4K1-5

顕著性マップを用いた画像の説明文自動生成

Image Captioning with Saliency Maps

吉井 和輝[*1] エリック・ニコルズ[*2] 船越 孝太郎[*2] 中野 幹生[*2] 青野 雅樹[*1]
Kazuki Yoshii Eric Nichols Kotaro Funakoshi Mikio Nakano Masaki Aono

[*1] 豊橋技術科学大学 Toyohashi University of Technology
[*2] (株)ホンダ・リサーチ・インスティテュート・ジャパン Honda Research Institute Japan Co., Ltd.

In recent years, due to advances in neural models for representing images and language, multimodal tasks like image captioning and visual QA have grown in popularity. Typical approaches are based in neural MT models where image region-text alignments are coupled with RNNs to generate captions. However, current approaches have difficulty covering all important regions in caption generation. To address this shortcoming, we introduce a novel image captioning system where salience maps are used to extract and featurize the most important image regions. In this paper, we propose several methods of generating features using saliency map and evaluate their impact on image captioning.

1. はじめに

人間とコミュニケーションを取るロボットや，先端的なナビゲーションなど，実世界を理解し，対話を行うシステムの実現は重要な課題になっている．我々の長期的な目標は，現実世界にも適用可能な柔軟なマルチモーダル表現を構築することである．画像からの説明文生成や Visual QA のようなタスクは，そのようなマルチモーダル表現を評価するのに適している．

近年，ニューラルネットにより画像や言語の表現力が向上したことと，[Lin 2015] など大量の説明文付き画像データセットの公開されたことから，画像からの説明文生成や画像に対する質問応答などの研究が盛んになっている．画像からの説明文生成の一般的なアプローチは，Convolutional Neural Networks (CNN)を用いて画像の特徴量を抽出し，単語ベクトルと共に Recurrent Neural Network (RNN)へ入力して説明文を予測する手法である[Vinyals 2014] [Karpathy 2015].

ニューラルネットを用いた手法は高い精度を達成しているが，画像特徴量を用いるのは説明文生成の最初の段階のみであるので，画像内の領域と生成される単語の対応が取りづらいという問題がある．この問題点を解決するために，単語の生成時に画像内の領域の情報を用いる Visual Attention モデルが提案されている[Xu 2015] [Jin 2015]. しかし，単語と画像内の領域の対応を持つアノテーションデータが少ないため，Visual Attention モデルは教師なしの手法で学習を行う必要があり，これは計算コストが高く，また誤った対応が学習される恐れがある．さらに，Visual Attention モデルは明示的な画像な領域の重要性という概念を持っていない．

画像内の領域の重要性に関する概念は，視覚心理学における視覚的顕著性の枠組みで研究がされている．これは，ある領域が周りに比べてどれくらい目立つのかという情報である．特に，ある画像の中の全ての領域の顕著性を推定する顕著性マップは，画像内の領域の重要性を決めるために有用だと考えられる．

本研究では，画像の説明文生成における顕著性の情報の有用性を検証するために，顕著性マップの情報を説明文生成に適用する手法を提案し，説明文生成に与える影響を調査する．手始めに，顕著性マップから画像内の最も顕著な領域を推定し，固定長の特徴量を計算し，説明文生成の手法に追加することを行う．実験の結果，顕著性マップの情報を用いることで精度が向上することが確認された．

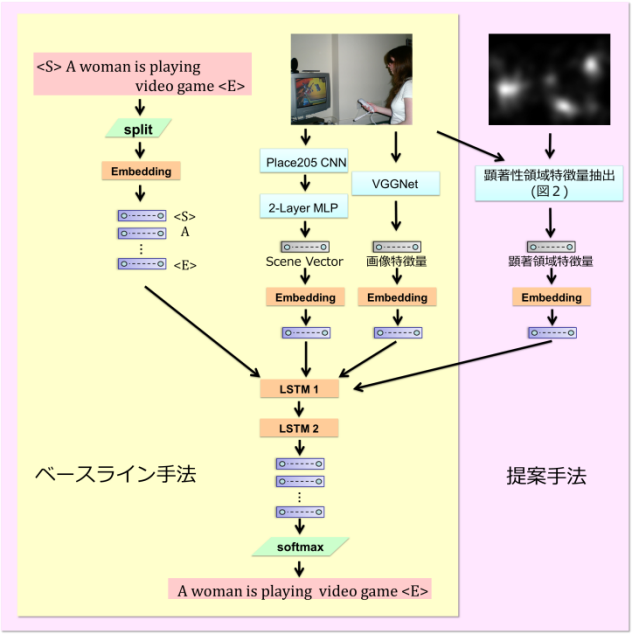


図 1 画像の説明文生成モデルの概要

2. 関連研究

画像説明文の自動生成というタスク自体は 2010 年頃から取り組まれている[Farhadi 2010]. しかし，近年効率的な学習手法が開発されたディープニューラルネットワークによって画像や言語の表現力が向上したことにより，このタスクも急速に発展した．

[Karpathy 2015]は，Region Convolutional Neural Network (R-CNN) を用いて物体検出と画像特徴量の抽出を，Bidirectional Recurrent Neural Network (BRNN)を用いて単語の特徴量の抽出を行い，二つの結果を統合してモデルの学習を行う手法を提案した．

[Jin 2015]は，単語生成時に画像内の小領域との対応の情報も用いる Visual Attention モデルを取り入れた．画像の小領域推定に物体検出手法を応用することで，可変サイズの小領域を用いる事が出来るようにしている．また，野球の試合やキッチンでの料理など，画像がどのような場面で撮影されたかを説明文のトピック分布を用いて定義し，これを Scene Vector 特徴量として文章の予測に活用する手法を提案した．

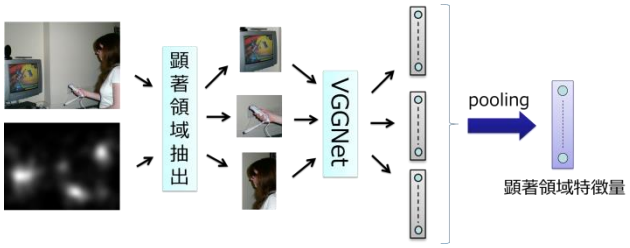


図 2 顕著性マップを用いた特徴量抽出

3. 提案手法

3.1 モデルの概要

図1に提案する説明文生成モデルの概要を示す. これは[Jin 2015]の Scene Vector を用いる手法に, 顕著性マップから作成する特徴量を追加したものとなっている. このモデルは画像や単語を特徴量に変換する符号化部分と特徴量から単語を予測する復号化部分に分けられる.

符号化部分では, 画像やテキストの情報を復号化部分へ入力するために特徴量の抽出や加工を行う. 画像については, 一般物体認識(ILSVRC)のタスクで事前学習された CNN である VGGNet [Simonyan 2015] を用いて画像特徴量の抽出と Scene Vector の予測, そして顕著性マップを用いて顕著領域の特徴量の抽出を行う. 顕著領域の特徴量は 3.2 節で, SceneVector は 3.3 節で詳細を述べる. 単語については, ある1単語が与えられた時, その単語を表す One-hot Vector を作成する. 各特徴量は, 復号化部分へ入力する前に, Embedding を行い特徴量の次元数を統一する.

復号化部分では, 3.4 節で述べる Long Short-Term Memory (LSTM) を用いて単語の予測を行う.

3.2 顕著領域の特徴量作成

顕著領域の特徴量の作成手順を図2に示す. 顕著性マップの情報を用いて画像内の顕著な領域を抽出し, そこから得られる画像特徴量を加工して, 顕著領域の特徴量とする.

顕著領域の抽出手順を図3に示す. はじめに対象とする画像から顕著性マップを計算する (図3(b)). 次に, 顕著性が τ 以上のピクセルが 1, それ以外のピクセルが 0 の二値画像を作成する(図3(c)). 次に, 二値画像の白色部分を囲う最小の領域を計算する(図3(d)). 最後に, 算出された領域を縦横 λ ピクセルずつ拡大して, 最終的な顕著領域とする(図3(e)). パラメータとなる τ, λ は様々な組み合わせを検証し, τ =0.4, λ =40 とした.

顕著領域の抽出後, 顕著領域特徴量の作成を行う. 各顕著領域毎に VGGNet を用いて特徴量の抽出を行い, それらを一つの特徴量にまとめたものを顕著領域の特徴量とする. 特徴量のまとめかたとして, 各特徴量の平均を取る average pooling (式(1))と, 各特徴量の各次元の最大値を取る max pooling (式(2)) の二つを提案する.

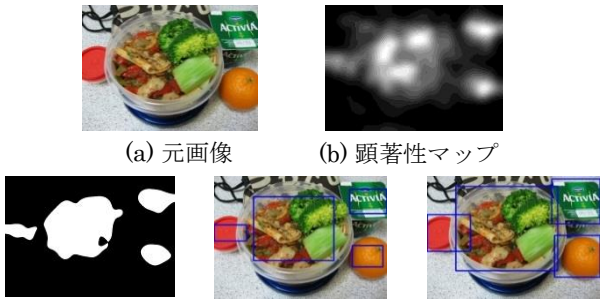
$$sa_{ave} = \frac{1}{N}\sum_{i=1}^{N} r_i \qquad (1)$$

$$sa_{\max(j)} = \max_{i \in R} \mathrm{r}_{i(j)} \qquad (2)$$

ここで, R は顕著領域の集合, N は顕著領域の総数, r_iは i 番目の顕著領域の特徴量, $\mathrm{r}_{i(j)}$は i 番目の顕著領域の特徴量の j 次元目の値を表す.

3.3 Scene Vector

この手法は[Jin 2015]で提案されたもので, 画像がどのような場面で撮影されたかを Scene Vector として表現し, それを説明文生成に用いる.



(a) 元画像　(b) 顕著性マップ

(c) 二値画像　(d) 領域抽出　(e) 領域の拡大

図 3 顕著領域の抽出手順

[Jin 2015]らは, 訓練データの画像に付与されている説明文を用いて Latent Dirichlet Allocation (LDA) を行い, トピック分布の予測モデルを学習した. その後, 説明文のトピック分布を画像特徴量から予測する隠れ層が 2 層の Multi Layer Perceptron (MLP)の訓練を行った. この MLP の出力を Scene Vector として復号化処理への入力とした. 画像特徴量の抽出には, 画像の撮影場所を当てるタスク(Places-205)で事前訓練された GoogLeNet [Szegedy 2015] を用いた.

上記の手順を再現し, トピック分布の中で最大の値の次元ごとにクラスタリングして可視化したところ, 複数の場面の画像が混在していた. そこで, Scene Vector の作成方法を以下の通りに改変した.

訓練データの画像から Places-205 で事前訓練された GoogLeNet を用いて画像毎に特徴量を抽出し, それらを用いて Kmeans 法によるクラスタリングを行う. そして, クラスタ id に対応する次元が 1, それ以外の次元が 0 となる one-hot vector を作成する. これを正解データとして MLP の訓練を行う.

3.4 LSTM

本研究では先行研究をもとに2層の LSTM を用いる[Jin 2015]. 1層目の LSTM を式(4), (5), (6), (7), (8), 2層目の LSTM を(9), (10), (11), (12), (13)に示す.

先行研究とは時刻毎に入力する特徴量が異なる. 時刻 t が 0 の場合, 1層目の LSTM へ画像全体の画像特徴量と顕著領域の特徴量を入力する. これにより LSTM の隠れ層の値が更新され, 次回以降にこの画像の情報を考慮した説明文の単語が順に出力される.

時刻 t が 1 以上の場合, 1層目の LSTM へ Scene Vector と直前に予測された単語の特徴量を入力する. 時刻 t=1 の場合は文章の開始を表す記号”<S>”を与える. これにより画像の場面を考慮しつつ次に出現する単語を予測する. この処理は文章の終了を表す記号”<E>”が出現するまで繰り返される.

$$i_t^{(1)} = \begin{cases} \sigma(\mathrm{W}_{imi^{(1)}} v_{im} + \mathrm{W}_{sai^{(1)}} v_{sa}) \ (t=0) \\ \sigma(\mathrm{W}_{sei^{(1)}} v_{se} + \mathrm{W}_{w_{t-1}i^{(1)}} v_{w_{t-1}} + \mathrm{W}_{h^{(1)}i^{(1)}} {h_{t-1}}^{(1)}) \ (t>1) \end{cases} \qquad (4)$$

$$f_t^{(1)} = \begin{cases} \sigma(\mathrm{W}_{imf^{(1)}} v_{im} + \mathrm{W}_{saf^{(1)}} v_{sa}) \ (t=0) \\ \sigma(\mathrm{W}_{sef^{(1)}} v_{se} + \mathrm{W}_{w_{t-1}f^{(1)}} v_{w_{t-1}} + \mathrm{W}_{h^{(1)}f^{(1)}} {h_{t-1}}^{(1)}) \ (t>1) \end{cases} \qquad (5)$$

$$c_t^{(1)} = \begin{cases} i_t^{(1)} \tanh(\mathrm{W}_{imc^{(1)}} v_{im} + \mathrm{W}_{sac^{(1)}} v_{sa}) (t=0) \\ f_t^{(1)} c_{t-1}^{(1)} + i_t^{(1)} \tanh(\mathrm{W}_{imc^{(1)}} v_{im} + \mathrm{W}_{sac^{(1)}} v_{sa} + \mathrm{W}_{h^{(1)}c^{(1)}} {h_{t-1}}^{(1)}) (t>1) \end{cases} \qquad (6)$$

$$o_t^{(1)} = \begin{cases} \sigma(\mathrm{W}_{imo^{(1)}} v_{im} + \mathrm{W}_{sao^{(1)}} v_{sa}) \ (t=0) \\ \sigma(\mathrm{W}_{seo^{(1)}} v_{se} + \mathrm{W}_{w_{t-1}o^{(1)}} v_{w_{t-1}} + \mathrm{W}_{h^{(1)}o^{(1)}} {h_{t-1}}^{(1)}) \ (t>1) \end{cases} \qquad (7)$$

$$h_t^{(1)} = o_t^{(1)} \tanh(c_t^{(1)}) \qquad (8)$$

$$i_t^{(2)} = \sigma(\mathrm{W}_{h^{(1)}i^{(2)}} h_t^{(1)} + \mathrm{W}_{h^{(2)}i^{(2)}} h_{t-1}^{(2)}) \quad (9)$$

$$f_t^{(2)} = \sigma(\mathrm{W}_{h^{(1)}f^{(2)}} h_t^{(1)} + \mathrm{W}_{h^{(2)}f^{(2)}} h_{t-1}^{(2)}) \quad (10)$$

$$c_t^{(2)} = f_t^{(2)} c_{t-1}^{(2)} + i_t^{(2)} \tanh(\mathrm{W}_{h^{(1)}c^{(2)}} h_t^{(1)} + \mathrm{W}_{h^{(2)}c^{(2)}} h_{t-1}^{(2)}) \quad (11)$$

$$o_t^{(2)} = \sigma(\mathrm{W}_{h^{(1)}o^{(2)}} h_t^{(1)} + \mathrm{W}_{h^{(2)}o^{(2)}} h_{t-1}^{(2)}) \quad (12)$$

$$h_t^{(2)} = o_t^{(2)} \tanh(c_t^{(2)}) \quad (13)$$

ここで, W_{xx}は LSTM の変換行列を, $v_{im}, v_{sa}, v_{se}, v_w$はそれぞれ, 画像特徴量, 顕著領域特徴量, 画像の場面の特徴量, 単語特徴量を, を表す. また, 各文字の右上は LSTM の層を, 右下は計算時の時刻を表す. σ, tanhはそれぞれシグモイド関数, ハイパボリックタンジェント関数を表す. LSTM で扱う特徴量の次元数は 512 とした.

3.5 単語の予測

2 層目の LSTM の出力を Embedding し, その値を Softmax 関数へ入力することで, その時刻での各単語の出現確率が算出される. これは式(14), (15)のように定義できる.

$$v_{out} = \mathrm{W}_{h^{(2)}} h_t^{(2)} \quad (14)$$

$$p_{(w_i)} = \frac{\exp(v_{out\,(i)})}{\sum_i^k \exp(v_{out\,(j)})} \; i = 1, \ldots, k \quad (15)$$

このとき, $p_{(w_i)}$は時刻 t での単語w_iの出現確率を表し, vはk次元のベクトルで$v_{(i)}$はベクトルvのi次元目の値を表す. kはこのモデルで用いる総単語数(辞書サイズ)となる.

3.6 モデルの訓練

画像の説明文生成モデルの目的関数は下記のとおりである [Jin 2015]. ある画像を与えた時に, その説明文の単語の出現確率の対数尤度を最大化する.

$$\frac{1}{N} \sum_{n \in S} \sum_{t=1}^{L_n} \log p\left(w_t^{(n)} \middle| w_{0:t-1}^{(n)}, I^{(n)}\right) \quad (16)$$

ここで, I は訓練を行う画像を, S は画像 I に付与されている説明文の集合を, $w_{0:t-1}^{(n)}$は t 番目の単語$w_t^{(n)}$より以前に出現した単語を, L_nは n 番目の説明文の総単語数を表す.

訓練の対象となるパラメータは各 Embedding の埋め込み行列と, LSTM の変換行列である.

4. 実験

4.1 実装

本手法の実装には Chainer [Tokui 2015] を用いた.

モデルの訓練には, 最適化アルゴリズムとして ADAM [Kingma 2014] を利用した. この時, ハイパーパラメータは α=0.00005, β_1=0.9, β_2=0.999 とした. また, ミニバッチ学習を行い, バッチサイズは 25 とした.

4.2 データセット

実験には Microsoft COCO [Lin 2014] (MSCOCO)のデータを用いる. MSCOCO では画像の説明文の自動生成タスクの精度を競うコンペティションが開催されており, そのためのデータセットが公開されている. ジャンルに依らない一般的な画像 1 枚あたりに 5 つの説明文が人手で付与されている. 訓練画像として 82,783 枚, 評価画像として 40,203 枚が用意されている.

顕著性マップの抽出には SALICON [Jiang 2015]データセットを使用する. これは MSCOCO のサブセットとして, 画像の顕著性マップを生成する精度を競うコンペティションのために用意されたデータセットであり, 画像に人手で顕著性マップが付与されている. 合計で 15,000 枚の画像が用意されている.

4.3 ベースライン手法

実験では SALICON データセットを用いるが, SALICON データセットで実験を行っている既存研究は見つかっていない. そのため, 先行研究を参考にベースライン手法を実装し, 結果の比較を行った.

ベースライン手法には[Jin 2015]の手法を用いた. ただし, 今回は Visual Attention モデルは未実装であり, また Scene Vector の作成方法を 3.3 節のものに改変した. 図1の黄色の枠がベースライン手法に対応する.

4.4 実験方法

(1) 実験 1:他の手法との比較実験

実装したベースライン手法と他のモデルとの比較実験を行った. この実験は, MSCOCO のコンペティションで行われているものと同様の条件で他のモデルと精度を比較し, ベースライン手法が十分な精度であることを確認することを目的とする.

この実験では MSCOCO のデータを用いてモデルの訓練と評価を行い, 顕著性マップの情報は用いない.

(2) 実験 2:顕著性情報の有用性検証実験

画像の顕著性情報の有用性を検証する実験を行った. この実験では顕著性の情報を用いた手法の限界を検証するために, 顕著性マップには SALICON データセットで提供されるオラクルを与えた. ベースライン手法と, ベースライン手法に顕著性の情報を組み込んだ提案手法を比較することで顕著性の情報の有用性を検証した.

この実験では SALICON データセットの 15,000 枚の画像wp用いて 5 分割交差検定(訓練画像 12,000 枚, 評価画像 3,000 枚)を行った.

4.5 実験条件

実験の前処理として, 説明文の整形と低頻出単語の削除を行った. 説明文の整形では, MSCOCO の訓練画像に付与されている説明文の全ての単語を小文字に変換し, 文末のピリオドの削除を行った. 低頻出語の削除では説明文出現する単語数を数え, 出現回数が 10 回以下の単語は”<unk>”という記号に変換した. また, 各説明文の文頭と文末に, 文章の始まりと終わりを表す”<S>”,”<E>”という特殊な記号を追加した. 上記の処理により最終的に, 7,229 単語の辞書が作成され, これを用いて実験を行った.

評価指標には, MSCOCO の評価指標として定義されている [Chen 2015] BLEU-1, BLEU-2, BLEU-3, BLEU-4, METEOR, ROUGE-L, CIDEr を用いた. いずれも, 予測された説明文が正解の説明文とどれだけ一致しているかを評価する指標となっている. 評価指標を計算するために, MSCOCO から公開されている Python evaluation API [Chen 2015]を用いた.

4.6 実験結果

他の手法との比較実験の結果を表1に, 顕著性の有無の比較実験結果を表2に示す.

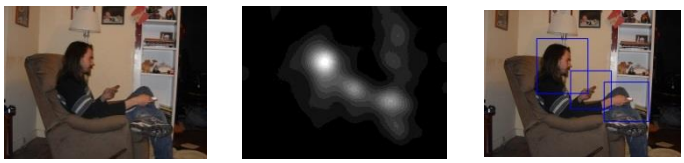
表1より, 今回実装したモデルは [Karpathy 2015]よりは高く, [Jin 2015]よりは低い精度となった. 表2より, 顕著領域の特徴量を使用した場合, いずれの評価指標でもそれを用いなかった場合より精度が向上することが確認できた. average pooling, max pooling はほぼ同精度となった.

表1 他の手法との比較実験結果

手法名	B-1	B-2	B-3	B-4	M	R	C
[Karpathy 2015]	0.651	0.464	0.321	0.224	0.211	0.475	0.697
[Jin 2015]	**0.697**	**0.519**	**0.381**	**0.282**	**0.235**	**0.509**	**0.838**
ベースライン手法	0.660	0.479	0.348	0.259	0.225	0.490	0.791

表2 顕著性の有用性検証実験結果

手法名	B-1	B-2	B-3	B-4	M	R	C
ベースライン手法	0.648	0.465	0.327	0.231	0.212	0.526	0.634
提案手法 (average pooling)	0.662	**0.484**	**0.344**	**0.245**	**0.221**	**0.536**	**0.688**
提案手法 (max pooling)	**0.663**	0.483	0.343	**0.245**	**0.221**	**0.536**	0.681



正解の説明文
a man in a chair holding a wii remote
a person that is playing a video game
a bearded man is sitting in a chair with a controller
the man is sitting in his brown recliner
a long haired man is sitting in a recliner playing video games

ベースライン手法
a man sitting on a couch with a laptop

提案手法 (average pooling)
a man is holding a wii remote while sitting on a couch

図４ 予測結果の具体例

5. 考察

5.1 実験 1:他の手法との比較実験

[Jin 2015]より低い精度であるのは，ベースライン手法で[Jin 2015]で用いられている Visual Attention モデルが未実装なためであると考える．[Karpathy 2015]の精度を上回っていることと，[Jin 2015]に近い精度が出ていることから，ベースラインのシステムとして十分な精度であると考える．

5.2 実験 2:顕著性情報の有無の比較実験

表1，表2より，実験1と比較して顕著性の情報がない提案モデルの精度が低いが，これは実験2で用いた訓練の枚数が少ないことに起因すると考える．

表2より，提案手法は全ての指標においてベースライン手法の精度を上回った．提案手法による画像説明文の具体例を図4に示す．図4の例は椅子に座ってゲームをする人の画像を入力しており，正解の説明文の多くがコントローラについて言及している．提案手法では手に持っているコントローラについて言及する説明文が生成できたが，既存手法ではできなかった．このような結果になったのは，顕著性の高い領域を抽出することで，ゲームのコントローラの情報も網羅出来るようになったためであると考える．このように，ベースライン手法では網羅できなかった顕著性の高いオブジェクトの情報をより多く取り入れられるようになったことが提案手法の精度が向上した大きな理由であると考える．

6. おわりに

本研究では，画像の説明文の自動生成の精度向上と画像の顕著性情報の有用性の検証を目的として，画像の顕著性を用いた画像説明文生成モデルを提案した．実験の結果，画像の顕著性は説明文の生成に有用であることが示唆された．

今後は，画像から説明文の生成のために効果的な顕著性マップを推定するモデルを組み込むことと，単語と画像内の小領域の対応を取る Visual Attention モデルを顕著性マップで拡張し，精度を向上させることを目指す．

謝辞：本研究の一部は JSPS 科研費基盤(B)26280038 の助成を受けた

参考文献

[Lin 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona and Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár, “Microsoft COCO: Common Objects in Context”, 2014.

[Vinyals 2015] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan, “Show and Tell: Neural Image Caption Generator”, Conference on Computer Vision and Pattern Recognition, 2015.

[Karpathy 2015] Andrej Karpathy and Li Fei-Fei, “Deep Visual-Semantic Alignments for Generating Image Descriptions”, Conference on Computer Vision and Pattern Recognition, 2015.

[Xu 2015] Kelvin Xu, Jimmy Ley Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S.Zemel, Yoshua Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”, Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015.

[Jin 2015] Junqi Jin, Kun Fu, Runpeng Cui, Fei Sha and Changshui Zhang, “Aligning where to see and what to tell: image caption with region-based attention and scene factorization”, arXiv, 2015.

[Farhadi 2010] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier and David Forsyth, “Every Picture Tells a Story: Generating Sentences from Images”, European Conference on Computer Vision, 2010.

[Simonyan 2015] Karen Simonyan and Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, International Conference on Learning Representations, 2015.

[Szegedy 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich, “Going Deeper with Convolutions”, Conference on Computer Vision and Pattern Recognition, 2015.

[Jiang 2015] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao, “SALICON: Saliency in Context”, Conference on Computer Vision and Pattern Recognition, 2015.

[Kingma 2014] Diederik Kingma and Jimmy Ba, “Adam: A Method for Stochastic Optimization”, arXiv, 2014.

[Chen 2015] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta and Piotr Dollar, C. Lawrence Zitnick, “Microsoft COCO Captions: Data Collection and Evaluation Server”, arXiv, 2015.

[Tokui 2015] Seiya Tokui, Kenta Oono, Shohei Hido and Justin Clayton, “Chainer: a Next-Generation Open Source Framework for Deep Learning”, Workshop on Machine Learning Systems in The Twenty-ninth Annual Conference on Neural Information Processing Systems, 2015.