



Universiteit
Leiden

The Netherlands

From pixels to patterns: AI-driven image analysis in multiple domains

Javanmardi, S.

Citation

Javanmardi, S. (2024, September 18). *From pixels to patterns: AI-driven image analysis in multiple domains*. Retrieved from <https://hdl.handle.net/1887/4092779>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/4092779>

Note: To cite this publication please use the final published version (if applicable).

Advancing Image Captioning via Deep Learning

This chapter is based on the following publications:

- S. Javanmardi, A. M. Latif, M. T. Sadeghi, M. Jahanbanifard, M. Bonsangue, and F. J. Verbeek, Caps captioning: a modern image captioning approach based on improved capsule network, *Sensors*, vol. 22, no. 21, p. 8376, 2022.
- S. Javanmardi, M. Jahanbanifard, M. Bonsangue and F. J. Verbeek, 2023. Using a Novel Capsule Network for an Innovative Approach to Image Captioning, *The Third AAAI Workshop on Scientific Document Understanding*, CEUR.

5.1 Chapter Summary

Following the advancements in biomedical image segmentation detailed in Chapter 4, this thesis now expands its analytical scope in Chapter 5 to encompass more complex data forms, namely the combination of image and text. This expansion leads us into the intricate field of image captioning, where the challenge lies not only in identifying objects within an image but also in understanding their relationships and contextually articulating this information. Despite the progress in recent years, existing methods in image captioning, predominantly based on Convolutional Neural Networks (CNNs), face limitations in accurately capturing positional and geometrical attributes.

In response, this chapter introduces a novel, sophisticated framework that shifts from traditional CNN approaches to a parallelized capsule network. This network aims to deepen the semantic interpretation of images by focusing on detailed spatial and geometrical aspects. A notable aspect of our approach is the utilization of a comprehensive vocabulary from Wikipedia, enhancing the variety and depth of the captions generated.

This research outlines a method that addresses the limitations inherent in CNNs and significantly enriches the descriptiveness of the output. Leveraging capsule networks, our framework emphasizes creating detailed and meaningful descriptions, considering the positions and intricate relationships of entities within images. The effectiveness of this approach is demonstrated through qualitative experiments on the MS-COCO benchmark dataset, where our model shows a marked improvement over existing state-of-the-art image captioning models, especially in accurately conveying the semantic content of images.

5.2 Introduction

Automatic image captioning is a challenging problem in computer vision, and it aims to generate rich content and human-understandable descriptions for given images (Wei et al., 2020). With the increase in the volume of digital images, we must deal with many different image resources on the Internet, i.e., news articles, advertisements, blogs, and the like. As a viewer of an image, one must interpret the content oneself, and most images have no description. It is challenging and time-consuming to consider the apparent equivalence between an image and thousands

of words. Moreover, this is not straightforward for a person with the ordinary perceptive capacity to manually interpret the content of a large volume of images. Therefore, computers try to employ automatic image captioning approaches to describe the content of images for various tasks.

Describing the content of images is essential for many application areas, such as automatic image annotation, scene understanding and image retrieval. Image captioning effectively allows blind people to comprehend and perceive their surroundings. This research area has several use cases, including biomedicine, business, education, and the latter is mainly used in digital libraries and web search engines (Asawa et al., 2021). Figure 5.1 shows an example of evaluated results by our best-presented model.



Generated sentence by our model: A group of people is standing together in a field.

Ground Truth: Many small children are posing together in the black and white photo.

Figure 5.1: An example of an image description with the proposed model.

The performance of image captioning models is closely related to the quality of extracted features from images and the power of the language model to generate accurate and meaningful descriptions related to image content. Considering the semantic relationships between the identified objects within the image is essential in the image caption generation task. However, in the recent image captioning models, not only identifying the objects (as the nouns in the captions) within the image is a challenge, but also how to find the interaction between the objects (as the verbs in the captions) is a big concern in this topic. Expressing this interaction by natural language as semantic knowledge, including the verbs and the adverbial compositions, would be a core issue in image captioning.

The visual content of the images alone cannot be trusted. Recently many different image captioning methods used deep learning algorithms to control the complexity and address these challenges of the image captioning process (Liu et al., 2020), (Wang et al., 2020), (Hossain et al., 2019), (Karpathy and Fei-Fei, 2015), (Bai and An, 2018). These approaches still have an issue with generating realistic descriptions that capture all image concepts. Many current proposed image

5. ADVANCING IMAGE CAPTIONING VIA DEEP LEARNING

captioning models mainly use convolutional neural networks as an image feature extractor that cannot significantly identify prominent image objects and their relationships to generate a meaningful description for the image. Therefore, our motivation in this chapter is to develop a novel method that can overcome the limitation of generating descriptions with a restricted variety of words for specific image contents and with the ability to describe the relationships between the objects. Hence, in this chapter, we proposed a novel encoder-decoder mechanism that addresses these challenges by developing a novel structure from a capsule network (CapsNet) (Sabour et al., 2017). Our proposed model considers the relationships between the objects and generates more meaningful and variant descriptions for the image via a language model.

CapsNet can effectively compensate for shortcomings in CNN by detecting tissue overlap characteristics (Ai et al., 2021). In CapsNet, more salient spatial features and geometrical attributes, such as direction, size, scale, and object attributions, can be represented for each input. This aspect of CapsNet contrasts with CNN since the lack of local invariance features produces excessive variations of global discriminating outputs (Hinton et al., 2018). In addition, our model employs Wikipedia as an external large-scale knowledge base, which aims to augment the diversity in textual training data to generate more meaningful and diverse captions.

In our model, an encoder-decoder system is employed to describe the content of images in natural language. Extracting features from identified attributes together with semantic relationships between those attributes is accomplished by CNN and CapsNet. The output of the encoder is three sequences of integers. The first one declares the visual content and high-level concepts within each image. The second sequence of integers is the corresponding textual information extracted from Wikipedia based on the predicted labels of the images, and the last sequence represents the descriptions of each image already present in the dataset. As input, these fixed-length attribute vectors are fed to the Recurrent Neural Networks (RNN) as a decoder to generate a caption by a language model. The main contributions of our work are as follows:

- Developing a novel parallel structure for a capsule network that can capture more comprehensive information about the objects within the image by considering their relationships. The model leads more accurate description

of the input image, and we performed a benchmark toward a list of existing state-of-the-art models.

- Applying the proposed method on a large-scale dataset, although the architecture of capsule networks requires a huge amount of processing resources when employed for a complex dataset like MSCOCO (Lin et al., 2014).
- Using Wikipedia as an external knowledge base to enrich all the textual training information. Wikipedia can help the image captioning model generate out-of-domain representation in describing the content of the image.

This chapter is organized as follows: Section 5.3 presents an overview of the related literature and models in image captioning. All the employed models and the proposed method with the design of the framework are presented in section 5.4. In sections 5.5 and 5.6, the reader can find the descriptions of all experiments and the study results, followed by the conclusions in section 5.7.

For assessing the results, we used standard discrete natural language processing metrics such as BLEU 1-4 (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005), showing a more accurate description of the input image when compared to existing state-of-the-art models.

5.3 Related Work

Image captioning is a popular research topic in computer vision and natural language processing. Generating an accurate textual explanation that describes the content of an image is accomplished by understanding the visual content of the image. Recent research recommends a model architecture for describing an image by a CNN as an encoder for extracting image contents, a module for embedding the words, and an RNN as a decoder for language modeling and creating image captions. The interest in image captioning has broadened with the development of benchmark datasets such as MS-COCO (Lin et al., 2014), Flickr 8K (Hodosh et al., 2013), and Flickr 30K (Young et al., 2014).

Current image captioning models can be categorized into template-based, retrieval-based, and neural network-based models. The template-based models (Farhadi et al., 2010), (Kulkarni et al., 2013), (Li et al., 2011) first detect all the image attributes using image classification and object detection methods. These methods generate captions by filling in predefined templates from the identified objects.

5. ADVANCING IMAGE CAPTIONING VIA DEEP LEARNING

This approach produces too flexible captions that cannot correctly describe the relationships between attributes (Jin et al., 2015).

Retrieval-based models (Kuznetsova et al., 2013), (Kuznetsova et al., 2014), (Ordonez et al., 2011) create a pool of similar images in an image database and rank the retrieved images by measuring their similarities. Subsequently, change the found image descriptions to create a new description for the queried image. The usefulness of this strategy is severely constrained when dealing with images that are not in the dataset and thus not classified, i.e., unseen.

The neural network-based models are the novel methods that have been proposed as encoder-decoder and inspired by the success of deep neural networks in machine learning tasks (Wu et al., 2017), (Rennie et al., 2017), (Kiros et al., 2014), (Mason and Charniak, 2014), (Devlin et al., 2015), (Vinyals et al., 2015), (Lebret et al., 2015), (You et al., 2016), (Johnson et al., 2016), (Yang and Liu, 2020). Kiros et al. (Kiros et al., 2014) proposed a multimodal language model that jointly learned the high-level image features and word representations. Their model can generate image captions without using any default template or structure, making the model more flexible. Nevertheless, their model could not learn latent representations of the interactions between the objects in the image. Moreover, they investigated a manual algorithm including multiple modules that cannot learn from each other during the training process.

Wu et al. (Wu et al., 2017) proposed a two-phase attribute-based model for the image captioning approach based on a CNN-LSTM framework. In their framework, the attributes as high-level semantic concepts are extracted by the CNN classifier to generate image captions. They significantly improved in generating rich captions, but their model has the problem of equally distributing semantic concepts in whole sentences (Hossain et al., 2019). They also implemented a visual question-answering model in the captioning phase using extracted information from an external knowledge base to answer a wide range of image-based questions based on the content of images.

Mason and Charniak (Mason and Charniak, 2014) proposed a graphic retrieval model to obtain the textual description of undescribed images based on the text descriptions of similar images with the highest rank in the dataset. The constant presence of the best matches description sentence to the query image is unrealistic. A word frequency model has been used to find a smoothed assessment of the visual

content of various captions. With the same challenge, Devlin et al. (Devlin et al., 2015) provided the nearest neighbor method for image captioning. They make a pool of captions based on training data and describe the query image based on the nearest neighbour images. Vinyals et al. (Vinyals et al., 2015) used a Neural Image Caption (NIC) model to generate a plain text description by maximizing the likelihood of the target sentence given the image. In their method, the words with the highest probability are selected from outputs to be formed as an image description.

Lebret et al. (Lebret et al., 2015) investigated a CNN-based image captioning approach to infer phrases that describe the image. Then all the predicted phrases are combined using a language model to create a caption. Their proposed model is an example-based method that makes the model like a large dictionary, and accurate relevant descriptions will not always be found in the data source. Therefore, these methods are not always fine for complex data, although they avoid critical mistakes in generating captions using a language model.

You et al. (You et al., 2016) proposed a combined bottom-up and top-down model that selects salient regions of an image via a bottom-up mechanism and then generates the captions by applying a top-down mechanism. A similar image captioning method has been proposed by Johnson et al. (Johnson et al., 2016). They employed a convolutional localization network to predict a set of captions across the important regions of the image and generate the label sequences using a recurrent neural network. The proposed method localizes the salient regions and generates captions for each region using a language model. Finding a relationship between all these regions is always a big challenge in these approaches.

Liu et al. (Liu et al., 2020) proposed an ontology to describe the scene construction of images. This constructed ontology can specify the object types and also the special information for the objects (e.g., location, velocity). This visual and special information can be transformed into meaningful project information for generating captions using integrated computer vision and linguistic models.

Various improvements are made in captioning models to make the network more inventive and effective by considering visual and semantic attention to the image. For example, in (Yang and Liu, 2020), Yang and Liu introduced a method called ATT-BM-SOM to increase the readability of the syntax and optimize the syntactic structure of captions. This framework operates based on the attention

5. ADVANCING IMAGE CAPTIONING VIA DEEP LEARNING

balance mechanism and the syntax optimization module and effectively fuses image information. Their model generates high-quality captions, compensating for the lack of image information selection and syntax readability.

Training large amounts of data affords machine learning models greater predictive performance. However, training massive data by machine learning may increase the execution time of the model, and it could memorize the data that causes the model overfitting. Moreover, data quality plays an important role in the performance of the model. The hypothesis is that more data may contain useful information. To this aim, Hossain et al. (Hossain et al., 2021) proposed a method that leverages a combination of real and synthetic data generated by the Generative Adversarial Network (GAN). This is an efficient alternative for the techniques requiring human-annotated images, as they are labor-intensive to generate and time-consuming.

Xian and Tian (Xian and Tian, 2019) employed a self-guiding model to extract textual features using the multimodal LSTM model. This model adequately describes the images without having a perfect training dataset. This is an important issue that we have considered in the research described in this chapter. Recently, Reinforcement Learning (RL) methods have been incorporated into image caption generation models. Rennie et al. (Rennie et al., 2017) proposed a reinforcement model for optimizing the process of image captioning. They considered a reward parameter on the results at the test time. Yan et al. (Yan et al., 2020) proposed a hierarchical model that uses the GAN together with the RL algorithm to produce more accurate captions for images. They measured the consistency between the generated captions and the content of images by the RL optimization process and the discriminator in the framework of GAN. For object detection and extracting salient regions from an image, they used faster R-CNN models, and then they used CNN to extract features from the proposed regions. They achieved significant improvement over the generated captions for the images.

In Section 5.4, the structure of the image caption generation models and the employed networks in our experiments will be discussed in more detail.

5.4 Dataset and Image captioning Methods

This section proposes an encoder-decoder model to generate descriptions from the images. Understanding the image requires recognizing the objects, properties, and

5.4 Dataset and Image captioning Methods

interactions in the encoder part. Moreover, producing sentences to describe images in the decoder requires understanding language syntax and semantics.

Figure 5.2 illustrates the employed Knowledge Discovery Database (KDD) in our research. In this model, images and descriptions proceed separately, after which all the information is fed into the caption predictor. The encoder-decoder framework is considered to create the captions of images and is inspired by the findings from previous work, as mentioned in section 5.3.

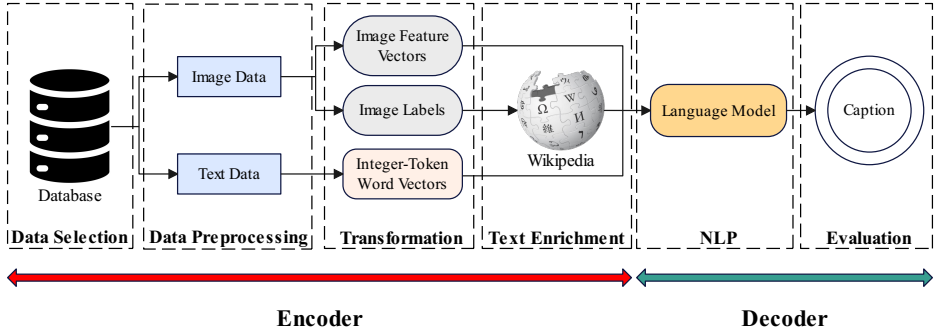


Figure 5.2: KDD methodology of the proposed model

This chapter presents a new version of the capsule network by parallelizing the basic structure of the capsule network to capture more comprehensive information about the objects within the image, which leads more accurate description of the input image. The base structure of the capsule network works well on a simple dataset such as MNIST, which includes images with a single object and only one channel. However, the network efficiency significantly decreases when applied to images with large special dimensions and complex datasets such as MS-COCO and Flickr. The presence of multiple channels and objects in the images increases the training time of the network and leads to weak results compared to the state-of-the-art. This problem happens due to inefficiency in capturing the underlying information of the image. To handle this issue, we extended the baseline network by parallelizing the convolutional layers and the primary capsules of the original CapsNet, followed by a concatenation approach to extract more complex and qualified features from the images. On the other hand, parallelizing the convolution layers reduces the dimensions of the fed features to the primary capsules and accelerates the learning process.

5. ADVANCING IMAGE CAPTIONING VIA DEEP LEARNING

In the proposed image captioning model, we use CNN and CapsNet architectures to incorporate visual context from an image which is then used as the input of a machine translation such as an RNN architecture to generate objective sentences in the decoder part of the framework. We applied cross-entropy loss to adjust the model weights during the sequential model training. In Figure 5.5, the entire flow of the proposed model is depicted.

We divide the dataset into train, validation, and test subsets. The train and validation sets are fed to the CNN and CapsNet to extract the visual features next. Transfer learning in CNN has been involved in retraining the MSCOCO dataset and extracting the visual attention of images. We have applied both the Inception-V3 and VGG16 as image feature extractors. These networks are trained on the ImageNet dataset with more than one million images of 1000 classes. Training the CapsNet is done from scratch and based on 80 categories of objects in Category Caps. Subsequently, the image features and captions are transferred to the RNN network to train the language model. We further explain the details of the model.

The proposed architecture uses Inception-V3 and capsule networks to extract visual information from the images and compare all our experiments to the result of the base models. The details of these networks are shown in Table 5.1.

Table 5.1: Specific parameters of the models in the evaluation.

Parameters	VGG-16	Inception-v3	CapsNet
Depth	16	48	8
Image size (pixel)	224×224	299×299	299×299
Solver (optimizer)	SGDM	RMSPProb	ADAM
Loss function	cross-ent.	cross-ent.	MSE
Batch size	32	64	128
Learning rate	0.001	0.0001	0.001
Learning rate drop factor	0.1	0.1	0.5
Learning rate drop period	10	10	10
Momentum	0.9	0.9	0.9
Gradient threshold method	L2norm	L2norm	L2norm

SGDM: Stochastic gradient descent with momentum; Adam: Adaptive momentum estimation;

RMSPProb: root mean square propagation; MSE: mean squared error.

5.4 Dataset and Image captioning Methods

Inception-V3: In 2015, Google introduced GoogleNet (Szegedy et al., 2015). This network reduces the computational burden of the network with a lightweight structure and has been shown to obtain better performance. The first version of the inception network includes filters of multiple sizes (1×1 , 3×3 , 5×5) to perform convolution on an input image. To reduce the network parameters and computational cost, Inception-V3 breaks down the kernels into smaller sizes (e.g., 5×5 kernels into two (1×5 , 5×1)). This solution can extend the depth of the network and helps to prevent computation and overfitting issues. Our research demonstrated the proper performance of Inception-V3 (Ashtiani et al., 2021). In the encoder phase, we used the extracted features from the last fully connected layer of the Inception-V3 network and the predicted labels from the SoftMax layer.

VGG16: This network is one of the two networks introduced by Simonyan and Zisserman in 2014 (Simonyan and Zisserman, 2014). This model has 13 convolutional layers of a 3×3 filter with a stride of 1 pixel followed by a max-pooling layer 2×2 filter of stride two and ReLU activation function. ReLU can reduce the gradient disappearance problem by providing more optimal error transmission than the sigmoid function. This network computes approximately 138M parameters and is considered an extensive network. A pretrained network on the ImageNet dataset extracts visual features from input images by applying the transfer learning method. VGG16 has five convolutional layers and pooling modules. These modules have respectively 64, 128, 256, 512 and 512 filters. The feature map size will be reduced in half after each module. Following (Wu et al., 2017) and its straightforward character, we considered this model a baseline. We employ the extracted features from the last fully connected layer to initialize the RNN network.

Capsule Network: A capsule is a set of neurons whose activity vectors indicate the posture characteristics of an entity, and the length of the vector denotes the chance of that entity existing. Unlike a convolutional network, capsules save comprehensive information about the location and pose of an entity. Hinton et al. (Sabour et al., 2017) claimed that regardless of the high capability of CNNs, this network has two main disadvantages: 1- lack of rotation invariant and 2- using a pooling layer. The former causes failure in recognizing spatial relations between the objects, and the latter causes information loss due to the maximum value selection of each region. Therefore Sabour et al. (Sabour et al., 2017) proposed a capsule network to address the issues mentioned above.

5. ADVANCING IMAGE CAPTIONING VIA DEEP LEARNING

There are different concrete components in a capsule network for learning the semantic representations within the image (Figure 5.3). These components map construction by reconstructing the discrepancy map from the input image. The major components of the capsule network involve as follows:

Primary capsules combine the features extracted by convolutional layers in the construction phase. Reshaping the extracted feature maps from the primary capsules. Squashing is a non-linear activation function that squashes the weighted input vector of a particular capsule. This function distributes the length of the output vector between 0 and 1. The dynamic routing layer produces output capsules with high agreements by automatically grouping input capsules. The pooling layers in the capsule network are replaced by a mechanism called "routing by agreement" in the routing layer: the output of each capsule in the lower level is sent to the parent capsules in the higher level only if their features have a dependency.

Category capsules with a marginal classification loss and a reconstruction sub-network with a reconstruction loss for recovering the original image from capsule representations.

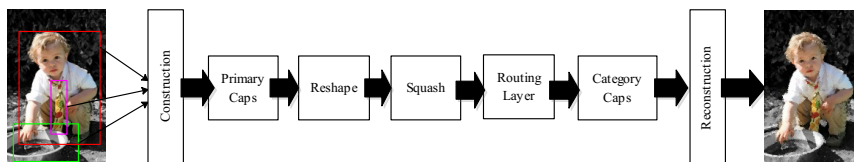


Figure 5.3: Capsule Network Architecture.

The operation of all these components is explained in this section in more detail. One important aspect of capsule networks is their ability to identify individual parts of objects in a single image and then represent spatial relationships between those parts. For example, in Figure 5.3, the CapsNet has identified three different parts of objects within the input image (tie, child, bin). The output image on the right side of the figure shows the result of the reconstruction subnetwork in the employed capsule network. Figure 5.4 shows the construction of a capsule and how data is routed between lower-level and higher-level capsules.

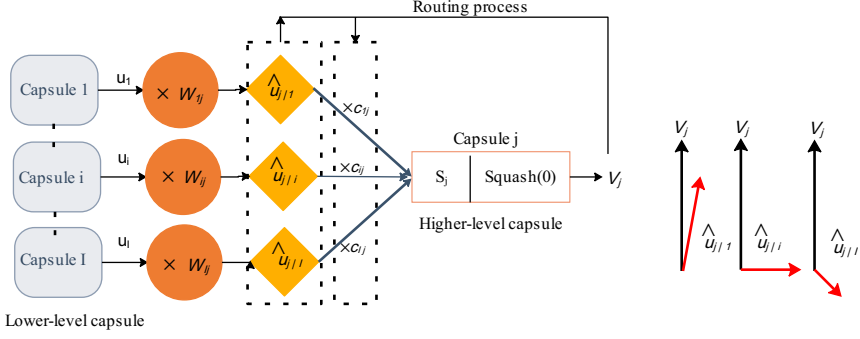


Figure 5.4: (a) Transferred information among the capsules from [1..I] (b) routing procedure.

In Figure 5.4a (left figure), each capsule finds the appropriate parent in the next layer during the dynamic routing procedure to send its output to those capsules in the above layer. The input and output of a capsule are vectors. Given u_i as the prediction vector of capsule i and $v_{j|i}$ as the output of parent capsule j in higher level will be computed by multiplying u_i with a weighted matrix W_{ij} :

$$\hat{u}_{j|i} = W_{ij} \cdot u_i \quad (5.1)$$

The length of u_i indicates the probability of predicting a component in the image even after changing the viewing angle. The direction of u_i represents several properties of that component, such as size and position. A weighted sum overall $\hat{u}_{j|i}$ and an intermediate coupling coefficient c_{ij} is calculated as the total input vector to capsule j by the following function:

$$s_j = \sum_i c_{ij} \hat{u}_{j|i} \quad (5.2)$$

Here, the coupling coefficient c_{ij} , is the class-specific likelihood calculated after flattening the vectors and is computed by a routing SoftMax function as follows:

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \quad (5.3)$$

5. ADVANCING IMAGE CAPTIONING VIA DEEP LEARNING

where b_{ij} represents the log probability of connection between capsules i and j . As shown in Figure 5.4 part (b) on the right side, the value of c_{ij} increases when the lower-level and higher-level capsules are consistent with their predictions and decreases when they are inconsistent. Based on the original paper, this parameter is initialized at 0 in the routing by agreement procedure. Instead of applying the ReLU activation function as in VGG16 and Inception-v3, the following non-linear squashing function (Sabour et al., 2017) will be calculated over the input vector in this network:

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \quad (5.4)$$

Where s_j is the input vector and v_j is the normalized output between 0 and 1. The log probability is updated along with the routing mechanism by calculating the agreement between v_j as the output of capsule j in the above layer and $\hat{u}_{j|i}$, as a prediction vector.

The loss function of the network for each capsule k is computed as follows:

$$L_k = T_k \max(0, l^+ - \|v_k\|)^2 + \lambda(1 - T_k) \max(0, \|v_k\| - l^-)^2 \quad (5.5)$$

where L_k is loss term for one prediction, T_k is a term equal to 1 when the class k is present; otherwise, it is 0. The upper and lower bounds of margin loss parameters, l^+ and l^- , are set to 0.9 and 0.1. It means that if an entity is present with a probability above 0.9, the loss is zero; otherwise, the loss is not zero. Regarding capsules that could not predict the correct label, if the predicted probability of all those labels is below 0.1, the margin loss is zero; otherwise, it is not zero. The parameter λ is set at 0.5 and is used for numerical stability to control the downweighting of the initial weights for the absent classes. $\|\cdot\|$ in all the equations denotes L2 norm.

Improved capsule network: In the improved version of the capsule network architecture, where we parallelized the convolution layers and primary capsules, the input image size is $229 \times 229 \times 3$. The different architecture of the capsule network distinguishes it compared to CNN. Except for the input and output layers, the capsule network consists of primary and category capsule layers. The output of the capsules is forwarded to the decoder. The networks prevent overfitting by rebuilding

the input image from the output capsules by minimizing the reconstruction loss as a regularization method in the decoder (Mandal et al., 2019).

The original capsule network has been tested on the MNIST dataset with one color channel (grayscale). However, the color of objects is an important factor in object detection and image captioning tasks. Therefore, we propose a parallelized capsule network that generates the descriptions of the images by passing the RGB images with three color channels through the three blocks of parallel convolutional layers and parallel primary capsules. The three-color channels of RGB images can store information and intuitively visualize content. Therefore, color analysis is also addressed in this parallelized structure of the capsule network, which makes the model more informative and improves the descriptiveness of image captions by extracting more qualified features from the image (Albawi et al., 2017). Adding more convolutional layers was not logical due to the increasing model complexity computational cost. The structure of the new network has been presented in Figure 5.5.

Gated Recurrent Unit: Our image captioning framework used a three-layer RNN network with a Gated Recurrent unit cell (Chung et al., 2014). This RNN is equipped with visual features in the feature maps of CNN and CapsNet. The proposed model generates a description for each image by maximizing the probability of the current word predicted in the caption according to the following formula:

$$\theta^* = \arg \max_{\theta} \sum_{(I,M)} \log p(M|I; \theta) \quad (5.6)$$

where θ are the parameters of the proposed model and M is the correct description of image I .

Suppose $\{m_0, \dots, m_{N-1}\}$ is a sequence of words in transcription M of length N , then $\log p(M|I)$ as the probability of generating a word for an image I , is as follows:

$$\log p(M|I) = \sum_{t=0}^N \log p(m_t|I, m_0, \dots, m_{N-1}, c_t) \quad (5.7)$$

where t is the time step and c_t is context vector. A two-step process feeds all the text data to the RNN network. The first step is tokenizing, and the second one is embedding. All the words in the sentences are converted into so-called integer-token vectors during tokenizing. This process is based on 10000 most frequent and unique words in the image captions.

5. ADVANCING IMAGE CAPTIONING VIA DEEP LEARNING

Throughout the embedding, all the integer-token vectors are transformed into floating-point vectors. We considered this part a decoder consisting of three GRU layers with an input size of 512. The embedding layer converts all the integer tokens into a 128-length vector. The output features initialize the GRU units from the encoder part. The governing equations in GRU are given as follows:

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \quad (5.8)$$

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \quad (5.9)$$

$$\tilde{h}_t = \tanh(W_h[r_t \circ h_{t-1}, x_t] + b_{\tilde{h}}) \quad (5.10)$$

$$h_t = z_t \circ \tilde{h}_t + (1 - z_t) \circ h_{t-1} \quad (5.11)$$

$$x_t = [E_w m_{t-1}, c_t] \quad (5.12)$$

where r_t is reset gate vector at instant t , h_t is output vector of the hidden layer, \tilde{h}_t is candidate activation vector, which is temporary output, z_t is update gate vector, and W_r , W_z , W_h are the weight matrices of the reset gate, the update gate, and the temporary output. All the biases corresponding to these weight matrices are represented by b_r , b_z , $b_{\tilde{h}}$. x_t is input vector at instant t , which is based on the input embedding matrices, E_w , and the one-hot encoder of the previous word, m_{t-1} . c_t is the context vector extracted by the feature maps of CNN and CapsNet. The concatenation operator is applied on $E_w m_{t-1}$ and c_t to make the input of the RNN network. \circ is an element-wise product. Eventually, we minimize the following standard cross-entropy loss function for the proposed captioning model with parameter θ and given a target ground truth $m_1^* : t$:

$$L_c^\theta = - \sum_{t=1}^T \log(p_\theta(m_t^* | m_1^* : t - 1)) \quad (5.13)$$

The performance of the implementation by different metrics is discussed in the section on evaluations and results.

External Knowledge: Many pipeline approaches have been proposed for image captioning by integrating knowledge in text script form. In this chapter, the generated caption of an input image is obtained using "beam search". i.e., in each iteration for training one image, we considered the top five attributes as a candidate for a query in a knowledge database to retrieve sentences. After extracting the visual features of each image using CNN and CapsNet, those five predicted attributes are used as queries to extract contextual information from the Wikipedia database for every image in the training dataset. We only selected the first three sentences for every attribute from all information retrieved from Wikipedia. Then by applying the automatic summarization method, we extract the first three sentences of retrieved text for each top-5 predicted label from CNN. By using this external knowledge, we enrich the descriptive information of each image. We then passed this information and all five available captions in the training set to the RNN network for generating a descriptive caption from the image.

We then passed this information and all five available captions in the training set to the RNN network for generating a descriptive caption from the image. Framework: The final model follows the encoder-decoder framework. The entire architecture of our proposed model is shown in Figure 5.5.

There are three primary phases in this model. The first phase includes extracting features from the images using two deep neural networks. In this step, CapsNet and inception-V3 are used for extracting visual content from the input image concurrently. In CapsNet, at first, three parallel levels with three convolutional layers in $72 \times 72 \times 96$, $34 \times 34 \times 96$ and $26 \times 26 \times 256$ sizes are applied to each channel of the image (Figure 5.5.a). As stated in section 3, a primary capsule block is followed by a reshaping and squashing process to take the concatenated features recognized by the convolutional and primary capsule layers and combine them to produce new features (Figure 5.5.b). Then "routing by agreement" mechanism is done rather than a pooling operation (Figure 5.5.c). Based on this mechanism, the output of each capsule in the lower level is sent to those parent capsules in the higher level with dependency on their features. The next layer is category capsules which indicate the membership probability of the input image in each category. The actual label masks the output of the categorical capsule layer by using the L2-norm to calculate the loss (Figure 5.5.d). The last part of the capsule network

5. ADVANCING IMAGE CAPTIONING VIA DEEP LEARNING

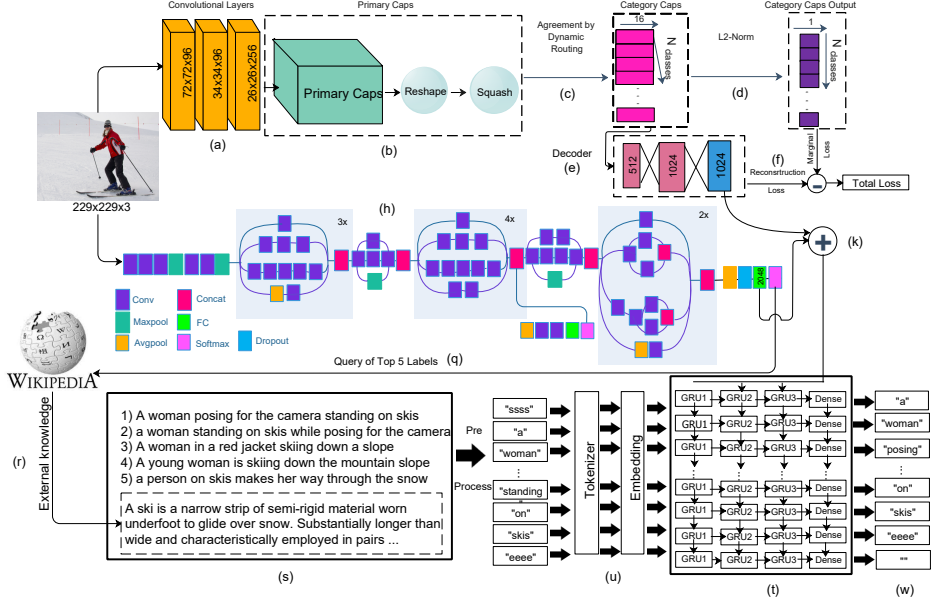


Figure 5.5: Our proposed model: a CNN and a CapsNet, are applied to a given image to produce the visual features and predict the attributes of the image (a-k). The textual information of each sample comprises the descriptions of the image and the aggregated data from the external database and a preprocessed method is applied to the text(l-n). After the tokenizing and embedding process, the visual attention of the image is fed to a GRU with 3 levels to generate a caption to explain the content of the image (p-r).

is the decoder which is used as a regularizer with two fully connected layers with sizes 512 and 1024 (Figure 5.5.e).

Capsules are forced to learn features that can be used to reconstruct the input image by the decoder based on the calculated reconstruction loss. The output of the second fully connected layer is used as the image visual features vector (Figure 5.5.f). At the same time, Inception-V3, as the second feature extractor, produces the features vector from the input image (Figure 5.5.h). A pretrained convolutional network is used in this step to handle the overfitting issue and increase the training time. Then both visual feature vectors are concatenated to feed the language model (Figure 5.5.g, Figure 5.5.i, Figure 5.5.j). All of these operations are done during the first phase. In the second phase, in addition to five captions of each image in the dataset, we extract external knowledge from Wikipedia based on the

top five labels of each image extracted from the CNN network (Figure 5.5.k). We use the first three sentences of the description retrieved by Wikipedia (Figure 5.5.l) for each label. Finally, the information from the first two-phase is fed to the last phase (Figure 5.5.m). In the last phase of the framework, we use the RNN network with three layers of GRU as a decoder (Figure 5.5.q). Tokenizing and embedding layers convert all the preprocessed textual data to an integer vector before feeding the descriptions to the language model (Figure 5.5.n, Figure 5.5.p). Finally, our model trains to describe all the textual and visual features of images by applying language modelling techniques (Figure 5.5.r). The model steps in Figure 5.5 are summarised as follows:

1. Partitioning the image set into train, validation, and test subsets randomly.
2. Applying image feature extractor models to extract visual features from the images (Figure 5.5.a-j)
3. Extracting external knowledge for each image by searching the predicted labels from the previous step as a query in Wikipedia and adding it to the captions that already exist for the images in the dataset (Figure 5.5.k-m)
4. Applying preprocessing methods to contextual data before feeding it to the RNN network, i.e., removing the punctuations, numbers and wrapping each sentence around with "ssss" and "eeee" tokens to specify the beginning and end of sentences for the network (Figure 5.5.n)
5. Transforming the textual features to the integers vector by tokenizing and embedding operations for training by the language model (Figure 5.5.p)
6. Training language model for certain epochs based on its performance on validation data. During the training phase, the model predicts the next word of each word in the caption (Figure 5.5.q-r)

After the training phase, the model is ready to evaluate test set images by extracting visual features and predicting the captions using a greedy search. Greedy search selects the word with the highest probability at each time step and uses it as the GRU input for the following time step until the end of the sentence is reached. In the next section, we will discuss the details of the experiments and the obtained results by the analyzed methods.

5.5 Experiments

This section reports the details of implementations and the results of the experiments conducted by different variations of models.

5.5.1 Dataset and implementation details

For this research, We use the MS-COCO dataset (Lin et al., 2014) to evaluate the proposed model in our experiments. MS-COCO contains 123,287k images with five captions and 80 object categories for each image annotated by Amazon Mechanical Turk (AMT) workers. Since there are no available annotations for the test set, In this work, we used publicly available splits provided by Karpathy et al. (Karpathy and Fei-Fei, 2015). We use 5,000 images for validation and testing and the rest for the training set. All the models are implemented in Python version 3.6 and using the capabilities provided by Keras version 2.2.5 and TensorFlow version 1.15.0 deep learning libraries. Table 1 shows the parameters set for each network. The training was done using a machine equipped with two GeForce RTX 2080 GPU cards with 8GB memory. Although, The machine was installed with two GPUs, for the experiments, only one was necessary.

Metric: To compare our results to other baseline models, we measure the performance of the implemented models by the commonly used metrics, BLEU 1-4 (Papineni et al., 2002), ROUGE (Lin, 2004) and METEOR (Banerjee and Lavie, 2005).

BLEU is one of the popular metrics to evaluate the correspondence between generated sentences by humans and machines. This metric measures the maximum number of co-occurrence n-grams between reference and candidate sentences. Here 'n' takes the value of 1, 2, 3, and 4 depending on the length of sentences.

ROUGE evaluates the performance of generated sentences by a machine based on their similarity to the reference sentences. This metric finds the longest subsequence of tokens between candidate and reference sentences and calculates how many tokens from the human reference summaries were duplicated in the machine-generated summaries. Unlike BLEU, which prioritizes precision, ROUGE is recall-oriented and can estimate correlated n-grams better than BLEU.

METEOR is the last evaluation metric in this chapter. This metric and exact word matching consider the stemmed and wordnet synonyms tokens between candidate and reference sentence alignment.

Baselines: We provide two baseline approaches to verify the effectiveness of the models. GRU model is used in the decoder part of the framework. Moreover, We used inception-V3 and VGG16 as the feature extractor method for the encoder part.

Our approaches: We assess different variations of our approach. CN + IncV3 utilizes the extracted features from the capsule network and inception-V3 as image feature extractors. CN + VGG16 uses a VGG16 network rather than inception-V3 in the encoder. The Wikipedia knowledge base enriches the contextualized language model in this model. So, CN + IncV3 + EK and CN + VGG16 + EK are the models that use relevant external knowledge from Wikipedia. We also have performed additional experiments to check the importance of the capsule network in describing the content of images. To that end, we implemented IncV3 + EK and VGG16 + EK methods to verify the effectiveness of the capsule network for image captioning models.

5.6 Results and Discussions

This section discusses the results from the implemented models and then compares them to state-of-the-art. Table 5.2 reports image captioning results for introduced methods on the MS-COCO dataset. The results demonstrate that the CN + IncV3 + EK model with capsule network and inception-V3 feature extractors can generate more human-like sentences by adding external knowledge to the language model. This model achieves significantly better results in the overall metrics.

For the sake of brevity in explaining the results, we label BLEU 1, BLEU 2, BLEU 3, BLEU 4, ROUGE, and METEOR as B1, B2, B3, B4, R, and M, respectively. Specifically, the calculated metrics, B(1-4), R, and M for CN+IncV3 +EK method are 0.89, 0.74, 0.61, 0.54, 0.66, and 0.45 respectively. This result shows that the performance of this model is significantly better than the other methods because it takes advantage of the capsule network and inception-V3 network as feature extractors and uses external knowledge to enrich the trainable contextual information for the language model. When we implemented the model without external knowledge, we faced almost 13.5% performance degradation in B1. The

5. ADVANCING IMAGE CAPTIONING VIA DEEP LEARNING

Table 5.2: The experimental results of implemented models. Bold text indicates the best overall performance.

Models	B1	B2	B3	B4	R	M
VGG 16 (Baseline)	0.33	0.24	0.18	0.16	0.21	0.24
IncV3 (Baseline)	0.36	0.26	0.21	0.17	0.23	0.28
CN + IncV3	0.77	0.54	0.43	0.35	0.47	0.35
CN + VGG 16	0.41	0.30	0.25	0.19	0.28	0.34
CN + IncV3 + EK	0.89	0.74	0.61	0.54	0.66	0.45
CN + VGG 16 + EK	0.59	0.44	0.37	0.29	0.31	0.38
IncV3 + EK	0.63	0.43	0.34	0.28	0.29	0.31
VGG 16 + EK	0.38	0.27	0.22	0.18	0.23	0.26

degradation for other evaluation metrics is about 27%, 29.5%, 35.2%, 28.8%, and 22.2% for B (2-4), R, and M respectively in CN + IncV3 model.

The performance decreases about 33.7%, 40.5%, 39.3%, 46.3%, 53%, and 15.5% for all the B (1-4), R, and M respectively in the case we implemented VGG16 rather than inception-V3 in CN + VGG16 + EK model. Comparing the results between CN + IncV3 + EK as the best model and IncV3 + EK shows that including a capsule network improves the results. In this case, performance improvement is about 41.27%, 72.1%, 79.4%, 92%, 127.5%, and 45.2% for all the B (1-4), R, and M metrics, respectively. Improving performance in these evaluation metrics when we implemented CN + VGG16 + EK and VGG16 + EK models is considerable. This improvement is as follows for B (1-4), R, and M respectively: 55.3%, 63%, 68.2%, 61.1%, 34.8%, and 46.1%. The results show that using VGG16 as a feature extractor is not as good as inception-V3 and decreases performance. Comparing CN + VGG16 and CN + VGG16 + EK models demonstrates adding external knowledge can enhance the performance of the language model. Comparing the evaluation metrics between these two models indicates 44%, 46.7%, 48%, 52.6%, 10.7%, and 11.8% improvement for B (1-4), R, and M respectively.

A comparison between the different models from our experiments demonstrates the effectiveness of CN + IncV3 + EK as our best model. In Figure 5.6, all the introduced models on MS-COCO are compared with other baselines across BLEU 1, BLEU 2, BLEU 3, BLEU 4, ROUGE and METEOR evaluation metrics. Comparing the results of applying all the models over the 100 training epochs

Table 5.3: Comparison of the best result to state-of-the-art models.

Models	B1	B2	B3	B4	R	M
Javanmardi et al.	0.89	0.74	0.61	0.54	0.66	0.45
(Aneja et al., 2018)	0.72	0.55	0.40	0.30	0.53	0.25
(Tan et al., 2019)	0.73	0.57	0.43	0.33	0.54	0.25
(Wu et al., 2017)	0.73	0.56	0.41	0.31	0.53	0.25
(Zhang et al., 2021)	0.75	0.62	0.48	0.36	-	0.27
(Yu et al., 2019)	0.81	0.67	0.52	0.40	0.59	0.29
(Lu et al., 2017)	0.75	0.58	0.44	0.33	0.55	0.26
(Anderson et al., 2018)	0.80	0.64	0.49	0.37	0.57	0.27
(Jiang et al., 2018)	0.80	0.65	0.50	0.38	0.58	0.28
(Yan et al., 2020)	0.73	0.53	0.39	0.28	0.56	0.25

shows that the performance of the model that includes external knowledge from Wikipedia and extracts image features by using inception-V3 and capsule network performs significantly better than the other models.

According to the plots, It is evident that most of the models have converged after 60 epochs. To prove the effectiveness of this model, we compare the result of the CN + IncV3 + EK method with state-of-the-art research. Table 5.3 shows that our best model outperforms previously published results on the MS-COCO "Karpathy" test split dataset.

Compared to our model, (Aneja et al., 2018) has proposed an attention mechanism to leverage spatial features of an image to find salient objects. Tan et al. (Tan et al., 2019) proposed a tuning model with a small number of parameters in the RNN. Their model can produce a very sparse decoder for generating a caption preserving the performance of the method compared to their baseline. Zhang (Zhang et al., 2021) et al. implemented a cooperative learning mechanism to combine two image caption and image retrieval modules while generating a caption. Then during a multi-step refining process, they refined the image-level and object-level information to produce a meaningful caption. Instead of using GRU as RNN, Yu et al. (Yu et al., 2019) proposed a model which employed a multimodal transformer as a language model in the decoder to generate a caption. Against our research, (Lu et al., 2017) and (Anderson et al., 2018) have focused on important image regions. Lu et al. (Lu et al., 2017) proposed an adaptive attention framework that could

5. ADVANCING IMAGE CAPTIONING VIA DEEP LEARNING

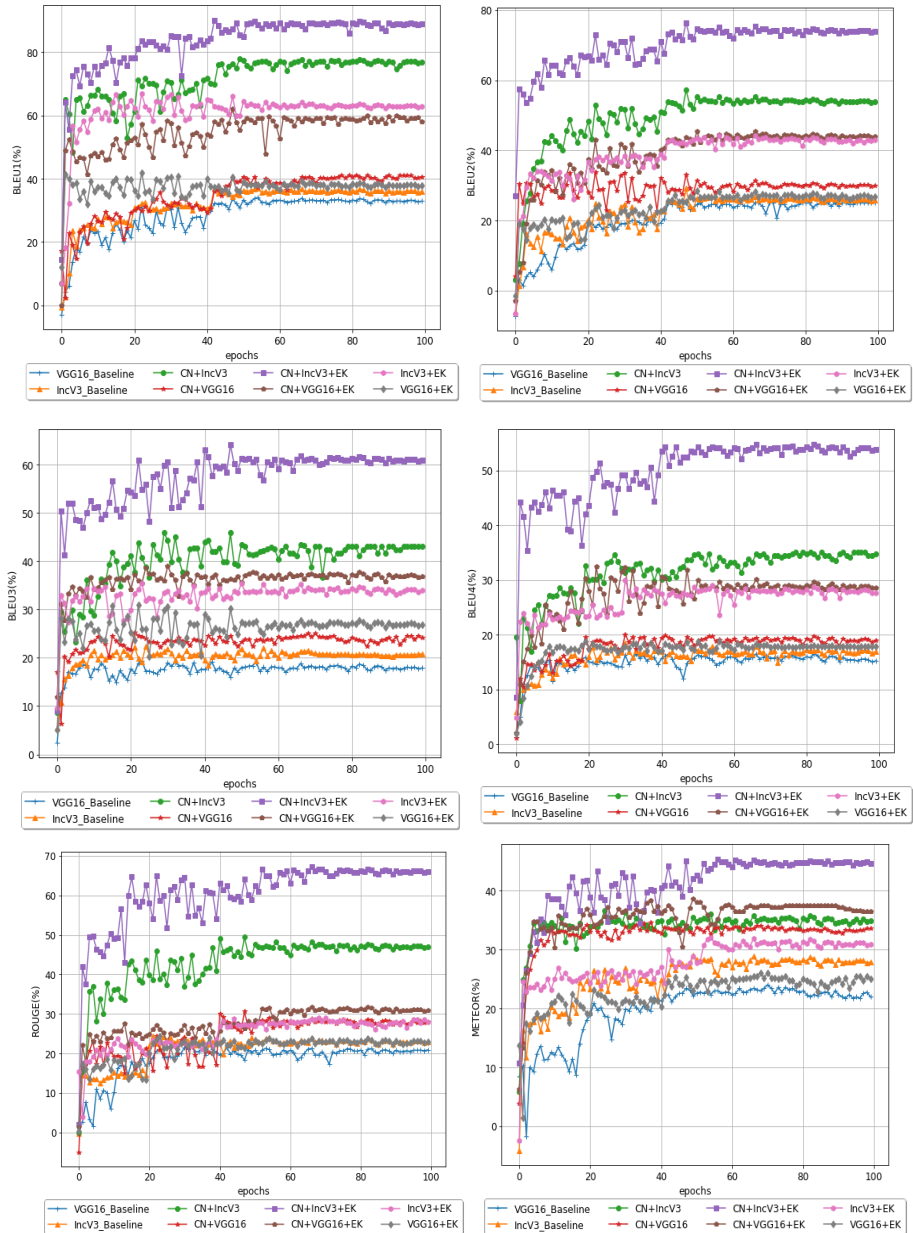


Figure 5.6: Comparative analysis on all the models based on evaluation matrices.

decide whether to rely on special attention to the image and when to attend to the textual image information. In (Anderson et al., 2018), Anderson et al. extracted a set of salient regions from the image by applying a bottom-up mechanism. They also implemented a top-down mechanism to determine the distribution of attention over the image to compute feature weightings in different regions.

Jiang et al. (Jiang et al., 2018) proposed a framework that includes a recurrent fusion network. This fusion procedure is implemented between the encoder and decoder to exploit interactions among the represented features from the encoder part for creating a new set of vectors from decoder outputs.

5.6.1 Qualitative Results

In this section, we present some examples to show the performance of the CN + IncV3 + EK method as our best model. We used the occlusion sensitivity function to visualize and localize the most important regions of the images for the network. The occlusion function computes sensitivity maps for CNNs. This function disturbs small input areas by replacing them with an occluding mask, typically a grey square, and moving the mask across the image to calculate the probability score of the given class. This method can highlight the most critical regions of the image for classification. Figure 5.7 shows some examples of occlusion sensitivity maps and the regions that provide more essential features for the network. Using occlusion sensitivity helps us better understand features used by the network and provides insight into the reasons for the misclassified images. These examples show that CN + IncV3 + EK, i.e., the best descriptor model, can generate more human-like sentences for each image. The generated caption for Figure 5.7.a photo indicates the good performance of our model. A remarkable result in this example is a plate of 'salad' in this image, which is not mentioned in the five captions for the image, while the network has considered it in the predicted caption. As expected, this image region is not recognized as an important region in the occluded image. Figure 5.7.b shows that our network has identified bird feeder as a tree since they are almost similar. Moreover, the bird feeder concept was not in the trained descriptions by the network. Recognizing similar objects is one of the challenges of image captioning models. The occluded image also shows that our model focused on the bird region. In Figure 5.7.c photo, a bus is at a bus stop, and our model could detect it well. In this example, the model appropriately distinguished the position and status of attributes relative to each other.

5. ADVANCING IMAGE CAPTIONING VIA DEEP LEARNING

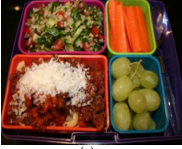
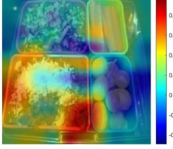

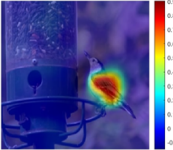

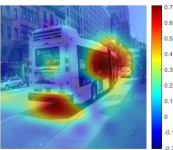

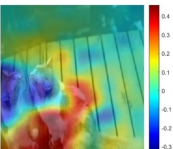

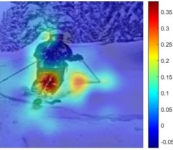

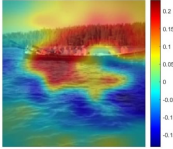
 <p>(a)</p>	<p>Truth Captions:</p> <ol style="list-style-type: none"> 1- These platters display healthy food choices of two entrees with a side vegetable and fruit. 2-A served tray filled with smaller plates of food. 3-A lunch tray with multiple compartments filled with food. 4-Four plastic containers filled with food on a table. 5-A four compartment tray holding various food items. <p>Predicted Captions:</p> <p>'ssss a plate of food and salad with vegetables and fruit eeee'</p>	
 <p>(b)</p>	<p>Truth Captions:</p> <ol style="list-style-type: none"> 1-A hummingbird standing on top of a green feeder. 2-A small bird resting and eating from a bird feeder. 3-a small bird on a bird feeder 4-The bird is standing on the rim of the bird feeder. 5-A small bird contemplates how to get some seeds. <p>Predicted Captions:</p> <p>'ssss a bird is sitting on a tree in the background eeee'</p>	
 <p>(c)</p>	<p>Truth Captions:</p> <ol style="list-style-type: none"> 1-A double city bus is pulled up to a bus stop 2-A city street scene with a bus and buildings 3-A city white bus stopped at a bus stop in front of tall buildings 4-A stopped bus pulled up to the bus stop 5-A city bus that is stopped at a bus stop <p>Predicted Captions:</p> <p>'ssss a large bus is parked on the side of a street eeee'</p>	
 <p>(d)</p>	<p>Truth Captions:</p> <ol style="list-style-type: none"> 1-A little cat looking at itself in a mirror. 2-A white and orange cat looking at itself in front of a mirror. 3-a brown and white cat looking it itself in a mirror 4-An orange and white cat standing in front of a mirror. 5-A cat on a porch looking at its reflection. <p>Predicted Captions:</p> <p>'ssss a cat is sitting on a bench with another cat eeee'</p>	
 <p>(e)</p>	<p>Truth Captions:</p> <ol style="list-style-type: none"> 1-A person on skis skiing down a mountain slope. 2-A man is skiing on the snow slopes 3-A skier is in the snow going downhill. 4-A person with green skis skiing down a big hill. 5-A person on skis is skiing down a snowy hill. <p>Predicted Captions:</p> <p>'ssss a person is skiing down a hill with a snow board eeee'</p>	
 <p>(f)</p>	<p>Truth Captions:</p> <ol style="list-style-type: none"> 1-Large canoe with many people on lake with trees lining shore. 2-A group of people paddle a long canoe in a clear lake bordered by pine woods. 3-Several people in a large rowboat with oars. 4-A big boat full of a lot of people. 5-A thick evergreen forest marks the boundary of a dark expanse of water, on which rests a long boat with packages at the rear and people to the fore, several holding long oars. <p>Predicted Captions:</p> <p>'ssss a canoe is a lightweight narrow vessel typically pointed at both ends and open on top propelled by one or more seated or kneeling paddlers facing the direction of travel eeee'</p>	

Figure 5.7: Generated examples by the best proposed model.

Information about the posture and location of attributes is one of the advantages of using a capsule network in our model. An interesting point about Figure 5.7.d photo is that our model has detected two cats in the image; however, the network did not notice one of them was the image of the first cat in the mirror. Moreover, the occluded image focused on the area of cats in the image. The photo of the skier person (Figure 5.7.e) has been described correctly, and the vital region of the image perfectly matches the generated caption in the occluded image. However, the ski board has been detected as a snowboard. Our model generates a longer and more detailed caption for Figure 5.7.f. Using the Wikipedia database to enrich the description of attributes in the image is, to some extent, noticeable.

In summary, the additional profit of our proposed framework is improving the image captioning performance by employing a network that can produce more comprehensive features concerning relational information between all the objects in the image. Therefore, the model generates denser and more diverse captions. Moreover, we compensated low resource language words by adding more external knowledge from Wikipedia to the dataset. So, the decoder can benefit from rich-resource captions through the training process. In terms of computation time, parallelizing the convolution layers in the enhanced version of the capsule network reduces the dimensions of the features fed to the primary capsules and accelerates the learning process.

5.7 Conclusions

In this chapter, we elaborated an encoder-decoder framework employing a parallelized capsule network as a feature extractor and the Wikipedia database as an external knowledge provider to establish if this approach can outperform state-of-the-art solutions. We implemented different architectures to produce contextual knowledge from images to achieve this. The models were trained on the MS-COCO dataset and evaluated based on BLEU (1-4), ROUGE, and METEOR scores. Two baseline models were included in our experimental setup and were compared with all different models in our study to obtain a baseline performance. Our novel approach demonstrated that using the proposed parallel capsule network as an encoder model provided a versatile image feature extractor.

We have demonstrated that the use of external knowledge further improved the results. Our best model was trained with the capsule network and inception-V3 as

5. ADVANCING IMAGE CAPTIONING VIA DEEP LEARNING

a feature extractor, with caption enrichment by an external contextual description. The results are the basis for future work that will generate more conceptual and specific descriptions by considering emotions in captions and using transformers in the decoder since transformers have extraordinary performance in image captioning (Choi and Choi, 2022). Our proposed approach paves the way to deal with these challenging extensions.