画像キャプション生成における JPEG 圧縮への頑健性の改善

遠藤洸亮　Zhishen Yang　岡崎直観
東京工業大学 情報理工学院
{kosuke.endo@nlp., zhishen.yang@nlp., okazaki@}c.titech.ac.jp

概要

本稿は画像キャプション生成タスクにおける JPEG 圧縮の影響を分析する。まず、高い圧縮率が適用された画像に対して、通常の画像キャプション生成モデルはその性能を維持できず、生成されるキャプションの品質が落ちてしまうことを示す。そこで、画像エンコーダと画像キャプション生成の二つのモデルの学習データに JPEG 画像を追加する手法を提案する。実験結果から、画像キャプション生成モデルの学習に JPEG 画像を追加しなくても、画像エンコーダの学習に JPEG 画像を追加するだけで、JPEG 圧縮に対して頑健な画像キャプション生成モデルを構築できることが分かった。

1　はじめに

画像キャプション生成は、画像を説明する文章を生成することを目的とした、自然言語処理とコンピュータビジョンの両分野にまたがるマルチモーダルなタスクである。近年の画像分類モデルと言語生成モデルの発展により、文章として自然で、画像の特徴をよく捉えた文章を生成できるようになった。ただ、現在の画像キャプション生成モデルは高品質な画像でのみ学習され、評価されることが多い [1, 2, 3, 4, 5]。

しかし、インターネット上の画像を取り扱う場合など、キャプション生成モデルを利用するときに与えられる画像が高品質であるとは限らない。高品質な画像はファイルサイズが大きくなるため、保存や転送により多くのコストがかかる。そのため、非可逆圧縮である JPEG [6] など、何らかのアルゴリズムで圧縮が行われ、画像の品質が劣化することが多い [7, 8, 9, 10]。

ImageNet [11] や COCO [12] といったデータセットに収録されている画像も JPEG で圧縮されているが、高品質な画像が多く、低品質な画像は少ない。JPEG 圧縮による画質の低下を PSNR と SSIM [13, 14] を用いて計測したものを図 1 と 2 に示す（数値データは付録の表 1 に掲載した）。このような JPEG 圧縮によって品質が低下した画像からのキャプション生成はあまり考えられてこなかった。

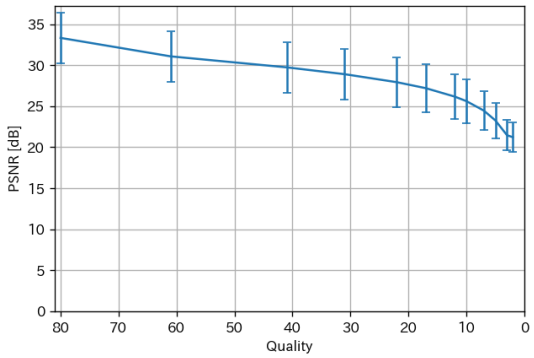

図 1　COCO の評価データの画像の JPEG 圧縮に対する PSNR の推移（Quality が小さいほど高圧縮・低品質である）

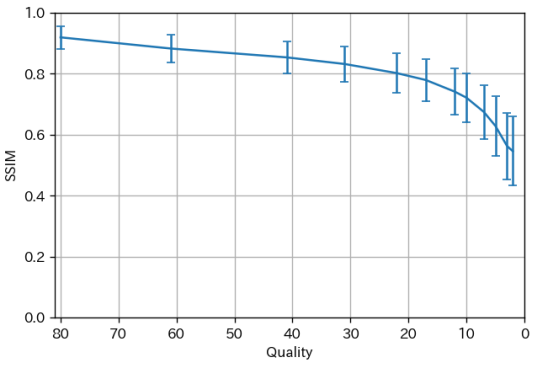

図 2　COCO の評価データの画像の JPEG 圧縮に対する SSIM の推移

本稿では、画像キャプション生成モデルにおける JPEG 圧縮の影響を定量的に分析する。また、画像キャプション生成モデルの頑健性を向上させるため、画像エンコーダと画像キャプション生成の二つのモデルの学習時に JPEG で圧縮された画像を追加する手法を提案する。

実験結果から、高圧縮率の JPEG 画像に対して

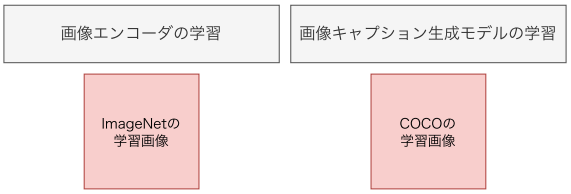

図 3　ベースラインの学習データセット

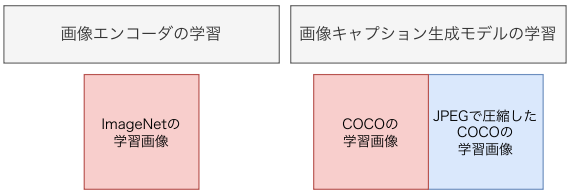

図 4　画像キャプション生成モデルの学習に JPEG 画像を追加する方法

ベースラインの画像キャプション生成モデルが十分な性能を発揮できないことが分かった。また、提案手法は高品質な画像に対して生成されたキャプションの BLEU スコアを維持しつつ、JPEG 圧縮で品質が劣化した画像に対して BLEU スコアを向上させることが確認された。画像エンコーダの学習に JPEG で圧縮された画像を追加する方法、画像キャプション生成モデルの学習に JPEG で圧縮された画像を追加する方法、両方のモデルの学習に JPEG で圧縮された画像を追加する方法の三つを比較したところ、画像エンコーダの学習に JPEG 画像を追加する方法と両方のモデルの学習に JPEG 画像を追加する方法は同程度の頑健性であり、画像キャプション生成モデルの学習に JPEG 画像を追加する方法は他の二つの方法より頑健性が低かった。このことから、画像エンコーダの学習にのみ JPEG で圧縮された画像を追加するだけでも、JPEG 圧縮に対する頑健性を持つ画像キャプション生成モデルを構築できることが分かった。

2　提案手法

通常の画像キャプション生成モデルは図 3 のように、ImageNet と COCO の高品質な学習画像からモデルを学習している。このため、高い圧縮率で品質が劣化した JPEG 画像に対しては、画像エンコーダの特徴量抽出が十分に行えなかったり、学習時の状況からの差が大きすぎるため、画像キャプション生成モデルが想定通りに動作しない可能性がある。そこで、画像エンコーダと画像キャプション生成モデルの二つの片方、もしくは両方の学習時に JPEG で圧縮された画像を追加し、JPEG 圧縮に対する頑健性を高めることを検討する。

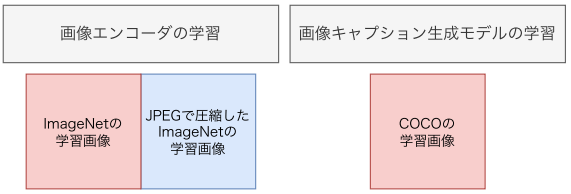

図 5　画像エンコーダの学習に JPEG 画像を追加する方法

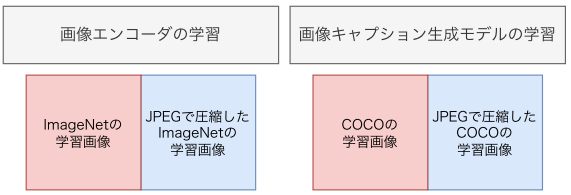

図 6　両方の学習に JPEG 画像を追加する方法

本稿で比較するのは、以下の三つの方法である。

1. 画像キャプション生成モデルの学習のために用いる COCO の画像を JPEG で圧縮し、学習データに追加する方法（図 4）
2. 画像エンコーダの学習のために用いる ImageNet の画像を JPEG で圧縮し、学習データに追加する方法（図 5）
3. ImageNet と COCO の両方の画像を JPEG で圧縮し、画像エンコーダと画像キャプション生成モデルの学習データとして、それぞれ追加する方法（図 6）

これらの方法では決められた圧縮率で学習データの全ての画像を圧縮し、学習データに追加する。これにより各モデルの学習に用いる事例の数が 2 倍になる。

3　実験

3.1　実験設定

画像分類　本研究では画像キャプション生成モデルの画像エンコーダを画像分類タスクで学習する。具体的には ImageNet [11] の画像分類タスクを用いて ResNet-18 [15] を学習する。実装には PyTorch のサンプルコード[1)]を利用した。ResNet-18 は深さが 18 層の画像分類モデルである。残差結合を用いることでより深い層を持つニューラルネットワークの学習を効率よく行えるようになっている。ImageNet は約 140 万件の画像からなり、画像分類において物体の画像を 1,000 個のカテゴリに分類する。本研究では評価のために開発データを二分割し、1,281,167

1）https://github.com/pytorch/examples

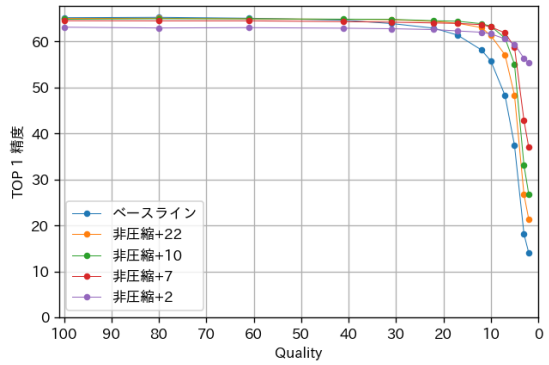

図 7 ベースラインと JPEG 画像を学習データに追加した画像分類モデルの JPEG 圧縮された評価データに対する TOP1 精度 quality=100 は元々の評価データを表す。

件の学習データとそれぞれ 25,000 件の開発データ、評価データを用いる（付録 C に分割の方法を記す）。

画像キャプション生成 PyTorch の実装[2)]を用い、MSCOCO '14 データセット [12] 上で Show, Attend and Tell [16] のモデルを学習する。Show, Attend and Tell は画像エンコーダと LSTM [17] デコーダからなる画像キャプション生成モデルである。画像に対する注意機構を導入し、重要な物体へ注意を向けながら単語を生成することで、画像の内容を捉えたキャプションを生成できる。COCO は約 16.4 万枚の画像と、それに付けられたキャプションの組からなる。本研究では評価のために Karpathy らの先行研究 [18] で用いられたデータセットの分割方法[3)]を用いる。これにより、113,287 件の学習画像とそれぞれ 5,000 件の開発画像、評価画像に分割する。生成されたキャプションの評価には BLEU-4 [19] を用いる。

画像分類モデルと画像キャプション生成モデルの 2 つの学習で、それぞれの開発データ上で最も性能が高いモデルを評価対象とした。

JPEG 圧縮 JPEG 圧縮には pillow ライブラリ[4)]を用いる。図 2 より quality が 22 より小さくなると SSIM が急激に降下を始めるため、追加する画像はデータセットの学習画像を 22, 10 ,7 ,2 の quality で JPEG 圧縮したのち、さらに 75 の quality で圧縮した画像である[5)]。

2) https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning
3) https://cs.stanford.edu/people/karpathy/deepimagesent/
4) https://pillow.readthedocs.io/en/stable/handbook/image-file-formats.html#jpeg
5) ライブラリの内部動作により、画像を指定された quality で JPEG 圧縮したのち、ファイルに書き出す際にさらに 75 の quality で JPEG 圧縮がかかっている。

評価に用いる画像 ImageNet, COCO 共に、元々の評価画像を 80, 61, 41, 31, 22, 17, 12, 10, 7, 5, 3, 2 の quality で JPEG 圧縮した画像を用いて評価を行う。

3.2 実験結果

画像分類 画像分類タスクの TOP1 精度を図 7 に示す。図中の非圧縮+[数字] は、ImageNet の画像を数字で示される quality で JPEG 圧縮し、その JPEG 画像をさらに 75 の quality で JPEG 圧縮した画像を学習データに追加したモデルを意味する。数値データは付録の表 2 に掲載した。Ehrlich ら [7] が示したように、高圧縮の JPEG 画像に対してベースラインモデルの TOP1 精度が低下する。また、JPEG で圧縮した画像を学習に追加したモデルは高品質な評価データに対する TOP1 精度を維持しつつ、quality が 17 未満の高圧縮の評価データ上でベースラインモデルの TOP1 精度を超える精度を記録した。そして、追加している JPEG 画像の圧縮率が高い程、高圧縮の評価データ上で TOP1 精度が高かった。この実験結果から、JPEG 画像を学習データに追加することにより、画像分類モデルの JPEG 圧縮に対する頑健性を強化できることが示された。

画像キャプション生成 画像キャプション生成タスクの BLEU スコアを図 8,9,10,11 に示す。数値データは付録の表 3 に掲載した。高圧縮の JPEG 画像に対して、ベースラインモデルの BLEU スコアが低下する。また、提案手法のどのモデルも高品質な評価データに対してベースラインモデルに匹敵する BLEU スコアを記録しつつ、quality が 17 より小さい高圧縮の評価データ上では、ベースラインモデルを超える BLEU スコアを記録した。

さらに画像エンコーダの学習にのみ JPEG 画像を追加する方法と画像エンコーダと画像キャプション生成モデルの二つの学習に JPEG 画像を追加する方法は、同様の性能を示している。一方、画像キャプション生成モデルの学習にのみ JPEG 画像を追加する方法はそれらを下回る性能であった。

4 関連研究

Bujimalla ら [20] は写真の品質の劣化原因としてモーションブラー（被写体ぶれ）を挙げ、画像キャプション生成タスクへの影響を調査した。そしてモーションブラーで学習データをデータ拡張することによりモーションブラーに対して頑健な画像キャ

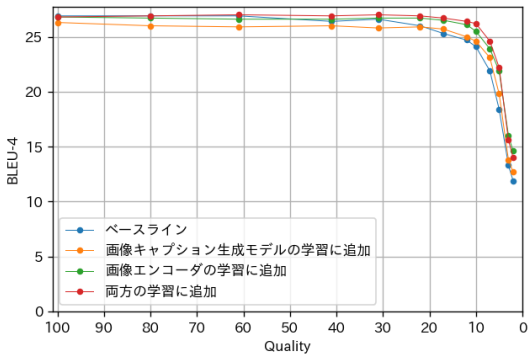

図 8　ベースラインと quality=22 で圧縮した画像をさらに quality=75 で圧縮した画像を学習に追加したモデルの JPEG 圧縮された評価データに対する BLEU-4

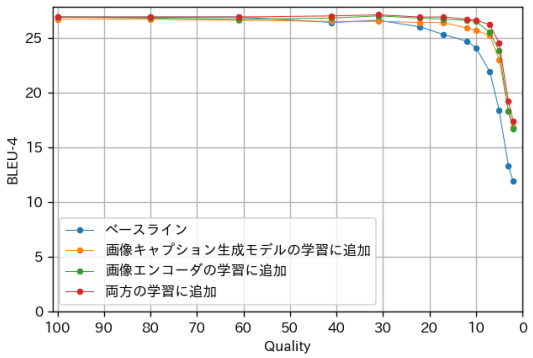

図 9　ベースラインと quality=10 で圧縮した画像をさらに quality=75 で圧縮した画像を学習に追加したモデルの JPEG 圧縮された評価データに対する BLEU-4

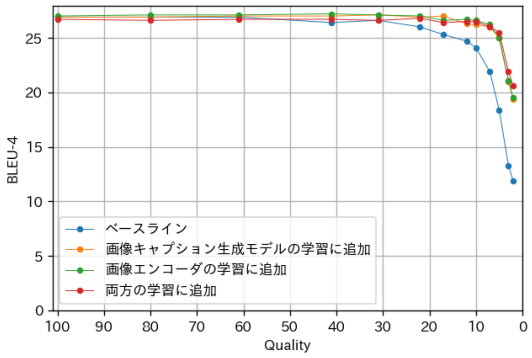

図 10　ベースラインと quality=7 で圧縮した画像をさらに quality=75 で圧縮した画像を学習に追加したモデルの JPEG 圧縮された評価データに対する BLEU-4

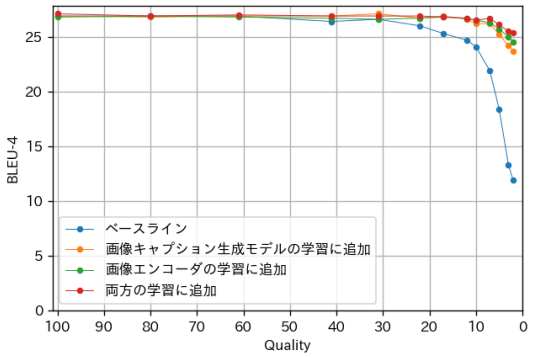

図 11　ベースラインと quality=2 で圧縮した画像をさらに quality=75 で圧縮した画像を学習に追加したモデルの JPEG 圧縮された評価データに対する BLEU-4

プション生成モデルの構築ができることを示した。しかし、JPEG 圧縮の影響は分析していない。

Ehrlich ら [7] は JPEG 圧縮の画像分類、物体検出・インスタンスセグメンテーション、セマンティックセグメンテーションの各タスクへの影響を分析し、高い圧縮率が適用された JPEG 画像に対してタスクの性能が低下することを示した。しかし、画像キャプション生成モデルへの影響に関して定量的な評価を行なっていない。

5　おわりに

本稿では、画像キャプション生成モデルへの JPEG 圧縮の影響を定量的に評価し、その影響を緩和する手法を提案した。具体的には、画像エンコーダと画像キャプション生成モデルの 2 つのモデルの学習に JPEG 画像を追加することで、JPEG 圧縮への頑健性の向上を目指した。実験結果から、高圧縮の JPEG 画像に対してベースラインモデルが生成するキャプションの品質が低下すること、提案手法によって学習された画像キャプション生成モデルが高品質な画像に対する性能を維持しつつ、高圧縮の JPEG 画像に対してベースラインよりも高品質なキャプションを生成できることが分かった。また、画像エンコーダと画像キャプション生成モデルの二つの片方、もしくは両方の学習時に JPEG 画像を追加する方法の三つを比較したところ、画像エンコーダの学習にのみ JPEG 画像を追加する方法と両方のモデルの学習に JPEG 画像を追加する方法は同程度の頑健性であり、画像キャプション生成モデルの学習にのみ JPEG 画像を追加する方法は他の二つの方法より頑健性が低かった。このことは JPEG 圧縮に対する頑健性を持った画像キャプション生成モデルの構築には、画像エンコーダの学習に JPEG 画像を追加することで十分であることを示唆している。

今後は JPEG 圧縮に対して頑健な画像エンコーダを視覚的質問応答や参照表現理解など他のダウンストリームタスクに適用し，JPEG 圧縮に対するモデルの頑健性を調査したい。

謝辞

本研究を進めるにあたり、東京工業大学大学院情報理工学院の Marco Cognetta 氏に、研究の方向性やデータの分析において日頃から貴重なご助言を頂きました。深く感謝いたします。

本研究成果は，国立研究開発法人情報通信研究機構（NICT）の委託研究「自動翻訳の精度向上のためのマルチモーダル情報の外部制御可能なモデリングの研究開発」（課題 225）により得られたものです.

参考文献

[1] Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, Hehong Chen, Guohai Xu, Zheng Cao, et al. mplug: Effective and efficient vision-language learning by cross-modal skip-connections. **arXiv preprint arXiv:2205.12005**, 2022.

[2] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. **CoRR**, Vol. abs/2202.03052, , 2022.

[3] Jia Cheng Hu, Roberto Cavicchioli, and Alessandro Capotondi. Expansionnet v2: Block static expansion in fast end to end training for image captioning. **arXiv preprint arXiv:2208.06551**, 2022.

[4] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pretraining for image captioning. In **2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, pp. 17959–17968, 2022.

[5] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali: A jointly-scaled multilingual language-image model, 2022.

[6] G.K. Wallace. The jpeg still picture compression standard. **IEEE Transactions on Consumer Electronics**, Vol. 38, No. 1, pp. xviii–xxxiv, 1992.

[7] Max Ehrlich, Larry Davis, Ser-Nam Lim, and Abhinav Shrivastava. Analyzing and mitigating jpeg compression defects in deep learning. In **2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)**, pp. 2357–2367, 2021.

[8] Samuel Felipe dos Santos, Nicu Sebe, and Jurandy Almeida. The good, the bad, and the ugly: Neural networks straight from jpeg. In **2020 IEEE International Conference on Image Processing (ICIP)**, pp. 1896–1900, 2020.

[9] Miklós Póth and Zeljen Trpovski. Analysis of jpeg digital image compression process. 2019.

[10] Xi Wang, Xueyang Fu, Yurui Zhu, and Zheng-Jun Zha. Jpeg artifacts removal via contrastive representation learning. In **European Conference on Computer Vision**, pp. 615–631. Springer, 2022.

[11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, Vol. 115, No. 3, pp. 211–252, 2015.

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft coco: Common objects in context. In **ECCV**, pp. 740–755. European Conference on Computer Vision, September 2014.

[13] Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In **2010 20th International Conference on Pattern Recognition**, pp. 2366–2369, 2010.

[14] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. **IEEE Transactions on Image Processing**, Vol. 13, No. 4, pp. 600–612, 2004.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, pp. 770–778, 2016.

[16] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. **arXiv preprint arXiv:1502.03044**, 2015.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. **Neural Computation**, Vol. 9, No. 8, pp. 1735–1780, 11 1997.

[18] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions, 2014.

[19] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In **Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics**, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[20] Shashank Bujimalla, Mahesh Subedar, and Omesh Tickoo. Data augmentation to improve robustness of image captioning solutions, 2021.

表 1　COCO の評価データの画像の JPEG 圧縮に対する PSNR と SSIM の推移

	Quality											
	80	61	41	31	22	17	12	10	7	5	3	2
PSNR [dB]	33.33±3.08	31.07±3.06	29.74±3.11	28.90±3.09	27.93±3.01	27.20±2.93	26.17±2.76	25.62±2.65	24.47±2.40	23.27±2.15	21.47±1.82	21.22±1.83
SSIM	0.92±0.04	0.88±0.05	0.85±0.05	0.83±0.06	0.80±0.06	0.78±0.07	0.74±0.08	0.72±0.08	0.67±0.09	0.63±0.10	0.56±0.11	0.55±0.11

表 2　ベースラインと ImageNet の学習データに JPEG 画像を追加したモデルの非圧縮の評価データと JPEG 圧縮された評価データに対する画像分類タスクの TOP1 精度

学習データの組成		Quality										
ImageNet	非圧縮	80	61	41	31	22	17	12	10	7	5	3	2
非圧縮	65.19	65.26	65.07	64.66	63.92	62.91	61.40	58.19	55.80	48.28	37.44	18.12	14.04
非圧縮+22	65.02	65.06	64.93	64.83	64.75	64.39	63.98	63.05	61.30	57.10	48.34	26.71	21.39
非圧縮+10	64.85	64.95	64.94	64.86	64.82	64.52	64.42	63.89	63.34	60.76	55.08	33.06	26.82
非圧縮+7	64.53	64.50	64.50	64.34	64.22	64.05	63.95	63.68	63.18	61.88	58.80	42.83	37.02
非圧縮+2	63.12	62.95	63.02	62.90	62.79	62.61	62.30	62.02	61.67	60.59	59.41	56.34	55.39

表 3　ベースラインと ImageNet と COCO の学習データに JPEG 画像を追加した画像キャプション生成モデルの JPEG 圧縮された評価データに対する BLEU-4

学習データの組成			Quality										
ImageNet	COCO	非圧縮	80	61	41	31	22	17	12	10	7	5	3	2
非圧縮	非圧縮	26.9	26.9	26.9	26.4	26.6	26.0	25.3	24.7	24.1	21.9	18.4	13.3	11.9
非圧縮	非圧縮+22	26.3	26.0	25.9	26.0	25.8	25.9	25.7	25.0	24.6	23.1	19.8	13.8	12.7
非圧縮	非圧縮+10	26.7	26.7	26.6	26.5	26.5	26.4	26.4	25.9	25.7	25.2	23.0	18.3	16.8
非圧縮	非圧縮+7	26.9	26.9	27.0	27.0	27.1	26.9	27.0	26.3	26.2	26.0	25.0	21.0	19.4
非圧縮	非圧縮+2	26.9	26.8	26.9	26.9	27.1	26.7	26.9	26.6	26.2	26.2	25.2	24.2	23.7
非圧縮+22	非圧縮	26.8	26.7	26.6	26.6	26.7	26.7	26.5	26.1	25.5	23.9	21.9	16.0	14.6
非圧縮+10	非圧縮	26.9	26.8	26.7	26.8	27.0	26.8	26.7	26.6	26.5	25.5	23.8	18.3	16.7
非圧縮+7	非圧縮	27.0	27.1	27.1	27.2	27.1	27.0	26.6	26.7	26.6	26.2	25.0	21.1	19.5
非圧縮+2	非圧縮	26.8	26.9	26.8	26.7	26.6	26.7	26.8	26.7	26.5	26.3	25.7	25.0	24.5
非圧縮+22	非圧縮+22	26.8	26.9	27.0	26.9	27.0	26.9	26.7	26.4	26.2	24.6	22.2	15.6	14.0
非圧縮+10	非圧縮+10	26.9	26.9	26.9	27.0	27.1	26.9	26.9	26.7	26.6	26.2	24.5	19.2	17.4
非圧縮+7	非圧縮+7	26.7	26.6	26.7	26.7	26.6	26.8	26.4	26.5	26.5	26.0	25.5	21.9	20.6
非圧縮+2	非圧縮+2	27.1	26.9	27.0	26.9	26.9	26.9	26.8	26.7	26.5	26.7	26.1	25.5	25.4

A　実験結果を表にまとめたもの

COCO における Karpathy [18] らの評価データの画像を JPEG 圧縮した時の PSNR と SSIM の推移を表 1 に示す。

画像分類タスクにおけるベースラインモデルと、学習画像に JPEG 画像を追加して学習したモデルの非圧縮の評価データと JPEG 圧縮された評価データに対する TOP1 精度を表 2 に示す。非圧縮+[数字] とは ImageNet の学習画像を指定された数字の quality で JPEG 圧縮し、さらに 75 の quality で JPEG 圧縮した画像を学習データに追加して学習したモデルを示す。

画像キャプション生成タスクにおけるベースラインモデルと、提案手法のモデルの非圧縮の評価データと JPEG 圧縮された評価データに対する BLEU スコアを表 3 に示す。非圧縮+[数字] とはカラム名のデータセットの学習画像を指定された数字の quality で JPEG 圧縮し、さらに 75 の quality で JPEG 圧縮した画像を学習データに追加して学習したモデルを示す。

B　学習

ResNet-18 の学習は 35 エポックを行った。

画像キャプション生成モデルの学習は安定化のために 2 つの段階に分けた。画像エンコーダをファインチューニングせずに、言語モデルを学習する第一段階と、画像エンコーダをファインチューニングさせて学習する第二段階である。第一段階と第二段階のそれぞれで 20 エポックと 6 エポックの学習を行なった。

C　ImageNet の開発データの分割

ImageNet の開発データを二分割して開発データと評価データを作る。全てのカテゴリにおいて、そのカテゴリに属する 50 枚の画像をそのファイル名で昇順ソートし、前半を開発データ、後半を評価データとして分割した。