with image ID as a key and corresponding of vectorized captions as values

Language model

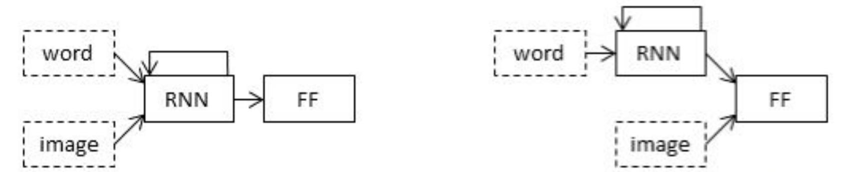
```
________________________________________________________________
Layer (type)                 Output Shape              Param #
================================================================
embedding_1 (Embedding)      (None, 80, 128)           2343936
________________________________________________________________
lstm_1 (LSTM)                (None, 80, 256)           394240
________________________________________________________________
time_distributed_1 (TimeDist (None, 80, 128)           32896
================================================================
Total params: 2,771,072
Trainable params: 2,771,072
Non-trainable params: 0
```

Full Architecture

05

Types of Architecture:

- In our problem, we have two inputs, the image we need to describe which will be feed to the CNN and the words which are produced as a sequence as the input to the RNN.
- Thus, we need to create a multimodal neural network.
- Since, we are dealing with two types of information, When should we introduce the image data vectors in the language model (RNN) ?
- We have two types of architecture, Injecting architecture and merging architecture.



(a) Conditioning by injecting the image means injecting the image into the same RNN that processes the words.

(b) Conditioning by merging the image means merging the image with the output of the RNN after processing the words.

Figure 5. Merging and injecting architecture

Types of Architecture:

- In the Injecting Architecture, The RNN trains on a mixture of language data and image data that is represented together.
- So RNN uses the mixture to predict the next word and finetunes the image information as well during the training at every step
- In the Merging Architecture, The image and the language data is encoded separately then presented together in a feed-forward network thus, creating a multimodal layer architecture

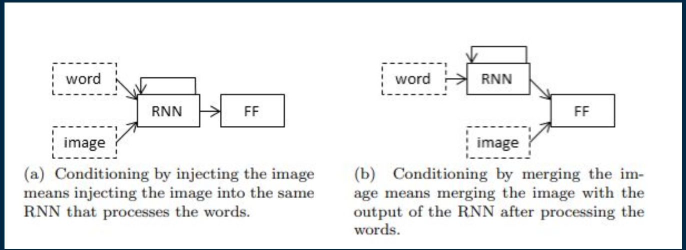
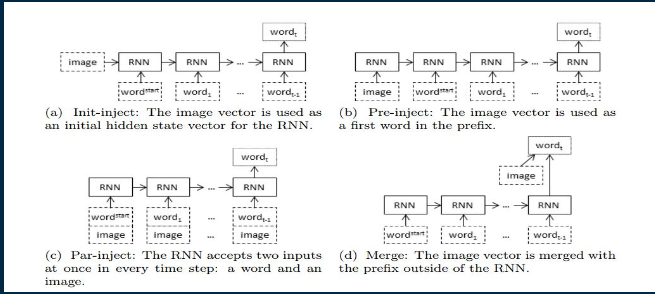


Figure 6. Merging and injecting architecture

Types of Architecture:

- The injection architecture has 3 variants, Init-inject, Pre-inject and Par-inject.
- We pass the image vector as the initial state of the RNN in the Init-inject.
- We pass the image vector as the first word in the Pre-inject. In the first time step, we send the image vector.
- We merge the image vector and the word vector into a similar-sized embedding space and pass it to the RNN.



(a) Init-inject: The image vector is used as an initial hidden state vector for the RNN.
(b) Pre-inject: The image vector is used as a first word in the prefix.
(c) Par-inject: The RNN accepts two inputs at once in every time step: a word and an image.
(d) Merge: The image vector is merged with the prefix outside of the RNN.

Figure 7. Different types of architecture

Our Architecture:

- In our Model, We are going to use the Pre-injection architecture initially.
- Then we are going to try a different architecture.

BLEU Score

06

BLEU Score

- BLEU scores were used widely as a metric to test the performance of Image captioning models.
- BLEU score gives an output between 0 and 1 to compare the similarities between some reference sentences and the translated one.

Modified Precision

- Reference Sentence: I am currently out of the office.
- Machine Translated Sentence: I am currently not in the office.

Modified Precision *(unigram)*:

$$p_1 = \frac{1+1+1+0+0+1+1}{7} = \frac{5}{7} = 0.71$$

Modified Precision *(bi-gram)*:

English: (I am), (am currently), (currently out), (out, of), (of the), (the office)

MT English: (I am), (am currently), (currently not), (not in), (in the), (the office)

$$p_2 = \frac{1+1+1+0+0+1}{6} = \frac{4}{6} = 0.66$$

Problem with Modified Precision:

- It tends to favor short translation

- Reference Sentence: I am currently out of the office.
- Machine Translated Sentence: I am

Modified Precision (unigram): $p_1 = 1$

Modified Precision (bigram): $p_2 = 1$

Modifier Precision (n-gram): $p_n = 1$

Brevity Penalty:

$$Brevity\ Penalty = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c <= r \end{cases}$$

Brevity Penalty(Image by Author)

- r: Length of the reference sentence.
- c: Length of the MT sentence.

- Reference Sentence: I am currently out of the office.
- Machine Translated Sentence: I am

$$BP = e^{(1-{}^{7}/_{2})} = 0.082$$

BLEU Score

$$\text{Bleu Score} = BP \cdot e^{\left(\frac{1}{N}\sum_{n=1}^{N} P_n\right)}$$

- BLEU-1 uses the unigram Precision score
- BLEU-2 uses the geometric average of unigram and bigram precision
- BLEU-3 uses the geometric average of unigram, bigram, and trigram precision
- BLEU-4 uses the geometric average of unigram, bigram, and trigram precision

Drawbacks of BLEU Score

- Sometimes the reference sentences might not include all the descriptors in the image, accordingly BLEU Score will disregard some correctly generated sentences.

State of the art:

- BLEU-1 BLEU-2 BLEU-3 BLEU-4
- 0.845 0.701 0.559 0.436

Ethical Issues 07

Ethical Issues:

- The model could learn toxic or rude or offensive language which could lead to generating offensive captions to given images.
- having bad words like curse words or violent language in the training data
- data-based or decoding-based methods aim to reduce toxicity.
- Data-based strategies are considered computationally expensive due to adding pre-training to the model and changing the parameters.
- Decoding-based methods have the advantage of being less expensive due to leaving the parameters unchanged and only modifying the decoding algorithm

Work So Far

08

Work Distribution

09

Thanks