



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Αυτόματη Δημιουργία Περιγραφών Εικόνων : Ποιοτική ανάλυση των περιγραφών. Natural Language Description of Images : A Qualitative Analysis.
Όνοματεπώνυμο Φοιτητή	Νικόλαος Παναγιάρης
Πατρώνυμο	Γρηγόριος
Αριθμός Μητρώου	ΜΠΣΠ/ 14067
Επιβλέπων	Ευθύμιος Αλέπης, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης **Οκτώβριος 2018**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Ευθύμιος Αλέπης
Επίκουρος Καθηγητής

Κωνσταντίνος Πατσάκης
Επίκουρος Καθηγητής

Μαρία Βίββου
Καθηγήτρια

Abstract

Image captioning is a challenging problem that lies at the intersection of computer vision and natural language generation. The task involves the generation of a fully-fledged natural language sentence that accurately summarizes the contents of an image. Image captioning is also the cornerstone towards real-world applications with significant practical impact, ranging from aiding visually impaired users to personal assistants to intuitive human-robot interaction.

The advance in image captioning has been marked as a prominent success of Artificial Intelligence. It has been reported that with certain metrics, like BLUE or CIDEr, state-of-the-art techniques surpasses human's performance. Thus, a natural questions that rises is : Do humans and machines speaking the same language?

An observation that well established in linguistics, is that different human speakers or the same speaker produce different descriptions when presented with an image. This observation has been overlooked by today's systems. However, this poses serious questions for both the development of algorithms and their evaluation. Therefore this thesis, tries to answer on which premises the the state-of-the-art algorithms for the generation of image captions are build upon. Are they trying to emulate or predict the behaviour of individual speakers in a given situation? With the aim of shedding light on this question, a model based on the encoder-decoder model was implemented. The output of the model was qualitatively analyzed towards two factors: (1) whether is biased towards frequent captions in the training set; (2) and whether better image representations enrich the language production.

ΠΕΡΙΛΗΨΗ

Η αυτόματη δημιουργία προτάσεων που περιγράφουν το περιεχόμενο μιας εικόνας, αποτελεί ένα σημαντικό πρόβλημα της τεχνητής νοημοσύνης. Συγκεκριμένα, βρίσκεται στην ένωση των επιστημονικών πεδίων της Υπολογιστής Όρασης και της επεξεργασίας φυσικής γλώσσας με μια σειρά από σημαντικές εφαρμογές όπως η αλληλεπίδραση ανθρώπου- ρομπότ.

Η επιτυχία αυτού του νέου επιστημονικού πεδίου έχει χαρακτηριστεί ως μια από τις σημαντικότερες επιτυχίες της τεχνητής νοημοσύνης έως τώρα. Συγκεκριμένα, δημοσιευμένες εργασίες παρουσιάζουν αποτελέσματα τα οποία είναι καλύτερα από αυτά που έχουν επιτύχει άνθρωποι. Επομένως, αξίζει κάποιος να αναρωτηθεί αν πλέον τα ευφυή συστήματα έχουν ισάξιες γλωσσικές ικανότητες με αυτές των ανθρώπων.

Πολλές μελέτες, στο πεδίο της γλωσσολογίας έχουν αποδείξει ότι οι άνθρωποι παράγουν διαφορετικές περιγραφές για μια εικόνα. Στην πραγματικότητα, ο ίδιος άνθρωπος μπορεί να παράξει διαφορετικές περιγραφές ανάλογα με την περίπτωση. Όπως είναι αντιληπτό, αυτή η ποικιλότητα στην παραγωγή γλώσσας δημιουργεί μια σειρά από προβλήματα στην δημιουργία αλγορίθμων αλλά κυρίως στο πως αυτοί οι αλγόριθμοι θα αξιολογηθούν. Αυτή η διπλωματική εργασία επιχειρεί να ερευνήσει ποιες αρχές διέπουν τους αλγόριθμους αυτόματης δημιουργίας περιγραφών εικόνας. Συγκεκριμένα προσπαθεί να απαντήσει την ερώτηση αν οι αλγόριθμοι μιμούνται η προβλέπουν την συμπεριφορά των ανθρώπων δεδομένης μιας εικόνας. Για να απαντηθεί αυτή η ερώτηση υλοποιήθηκε ένα μοντέλο περιγραφής εικόνας του οποίου τα αποτελέσματα εξετάστηκαν ποιοτικά και ποσοτικά ως προς το αν αναπαράγουν τις περιγραφές πάνω στις οποίες εκπαιδεύτηκε το μοντέλο και αν καλύτερες αναπαραστάσεις εικόνας βελτιώνουν το γλωσσικό αποτέλεσμα.

Table of contents

List of figures	v
List of tables	vii
1 Introduction	1
1.1 Structure of the Thesis	3
2 Background	4
2.1 Neural Networks	4
2.2 Perceptron	5
2.2.1 Activation	6
2.2.1.1 Logistic or Sigmoid function	6
2.2.1.2 Hyperbolic tangent (tanh)	7
2.2.1.3 Rectified linear unit (ReLU)	7
2.2.1.4 Softmax	8
2.2.2 Multi-Layer Perceptron	9
2.2.3 Training	10
2.2.3.1 Loss Function	10
2.2.3.2 Backpropagation	11
2.2.3.3 Optimization Algorithms	11
2.3 Deep Learning	12
2.3.1 Feed-Forward Neural Networks	13
2.3.2 Convolution Neural Networks	15
2.3.2.1 AlexNet	20
2.3.2.2 VGG-16 and VGG-19	21
2.3.2.3 GoogleNet or Inception	23
2.3.2.4 ResNet	25
2.3.2.5 Recurrent Neural Networks	26

2.3.2.6	Long Short-Term Memory	27
3	Image Captioning: A survey of Models and Datasets.	30
3.1	Description as Generation from Visual Input	31
3.2	Description as a Retrieval in Visual Space	33
3.2.1	Description as a Retrieval in Multimodal Space	35
3.2.2	Retrieval and template based methods augmented by neural networks	35
3.2.3	Image captioning based on the encoder-decoder framework	37
3.2.4	Adversarial Generation of image captions	39
3.3	Datasets	40
4	Methodology	42
4.1	Model Architecture	42
4.1.1	Visual Feature Extraction	44
4.1.2	Language Model	45
4.1.2.1	LSTM Cell	45
4.1.2.2	Proposed LSTM Language Model	47
4.1.3	Training and Regularization	48
4.1.4	Inference	49
4.2	Evaluation Metrics	50
4.2.1	BLEU	51
4.2.2	ROUGE-L	51
4.2.3	METEOR	51
4.2.4	CIDEr	52
4.2.5	Automatic Metrics Discussion	53
5	Experiments	55
5.1	Implementation Details	55
5.2	Image captioning results	58
6	Conclusion & Future Work	61
	References	64

List of figures

1.1	This figure shows an image with captions generated by humans, and an image captioning model [23].	2
2.1	The role of the activation function in the neural network model.	6
2.2	Standard Sigmoid function.	7
2.3	Standard hyperbolic tangent function.	8
2.4	An an example arrangement of neurons in a 1-layer neural network.	14
2.5	Architecture of LeNet-5 [75]	15
2.6	Computing the output values of a 2×2 average pooling operation on a 4×4 input using 1×1 strides.	17
2.7	Performance improvement on the ILSVRC ImageNet.	19
2.8	An illustration of the architecture of AlexNet,explicitly showing the flow of tensors between two GPUs [65].	20
2.9	An illustration of the Inception module [117].	23
2.10	A simplified residual unit which consists of two stacked convolutional layers activated using ReLUs. The input of the first convolutional layer is additively merged with the output of the last convolutional layer, which is what, in this document referred as a shortcut.	25
2.11	Basic LSTM memory block.	27
3.1	General structure of compositional image captioning methods [4].	37
3.2	The encoder-decoder model proposed by [63].	38
3.3	The GAN framework for image captioning proposed by [15].	39
4.1	A high-level block diagram of the visual captioning pipeline.	43
4.2	The steps required for processing images before are fed to the VGG-16.	45

4.3	A memory block contains a cell c which is governed by three gates. The dotted lines to the rectangle are indicating the multiplications used as gate controls, while the solid line is the data flow representations. The triangles are non-linearities that model is fused with.	46
4.4	The LSTM-based language model unrolled in time	47
4.5	Probability density predicates of BLEU, Meteor, ROUGE, and CIDEr scores compared to human judgment in the Flickr8K dataset [7]. The y-axis depicts the probability density while the x-axis is the score of the automatic metrics.	54
5.1	Greedy Search: A dog runs through the grass. Beam Search = 3: A brown dog runs through the grass. Beam Search = 7: A dog runs through the grass. Beam Search = 11,20 : A dog.	56
5.2	The fifty most frequently appearing words in the training captions [Win]	57
5.3	The fifty least frequently appearing words in the training captions [Win]	57
5.4	The image on the left is the one from the training images. The model produced as the a caption “ a man and a woman are sitting on the edge of a lake ”. The right image is one of the training test and its accompanied caption is “A man and woman sitting on a deck next to a lake”	59
5.5	For the left image the caption produced by the model is “a football player in red is challenging the player in red ” while for the image from the right is “a group of football players are tackling a football player’	60
5.6	The left image is derived from the test set while the right image is derived from the training set.	60
5.7	Successful captions produced by the implemented image captioning model.	60

List of tables

2.1	AlexNet network architecture implementation details [65]	21
2.2	VGG networks architecture implementation details [110].	22
2.3	GoogLeNet incarnation of the Inception architecture [117].	24
3.1	Image captioning datasets.	40
5.1	The overall hyper-parameters used in the final model.	58
5.2	Evaluation Results	58

Chapter 1

Introduction

Since the advent of Deep Learning (DL) methods, the fields of Natural Language Processing and Computer Vision (CV) have greatly advanced towards their core goals of understanding and generating text and images respectively. Despite that both fields using similar methods, they have traditionally been developed separately. However, recent years the staggering growth of the use of multimedia on the internet has lead to an upsurge of interest in problems that require the processing of both linguistic and visual information. Nowadays, the web provides a vast amount of data that combines both linguistic and visual information such as tagged images, video with subtitles or multimedia feeds across social media platforms. In order such vast amount of data to be exploited, the NLP and CV communities have been move closer together and the field of language-vision emerged.

In the language-vision field, automatic image description has received the most scholarly attention. Loosely speaking, this task involves the analysis of the visual content of an image and the generation of a textual description (typically a sentence) that conveys the most salient aspects of the image. Apparently, this task is notably challenging from a Computer Vision point of view since the description could in principle refer to any visual aspect depicted in the image (e.g. mention objects, their attributes or their relation). However, more challenging the description could even contain information that is not explicitly depicted in the image (e.g. it can describe people waiting for a bus, even when the bus is not visible due to the fact that has not arrive yet). In other words, in order for an image captioning system to produce an accurate description the full understanding of the image is required.

Although the starting point of every image captioning system is the image understanding is not sufficient enough. For example, suppose that a cascade of state-of-art vision models are applied to localize objects depicted in the images, determine their attributes, compute scene properties in order to decide which of those convey the most salient information. As one would expect, the output of such pipeline would be raw detections (i.e. labels), which

would be unsuitable to be considered as a fully-fledged sentence. On the contrary, a caption, has to be comprehensive but concise (includes only those properties that individualize the image), has to be formally correct (i.e. well formed sentences) but also able to capture the variability that humans speakers show in order to sound human-like [7]. Consequently, automatic caption generation does not only requires in depth image understanding, but also sophisticated NLG systems. Given its importance from a application point of view it also poses challenging questions to both Computer Vision but also NLG communities. Driven by the existence of mature vision but also NLG technologies and the availability of relevant datasets, the past three years an increasing number of works that use Deep Learning models has been produced.


	Automatically produced captions.	A train that is pulling into a station.
		A train that is going into a train station.
		A train that is parked in a train station.
	Human produced captions.	A train pulling into a station outside during the day.
		A passenger train moving through a rail yard.
		A long passenger train pulling up to a station.

Fig. 1.1 This figure shows an image with captions generated by humans, and an image captioning model [23].

State-of-art image captioning models today tend to be monolithic neural models, essentially of the “encoder-decoder” paradigm. Images are encoded into a vector with a convolutional neural network (CNN) , and captions are decoded from this vector using a Recurrent Neural Network (RNN) (see Chapter 2), with the entire system trained end-to-end. It has been reported that with certain, metrics like BLUE [94] or CIDEr [118] those techniques have already surpassed human’s performance. A natural question, thus, rises: *has the problem of generating image captions has been solved?* In order this question to be answered, one should take a step back and look at the samples in Figure 1.1 produced by the dominant approach in image captioning, that is the “encoder-decoder” framework proposed in [119] and followed in this thesis. Although the captions, faithfully describing the content of the images, those captions are not novel. In particular, the captions use n-grams that appeared in the training samples and have a smaller vocabulary. In Figure 1.1 becomes apparent that the model reveals itself especially when multiple captions are shown in succession.

Therefore, this thesis explore the following questions:

- RQ 1: Do different images representations affect the language production?
- RQ 2: Is the decoder-encoder model produces novel image captions?

In order to answer those questions, the implementation and analysis of a popular architecture proposed in literature [119] was performed. In particular, in order to answer the RQ-1 the model was exposed to different image representations extracted with the use of state-of-the-art CNN models that were trained on the ImageNet dataset [26] (see Chapter 2). Towards answering the second research question, the captions produced by the model were analyzed with the use of various evaluation metrics adopted from the field of machine translation as is proposed in the literature. However, evaluating an image caption system and a Natural Language Generation system is a non-trivial task [7]. Thus, n-gram statistic and word usage as a proxy for measuring how closely the generated captions mirror the distribution of the training samples was performed.

1.1 Structure of the Thesis

The rest of the thesis is organized as follows. The Chapter 2 elaborates on the theoretical background on Neural Networks and Deep Learning methods. An extensive review of CNN from object recognition is also given. The third chapter presents an extensive literature in the image captioning field. Several, related works on visual captioning are presented and discuss the datasets available to train such models. The details of the implemented caption generator used in this thesis and the automatic metrics used to evaluate a captioning system are discussed in Chapter 4. Chapter 5 contains results from several experiments conducted from the purposes of this thesis. In Chapter 6, shortcomings of the implemented visual captioning systems are identified and a few ways to address these issues are discussed.

Chapter 2

Background

2.1 Neural Networks

Neural networks are a machine learning technique incorporating principles that have been observed in the biological nervous system. A biological neuron is a cell consisting of a body, multiple smaller protrusions called dendrites and a single long protrusion called an axon. On top of the dendrites are found numerous spots, known as synapses, where axons of other neurons are creating a connection with the neuron. Often, axons are producing an electric pulse which affects the permeability of cell's which leads to the slight increase of the voltage inside the neuron body. The more activations come from other neurons, the higher the voltage grows. Given a certain threshold, if the voltage is below that threshold nothing particular happens. When this voltage exceeds these particular threshold the neuron produces an action potential meaning that a signal starts propagating through the axon. This axon can be attached to the dendrites of one or more other neurons, where the process is repeated in an analogous way.

This simplified function of biological neurons is stimulated by the artificial neuron networks meaning that the artificial neurons are units which contain one or more inputs and produce a single output. The activation of an artificial neuron depends on how much of non-zero inputs the neuron receives. It can remain either inactive and produce a zero output or it can produce an action potential which is represented by a non zero output. Multiple neurons are connected to one another, producing an (artificial) neural network.

Artificial neural networks (ANNs), also known as neural networks (NNs) are data-driven machine learning algorithms that process information by their dynamic state response to external inputs [13]. Its origins are dated back to 1943, to a study of mathematical representations of information processing in biological systems by McCulloch and Pitts [113]. In its most simple form a NN is a network of interconnected nodes and simple

processing units known as neurons, the neurons are joined with weighted connections and calibrate the strength of the transmitted signals in a similar manner like synapses in human brain. Its resemblance with the brain is denoted by two aspects; that the knowledge is acquired through the learning process and that it is stored in connections between neurons [47].

Neural networks have been used in a wide range of machine learning tasks such as pattern classification or regression. The principal components of neural networks are neurons, layers, and activation functions despite the variety of the architectures that have been introduced over the years

2.2 Perceptron

The fundamental unit of most neural networks is called a perceptron or simply neuron. Like its biological counterpart, a neuron can be seen as a sort of a “black box” which takes a fixed number of inputs and produces a single output. To each input, a weight is assigned indicating the extent to which this particular connection affecting the resulting output. The rough idea described above can be formalized as follows:

Let $X = (x_1, \dots, x_n)$ represent the vector of input data and $w = (w_1, \dots, w_n)$ be the vector of corresponding weights. In order the output to be calculated, we first need to compute the inner potential of neurons which is given as follows :

$$\xi = w_X = \sum_{i=1}^n w_i x_i \quad (2.1)$$

The activation functions are task at hand depended. In case of classification, as in our case, threshold activation functions are used. A threshold activation function in its primitive form can be given:

$$\sigma(\xi) = \begin{cases} 1 & \xi \geq h \\ 0 & \xi < h \end{cases} \quad (2.2)$$

Where h is a fixed arbitrary real-valued parameter (threshold). Another form of artificial neuron is a neuron with bias. The bias neuron an extra input x_0 and a corresponding weight w_0 are considered in order the weighted to be computed. If the $w_0 = -h$ then the activation function takes the following form:

$$\sigma(\xi) = \begin{cases} 1 & \xi \geq 0 \\ 0 & \xi < 0 \end{cases} \quad (2.3)$$

Where ξ now starts from zero instead of one. This approach is the dominant approach in neural network literature henceforth any mention of term “neuron” is only to be as a neuron with bias.

2.2.1 Activation

The output of each node is calculated by the node’s activation function that takes weighted inputs of the node as parameters transformed from a transfer function as can be seen in Figure 2.1. The transfer function aims to create a linear combination of weighted inputs in order to provide them to the activation function. In the sections that follow briefly, we discuss activation function that we are going to use later in this thesis.

2.2.1.1 Logistic or Sigmoid function

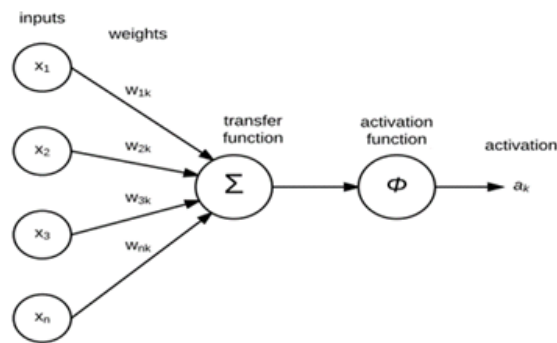


Fig. 2.1 The role of the activation function in the neural network model.

A sigmoid function [22] is a monotonically increasing function (see Figure 2.3), which reached an asymptote at some finite value as $\pm\infty$ is reached. In terms of neural networks, the most widely used sigmoid function is the standard logistic defined as:

$$\sigma(\chi) = \frac{1}{1 + e^{-\chi}} \quad (2.4)$$

The logistic or sigmoid function is a smoother approximation of the step function that was used in the early versions of neural networks. The output values are in range $[0, 1]$ hence it is suitable for output neurons that perform classification task. However, an important problem arises, when the weights of the network are initialized with small values almost zero. The initial activations for the standard logistic function will be then set 0.5 on average. Thus, sigmoid that are symmetric about the origin are preferred to be used because tackle this problem by producing always positive outputs and help gradient-based optimization.

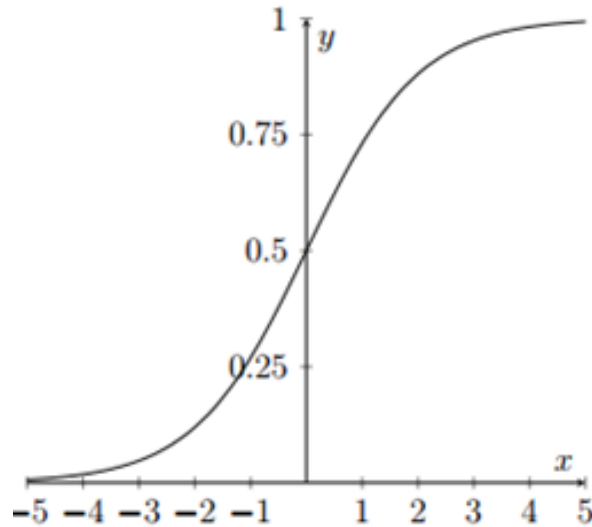


Fig. 2.2 Standard Sigmoid function.

From optimization point of view, another problem occurs. The problem is that the derivatives for sigmoid are vanishing near saturation points, Thus, this makes it harder for a neuron to propagate the error signal and move out from the saturation points.

2.2.1.2 Hyperbolic tangent (tanh)

The Hyperbolic tangent (see Figure) is a non-linear S-shaped function like the sigmoid mentioned before but its lower horizontal asymptote is at -1 instead of 0 . The fundamental difference between those two is that the tanh is zero-centered with a steeper rise which leads the classification models to reduce the number of misclassified samples. The tanh functions are differentiable and is given:

$$\tanh(\chi) = \frac{e^{\chi} - e^{-\chi}}{e^{\chi} + e^{-\chi}} \quad (2.5)$$

2.2.1.3 Rectified linear unit (ReLU)

The rectifier activation function has been shown that can improve the discriminative performance of convolutional networks[73]. The rectifier non-linearity is defined as follows:

$$\text{ReLU}(\chi) = \max(0, \chi) \quad (2.6)$$

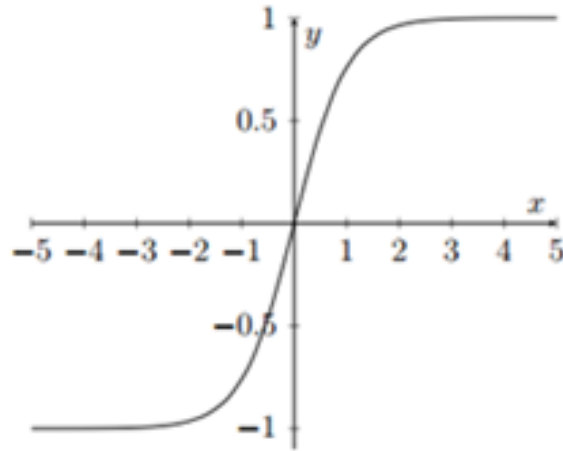


Fig. 2.3 Standard hyperbolic tangent function.

It has been shown that ReLU gives true sparsity to the model due to zero constrain, which means that only a small set of values are nonzero. Moreover, it does not present vanishing gradient problem because the function is becoming linear when $x > 0$. Additionally, sparse representation is biologically plausible which means that for a given set of neurons, most of them are inactive and just a few are activated by some input. The main reason that makes sparsity desirable to machine learning algorithms is that sparse features are effectively lower the number of dimensions and sufficiently tackles the so called curse of dimensionality. Thus, sparse representations are more robust in small changes to the inputs compared to dense representations. It has been proven that the discontinuity at zero can affect optimization techniques. However, when smoother versions of rectifier were applied the performance was worse due to the fact that that exact sparsity was lost.

2.2.1.4 Softmax

The Softmax activation function is widely used in the last layer of the networks, it aims to interpret an arbitrary real value to the posterior probability of the class in range $(0, 1)$:

$$p(c_k|x) = \frac{e^{a_k}}{\sum_{i=1}^m e^{a_i}} \quad (2.7)$$

Where m is the number of output neurons (classes) a_k is the activation value of the k th node:

$$a_i = \sum_{j=0}^d w_{ij}h_j(x) \quad (2.8)$$

Where w_{ij} is the i -th node's weights and the $h_j(x)$ is the output of the previous layer.

2.2.2 Multi-Layer Perceptron

In case we have only one neuron unit, the output value is used as the final indicator of the assigned class to the input vector. Someone, could feed this output to another neuron instead. Hence, in that way it is possible a network consisted of interconnected neurons to be created. Numerous different topologies have been proposed from different perspectives and different purposes. However, the most widely used topology of neural networks are the multi-layer feedforward networks.

Having explained some baseline information about neural networks now we are about to introduce formally the simplest version of neural network, the multi-layer perceptron (MLP) [104]. The network consists of a certain number of neurons that are then split into several disjoint sets, the so called layers. The layers in MLP architecture are ordered sequentially and each layer receives input from the preceding layer. The input layer is not considered as a “true” layer because no computation is performed in it. It receives problem-specific inputs from the outside world. An MLP contains one or more hidden layers, which receive inputs from preceding layers (input or hidden layers) and their outputs connect to the next layers (output or hidden layers). Each neuron in a hidden layer employs a nonlinear activation function that is differentiable. The output layer presents the final result of the computation performed by the network to the outside world. For a given input layer x , the network computes the hidden activation vector h and the output vector y as follows:

$$h = f(W^x h x + b^h) \quad (2.9)$$

$$y = G(W^x y h + b^y) \quad (2.10)$$

Where W are the weight matrices of two connected layers, $W^x h$ is the matrix that contains the weights of input layer, $W^h y$ is the matrix that contains weights of hidden layer, b are the bias vectors and F and G are the activation functions. The network is highly connected thus all the neurons in one layer are connected to all neurons in the following layer. Hence, multi-layer perceptron is a feed-forward NN which means that the information inside the network moves from input neurons through hidden layers’ neurons to the output neurons. The MLP’s expressive power is given by the universal approximation theorem [6]. The theorem states that a single hidden layer MLP and sufficient number of nonlinear units are sufficient to approximate any smooth function and a compact input domain with arbitrary precision. However, the number of hidden units that are required is unknown and sometimes can be so large and thus inapplicable. The use of hidden layers helps the partition of input space into exponentially more linear regions than a shallow network, with same number

neurons does. That ability allows them to represent easily highly structured and complex functions.

2.2.3 Training

2.2.3.1 Loss Function

The optimization objective for supervised neural network training is based on a loss function that measures how well the network performs on the training data. As the name implied, the output of the loss function needs to be minimized to receive the best-performing model (as opposed to maximization of the likelihood of the data under the model as it is common for MaxEnt). There are multiple loss functions which are common in the literature, and which are usually preferably used with specific output activation functions. We will briefly introduce the two most common functions.

Mean squared error (MSE): The MSE function assumes that the errors are normally distributed. It is defined as:

$$MSE(y, \hat{y}) = \frac{1}{N} \sum_{n=1}^N (y_n - \hat{y}_n)^2 \quad (2.11)$$

MSE assumes that the errors are normally distributed. It is frequently used in combination with the tanh and linear activation functions.

Cross-entropy error (CEE): The CEE function measures the cross-entropy of the output and the target values. As a consequence, it is only applicable if the predictions and targets are probabilities. It is a natural choice for sigmoid and Softmax activation functions. When used in combination with one of these functions, the computation of the derivatives in backpropagation can be simplified significantly. The CEE function for the multiclass case, i.e., when using Softmax activation, is defined as follows:

$$CEE(Y, \hat{Y}) = \sum_{n=1}^N \sum_{d=1}^D \hat{y}_{nd} \ln y_{nd} \quad (2.12)$$

2.2.3.2 Backpropagation

The concept of backpropagation algorithm was first introduced by Paul Werbos in his 1974 PhD thesis. Backpropagation is a neural network training strategy. In supervised learning, Target classes are essential the baseline for error calculation. Then errors are backpropagated to each node in previous layers. Error e is obtained as gradient of the loss function L with respect to each layer's weights w_{kj} given input of the node x and an activation function.

$$a_j = \phi\left(\sum_{k=1}^n w_{kj}x_k\right) \quad (2.13)$$

$$e = \frac{L}{w_{kj}} = \frac{L}{w_{kj}} \frac{a_j}{w_{kj}} \quad (2.14)$$

Gradient computation demands application of the chain rule in order to compute partial derivative of the loss function L with respect to particular weight w_{kj} . Using the error, weights are updated by an optimization algorithm such as stochastic gradient descent. Basic stochastic gradient it usually leads to slow convergence of the network. Thus, several techniques have been proposed in literature for optimization algorithms such as Adagrad [34], Adatadelta [128], RMSprop, and Adam [61] all of the which can improve greatly the speed convergence. Here we briefly describe RMSprop and Adam which will be used in the subsequent sections of this work.

2.2.3.3 Optimization Algorithms

RMSprop: RMSprop is an optimizer having per-parameters adaptive learning rate. Incorporates a moving average of the magnitude of recent gradients in order to normalize the current gradients. The normalization is performed over the root mean soared gradients. The learning rate or the step rate and the running average term $r(\tau)$ is added to the weight update equation:

$$r(\tau) = \gamma r(\tau - 1) + (1 - \gamma) \left(\frac{E}{w_i}\right)^2 \quad (2.15)$$

$$\Delta w(\tau + 1) = \frac{-\eta}{\sqrt{r(\tau)}} * \frac{E}{w_i} \quad (2.16)$$

Where γ is the decay value which calibrates the contribution of new gradients for the running average $r(\tau)$. One key aspect is to initialize weights W and biases b to positive and negative values close to zero, in order the sigmoidal activation functions to operate on the central region which leads to larger propagated gradients [43]. Usually, the values are

drawn in a random and independent order from uniform or Gaussian distributions. The input features, for the reason we mentioned above, are normalized so as to have zero mean and unit variance in each dimension.

Adaptive Moment Estimation (Adam) Adam [62] is the second method that we examine. This method computes the adaptive learning rates for each parameter. But in contrast of RMSprop, Adam is not only keeps an exponentially decay average of the past squared gradients u_t but it keeps also an exponentially decaying average of the past gradients m_t :

$$m_t = \beta_1 m_{(t-1)} + (1 - \beta_1) g_t \quad (2.17)$$

$$u_t = \beta_2 u_{(t-1)} + (1 - \beta_2) (g_t)^2 \quad (2.18)$$

m_t, u_t are the estimates gradients of first and second moment respectively. The Adam update rule is given as follows:

$$\theta_{+} = \theta_{\tau} - \frac{h}{\sqrt{\hat{u}_t}} \hat{m}_t \quad (2.19)$$

Usually the default values are 0.9 for β_1 10^{-8} for β_2 and 0.999 for β_2

2.3 Deep Learning

The past few years a number of fields have been revolutionized by the use of Neural Networks. For instance, the field of visual recognition, or related tasks such as image segmentation and object recognition, witnessed drastic changes. In particular, the state-of-the-art computer vision models based on Convolutional Neural Networks [75] have become capable of distinguishing thousands of visual categories performing comparable to humans, or even, outperform them at accuracies in some fine-grained categories such as breeds of dogs [106]. The two fundamental blocks of these models are based on techniques developed starting in the 1940s and have been described in the previous sections. In particular, early neuron models with adjustable synaptic strengths and learning rules were developed under the name of “cybernetics” [87, 102, 122]. In the mid 1980s, under the name of “parallel distributed processing” [105] introduced the use of back-propagation for training networks of neurone-like units as presented earlier. However, the first model that successfully combines those two techniques for visual recognition problems (digit recognition) is LeNet-5 [74]. This can be seen as a primitive description of the modern Convolutional Neural Networks

(CNNs). However, their success on large-scale visual recognition problems, such as the Imagenet challenge [105] was only possible a few decades later with significantly more computational power and the availability of enormous datasets [65, 73, 44]. Those models are now known under the name of “Deep Learning”.

Broadly speaking, CNNs describe a function that maps an input space (e.g. images) to an output space (e.g. probability of classes that best describes an image) and the parameters of this functions are learned on a large collection of labeled images (e.g. ImageNet dataset). This approach has two advantages: (1) is trained on a “end-to-end” fashion, that is, the objective of the entire computational process shares the same objective (correctly classify images); and more importantly (2) once those models are trained on a particular dataset can be used as a fixed feature extractor mechanism for images. Most of the work on image captioning with Deep Learning utilizes the latter advantage of CNNs directly.

In addition to progress in visual recognition, the field of Natural Language Processing has witnessed similar successes. In particular, Recurrent Neural Networks (RNN)[121, 50] have proven efficient in modeling natural language. Typically, such models use image features extracted from the pre-trained CNNs to sample terms from a vocabulary in such a way that a sequence of those terms to result in a concrete conceptual description 1 either of image as a whole or of an object within that image. Their main advantage over n-gram models is that they can represent sequences of variable length, while avoiding data sparsity and reducing the number of parameters through jointly representations of similar histories (previous time states) [41]. Moreover, are able to handle long-range dependencies. In the next sections a detailed presentation of those models are given since they are fundamental blocks of the proposed method.

2.3.1 Feed-Forward Neural Networks

Before describing the Convolution Neural Networks, a presentation of the basic feed-forward Neural Network is given. In a supervised learning scenario, a set of labeled data is given in the following form : $\{(x^{(i)}, y^{(i)})\}$. Broadly speaking, neural networks provide a mechanism of representing a complex nonlinear function $h_W(x)$ for the input variable x . The function $h_W(x)$ is parameterized by a weights matrix \mathbf{W} that is tuned in such way to fit the input data.

A Neural Network is consisted of multiple layers. For example the network depicted in Figure 2.4 is comprised of three layers: the input layer, the hidden layer, and the output layer. Neurons in one layer have connections to all neurons in the previous layer but are not connected to each other (i.e. the network is fully connected). The arrangement of neurons of a neural network is often referred to as the network’s architecture.

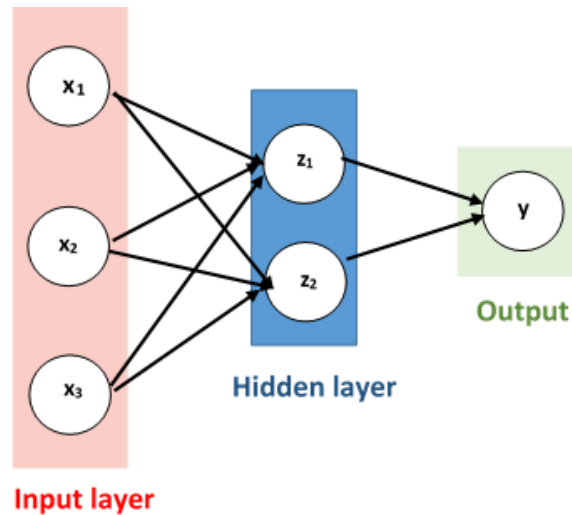


Fig. 2.4 An an example arrangement of neurons in a 1-layer neural network.

Apart from the neurons in input layer, each neuron x_i in the neural network is a computational node: it takes as input the values of the neurons of the preceding layer connected to it. Concretely, the inputs to the neuron labeled z_1 (Figure 2.4) are: x_1 , x_2 and x_3 and the input to the last layer is z_1 , z_2 . Each node computes a weighted linear combination of its input. More formally, let x_1, \dots, x_n be the inputs to a neuron z_j the computation is performed is the following:

$$a_j = \sum_{i=1}^n (w_{ij}x_i + b) \quad (2.20)$$

where w_{ij} are the weights, i.e. parameters that describe the interaction between the connected nodes and b_j term is the bias associated with the neuron z_j . The a_j are known as activations ¹. Common settings for the non-linearities are tanh, the sigmoid function $1/(1 + ex)$ and the rectified linear unit (ReLU) $\max(0, x)$. Since the activation of each neuron depends only upon the values of neurons in preceding layers, the computation of the activations starting from the first hidden layer (which depend only upon the input values) and proceed layer-wise through the network. This process where information propagates through the network is called the forward-propagation step.

The parameters W in the neural network are the weight terms for each of the edges as well as a bias term for each of the nodes ². The objective on which the parameters are learned is to minimize some objective or loss function. The commonly used approach to learn the

¹Note that the last layer of the neural network normally does not contain the activation function

²Excluding the ones in the inputlayer

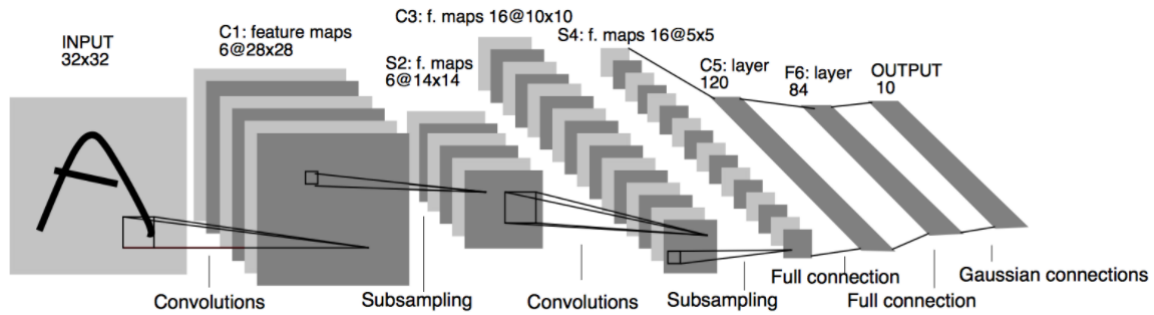


Fig. 2.5 Architecture of LeNet-5 [75]

parameters W in such a way that the objective function is minimized is through the error backpropagation algorithm.

2.3.2 Convolution Neural Networks

LeNet-5 [75] is considered to be the first Convolutional Neural Network architecture proposed in literature. In recent years, a booming number of architectures has been proposed which improves LeNet-5 in a number of aspects, which will be presented in the following section, but all of them use the same concepts as LeNet-5. This section briefly describes those concepts as depicted in Figure 2.5.

In its most primitive form, a Convolution Neural Network (CNN or Convnet) is a multilayer, hierarchical neural network [120]. There are, however, three fundamental differences that distinguish the CNN from the simple feed-forward neural network described in the previous section: local receptive fields, weight sharing, spatial pooling or subsampling layers.

In the simple neural networks, every neuron is connected to each of the neurons in the next layer, i.e. each neuron in the hidden layer computes a function that depends upon the values of every node in the input layer. However, in case the input is an image the dimensionality is high, therefore, the use of fully connected layers increases drastically the number of parameters and the processing time. Hence, in such cases the network's architecture should be aware of the spatial layout of the input and exploit the local structure within the input. For example, neighboring pixels within an image tend to be highly correlated while distanced pixels tend to be weakly or even uncorrelated. This observation is not new, many standard feature representations used in computer vision use the sliding windows technique to extract local features within the image [24]. The CNN architecture exploits the local hierarchy within the image by using a local connectivity pattern between neurons of neighboring layers. In particular, it constrains each neuron to depend only on a spatially local subset of the variables of the preceding layer. More concretely, if the input to the CNN is a 32×32 image

patch, a neuron in the first hidden layer might depend only on a 3×3 subwindow instead of the whole input. Those nodes in the input layer that connect each neuron to only a local region of the input volume are called *receptive field* or *filters*. Loosely speaking, the receptive field defines the region in the input space that a particular node is “looking at”. Therefore, this characteristic leads to a sparser set of edges since adjacent layers are not always fully connected.

The second important feature of CNNs, is that they exploit weighting sharing schemes across deferent neurons in the hidden layers. As mentioned in Section ??, each neuron in the networks computes a weighted linear combination based on its inputs. However, this process can be seen as evaluating a linear filter over the input values [24]. In this view, sharing weights across neurons is the equivalent of evaluating the same filter over multiple subwindows of the input image. Therefore, CNN networks effectively learn a set of filters each of which is applied in all the subwindows of the input image. This procedure is taking place within the building block of a CNN, that is, the convolution layer. Evaluating a filter across all spatial positions of the input tensor amounts to *convolve* the image with the filter. The result of the *convolution step* of the CNN is the *convolution response map* or *activation map*. Since the convolution layer is the core computational module of a CNN a concrete example is given.

An image can be represented as a multi-dimensional matrix (i.e. tensor) of pixel values. Suppose that the input of the CNN is a color image I of width and height of 32 which is represented as a $32 \times 32 \times 3$ tensor. Consider also a $5 \times 5 \times 3$ filter f . The network convolves this filter by sliding it over all spatial positions of the input image. The convolution produces an activation map with size of 28×28 , where each element is the result of the dot product between the filter and the input. The size of the activation map, however, is defined by three parameters: (1) the depth, which in practice is the number of filters to be used in convolution; (2) the stride, which is the number of pixels that the filter slides; and (3) whether it is decided to pad the input.

The use of the same set of filters over the entire image results in a more abstract representation of the input data. The constraints that are enforced in order the weights to be equal through the network, have a regularizing effect on the CNN. Moreover, an obvious advantage of the weight sharing scheme is that reduces drastically the number of parameters that have to be learned, making markedly easier and more efficient to train such architectures since reduces over-fitting. In turn, this allows the network to generalize better in many visual recognition settings.

The third distinct characteristic of a CNN is the presence of a pooling layer which also plays an important role for further improving the regularization of the network. This

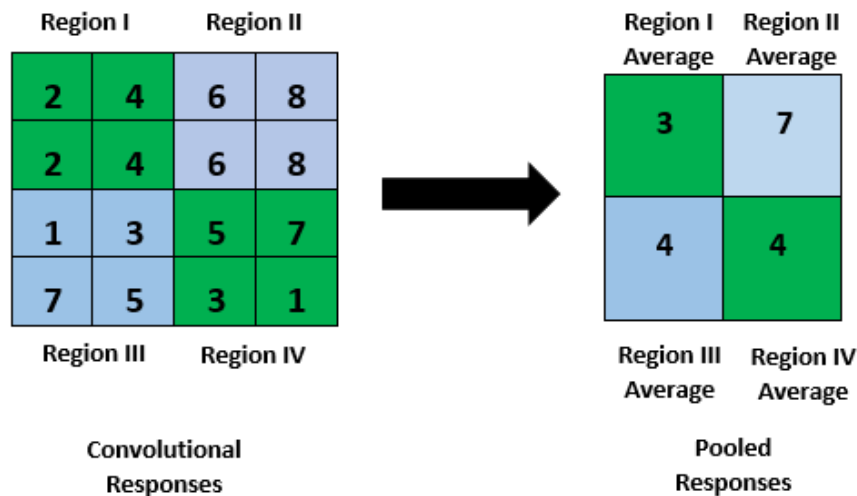


Fig. 2.6 Computing the output values of a 2×2 average pooling operation on a 4×4 input using 1×1 strides.

is achieved by reducing the dimensionality of the activation map and inject a degree of invariance into the model. The most commonly used pooling strategy is the spatial [10]. In this approach, the activation map is divided into a set of $m \times n$ blocks. Then, a pooling transformation function is applied over each of those blocks yielding to a reduced activation map of size $m \times n$. In the case of max pooling, the activation value for each block is taken to be the maximum value over the block of activations, and in the case of average pooling, the activation value is taken to be the average value of the block activations. In Figure 2.6 an example of average pooling is illustrated.

In a typical CNN architecture, there are multiple layers, alternating between convolution and pooling followed by one or several fully connected layers [82]. For example, another convolution-pooling layer can be stacked on top of the outputs of the first convolution-pooling layer. By doing so, the outputs of the first set of convolution-pooling layers are treated as the input to the second set of layers. In this way, a multi layered or deep architecture is constructed. Broadly speaking, the early convolutional filters, such as those in the first convolutional layer, can be thought of as providing a low-level encoding of the input data. In the case of image data, these low-level filters may consist of simple edge filters. The following layers, however, begin to learn more and more complicated structures. In practice, the depth of the network increases its expressive power. Since CNNs are a kind of feed-forwards networks they use the standard technique of error backpropagation used to train neural networks [9].

Convolutional Neural Networks have been shown to provide excellent performance in multiple areas, ranging from simple handwriting recognition [76], visual object detection [19] to character recognition [107]. Combined with the drastic advancements in distributed and GPU (graphics processing units) computations, it was made possible to train much deeper CNNs that achieve state of the art results on traditionally very challenging datasets for objects recognition such as ImageNet. Despite this encouraging progress, training such deep learning structures is still remains a challenge. The performance of the design is very sensitive to the implementation details, which is often the case [14]. Fortunately, using already trained structures is straightforwardly and the expressive power of the network's depth can be applied to other scenarios; offering great advantages. Note that from a high-level point of view, a convolutional network is architecturally split into two important parts, each designed to fulfill a different purpose. The first part of the network (convolutional and pooling layers) performs feature extraction and the second part performs classification (fully connected layers) based on the extracted features (see Figure 2.5).

Deep learning approaches have been shown to be able to capture abstract feature representations providing both representative and discriminative information from images to facilitate different tasks that includes vision modules (e.g. image captioning). A possible explanation for this ability is given in Dicarlo's hypothesis [31], where the image distributions are extricated as they move through the layers of the network. In many tasks that involve a computer vision module, direct application of the deep learning models is not possible due the lack of sufficient amount of data. The observations, however, made in [33, 129] indicate that, as the features are transformed from an overlapped space to separable space into the network, indeterminate representations can be used as generic highly expressive features that convey semantical information that describe the objects in the image. These features are the activations of the last layer of a CNN and can be used effortlessly, as a transfer learning technique, to inform models that require highly expressive image features such as image captioning [8] or referring expression generation [84]

Those Deep Learning architectures owning their success partly to the collection of vast amount of data. In particular all these model are trained on ImageNet. This is a large image dataset organized primarily by the Standford Vision Lab. The dataset contains more than 15 million high-resolution images belonging to around 22,000 categories. The images are collected from the Internet and labeled by humans using a crowd-sourcing tool. This dataset has become an invaluable resource for computer vision and machine learning researchers. The ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [106] uses a subset of ImageNet, containing 12 million training images and 50 thousand validation images, with roughly the same number of images per image category. This annually held competition has

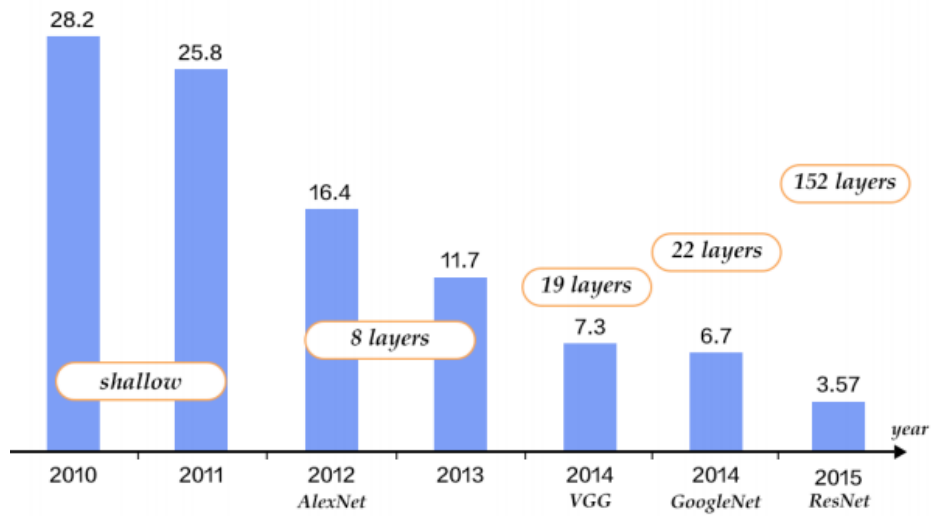


Fig. 2.7 Performance improvement on the ILSVRC ImageNet.

seen state-of-the-art image classification accuracies by deep networks such as AlexNet by [65], VGG [110], GoogleNet [117] and ResNet [48]. All winners since 2012 (Figure 2.7) have used deep CNN. Since those models have been used extensively as feature extraction methods in the following sections their key characteristics are presented.

2.3.2.1 AlexNet

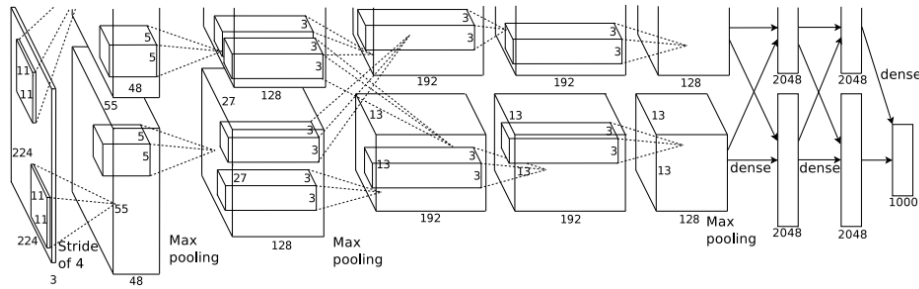


Fig. 2.8 An illustration of the architecture of AlexNet, explicitly showing the flow of tensors between two GPUs [65].

The first CNN which achieved major improvements on the ImageNet was AlexNet [65]. It was submitted to the ImageNet ILSVRC challenge of 2012 and significantly outperformed the other hand crafted models (accuracy top5 of 84% compared to the second runner-up with 74%). Figure 2.8 illustrates the architecture of AlexNet and its details are illustrated in Table 2.1.

The model is consisted of 8 trainable layers with adjustable weights. In particular, the first five layer are convolutional and the last three are fully connected. The first two convolutional layers are followed by local response normalization layers. Three maxpooling layers are placed after the two normalization layers and the fifth convolutional layer. All maxpooling layers use a 3×3 pooling window with a stride of 2 (see Section 2.3.2). As depicted in the Figure 2.8 the filters of the third convolution layer are applied to all the feature maps in the second layer. The model has around 60 million parameters and 650000 neurons. No data augmentation is used during the training that lasts for 6 days.

More details about the training procedure:

- Batch size = 128.
- Data augmentation: No.
- Weight decay : 0.0005.
- Learning rate: 0.01
- Learning rate decay schedule: decrease by factor of 10 when validation accuracy is steady.
- Momentum with rate 0.9.
- Relu is used after every convolution and linear layer.

Layer Type	Kernel size	Stride	Output size
input			224x224x3
conv	11x11	4	55x55x96
lrn			55x55x96
max pool	3x3	2	27x27x96
conv	5x5	1	27x27x256
lrn			27x27x256
max pool	3x3	2	13x13x256
conv	3x3	1	13x13x384
conv	3x3	1	13x13x384
conv	3x3	1	13x13x256
max pool	3x3	2	6x6x256
linear			1x1x4096
drop 0.5			1x1x4096
linear			1x1x4096
drop 0.5			1x1x4096
linear			1x1x4096
softmax			1x1x4096

Table 2.1 AlexNet network architecture implementation details [65]

2.3.2.2 VGG-16 and VGG-19

VGG networks [110] had the second best results in image classification task in the ILSVRC 2014 challenge with an error rate of 7.3%. In contrast to the AlexNet, this architecture utilizes filters with a very small receptive field of size 3x3. Although the size of the field is smaller it succeeds in keeping the attributes of a larger field.

The models are consisted of 16 and 19 trainable layers with adjustable weights. In particular, the models are using only 3x3 convolutional layers stacked on top of each other in increasing depth. Each of those stacked layers (either in tuples, triples or quartet) is followed by a maxpooling layer of size 2x2 and stride 2. The networks have a total number of 134 and 144 millions parameters. Unlike AlexNet, those models are using scale jittering augmentation during the training which lasts up to three weeks. More details about the network's architecture can be seen in the Table 2.2

Specific details about the training procedure as reported in [110]:

- Batch size = 256.
- Data augmentation: scale jittering.
- Weight decay : 0.0005.

- Learning rate: 0.01.
- Learning rate decay schedule: decrease by factor of 10 when validation accuracy is steady.
- Momentum with rate 0.9.
- Relu is used after every convolution and linear layer.

Layer Type	Kernel size	Stride	Output size	Kernel size	Stride	Output size
	VGG-16			VGG-19		
input			224x224x3			224x224x3
conv	3x3	1	224x224x64	3x3	1	224x224x64
conv	3x3	1	224x224x64	3x3	1	224x224x64
max pool	2x2	2	112x112x64	2x2	2	112x112x64
conv	3x3	1	112x112x128	3x3	1	112x112x128
conv	3x3	1	112x112x128	3x3	1	112x112x128
max pool	2x2	2	56x56x128	2x2	2	56x56x128
conv	3x3	1	56x56x256	3x3	1	56x56x256
conv	3x3	1	56x56x256	3x3	1	56x56x256
conv	3x3	1	56x56x256	3x3	1	56x56x256
conv				3x3	1	56x56x256
max pool	2x2	2	28x28x256	2x2	2	28x28x256
conv	3x3	1	28x28x512	3x3	1	28x28x512
conv	3x3	1	28x28x512	3x3	1	28x28x512
conv	3x3	1	28x28x512	3x3	1	28x28x512
conv				3x3	1	28x28x512
max pool	2x2	2	14x14x512	2x2	2	14x14x512
conv	3x3	1	14x14x512	3x3	1	14x14x512
conv	3x3	1	14x14x512	3x3	1	14x14x512
conv	3x3	1	14x14x512	3x3	1	14x14x512
conv				3x3	1	14x14x512
max pool	2x2	2	7x7x512			7x7x512
linear			1x1x4096			1x1x4096
drop 0.5			1x1x1024			1x1x1024
linear			1x1x4096			1x1x4096
drop 0.5			1x1x1024			1x1x1024
linear			1x1x1000			1x1x1000
softmax			1x1x1000			1x1x1000

Table 2.2 VGG networks architecture implementation details [110].

2.3.2.3 GoogleNet or Inception

GoogleNet [117] sets the new state-of-the-art for both classification and detection in ILSVRC 2014 with 6.67% top-5 error accuracy. The main contribution of this architecture to the field is the improved utilization of the computing resources within the network. In order to achieve that, they derived from the well-established CNN paradigm which requires stacked convolutional layers (optionally followed by contrast normalization and maxpooling) that are followed by at least one fully-connected layer. Authors based their work in Network-in-Network approach proposed by [80], which builds micro networks with complex structures for better data abstraction within the receptive field. Authors in [117] called those micro neural networks as inceptions.

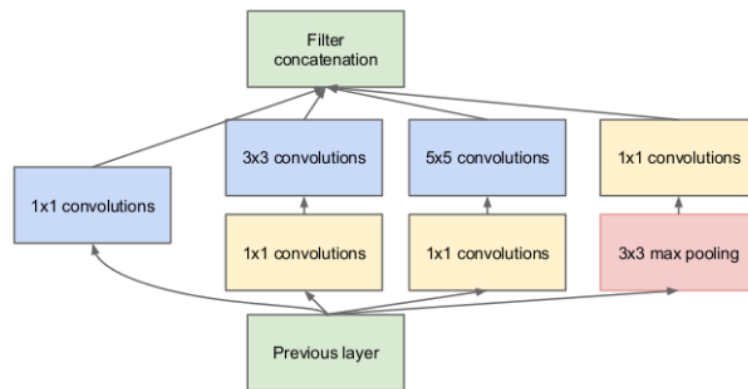


Fig. 2.9 An illustration of the Inception module [117].

The main use of inception module is dimension reduction in order to alleviate computational bottlenecks, that would strongly limit the depth of the network; this results not only in an increase of the depth, but also the width without significant performance limitations. The inception module acts as multi level feature extractor by computing 1×1 , 3×3 , and 5×5 convolutions within the same module of the network. Then the output of those filters, is concatenated to the channel dimension before moves to the next layer.

The network is 22 layers deep when counting only trainable layers. In total, the networks consists of about 100 layers. It is used average pooling with 3×3 filter size and stride of 2. However, its incredible depth affects its ability to propagate gradients through all the layers. In order to solve this problem, auxiliary classifiers networks were used in the intermediate layer of the network. Authors argued that the use of auxiliary classifiers, ensures discrimination in the lower stages in the classifier by increasing the strength of the gradient that gets propagated back. These classifiers take the form of smaller convolutional networks put on top of the output of the Inception module shown in Figure 2.11.

Type	Kernel Size	Stride	Output size	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj
convolution	7×7	2	112×112×64						
max pool	3×3	2	56×56×64						
convolution	3×3	1	56×56×192		64	192			
max pool	3×3	2	28×28×192						
inception (3a)			28×28×256	64	96	128	16	32	32
inception (3b)			28×28×480	128	128	192	32	96	64
max pool	3×3	2	14×14×480						
inception (4a)			14×14×512	192	96	208	16	48	64
inception (4b)			14×14×512	160	112	224	24	64	64
inception (4c)			14×14×512	128	128	256	24	64	64
inception (4d)			14×14×528	112	144	288	32	64	64
inception (4e)			14×14×832	256	160	320	32	128	128
max pool	3×3	2	7×7×832						
inception (5a)			7×7×832	256	160	320	32	128	128
inception (5b)			7×7×1024	384	192	384	48	128	128
avg pool	7×7	1	1×1×1024						
dropout 0.4			1×1×1024						
linear			1×1×1000						
softmax			1×1×1000						

Table 2.3 GoogLeNet incarnation of the Inception architecture [117].

Futher details about the training procedure as reported in [117]:

- Weight decay : 0.0001 - 0.0005.
- Learning rate: 0.0015.
- Learning rate decay schedule: after every eight epochs is multiplied by 0.96
- Momentum with rate 0.9.
- Weights initialization: sampled from normal distribution and centered around 0 with small variance.

2.3.2.4 ResNet

All the architectures after AlexNet exploit “very deep” models. The depth of a neural network has been shown to be of crucial importance [110, 112]. However, as the depth and the width of the models growing, its accuracy tends to saturate and degrade rapidly. The authors [48] observed, no matter how deep a network is, the training error should be no higher than that of the shallower. They argued that, a neural network could approximate any complex function and therefore could possibly learn an identity function that maps input to output by effectively skipping some layers.

The Resnet uses “*skip or shortcut connections*” in order to improve the convergence rate and classification accuracy of very deep CNNs. The reason behind the shortcuts is that by letting the signal flow more easily through the network the problem of exploding or vanishing gradients can be reduced, which in turn makes the deep network easier to train. Responsible to perform those shortcuts are micro neural networks called residual units (an illustration of this can be seen in Figure 2.1). Residual units are basically stacked convolutional and normalizing layers. The input of each unit is additively merged with the last stacked layer in order to simulate a shortcut or a *residual mapping*.

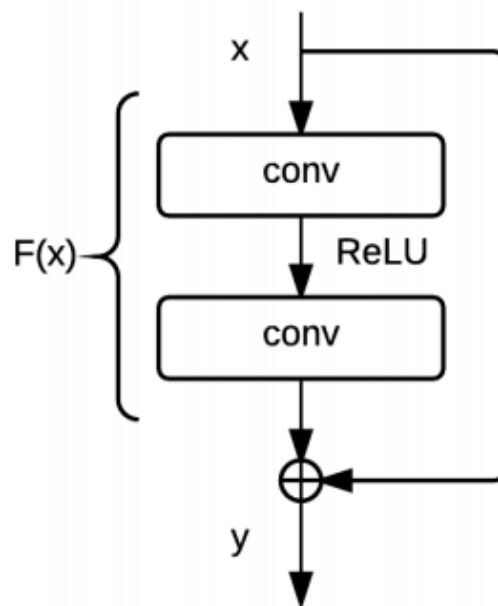


Fig. 2.10 A simplified residual unit which consists of two stacked convolutional layers activated using ReLUs. The input of the first convolutional layer is additively merged with the output of the last convolutional layer, which is what, in this document referred as a shortcut.

The 152 layer residual net is the deepest model ever to be trained on and applied to ImageNet. It performs a 4.49% top-5 error, which outperforms all previous models. An ensemble of six residual models, including two 152 layer deep ones, has managed to perform a top-5 error of 3.57%. This ensemble won first place in the ILSVRC 15 classification competition.

Further details about the training procedure as reported in [48]:

- Batch size = 256.
- Weight decay : 0.0005.
- Learning rate: 0.1
- Learning rate decay schedule: divided by 10 when validation accuracy plateaus.
- Weights initialization: Xavier [55]

2.3.2.5 Recurrent Neural Networks

Feedback connections in a neural network can produce past context information. This network architecture is known as recurrent neural network (RNN). Many versions of recurrent neural networks have been developed or adapted in order to achieve excellent results in different machine learning domains. For a sequence of input vectors x_1, \dots, x_T a basic RNN computes the sequence of hidden activations h_1, \dots, h_T and the output vectors as y_1, \dots, y_T as:

$$h_t = f(W^{hx}_t + W^{hh}h_{t-1} + b^h) \quad (2.21)$$

$$y_t = G(W^{hy}h_t + b^y) \quad (2.22)$$

For all times steps $t = 1, \dots, T$ where W are the weights matrices of two connected layers, and B denotes the bias terms as F and G are the activation function used. For a deep RNN with several stacked hidden layers³, each hidden layer receives the output of the previous hidden layer. Despite this minor modification, the effects are profound: In a RNN information obtained from previous time steps can loosely speaking circulate indefinitely inside the network through the directed cycles, where hidden layer can play the role of memory. These hidden activations are making an internal state where can be represented as h_t vector given a certain time step for each hidden layer. The output of the network at time t is a function of all received inputs vectors till that time. This RNN as the we one we described is also known as a vanilla RNN.

From the scope of memory, RNNs are more computationally expensive than FNNs. As we stated above, a feed forward neural network can approximate any non-linear function on a compact domain with arbitrary precision. The equivalent to MLPs approximation theorem in RNNs is that with sufficient number of hidden units any dynamical system can be approximated. Potentially a RNN is computationally as expressive as any Turing machine

In order the RNN to be trained a straightforward extension of the backpropagation algorithm is applied known as backpropagation through time (BPTT). The basic idea behind this training strategy is to unfold the network through time and then to use the standard backpropagation algorithm as it was a classic MLP. This unfolded presentation denotes that a RNN can be else seen as a very deep FNN but with a layer for time step and share weights across time.

The above vanilla structure is not commonly used because of a problem known as vanishing gradient problem and a subsequent problem called exploding gradient problem. Several techniques have been presented to overcome the difficulties of training RNNs, such as training with second order optimization methods

2.3.2.6 Long Short-Term Memory

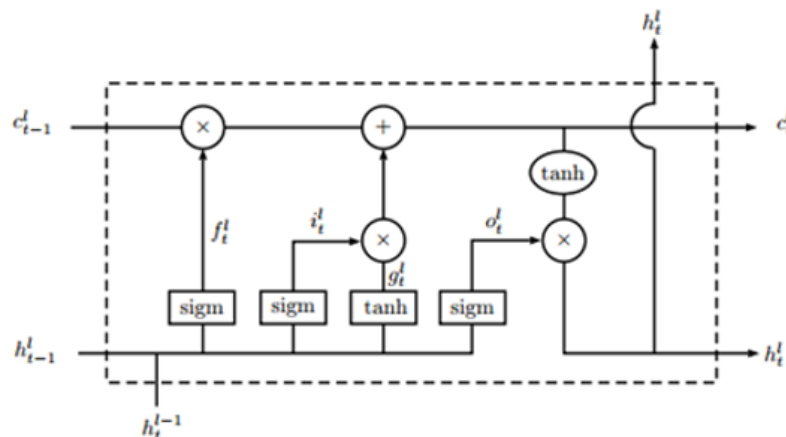


Fig. 2.11 Basic LSTM memory block.

The RNN owes its memory to the fact that it feeds its stated forward in time, in practice the network easily forget after a couple of steps. It has been shown that to train a standard RNN to find patterns that occurs after a large number of steps is a trivial task. In order this problem to be addressed an extension of standard RNN was proposed, the so-called Long Short-Term Memory (LSTM).

An LSTM network architecture is built upon LSTM memory blocks just as a RNN is based on McCulloch-Pitts neurons. LSTM as almost all feedforward neural networks is

comprised of multiple layers with many cells in each layer. In Figure 2.11 it is depicted a basic LSTM memory block. This is a single cell k in layer l at time step t . As we can see the output is a single entry in the input vector to the next layer and next time step. There have been several proposed architectures, we choose to describe here the model as presented in its first form in 1997.

The rough idea behind the LSTM cell is the use of the so called gates to control which information the cell will remember, forget and produce. These gates allow cell information longer than the neuron of an RNN. In time step t the output of l is defined as h_t^l . Similarly, for the LSTM memory cells c_t^l defines the memory states for layer l at time step t . Formally the calculation of new memory state is as follows:

$$c_t^l = f_t^l c_{(t-1)}^l + i_t^l g_t^l \quad (2.23)$$

Where \cdot denotes element-wise multiplication. Furthermore, the previous memory state is element-wise multiplied by f_t , the forget mechanism. The forget mechanism is responsible to choose which properties of $c_{(t-1)}^l$ and is given as follows:

$$f_t^l = \left[T_{(m+n,n)} \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix} + b_f \right] \quad (2.24)$$

The size of layer $l-1$ and l is m and n respectively, thus the transformation T is done on a vector of size $m+n$ into a vector of size n . The use of element wise logistic function secures that the values are between 0 and 1 where zero denotes that the cell must completely forget and 1 to remember what in time t is stored in the memory state. The additional bias term is used so as the model to store information easily during the first steps of training.

When the information from input and the previous output $i_t^l g_t^l$ is added to the memory gate then the gate decides which features g_t^l of each input should be added. The values of i_t^l, g_t^l are given from the following Equations:

$$i_t^l = \left[T_{m+n,n} \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix} \right] \quad (2.25)$$

$$g_t^l = \tanh \left[T_{m+n,n} \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix} \right] \quad (2.26)$$

Note that g_t^l use the tanh function instead of the logistic function. This allows the input values to the LSTM to take on values between -1 and 1. After the memory state is updated, it needs to be decided what the cell should output. This is done with the last gate o_t , which

regulates what properties of our memory state we will use in the output and is described in the following equation:

$$o_t^l = \sigma \left[T_{m+n,n} \left(\begin{matrix} h_t^{l-1} \\ h_{t-1}^l \end{matrix} \right) \right] \quad (2.27)$$

The output for the current layer h_t^l is sent to the next layer and time step. The tanh function is first applied to the memory state to normalize it to the interval $[-1, 1]$ and the result is then scaled with the output gate.

$$h_t^l = o_t^l * \tanh(c_t^l) \quad (2.28)$$

In equation 2.29, $lstm(\cdot)$ is denoted as a function calculating the next output of an LSTM-cell according to equations 2.28, 2.23 (The internal memory state is implicitly updated).

$$h_t^l = lstm_t^l(h_1^{(l-1)}, h_{(t-1)}^l) \quad (2.29)$$

Chapter 3

Image Captioning: A survey of Models and Datasets.

Recently automatic image description generation has received a large amount of scholarly attention from both the computer and natural language processing communities. This chapter provides an overview of the work has been conducted in the field. In particular, the classification proposed in [7] is followed. Specifically the approaches are classified based on how the problem is conceptualized. The first group of models follows the classical pipeline that is, first the image content is represented in terms of objects, attributes, scene types based on a set of visual features. Then, this content information is used by a natural language generation system that produces the final image caption. Those approaches are coined as direct generation models.

The second group of approaches, treat the problem of image captioning as a retrieval. The basic idea underlying the works of those models is that, firstly a search to find similar images with the one that is to be described is conducted. Then, those model can either reuse the caption of the most similar image, or they can combine the descriptions of similar images to produce a novel caption. Those models can further be divided based on which criteria the search is conducted. Specifically, the first subgroup conducts search in the visual space and the second in a multimodal space that represents both images and text.

In the following subsections, a comprehensive overview of the state-of-the-art approaches in image captioning generation is given using the three categories of models as described before : *direct generation models, retrieval models from visual space, and retrieval model from multimodal space..* However, strong focus will be given to retrieval models from multimodal space. Additionally, existing datasets available for training such captioning models will be presented.

3.1 Description as Generation from Visual Input

The models that belong in this category, first analyze the visual content and then generate a sentence that conveys that meaning. This is achieved by following the next pipeline:

- A cascade of computer vision techniques are used in order to extract information such as: scene type, recognize object within the image, predict their attributes and their relationships and/or predict actions that taking place.
- As a second step a generation phase is taking place that converts the raw computer vision detection into words or phrases. Then, by using well established techniques from natural language generation such as templates or grammar rules the final captions are produced.

The fundamental difference of the approaches in this section compared to those that are presented in the next two sections is that they perform an explicit mapping between images and captions. The latter approaches use implicit vision and language models. However, one disadvantage that stems from explicit architectures is that they are tailored to the problem at hand, thus the generated descriptions are constrained to predefined sets of visual objects (e.g. scenes.). More importantly, those architectures heavily rely on the accuracy of object detector and consider the detections absolute, something that rarely holds in practise [7, 5].

In this category of models many different methods representing the images have been proposed that follows closely the advantages of the computer vision at the time. In particular, images has been represented by using spatial relationships [40], corpus-based relationships [60], or spatial and visual attributes [66]. Another trend, combines different computer vision system output in terms of tuples such as the objects detected, along with the attributes of those objects, the spatial relations between them, and the scene type [Mitchell et al., 67, 40, 78]. The most notable exception is the work of [39], which does not rely on object attributes annotations. In order to extract such information, they use multiple instance learning to train visual detectors for words that have been observed to occur in training captions. The word detector output serves as conditional input to a maximum-entropy language model.

The first attempt to correlate the structure of an image to its corresponding description is the Visual Dependency Representation method proposed in [37]. In particular this methods captures the spatial relationship of the object depicted with the use of a dependency graph. Then, this graph can be correlated with the syntactic dependency tree of the corresponding captions. More recent work, extended this method by automatically produce the dependency representation based on the output of an object detector [35] or scene detector.

How the generation process is handled, is another dimension that existing work varies. In particular, there are language generation models that generating descriptions with the use of n-gram language models that are trained on subsets of Wikipedia such as [66, 78]. Mainly this approach first determines the attributes and the relationships of image patches as region–preposition–region triples. The language model that is based on the n-gram method is used to generate the image caption. Again notable exception is the work presented in [39], that makes use of a maximum entropy language model that effectively makes use of the output of the words detectors that is the main novelty of this work.

Another research thread, generates captions use the use of templates. Mostly, those templates are pre-defined sentence frames with gaps that are required to be filled with the output of computer vision systems. [60] fill in a sentence template by selecting the likely objects, verbs, prepositions, and scene types based on a Hidden Markov Model. Verbs are generated by finding the most likely pairing of object labels in the Gigaword external corpus. The generation model of [37] parses an image into a VDR, and then traverses the VDRs to fill the slots of sentence templates. This approach also performs a limited form of content selection by learning associations between VDRs and syntactic dependency trees at training time; these associations then allow to select the most appropriate verb for a description at test time.

Lastly other approaches make use of more linguistically informed language model structures. The most notable work of this thread is [Mitchell et al.]. The basic idea behind this work is the over-generation of syntactically well-formed sentence fragments which are combined with the use of a tree based grammar. Similarly [69] used tree-fragments which are learned directly from the training set. At test time, the model combines those tree-fragments to produce novel descriptions.

The aforementioned systems aim solely on generating novel captions given an input image. However, authors in [51] argued that evaluating the quality of novel generated captions poses a number linguistics difficulties that stems from the task at hand. In other words, there are many different ways to describe an image. Furthermore, evaluating natural language generation system is know to be a non-trivial task [101]. Authors in [51] proposed as a solution to evaluate the translation of the two modalities independently from the generation process. Specifically, they re-framed the task of image captioning as retrieval problem by correlating images with candidate description by retrieving a set of images with their corresponding captions. Those candidate captions can be either reused or a description can be generated by using different parts derived from the candidate captions. As mentioned earlier, the search can be conducted in two different spaces: (1) visual space, in which only the similarity between the images is considered; or (2) in a multimodal space where the

image and textual features are projected into a common space. In the following section works included in those categories are presented but with a strong focus on those model that search in the multimodal space.

3.2 Description as a Retrieval in Visual Space

The large body of the work conducted in this group treats the problem of automatically generate a caption for an image by retrieving images with high resemblance with the image that is required to be described. Broadly speaking, those model consider similarity in visual space in order to associate captions with the image to be described. The generated caption can either be a sentence that has already existed or a sentence composed from the retrieved ones. The constituent parts of the models belonging in this category follow the following pipeline [7]:

- Encode image to a suitable form
- Retrieving one or a set of candidate images based on an appearance similarity measure.
- Generate captions by either reuse a sentence from the pool of the candidate captions or compose a sentence from the retrieved ones.

The first to pioneer in this type of models was [93] with the In2Text model. First they employ global image descriptors to retrieve a set of images from a web-scale collection of captioned photographs. Then, they utilize semantic contents of the retrieved images to perform re-ranking and use the caption of the top image as the description of the query.

[46] used the Stanford CoreNLP toolkit to process the captions in the dataset in order to produce a list of phrases for each image. Then, their system in order to generate a caption, first it performs image retrieval based on global image features. Then, a model trained to predicate phrase relevance is used to select phrases from the ones associated with retrieved images. Finally a description sentence is generated based on the selected relevant. phrases.

In a similar vein [70] proposed a tree-based method to produce image captions by composing image descriptions from a pool of web images. Authors, as first step performed image retrieval and phrase extraction. Then, the extracted phrases were used as tree fragments and the problem of generating a caption was framed as a constraint optimization problem, and implemented as an Integer Linear Programming [20, 103] and solved by using the CPLEX solver2. This method is similar to the one described before [68]

[96] introduced a large-scale scene attribute dataset that consists of 14,340 images belonging in 707 scene categories. Those images were further annotated from a pool of

102 discriminative attributes such as materials, surface properties, lighting, affordances, and spatial layout. Consequently, were able to train attribute specific classifiers the output of which can be used as a global image descriptor. They argued that this image encoding captures the semantic content in a more refined way when compared to the standard global image vector.

The work described in [86] differs from the work presented that far since formulates the problem of captions generation as an extractive summarization. The output descriptions is selected based on only textual information. Specifically, the images are represented by using the scene attributes descriptor used in [12]. As a first step the similar images with the query image are identified and in the next step the conditional probabilities of including a word in the final caption are estimated with the use of nonparametric density estimation. Then the final caption is handled with the use of two different extractive summarization techniques, one depending on the SumBasic model [92] and the other based on Kullback-Leibler divergence [99] between the word distributions of the query and the candidate descriptions.

In [124] authors proposed a novel query expansion approach for improving transfer-based automatic image captioning which is based on the compositional distributed semantics. To encode the images, unlike any other work described so far they use activations of the last layer of the VGG-16 model [110]. Then, the original query is expanded as the average of the distributed representations of retrieved descriptions, weighted by their similarity to the input image.

In a similar vein, the work of [30] also encodes the images with CNN activations. Then those image features are fed to a k-nearest neighbor model that determines which images from the training set are visually similar to the input image. A description then is selected from the pool of candidate descriptions associated with the retrieved images that best describes the images that are similar to the query image, in a similar vein as [124]. However, their approach differs on how the similarity is defined. Specifically, in [30] it is proposed the descriptions similarity to be computed based on the n-gram overlap F-score between the descriptions [7]. They suggest that the output of the system should be the caption with the highest mean n-gram overlap with the other captions in the pool (k-nearest centroid captions).

3.2.1 Description as a Retrieval in Multimodal Space

The works presented in Sections 3.1, 3.2 are representative examples of early work. Thus, in this thesis such methods are not considered and was only given a brief overview of those methods. However, due to the unrepresented progress in field of deep learning [18], [90], works from the early 2013 begin to rely exclusively on the use of deep learning methods to address the image captioning problem. In this section, such methods will be reviewed. Despite the fact that deep neural networks are now widely adopted for tackling the image captioning task, different methods may be based on different frameworks. In this subsection, a further classification is been used based on the main framework they use and each of their subsequent components is discussed.

3.2.2 Retrieval and template based methods augmented by neural networks

Driven by the advances in the field of deep learning neural networks, instead of relying on the hand crafted image features and shallower architectures that were presented in the previous section, deep learning neural networks are used in the task of image caption. The models in this era are driven the premises of retrieval based models, but they utilize a multi-modal retrieval space.

In [111] authors in order to retrieve a caption for a query image, they utilized dependency-tree recursive neural network to encode the phrases and sentences into compositional vectors. In order to encode images they made use of another deep learning neural network introduced in [71]. Then the multimodal features are projected into a common space with the use of a max-margin objective function. It is assumed that the candidate image sentences pairs will have larger inner products while the dissimilar image-sentence pairs will score lower. Finally, the caption retrieval is performed based on a similarity metric withing the common space that images and sentences are projected.

In [59] it was proposed that sentence fragments and image fragments should be embedded into a common space for ranking the similarity of the sentences given a query image. In order to encode the sentence fragments they made use of the dependency tree relations proposed in [25], and from representing the image they used the Region Convolutional Neural Network network [42]. In order, to represent both both image fragments and sentence fragments as feature vectors, the authors used a structured max-margin objective, which includes a global ranking term and a fragment alignment term, to map visual and textual data into a common space. In the common space, similarities between images and sentences are computed based on fragment similarities, as a result sentence ranking can be conducted at a finer level.

In [83] authors used a multimodal Convolutional neural to define the similarities between images and captions by modeling the different level of interaction between them. In particular, their three staged framework consists of a cascade of CNNs that encodes the images [110, 116], in the next step the output of the CNNs is matched in order to jointly represent visual and textual data [57]. As a final stage, the authors used multi-layered perceptrons that ranks whether the visual and textual data are compatible. To capture the variability of the image and textual data authors used a number of different CNNs. Then, the final matching score was based on an ensemble of multimodal Convolutional Neural Networks.

In [125] authors proposed the use of deep canonical correlation analysis [3] in order to align sentences and images. The visual features were again extracted with the use of a CNN [65]. In order to encode sentences they used a stacked network that is fed with Frequency-Inverse Document Frequency representations of sentences. The Canonical Correlation Analysis objective is responsible from projecting the visual and textual features in a common space, that maximizes the correlation between the paired features. In the joint latent space, similarities between an image feature and a sentence feature can be computed directly for sentence retrieval.

The use of deep learning methods was also attempted to the models that perform caption generation from the visual space. A notable work is of [72]. Their proposed method utilizes a soft-template to generate captions. Specifically, the SENNa software is used to extract phrases from the training captions. Then, those phrases were encoded by high-dimensional vectors following the methods proposed in [88, 89]. As a metric between images and textual phrases, the authors trained a bi-linear model that given an image the model can infer the phrases. Then, those phrases along with the utilization of corpus statistics lead the model to produce a caption.

With the utilization of deep neural networks, performances of image captioning methods are improved significantly. However, introducing deep neural networks into retrieval based and template based methods does not overcome their disadvantages. Limitations of sentences generated by these methods are not removed. In particular Retrieval based and template based image captioning methods impose limitations on generated sentences. Thanks to powerful deep neural networks, image captioning approaches are proposed that do not rely on exiting captions or assumptions about sentence structures in the caption generation process.

Such methods can yield more expressive and flexible sentences with richer structures. Using multimodal neural networks is one of the attempts that rely on pure learning to generate image captions.

3.2.3 Image captioning based on the encoder-decoder framework

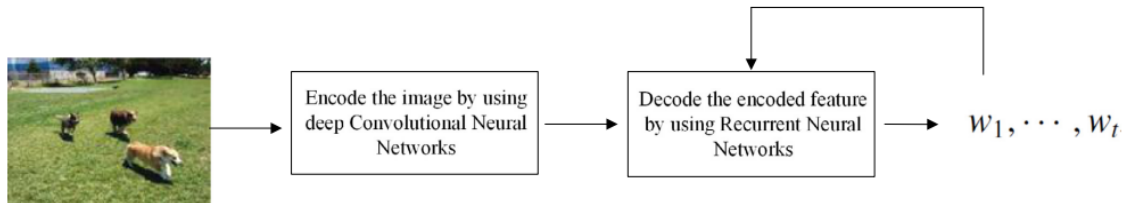


Fig. 3.1 General structure of compositional image captioning methods [4].

Building upon the recent advances in neural machine translation [17, 56, 114], the encoder-decoder model was adopted to generate image captions. The general structure of how the encoder-decoder model is adjusted from the purposes of image captioning is depicted in the Figure 3.1. In its original form, the model is designed to translate on sentence from one language into the target language. The adaption of this framework to image captioning, assumes that the problem is in fact a translation problem. However, the source modality is an image and the target modality is a natural language. The majority of the works in image captioning that uses this framework, first encode an image into an intermediate representation, and then the decoder model is responsible to generate a caption token by token. This framework is adopted in this thesis as well.

The first to use the encoder-decoder framework in image captioning was [63]. In particular, they unified the joint image-text embedding models and multimodal neural language models, so that given an image input, a sentence output can be generated word by word like language translation. In order to encode textual data they used an LSTM and for the extraction of image representation they used a deep Convolutional Neural. Then, through optimizing a pairwise ranking loss, encoded image features are projected into the embedding space of the LSTM hidden states. In the embedding space, the decoder which is a structure-content neural language model is able to generate novel descriptions in an end-to-end fashion word by word. Their approach is depicted in the Figure 3.3

Concurrently to the work of [63] another work that is based on encoder-decoder framework was presented by [32]. The basic difference is that the authors rather than projecting the vision space into the spanned by LSTM hidden states, the model receives as input a static image without been preprocessed by any CNN, which is then fed to a four layer LSTM. An extension of this approach was presented in [54] where the authors augmented the input to the LSTM by adding semantic image information.

[83, 85] used a Recurrent Neural Network language model to multimodal cases for directly modelling the probability of generating a word conditioned on a given image and

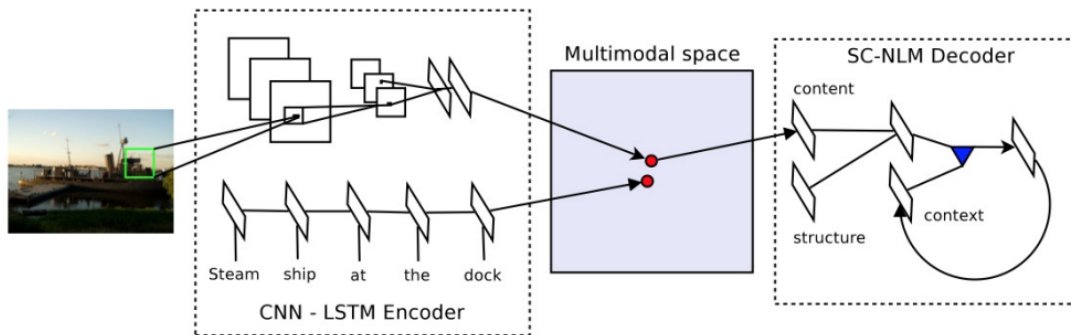


Fig. 3.2 The encoder-decoder model proposed by [63].

previously generated words. Under their framework, a deep Convolutional Neural Network [65] is used to extract visual features from images, and a Recurrent Neural Network [108] with a multimodal part is used to model word distributions conditioned on image features and context words. There are a few similar work with the aforementioned works such as [119] and will be not further explored.

[58] differentiates from previous approaches by introducing a deep visual semantic alignment model with a simpler architecture and objective function. Their basic assumption is that part of the sentences referring to particular but not known in advance regions of the images. Thus, their model infers the matching of sentence segments and image regions. The fundamental components of their approach is a convolutional neural network that encodes images, and a bidirectional RNN that encodes sentences and is trained on a structured objective that matches the two modalities. The two modalities are projected into a common multimodal embedding space. Then, the multimodal RNN uses the previous inferred matches to generate the novel descriptions. They condition the language model to image features only its first state.

In sharp contrast to the work presented in this section, the model of [16] dynamically generates a scene representation while the image caption is being generated. Specifically, while a word is generated the visual representation is adjusted accordingly. In a similar vein, [123] uses an RNN-based architecture in which the visual representations are dynamically updated. However, they further augment their approach by incorporating an attention component in their framework. Theirs model attention is visual that is, the attention is used to determine whether a region in an image is salient or not. If the region is found to be salient then, the captions are conditioned on those regions.

3.2.4 Adversarial Generation of image captions

Generative adversarial networks (GANs) have drawn great attentions since [45] introduced the framework for generating the synthetic data that is similar to the real one. The main idea underlying this frameworks is that there are two neural networks models namely, a discriminator and a generator, posed in an adversarial game during training. The purpose of the discriminator is to distinguish the synthetic data from the real, while the generator is training to confuse the discriminator by generating synthetic data with high resemblance to the real one. During learning, the gradient of the training loss from the discriminator is then used as the guidance for updating the parameters of the generator. Since then, GANs achieve great performance in computer vision tasks such as image synthesis [29, 53, 77, 98]. Their successes are mainly attributed to training the discriminator to estimate the statistical properties of the continuous real-valued data (e.g., pixel values).

The adversarial learning framework has been proven to be a strong candidate in generating high quality image captions. However, there are a few issues that restrict the use of GANs in image captioning and in natural language generation in general. By design, GANs facing difficulties in dealing with discrete data (e.g., text sequences [11]). Text sequences are discrete tokens whose values are nondifferentiable which poses severe limitation of the training of GANS. There is a growing thread in image captioning that utilizes GANs and are going to be described in this section. The overall architecture of this type of models is depicted in Figure

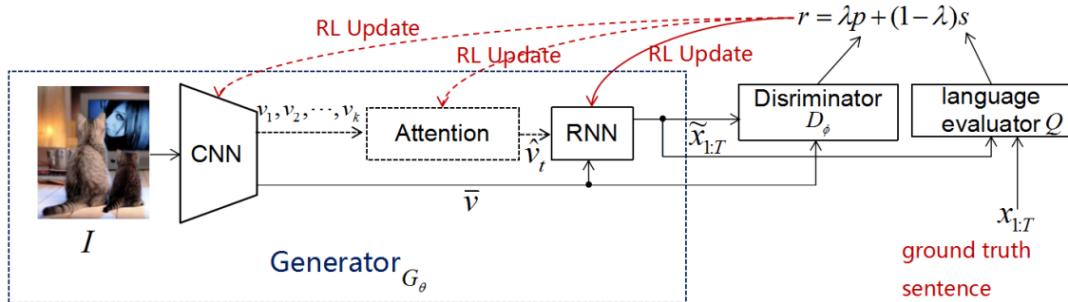


Fig. 3.3 The GAN framework for image captioning proposed by [15].

One of the first to adapt the GAN framework in image captioning is [23]. Specifically, they propose a new framework based on Conditional Generative Adversarial Networks (CGAN) [90], which jointly learns a generator to produce descriptions conditioned on images and an evaluator to assess how well description fits the visual content. In order to train their model, they used Policy Gradient [115], a strategy stemming from Reinforcement Learning, which allows the generator to receive early feedback along the way. A second approach was

Name	Images	Captions per image
Pascal1K [100]	1,000	5
VLT2K [36]	2,424	3
Flickr8K [52]	8,108	5
Flickr30K [97]	31,783	5
MS COCO [81]	164,062	5

Table 3.1 Image captioning datasets.

introduced in [109]. Specifically, they employ adversarial training in combination with an approximate Gumbel sampler to backpropagate the errors in the framework and make its training feasible. However, what these papers concern about most is naturalness and diversity of descriptions while sacrificing the fidelity, which results in much lower language metrics scores compared with other image captioning algorithms.

In order to alleviate the restrictions of the previous work [15] proposed a novel conditional-generative adversarial-nets-based image captioning framework as an extension of traditional reinforcement-learning (RL)-based encoder-decoder architecture. To deal with the inconsistent evaluation problem among different objective language metrics, they adjusted the “discriminator” networks to automatically and progressively determine whether generated caption is human described or machine generated. Two kinds of discriminator architectures (CNN and RNN based structures) are introduced since each has its own advantages. They argued that the proposed algorithm should be generic so it can enhance any existing RL-based image captioning framework.

3.3 Datasets

There is a wide range of dataset suitable from automatic image captioning. Mostly, such datasets contain images associated with caption. Most of the times, those dataset are significantly different from each other in terms of size, the format of the captions or the way those captions were harvested. In this section an extensive overview of the available datasets is given. The table 3.1 summarizes the datasets that are going to be described in this section.

Pascal1K The Pascal1K sentence dataset [100] has been commonly used a benchmark dataset when it comes to evaluate the the quality of the produced captions. It contains 1,000 images extracted from the Pascal 2008 object recognition dataset [Everingham et al.] and includes objects from different visual classes, such as humans, animals, and vehicles. For

each image five captions are associated and where collected on the Amazon Mechanical Turk platform.

Visual and Linguistic Treebank The VLT2K [36] contains images derived from the Pascal 2010 action recognition dataset. Those images are accompanied with three, two-sentence captions per image. Similarly to [100] the captions were collected on the Amazon Mechanical Turk platform. However unlike [100] the turkers were given specific instruction to describe the main action that is depicted in the image and which actors are involved in the action (in the first sentence) and which objects are depicted in the image (second sentence).

Flickr8K, Flickr30K The Flickr8K dataset [52] and its extended version Flickr30K dataset [97] contain images from Flickr, comprising approximately 8,000 and 30,000 images, respectively. The images in these two datasets were selected through user queries for specific objects and actions. These datasets contain five descriptions per image which were collected from AMT workers using a strategy similar to that of the Pascal1K dataset. In this thesis the Flickr8K is used mainly due to the realistic images and that contains and it small size. The image captioning model require a lot of computational power that they author of this thesis has not access.

MS-COCO The MS- COCO dataset [81] consists of 123,287 images with five different descriptions per image. Images in this dataset are annotated for 80 object categories, which means that bounding boxes around all instances in one of these categories are available for all images. The MS COCO dataset has been widely used for image description, something that is facilitated by the standard evaluation server.

Chapter 4

Methodology

In this chapter, the details of all the components of the implemented image captioning model that used in this thesis are given. In particular, the model is an adaption of the first image captioning model that proposed in literature that has the ability to be trained end-to-end [119]. After the presentation of all the constituent parts of the model is given, a discussion on the automatic evaluation metrics that has been used in the literature and used in this thesis is given.

4.1 Model Architecture

As was shown in the Chapter 2, recent advances in statistical machine translation proved that granted a powerful sequence model it is possibly to directly maximize the probability of a correct translation, in an end-to-end fashion, given an input sentence. As it was shown, those models make use of a RNN network which is responsible to encode the variable length input sentence into a fixed dimension vector and the same time to decode it to the desired target sentence. In other words, the adaption of this encoder-decoder model in image captioning that was proposed in [119] and is followed in this thesis is that, instead of using an input sentence the model translates the image features into descriptions.

Therefore, in this thesis it was directly maximized the probability of the correct description given an image by using the following formulation [119]:

$$\theta^* = \arg \max_{\theta} \sum_{(I,S)} \log p(S|I; \theta) \quad (4.1)$$

where θ are the parameters of the image captioning generator, I is the input image, and S its a human produced caption. However, due to the fact that captions can convey any visual

features of the image their length is unbound. Thus, the chain rule is applied to model the joint probability over S_0, \dots, S_N as follows:

$$\log p(S|I) = \sum_{t=0}^N \log p(S_t|I, S_0, \dots, S_{t-1}) \quad (4.2)$$

For notation simplicity, the model's dependency towards its parameters θ it was not included. Granted a training sample pair (S, I) the model is optimized based on the sum of the log probabilities as presented in Equation 4.2 over the whole training set using stochastic gradient descent as presented in the Chapter 2.

Apparently, the $p(S_t|I, S_0, \dots, S_{t-1})$ was modeled with an RNN, where the variable number of words were conditioned upon up to timestep $t - 1$ and were represented as a fixed length hidden state h_t . This state is updated every time it sees a new input x_t with the use of a non-linear function f :

$$h_{t+1} = f(h_t, x_t) . \quad (4.3)$$

Concretely, for the implemented image captioning model two crucial designs choices were made: how will f be best modeled and how the image features along with words are going to fed as inputs to the system. For the former it was followed the same modeling idea as in many image captioning models proposed in literature and f was implemented with the use a Long-Short Term Memory (LSTM) net, which has shown state-of-the-art performance on sequence tasks such as translation. Next section elaborates in the details of the language model. For the latter, in order to extract image representations, CNNs were used (See Chapter). The words are represented with an embedding model. The overall architecture of the model is depicted in Figure 4.1



Fig. 4.1 A high-level block diagram of the visual captioning pipeline.

4.1.1 Visual Feature Extraction

In encoding image into vectors, the activation values extracted from the Convolution Neural Networks layers are only used in this thesis as image representations. As argued in Chapter 2, networks pre-trained on the Image-Net data-set were found to transform the image into meaningful representations [27]. In principle, in order one CNN to be used as a feature extractor, one should remove its last layer which computes the 1,000 probabilities of different ImageNet classes, but keep all the other layers and parameters intact. By doing so, this CNN is transformed into a feature extraction function $CNN_{(\theta_c)}(I)$, which receives as an input image pixels I and has parameters θ_c . For example, VGG-16 [110] has approximately 123 million parameters θ_c and the $CNN_{(\theta_c)}(I)$ is a 4096-dimensional vector. This representation can be extracted from a layer that is located after a non-linearity (e.g. RELU in VGG) and just before the fully connected layers that play the role of the ImageNet classifier.

In practise it is common to use the pretrained CNN as a standalone feature extractor and extract the features for all the images in the dataset. Thus, the image encoding V takes the form:

$$V = W[CNN_{\theta_c}(I)] + b \quad (4.4)$$

In other words, in Equation 4.4 the image is encoded by taking its feature vector and passing it through a linear transformation. The parameters W, b will be trained and the vector V will continue to further processing in the network. However, another method for extracting visual vectors exists but it was not used in this thesis. Specifically, one can also backpropagate through the CNN and adjust the parameters θ_c instead. This process is coined as fine-tuning. Despite its intuitive idea most of the times is extremely computationally expensive since the convolutions taking place in the convolutions layers of CNN are more computationally expensive compared to simply referencing a precomputed 4096 dimensional vector for any image.

Before the images are fed to the CNNs a pre-processing step should take place. The pre-processing pipeline is depicted in Figure 4.2. In detail the steps are the following:

- As a first step images are resized along the short side to a size of 256 pixels followed by a centre crop with a dimension of 224×224 pixels for the VGG-16 and with a dimension of 299×299 for the Inception V3.
- As a second step, the images are transposed in the following form Channels X Height X Width.
- As a third step, for each image the dataset mean is subtracted.

- As a final step, the channels are switched from RGB to BGR. The reason for this step is mainly technical.

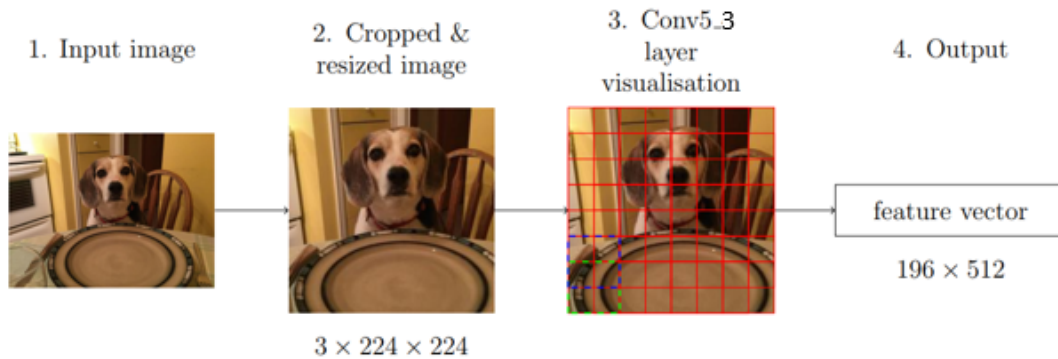


Fig. 4.2 The steps required for processing images before are fed to the VGG-16.

After the preprocessing step is completed, the preprocessed image is fed into the network and the final representation for each image is a vector of 4096 or 2048 dimension for VGG-16 and Inception V3 respectively.

4.1.2 Language Model

The next stage of the pipeline is a conditional language model that receives the images features and produces a caption. The most popular architecture in literature to model the probability of a caption S , given an image feature V as $P(S|V)$ is Long-Short Term Memory (see Chapter 2) network. The following subsection elaborate on the details of the LSTM cell as used in image captioning.

4.1.2.1 LSTM Cell

Recall from Equation 4.3 that the form of f plays the most significant role in the image captioning pipeline for two reasons: (1) first the model that will be chosen should be able to address the problem of vanishing and exploding gradients that is the most common challenge while training the RNNs [50]; and (2) it should be able to handle sentences of arbitrary length a characteristic that stems from RNNs design.

As mentioned in Chapter 2, the fundamental block of the LSTM model is a memory cell c the role of which is to store which input has been seen up to this step (see Figure 4.3). Recall now, the the behavior of the cell is governed by three gates or layers, that are used multiplicatively. Concretely, the value of the cell at any time step t is depended on the current

$$i(t) = \sigma(W_{ix}x(t-1) + W_{iy}y(t-1)) \quad (4.5)$$

$$o(t) = \sigma(W_{ox}x(t-1) + W_{oy}y(t-1)) \quad (4.6)$$

$$f(t) = \sigma(W_{fx}x(t-1) + W_{fy}y(t-1)) \quad (4.7)$$

$$m(t) = f(t) \odot m(t-1) + i(t) \odot \tanh(W_{mx}x(t) + W_{my}y(t-1)) \quad (4.8)$$

$$y(t) = o(t) \odot m(t) \quad (4.9)$$

$$p_{t+1} = \text{Softmax}(y(t)) \quad (4.10)$$

where \odot represents the product with a gate value, the various W_{\cdot} matrices are trained parameters. The nonlinearities are sigmoid $\sigma(\cdot)$ and hyperbolic tangent $h(\cdot)$. The Equation 3.9 is what will be feed to a Softmax function, which will calculate the probability distribution p_t over all words.

4.1.2.2 Proposed LSTM Language Model

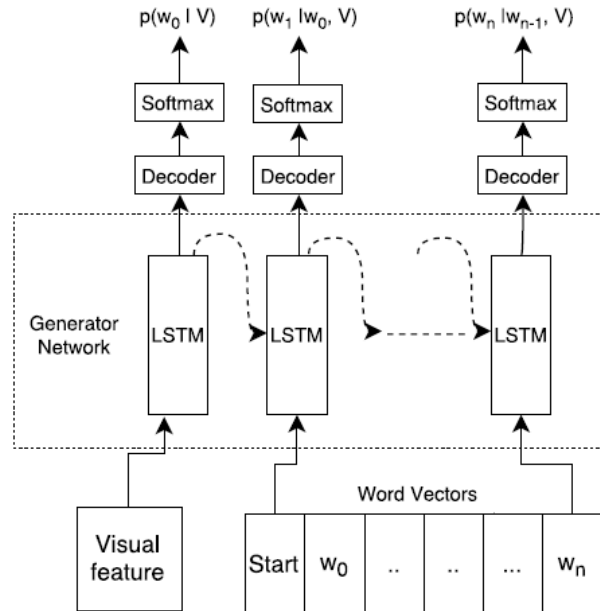


Fig. 4.4 The LSTM-based language model unrolled in time

The architecture of the language model that was implemented for the purposes of this thesis is depicted in Figure 4.4. The LSTM model is trained to predict each word of the

sentence after it has been fed with the image features and the words that were previously predicted as $p(S_t|I, S_0, \dots, S_{t-1})$. Concretely, consider the LSTM in unrolled form, i.e. for the image and each word in the sentence a copy of the LSTM memory is created. All the copies share the same parameters and the output y_{t-1} of the LSTM at time $t-1$ is fed to the LSTM at time t . In the unrolled version, the recurrent connections are used as feed-forwards connection. Specifically, the input image is denoted by I , and let $S = (S_0, \dots, S_N)$ to be an training sample for that image, then the procedure that is followed in the unrolled LSTM is the following:

$$x_0 = \text{CNN}(I) \quad (4.11)$$

$$x_t = W_e S_t, \quad t \in \{0 \dots N-1\} \quad (4.12)$$

$$p_{t+1} = \text{LSTM}(x_t), \quad t \in \{0 \dots N-1\} \quad (4.13)$$

Each words is represented as a one-hot vector S_t the dimensions of which are equal to the size of vocabulary. Both the image and the words are mapped to the same space, the image by using a vision CNN, the words by using word embedding W_e . The image I is only input once, at $t = 0$, to inform the LSTM about the image contents.

Therefore, the training objective of the model is to assign the highest probability of the next ground truth word given the current input and the hidden state. Then the whole model is trained to minimize the negative log likelihood, which is equivalent to maximize the likelihood as follows:

$$L(I, S) = - \sum_{t=1}^N \log p_t(S_t) . \quad (4.14)$$

The above loss is minimized w.r.t. all the parameters of the LSTM, the top layer of the image embedder CNN and word embeddings W_e .

4.1.3 Training and Regularization

In order the language model to minimize the negative log likelihood cost function shown in (4.14) it is optimized with the use of backpropagating through time and adjusting the LSTM parameters using gradient descent. Specifically, the ADAM variant of the stochastic gradient descent was used. The learning rate was set to 0.00051. So far it was only discussed the training on a single example. However, the training samples were used in random mini batches of sentence–image pairs. It was found that the captions within each mini-batch had to be padded in order all the sentences to have the same length otherwise the loss function

was affected. The size of the batch was found to affect the language model the most among the parameters. The best size batch was 32.

Secondly, dropout was used for regularization of the caption generation. Broadly speaking, the dropout method drops the output of some neurons within a layer before feeding the to the next layer. The basic idea behind this method is that the neurons are more robust since they are less co-dependent. Following the premises of [126], the dropout was used only at the input and output of the LSTM. It was attempted to be used to the recurrent connections as well but the model did not performed at it was expected. Additionally, it was found that using a drop probability of 0.22 increases the generalization capacity of the model.

4.1.4 Inference

At the inference phase the model is expected to generate the most likely caption S^* given an input image. More formally:

$$\begin{aligned}
 S^* &= \arg \max_s \log p(S|I) \\
 &= \arg \max_s \sum_{t=0}^N \log p_t(S_t|I, S_0, \dots, S_{t-1}) \\
 &= \arg \max_s \sum_{t=0}^N LSTM_t(S_t|I, S_0, \dots, S_{t-1})
 \end{aligned} \tag{4.15}$$

However, during the test phase of the image caption generation, there are not reference captions available. Thus, it is required to sample the words of those captions from the distribution $P(S|I)$. Concretely, similar to the training phase the image features are fed to the language model at time-step $t = 0$ accompanied with the a special symbol that indicates the start of the sequence. Then, at time-step $t = 1$ the word with the highest probability it the word that model considers to be the first word. At time-step $t = 2$ the corresponding embedding of the word is fed to the language model. This process continues until the special end-of-sequence token is sampled or a maximum length is reached.

One major disadvantage that stems from this approach is that, it is not guaranteed that by finding the most likely word at each time will lead to the most likely sentence. Ideally, the model should exhaustively search the entire space of possibly sentences. However, such a search is not feasible since the time complexity will be exponentially to the caption's length.

In literature, beam search has been proposed to address this issue. The basic idea underlying beam search is to maintain b partial sentences R_t at each step. Concretely, the log

probability distribution of a partial caption R_t granted an image I is:

$$\begin{aligned}
 \log p(R_t | I) &= \sum_{u=0}^t \log p_u(S_u | I, S_0, \dots, S_{u-1}) \\
 &= \log p(R_{t-1} | I) + \log p_t(S_t | I, S_0, \dots, S_{t-1}) \\
 &= \log p(R_{t-1} | I) + \log p_t(S_t | I, R_{t-1}) \\
 &= \log p(R_{t-1} | I) + \log LSTM_t(S_t | I, R_{t-1})
 \end{aligned} \tag{4.16}$$

In particular, for those top- b sentences only the extensions with top- b words are considered and the partial sentences are re-ranked accordingly. This process is repeated until there no other available search beams or the maximum length of caption is reached. The final output of this procedure is b generated candidate captions ranked based on the log likelihood as been calculated in 4.16. For the purposes of this thesis the beam search was used.

4.2 Evaluation Metrics

Evaluating the quality of the output of image captioning systems, and Natural Language Generation systems in general, is a fundamentally difficult task. Sometimes, multiple answers are correct and deciding which mapping is better is often subjective. Fortunately, there have been a number of evaluation measures that have been proposed in literature that fell into the following two categories: extrinsic or intrinsic evaluation methods. Intrinsic evaluation measures the performance of a system in terms of the similarity of its output to a reference model or on quality criteria. Such measures are not concerned with the effectiveness of the system in relation to its users.

In image captioning literature the most used evaluation metrics are borrowed from machine translation, namely BLEU [95], ROUGE-L[79] and METEOR [28]. The rationale behind the use of those automatic metrics is that human evaluation can be time consuming and expensive. These measures were originally developed and used to evaluate the output of machine translation and/or text summarization methods. However, CIDEr [118] was developed specifically for the evaluation of image captioning systems. The basic idea behind those automatic evaluation metrics is to compare the output of a machine translation system against reference human translations.

4.2.1 BLEU

BLEU [95] attempts to capture legitimate variation in the wording or the order of words of the output of the system through the use of multiple reference translations [64]. In other words, “the closer a machine translation is to a professional human translation, the better it is”. More formally, blue is defined as the geometric mean of n-gram precision scores multiplied by a brevity penalty factor [64]. This factor is required due to the fact that if only the precision it was used, the sentences containing only one word will be always score better than longer sentences. The final BLEU score is given by

$$\text{BLEU-n} = BP \cdot \exp\left(\sum_{n=1}^N w_n \log(p_n)\right), \quad (4.17)$$

4.2.2 ROUGE-L

ROUGE [79] was initially proposed for the evaluation of the summarization systems. It compares an automatically produced text against a single or a set of references produced by humans. In practice ROUGE is a family of metrics that solely relies on overlapping n-grams, words sequences and word pairs. The calculation of ROUGE in image captioning is two-fold. Firstly, the recall and the precision scores of the longest most used subsequences observed between the reference and the candidate sentences and is calculated:

$$R_{lcs} = \frac{LCS(Cand, Ref)}{Reference\ length}, \quad (4.18)$$

$$P_{lcs} = \frac{LCS(Cand, Ref)}{Candidate\ length}, \quad (4.19)$$

where R_{lcs} and P_{lcs} are recall and precision scores respectively, $LCS(Cand, Ref)$ is the longest most used subsequence among the candidate $Cand$ and reference Ref .

Finally, the final ROUGE-L score is computed as :

$$\text{ROUGE-L} = F_{\beta} = \frac{(1 + \beta^2)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}}. \quad (4.20)$$

4.2.3 METEOR

In order one to calculate the METEOR [28] score, first has to align the candidate and reference sentences, where each word in the candidate sentence should be matched with exactly one word in the reference set. However, despite the matching of words between

the candidate and the reference sentences, stemmed token and paraphrase matches are also considered. The basic idea underlying this process is to minimize the sequences of words that are contained in both the sentences. Then, as a second step, the weighted precision and recall are computed for the aligned words. Note that the weights are set based on the type of alignment. Then the final Meteor score is computed as the penalized product of the number of sequences in the alignment and the F-score derived from the weighted precision and recall:

$$Pen = \gamma \cdot \left(\frac{ch}{m} \right)^\theta, \quad (4.21)$$

$$METEOR = (1 - Pen) \frac{P_m R_m}{\alpha P_m + (1 - \alpha) R_m}, \quad (4.22)$$

where the terms R_m and P_m are the recall and precision, ch is the number sequences the alignment is consisted, m is the length of the candidate sentence and α , γ and θ are factors that minimize the correlation of the metric with human collected judgments. Note that in case there are many reference sentences, the final METEOR score is the maximum of the candidate sentence and any given reference sentence.

4.2.4 CIDEr

CIDEr [118] was the first evaluation metric proposed for the task of image captioning. In particular, its goal is to evaluate how well a candidate sentence matches the consensus of a set of reference sentences. Towards this direction, for each candidate and reference sentence a vector with the well known vectorization process of frequency inverse document frequency (TF-IDF) is computed. In this context, TF plays the role of the consensus since its considering the frequently occurring words in the reference sentences while IDF penalizes the most common words that occur to captions for different images.

The $CIDEr_n$ score is calculated by averaging the cosine similarity between the TF-IDF vectors of the candidate sentences and all reference captions. The n indicates the n -gram size which was used to calculate the TF-IDF vector. The Final CIDEr score is the mean of the four $CIDEr_n$ metrics, with $n = 1, 2, 3, 4$.

There were a few modification to the original CIDEr metric in order to prevent of leading systems to produce captions that achieve high scores but those scores are not verified by the human judgment. In literature, this metric is coined as CIDEr-D and is the most widely used version in image captioning.

4.2.5 Automatic Metrics Discussion

Loosely speaking, the aforementioned metrics evaluate the suitability of a caption with respect to the visual content of the image. They achieve that, by comparing how well a candidate caption is correlated with a reference caption. It was found, that when the number of the reference captions is increased the better those metrics perform [118]. Additionally, in [118] it was reported that CIDEr and METEOR are the metrics that represent the human judgment better, while BLEU-4 it was found that has no correlation at all. Those findings were further certified in the first MS-COCO image captioning challenge. In particular, [21] reaffirmed the observations made by [118].

Introducing automatic measures that can mimic human judgments in evaluating the suitability of image descriptions is most likely the most urgent need in the area of image captioning [7]. However, since conducting human judgment experiments is costly, there is a major need for improved automatic measures that are more highly correlated with human judgments. In Figure 4.5 it is shown for the Flickr8k dataset, the BLEU is confirmed to be unable to discriminate between the lowest three human judgments, while consistent to previous works [118, 21] Meteor and CIDEr show signs of a clear discrimination. Since in this thesis the Flickr8k is used, most importance will be given to METEOR and CIDEr metrics.

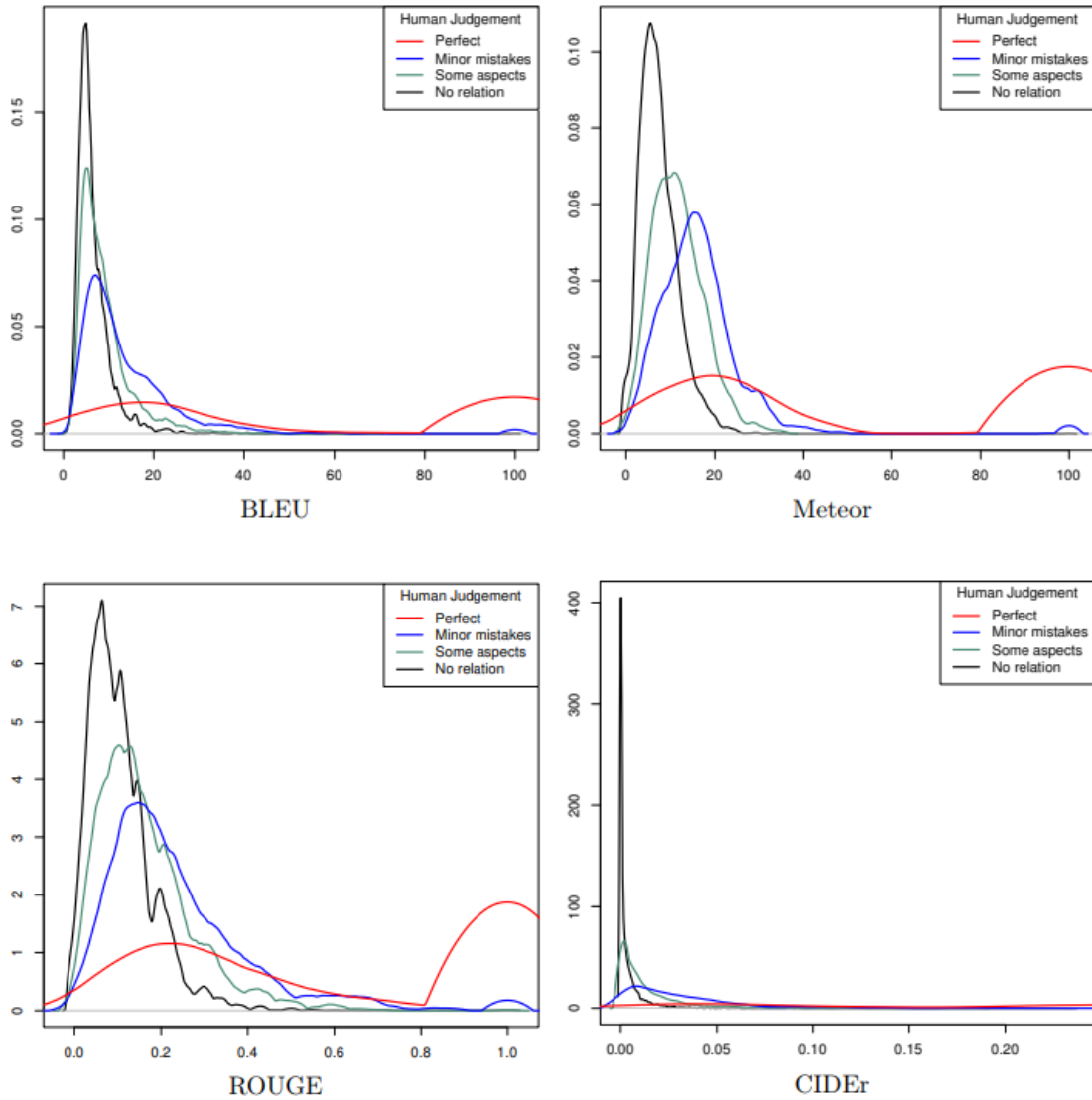


Fig. 4.5 Probability density predicates of BLEU, Meteor, ROUGE, and CIDEr scores compared to human judgment in the Flickr8K dataset [7]. The y-axis depicts the probability density while the x-axis is the score of the automatic metrics.

Chapter 5

Experiments

In the previous chapter, it was formalized the model that was used for the purposes of this thesis. In this chapter the results from the experiments of the Flickr8k dataset are presented. The evaluation is relied on the four evaluation metrics described in Section 4.2, namely BLEU, METEOR, ROUGE-L and CIDEr, to evaluate the quality of the produced captions.

Although there has been immense progress in the field of image captioning, those models require notable computational power. Therefore, in this thesis experiments that require significant changes on the models were omitted. To illustrate the computational needs, in [119] they reported that training model took over three weeks with a high-end GPU. The implemented model builds upon their model. Training the model on an average computer required 50 hours of training.

Additionally, [49] argue that the performance of image captioning models mostly restricted due to language models. In order to prove their point, they trained an image encoder model gradually while they kept the same language. They reported that each time step the results are improved. Conversely, when they used the image encoder as a fixed model, training the language model yields better captions to a certain level after which additional training does not improve the results. Therefore, this thesis focuses on the experiments on quality of captions where it was assumed that there is room for improvements.

5.1 Implementation Details

Before presenting the evaluation results, a brief discussion on implementation platforms details, and hyper-parameters choices is given.

Language Model: The proposed LSTM language model used as generator is implemented using the Keras library [18] running on tensorflow backend [2]. The language models are

trained using stochastic gradient descent with the Adam [62] algorithm and the dropout regularization is implemented as described in [127]. The error is back-propagated to all the language model parameters and word embedding matrices, but the image feature extraction models are kept constant due to computational limitations. The language model is trained by minimizing the negative log-likelihood assigned by the model to the training samples.

In all experiments beam search was used of size $l = 3$. Different sizes of beam search were used. However, despite the expectations, when the size of search was increased the captions were shorter and most of the time incomplete. For example for the image depicted in Figure 5.1 captions for different sizes are depicted but also those produced by greedy search.



Fig. 5.1 Greedy Search: A dog runs
Beam Search = 3: A brown dog runs through the grass.
Beam Search = 7: A dog runs through the grass.
Beam Search = 11,20 : A dog.

Flicker8k dataset: In all of the experiments the Flicker8k dataset was used. The reason that this dataset was chosen is that is relatively small dataset and that allowed to make the training of the model feasible. Although, the MS-COCO is the most widely used dataset in image captioning, its size make the training of a captioning model on an average computer impossible. The flicker dataset consists of 6000 training images, 1000 validation and 1000 test images. The training and validation sets have five reference captions for each image annotated by humans.

Before using the reference captions of flicker dataset for training, the text is tokenized in a way that the symbols and numbers are removed. Furthermore, words that occurring less than five times are also removed in order to avoid to fed the model with spelling mistakes

and also to avoid extremely rare words. Those words are depicted in Figures 5.2,5.7. The final vocabulary is of 8763 tokens.

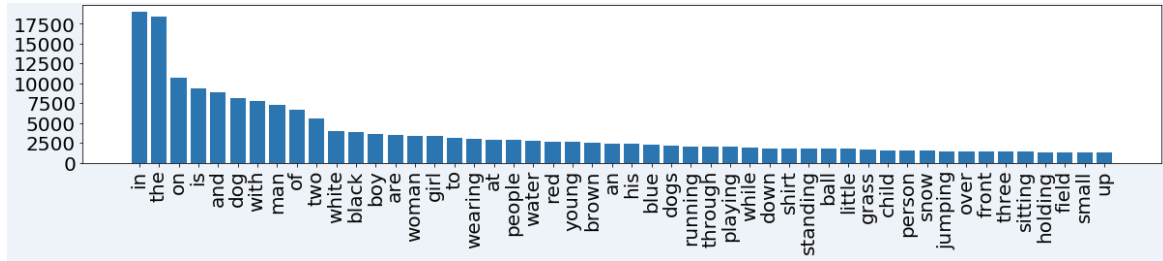


Fig. 5.2 The fifty most frequently appearing words in the training captions [Win]

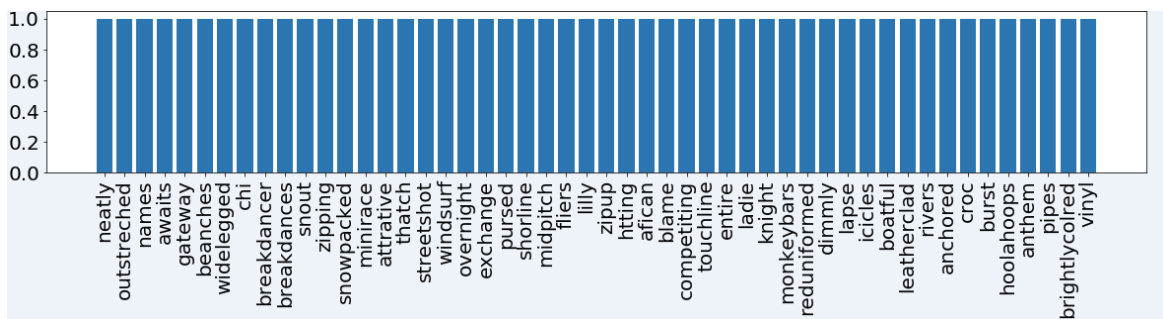


Fig. 5.3 The fifty least frequently appearing words in the training captions [Win]

Visual Feature Extraction: The CNN models that were used are the VGG-16 and Inception V3 and are based on the Keras Library and are trained on ImageNet dataset [26]. In order to obtain a global image representation, for VGG-16 the activation of its last fully connected layer was used leading to 4096 dimension image representation. For Inception, activation of pool_3:0 layer were used leading to a 2048 dimension image presentation. Then this representation is compressed into a fixed-length dimensional vector. It was found that by transforming the image representation into an image embedding yields better results.

The overall hyper-parameters of the final model are given in Table 5.1

Hyperparameter	Value
Learning rate	0.00051
Batch size	32
Epochs	33
Dropout rate	0.22
Embedding size	300
LSTM output size	300
LSTM layers	3

Table 5.1 The overall hyper-parameters used in the final model.

Evaluation Metric	Baseline	Main Model
BLEU-1	58.00	61.75
BLEU-2	34.33	40.79
BLEU-3	25.30	27.84
BLEU-4	13.40	18.95
METEOR	14.34	21.49
CIDEr	37.56	41.5

Table 5.2 Evaluation Results

5.2 Image captioning results

This section presented the evaluation findings of the proposed method, along with the baseline model which is the one proposed in [119] but instead of using the Inception V1 as image encoder the VGG-16 was used. Table shows the evaluation results on different image captioning metrics.

As it is shown in the table, the different image features affecting the image captioning model significantly, not only in term of text quality but also on how accurately describe the image. However, both models producing captions that are redundant. Furthermore, it was found that the produced captions are identical in terms of textual similarity but also identical with those in the training set. In particular, if the best candidate is taken into consideration, the sentence is present in the training set 84% of the times. This however, should not come as a surprise. The objective that the model is trained but also the evaluation metrics encouraging the use of exact wording with the image captions that are in the training set. In other words, the model is trained on such objectives encourage the similarity in the n-gram space rather than in the semantic space. That is apparent the Figure 5.4. Where as it can be seen the model produced an identical caption. In particular, the model overlooked the child in the image since the 12 similar scenes in the training set where not depicting a child.



Fig. 5.4 The image on the left is the one from the training images. The model produced as the a caption “ a man and a woman are sitting on the edge of a lake ”. The right image is one of the training test and its accompanied caption is “A man and woman sitting on a deck next to a lake”

Another significant finding is that, only a limited number of words from the the training vocabulary is retrieved. Concretely, the proposed model was trained on a vocabulary that contains approximately eight thousands words but it was only able to produce approximately 2000 words and in case of baseline almost 1000. An interpretation of these results is that the better the image encoding is the more words are used. However, it should be noted that all the image captioning models, even those that claim state-of-the art results are tuning out secondary visual information. Note that in this thesis, also only a global image encoding was used. For example other information such as object, scenes detections were not incorporated into the image encoding. By doing so, the model is not exposed to all the available information and therefore, it is not possible to produce descriptions that containing other visual information since there not available to it. In other words, the models are ought to be able to talk about objects in the images that given the powerful CNN models can be actually detected as opposed to letting the language model to hallucinate whether which object is depicted. Apparently, those limitation lead the models to produce smaller vocabulary and significant deviations from the training data word distributions. This problem becomes apparent especially when images are shown in succession as in figure 5.5. It can be seen that the model reveals itself by repeating the same degraded n-gram distribution.

Furthermore, what is more important is that the captions that fail to describe the content of an image accurately is because that are scene dependent. For example, the training set distribution of the following term was examined: “A man and a woman”. In Figure 5.6 are depicted two images with a similar content. For the image on the left which a test image the model produced as a caption “a man and a woman are sitting in a kitchen ” while for the ground truth image the caption is “A man and a woman are sitting down and eating ”. It is apparent that model produced a caption towards which is biased due the plethora of similar examples in the training set. In this case however, the woman is depicted in television



Fig. 5.5 For the left image the caption produced by the model is “a football player in red is challenging the player in red ” while for the image from the right is “a group of football players are tackling a football player’

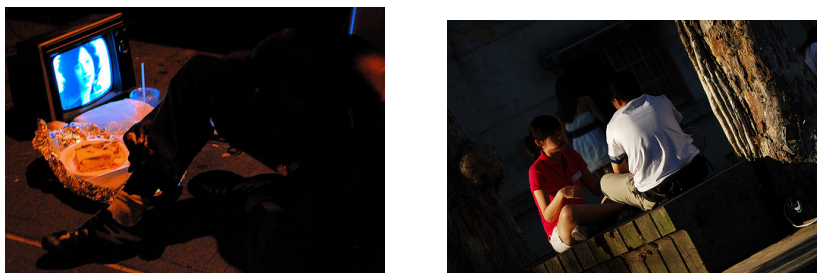


Fig. 5.6 The left image is derived from the test set while the right image is derived from the training set.

and the model was unable to understand that the woman is not actually present in the image. There are many of those examples in which when the angle or the context of the image bears some visual similarity the model will produce a captions that fails to capture its context by reproducing those are in the training set. The reason for such behavior is as mentioned above, that the global representation does not convey enough information so the language model to produce an accurate caption. It disregards completely the scene context. Thus, instead of allowing the language model to do the heavy lifting, image captioning model should rely more on the image captioning models.

Captions that score more than 60 in terms of blue are depicted in the image Figure 5.7

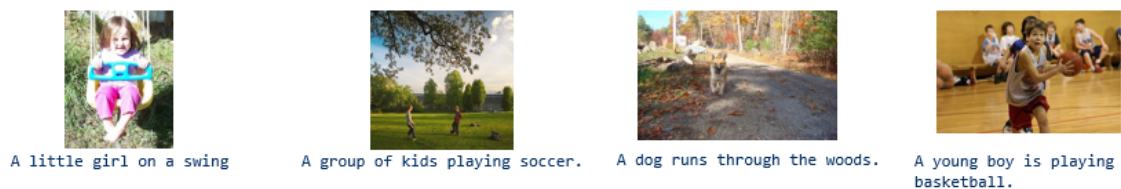


Fig. 5.7 Successful captions produced by the implemented image captioning model.

Chapter 6

Conclusion & Future Work

As this thesis demonstrates, the Computer Vision and Natural Language fields have witnessed an upsurge in interest in image captioning systems. This is largely owned to the recent advances in deep learning models for images and text, which has allowed the models to be trained in an end-to-end fashion. Nevertheless, a series of challenges for image captioning remain. This documents shed light on the quality of the produced captions and how the captions feel rigid, dry, and lacking in vitality.

Current algorithms rely on direct representations of the descriptions they are exposed at the training time, and thus their output at test time is very similar. This results in many repetitions and limits the diversity and the naturalness of the generated text. Although these models have led to a constant increase of performance on evaluation metrics, yet the machine produced captions are easily distinguished by humans. This issue becomes apparent when multiple captions are shown in succession. This is not surprising. The extensive survey of the existing approaches shows that existing efforts primarily focus on veracity rather than other fundamental qualities of human language, and behavior in general, that is the naturalness and diversity. After the experiments for the purposes of this thesis, four different axons have been identified that prevents the models of producing such output: (1) the aim of the models, which is best modeled in their objective functions; (2) the representation those models use for the generation; (3) the absence of information fusion between language and image features and the way the image features are fused during the generation process; and (4) that today's state-of-the-art image captioning systems are evaluated to produce a single RE for each object.

Firstly, with a closer look on the objective function that the models are trained (See Chapter 4), it will become apparent that are mostly trained with the maximum likelihood, as the model that used in this thesis. Such an objective exclusively enforces that the caption will unambiguously describe the image by forcing the model to use the n-grams high

resemblance to the ground truth in terms of detailed wording disregarding the visual input. Consequently, the model will ignore captions that are similar in semantic space but will encourage resemblance in the n-gram space which is only a small subset of the former. Therefore, those systems often “reveal themselves” because they generating a particular word distribution and using a smaller vocabulary. However, intuitively it seems desirable to take into account not only the dependency of the produced caption in respect with a particular image, but also the inverse, the likelihood that a particular caption will be produce given images with similar content.

Furthermore, systems are designed to produce a single caption. Yet, multiple unambiguous captions are typically correct for a single image. This property, is profound in humans speaker language formation process. However, this diversity has been neglected completely in state-of-the-art systems. Firstly, the recent progress on image captioning has greatly deranged the well-known saying that *a picture is worth a thousand words*. Arguably, the starting point for every captioning system is to understand the image, i.e. recognizing objects within it, reasoning about the relationship between them, and focusing on the most salient content of the image. However, the desiderata to use only the most salient information disregards secondary information (e.g. scene specific features) in the generated captions. In other words, the models are exposed to a particular context and hence the produced captions will always belong to the same *semantic space*. In this thesis, this hypothesis was verified. In particular, when it was attempted with the use of beam search to sample multiple captions, all the top-ranked captions in the beam were related to each other, sometimes differing only in grammatical arrangements, and thus still only cover a small portion of the available visual information. However, a system that produces multiple captions, that belong in the same semantic space seems to be the only way forward. One way to design such a system would be to further condition the caption generation with an vector" which encodes the visual information already described in previously generated captions.

Thirdly, the aim of the systems not only affects their objectives but also the representation used in the generation process. Specifically, the primary purpose of the features that are used in image captioning is to facilitate the production of a caption that describes accurately the image by encoding global context and differences in terms of visual appearance and attributes of objects depicted in the images. However, such representation makes the assumption that CNN features encode hierarchical properties that represent scene context or other information in the scene. Even if that assumption holds, it is no-trivial for a language model to learn how to utilize this implicit information. Thus, there could be benefits from extracting explicitly contextual features. In particular, this document makes the following distinction between two main kinds of visual features that might play a role if the goal is to generate human-like

image captions: (1) “traditional” image features that are concerned to convey the properties that will produce a caption that achieves the goal of describing the image with brevity; and (2) scene-specific features that capture higher-level semantic information encoded in an image in explicit manner. For instance, it is unlikely to refer to a plane as boat if the scene is about airports. Thus the caption ought to be able to describe a plane based on the scene that has been actually detected as opposed to letting the language model hallucinate an airport or a plane because that sounds plausible. This is largely owned that the existing systems, more often than not, rely only on just a first glance gist of the scene. Finally, the assumption to keep only the most salient information neglects secondary information in the generated caption. This resulting in unnatural language but the captions often achieve their goal.

However, determining which of aspects of the scene increase the discriminatory power or the naturalness of the captions to be produced requires commonsense knowledge. Such commonsense knowledge can be extracted by using the captions in the training set due to the fact that captions only contain information that is inherently salient. Additionally, there abstract concepts that an object recognition model is unable to capture. For instance, concepts with vague properties such as short or tall may be highly correlated to specific visual patterns. Or, the language model can learn, based on particular visual patterns, whether a person rides a bike. Exploiting the parallel structure of image and language features it is possible the model to disambiguate noisy visual detections. Note that, one fundamental assumption that is made in this line of work is that the visual detections are accurate. This assumption, however, does not always holds. Lastly, by learning a joint multi-modal representation for both image features and language is it possible to measure the similarity between them and select those that are most suitable to be used. However, rarely works in the image captioning use joint representation or in fact considers language features. Instead, language is considered holistically.

Undeniably, the progress in image captioning is tremendous. However, the systems are far from perfect. If one wants to draw an analogy with the way of humans acquire language will notice that in the first stage, humans learn to utter a few words in order to refer to particular objects in their environment. Then, as a second stage humans learn to repeat and recombine sentences that hear from the people surrounding them. Only in later stages, human acquire the skill to precisely reason about the words they use, and to be able to produce a description or even long essays about an single entity such as image. As this thesis proved, the image captioning models are currently at the second stage. That is those systems tend to repeat or recombined already seen image captions from the training. Although there is a small number of works that trying to make the generated image captions as human-like as possible they doing so by sacrificing their unambiguous context.

References

- [Win] Image captioning from scratch., howpublished =
https://fairyonice.github.io/develop_an_image_captioning_deep_learning_model_using_flickr8k_data.html. Accessed 2010-09-30.
- [2] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- [3] Andrew, G., Arora, R., Bilmes, J., and Livescu, K. (2013). Deep canonical correlation analysis. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1247–1255, Atlanta, Georgia, USA. PMLR.
- [4] Bai, S. and An, S. (2018). A survey on automatic image caption generation. *Neurocomputing*, 311:291 – 304.
- [5] Baltrusaitis, T., Ahuja, C., and Morency, L. (2017). Multimodal machine learning: A survey and taxonomy. *CoRR*, abs/1705.09406.
- [6] Bengio, I. G. Y. and Courville, A. (2016). Deep learning. Book in preparation for MIT Press.
- [7] Bernardi, R., Cakici, R., Elliott, D., Erdem, A., Erdem, E., Ikizler-Cinbis, N., Keller, F., Muscat, A., and Plank, B. (2016a). Automatic description generation from images: A survey of models, datasets, and evaluation measures. *J. Artif. Int. Res.*, 55(1):409–442.
- [8] Bernardi, R., Çakici, R., Elliott, D., Erdem, A., Erdem, E., Ikizler-Cinbis, N., Keller, F., Muscat, A., and Plank, B. (2016b). Automatic description generation from images: A survey of models, datasets, and evaluation measures. *CoRR*, abs/1601.03896.
- [9] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [10] Boureau, Y. L., Bach, F., LeCun, Y., and Ponce, J. (2010). Learning mid-level features for recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2559–2566.

- [11] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Józefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space. *CoRR*, abs/1511.06349.
- [12] Brown-Schmidt, S. (2009). Partner-specific interpretation of maintained referential precedents during interactive dialog. *Journal of memory and language*, 61(2):171–190.
- [13] Caudill, M. (1987). Neural networks primer, part i. *AI Expert*, 2(12):46–52.
- [14] Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. *CoRR*, abs/1405.3531.
- [15] Chen, C., Mu, S., Xiao, W., Ye, Z., Wu, L., Ma, F., and Ju, Q. (2018). Improving image captioning with conditional generative adversarial nets. *CoRR*, abs/1805.07112.
- [16] Chen, X. and Zitnick, C. L. (2015). Mind’s eye: A recurrent visual representation for image caption generation. In *CVPR*, pages 2422–2431. IEEE Computer Society.
- [17] Cho, K., van Merriënboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.
- [18] Chollet, F. (2015). keras. <https://github.com/fchollet/keras>.
- [19] Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., and Schmidhuber, J. (2011). High-performance neural networks for visual object classification. *CoRR*, abs/1102.0183.
- [20] Clarke, J. and Lapata, M. (2008). Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research (JAIR)*.
- [21] Cui, Y., Ruggero Ronchi, M., and Lin, T.-Y. (2015). 1st captioning challenge slides. Large-scale Scene UNderstanding Workshop.
- [22] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314.
- [23] Dai, B., Lin, D., Urtasun, R., and Fidler, S. (2017). Towards diverse and natural image descriptions via a conditional GAN. *CoRR*, abs/1703.06029.
- [24] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 1, pages 886–893 vol. 1.
- [25] De Marneffe, M.-C., MacCartney, B., Manning, C. D., et al. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454. Genoa Italy.
- [26] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L. (2009a). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255.
- [27] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L. (2009b). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

- [28] Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.
- [29] Denton, E. L., Chintala, S., Szlam, A., and Fergus, R. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. *CoRR*, abs/1506.05751.
- [30] Devlin, J., Cheng, H., Fang, H., Gupta, S., Deng, L., He, X., Zweig, G., and Mitchell, M. (2015). Language models for image captioning: The quirks and what works. *CoRR*, abs/1505.01809.
- [31] DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron*, 73:415–34.
- [32] Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*.
- [33] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2013). Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531.
- [34] Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159.
- [35] Elliott, D. and de Vries, A. P. (2015). Describing images using inferred visual dependency representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 42–52.
- [36] Elliott, D. and Keller, F. (2013). Image description using visual dependency representations. In *EMNLP*, pages 1292–1302. ACL.
- [37] Elliott, D., Lavrenko, V., and Keller, F. (2014). Query-by-example image retrieval using visual dependency representations. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 109–120.
- [Everingham et al.] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. <http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html>.
- [39] Fang, H., Gupta, S., Iandola, F., Srivastava, R., Deng, L., Dollar, P., Gao, J., He, X., Mitchell, M., Platt, J., Zitnick, L., and Zweig, G. (2015). From captions to visual concepts and back. In *CVPR*.
- [40] Farhadi, A., Hejrati, M., Sadeghi, M. A., Young, P., Rashtchian, C., Hockenmaier, J., and Forsyth, D. (2010). Every picture tells a story: Generating sentences from images. In *ECCV*, pages 15–29.

- [41] Gatt, A. and Krahmer, E. (2017). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *CoRR*, abs/1703.09902.
- [42] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587.
- [43] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feed-forward neural networks. In *JMLR W&CP: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, volume 9, pages 249–256.
- [44] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- [45] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.
- [46] Gupta, A., Verma, Y., and Jawahar, C. V. (2012). Choosing linguistics over vision to describe images. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI’12, pages 606–612. AAAI Press.
- [47] Haykin, S. (2009). *Neural Networks and Learning Machines*. Number τ . 10 in Neural networks and learning machines. Prentice Hall.
- [48] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.
- [49] Hessel, J., Savva, N., and Wilber, M. J. (2015). Image representations and new domains in neural image captioning. *CoRR*, abs/1508.02091.
- [50] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.
- [51] Hodosh, M., Young, P., and Hockenmaier, J. (2013a). Framing image description as a ranking task: Data, models and evaluation metrics. *J. Artif. Int. Res.*, 47(1):853–899.
- [52] Hodosh, M., Young, P., and Hockenmaier, J. (2013b). Framing image description as a ranking task: Data, models and evaluation metrics. *J. Artif. Int. Res.*, 47(1):853–899.
- [53] Isola, P., Zhu, J., Zhou, T., and Efros, A. A. (2016). Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004.
- [54] Jia, X., Gavves, E., Fernando, B., and Tuytelaars, T. (2015). Guiding long-short term memory for image caption generation. *CoRR*, abs/1509.04942.
- [55] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R. B., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *CoRR*, abs/1408.5093.

- [56] Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. Seattle. Association for Computational Linguistics.
- [57] Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland. Association for Computational Linguistics.
- [58] Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [59] Karpathy, A., Joulin, A., and Fei-Fei, L. F. (2014). Deep fragment embeddings for bidirectional image sentence mapping. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 1889–1897. Curran Associates, Inc.
- [60] Ke, T., Yang, B., Zhen, L., Tan, J., Li, Y., and Jing, L. (2012). Building high-performance classifiers using positive and unlabeled examples for text classification. In Wang, J., Yen, G. G., and Polycarpou, M. M., editors, *Advances in Neural Networks – ISNN 2012*, pages 187–195, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [61] Kingma, D. P. and Ba, J. (2014a). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- [62] Kingma, D. P. and Ba, J. (2014b). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- [63] Kiros, R., Salakhutdinov, R., and Zemel, R. S. (2014). Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*.
- [64] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L., Shamma, D. A., Bernstein, M. S., and Li, F. (2016). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *CoRR*, abs/1602.07332.
- [65] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS’12, pages 1097–1105, USA. Curran Associates Inc.
- [66] Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A. C., and Berg, T. L. (2011). Baby talk: Understanding and generating image descriptions. In *Proceedings of the 24th CVPR*.
- [67] Kulkarni, G., Premraj, V., Ordonez, V., Dhar, S., Li, S., Choi, Y., Berg, A. C., and Berg, T. L. (2013). Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903.
- [68] Kuznetsova, P., Ordonez, V., Berg, A. C., Berg, T. L., and Choi, Y. (2012). Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL ’12, pages 359–368, Stroudsburg, PA, USA. Association for Computational Linguistics.

- [69] Kuznetsova, P., Ordonez, V., Berg, T., and Choi, Y. (2014a). Treetalk: Composition and compression of trees for image descriptions. *Transactions of the Association for Computational Linguistics*, 2:351–362.
- [70] Kuznetsova, P., Ordonez, V., Berg, T. L., and Choi, Y. (2014b). Treetalk: Composition and compression of trees for image descriptions. *TACL*, 2:351–362.
- [71] Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8595–8598.
- [72] Lebrecht, R., Pinheiro, P. O., and Collobert, R. (2015). Phrase-based image captioning. *arXiv preprint arXiv:1502.03671*.
- [73] LeCun, Y., Bengio, Y., and Hinton, G. E. (2015). Deep learning. *Nature*, 521(7553):436–444.
- [74] LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In Touretzky, D. S., editor, *Advances in Neural Information Processing Systems 2*, pages 396–404. Morgan-Kaufmann.
- [75] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- [76] LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.
- [77] Ledig, C., Theis, L., Huszar, F., Caballero, J., Aitken, A. P., Tejani, A., Totz, J., Wang, Z., and Shi, W. (2016). Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802.
- [78] Li, J., Wang, G. A., and Chen, H. (2011). Identity matching using personal and social identity features. *Information Systems Frontiers*, 13(1):101–113.
- [79] Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. *Text summarization branches out: Proceedings of the ACL-04 workshop*, 8.
- [80] Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *CoRR*, abs/1312.4400.
- [81] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.
- [82] Liu, M., Shi, J., Li, Z., Li, C., Zhu, J., and Liu, S. (2016). Towards better analysis of deep convolutional neural networks. *CoRR*, abs/1604.07043.
- [83] Ma, L., Lu, Z., Shang, L., and Li, H. (2015). Multimodal convolutional neural networks for matching image and sentence. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2623–2631.
- [84] Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A. L., and Murphy, K. (2015). Generation and comprehension of unambiguous object descriptions. *CoRR*, abs/1511.02283.

- [85] Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., and Yuille, A. (2014). Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*.
- [86] Mason, R. and Charniak, E. (2014). Nonparametric method for data-driven image captioning. In *ACL (2)*, pages 592–598. The Association for Computer Linguistics.
- [87] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.
- [88] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.
- [89] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- [90] Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *CoRR*, abs/1411.1784.
- [Mitchell et al.] Mitchell, M., Han, X., Dodge, J., Mensch, A., Goyal, A., Berg, A., Yamaguchi, K., Berg, T., Stratos, K., and Iii, H. D. Midge: Generating image descriptions from computer vision detections.
- [92] Nenkova, A. and Vanderwende, L. (2005). The impact of frequency on summarization. Technical report, Microsoft Research.
- [93] Ordonez, V., Kulkarni, G., and Berg, T. L. (2011). Im2text: Describing images using 1 million captioned photographs. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 1143–1151. Curran Associates, Inc.
- [94] Papineni, K., Roukos, S., Ward, T., and jing Zhu, W. (2002a). Bleu: a method for automatic evaluation of machine translation. pages 311–318.
- [95] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002b). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02*, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [96] Patterson, G., Xu, C., Su, H., and Hays, J. (2014). The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1):59–81.
- [97] Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. (2015). Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), ICCV '15*, pages 2641–2649, Washington, DC, USA. IEEE Computer Society.
- [98] Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434.

- [99] Raiber, F. and Kurland, O. (2017). Kullback-leibler divergence revisited. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR '17*, pages 117–124, New York, NY, USA. ACM.
- [100] Rashtchian, C., Young, P., Hodosh, M., and Hockenmaier, J. (2010). Collecting image annotations using amazon’s mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk, CSLDAMT '10*, pages 139–147, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [101] Reiter, E. and Belz, A. (2009). An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Comput. Linguist.*, 35(4):529–558.
- [102] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386.
- [103] Roth, D. and Yih, W.-t. (2004). A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*.
- [104] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA.
- [105] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1988). Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA.
- [106] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(3):211–252.
- [107] Saidane, Z. and Garcia, C. (2007). Automatic scene text recognition using a convolutional neural network. In *In Proceedings of the Second International Workshop on Camera-Based Document Analysis and Recognition (CBDAR)*.
- [108] Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.
- [109] Shetty, R., Rohrbach, M., Hendricks, L. A., Fritz, M., and Schiele, B. (2017). Speaking the same language: Matching machine to human captions by adversarial training. *CoRR*, abs/1703.10476.
- [110] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.
- [111] Socher, R., Karpathy, A., Le, Q. V., Manning, C. D., and Ng, A. Y. (2013). Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*.

- [112] Srivastava, R. K., Greff, K., and Schmidhuber, J. (2015). Highway networks. *CoRR*, abs/1505.00387.
- [113] Stevens, S. S., Volkman, J., and Newman, E. B. (1937). A Scale for the Measurement of the Psychological Magnitude Pitch. 8(3):185–190.
- [114] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- [115] Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems, NIPS’99*, pages 1057–1063, Cambridge, MA, USA. MIT Press.
- [116] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*.
- [117] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions. *CoRR*, abs/1409.4842.
- [118] Vedantam, R., Lawrence Zitnick, C., and Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [119] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [120] Wang, T., Wu, D. J., Coates, A., and Ng, A. Y. (2012). End-to-end text recognition with convolutional neural networks. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3304–3308.
- [121] Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1(4):339 – 356.
- [122] Widrow, B. and Hoff, M. E. (1988). Neurocomputing: Foundations of research. chapter Adaptive Switching Circuits, pages 123–134. MIT Press, Cambridge, MA, USA.
- [123] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France. PMLR.
- [124] Yagcioglu, S., Erdem, E., Erdem, A., and Cakici, R. (2015). A distributed representation based query expansion approach for image captioning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 106–111. Association for Computational Linguistics.

- [125] Yang, Y., Teo, C. L., Daumé III, H., and Aloimonos, Y. (2011). Corpus-guided sentence generation of natural images. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 444–454. Association for Computational Linguistics.
- [126] Zaremba, W., Sutskever, I., and Vinyals, O. (2014a). Recurrent neural network regularization. *CoRR*, abs/1409.2329.
- [127] Zaremba, W., Sutskever, I., and Vinyals, O. (2014b). Recurrent neural network regularization. cite arxiv:1409.2329.
- [128] Zeiler, M. D. (2012). ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701.
- [129] Zeiler, M. D. and Fergus, R. (2013). Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901.