動画中の動きを考慮した Transformerベースビデオキャプショニング

木村 航也[1,a)]　齊藤 燎[2,b)]　山森 一人[3,c)]

概要：ビデオキャプショニングは、動画の内容などを自然言語で記述するタスクであり盛んに研究されているが、動画内に現れるオブジェクトの「動き」に注目する研究は少ない．本研究の目的は、動画内に現れるオブジェクトの「動き」を反映したキャプションを生成することである．具体的には、オブジェクトの動き情報である Optical Flow と色情報である RGB 値の関連度を学習する Cross Attention を Transformer の Encoder に組み込む．MSVD と MSR-VTT の 2 つのデータセットを用いて実験を行い、BLEU と METEOR を用いて評価を行った．実験の結果、提案手法は従来法と同程度、またはそれ以上の結果を得ることができ、文法的、意味的に許容可能なテキストを生成することが出来た．

キーワード：深層学習、Transformer、Video Captioning、MSVD、MSR-VTT

Transformer-based video captioning with object motion in video

Abstract: Video captioning is the task of describing the events in a video in natural language. However, few recent studies have focused on the ”motion” of objects appearing in video. This study aims to generate captions considering the ”motion” of objects in videos. The proposed method incorporates Cross Attention that learns the relationship between Optical Flow and RGB values into Transformer. Our method achieves the same or better results than the conventional methods for evaluation metrics BLEU and METEOR. Our proposed method were able to produce grammatically and semantically consistent text.

Keywords: Deep Learning, Transformer, Video Captioning, MSVD, MSR-VTT

1. はじめに

デジタルカメラやスマートフォンなどの映像機器や端末が幅広く普及し、多くの人が写真や動画の撮影を楽しんでいる．撮影された写真や動画の一部はインターネット上で共有され様々に利用されている．一方、動画を閲覧したい人がテキストで検索する際、内容に関連するタグやキャプションが不正確なため、無関係な動画が検索結果として表示されてしまうことがある．動画から情報を抽出し正確なタグやキャプションを付け整理することは重要なタスクと言えるが、未整理の動画を全て視聴し手作業でタグやキャプションを付与することは現実的ではない．そこで、動画の内容を自動で要約する技術である「ビデオキャプショニング」が注目されている．ビデオキャプショニングは盛んに研究されている分野だが、高価な計算資源が必要で一般的とは言えない．また、動画を対象としているにも関わらず、動画内に出現するオブジェクトの「動き」に注目する研究は少ない．

本研究の目的は、動画内に現れるオブジェクトの「動き」を反映したキャプションを生成することである．本研究では、動画内オブジェクトの動き情報である Optical Flow と、色情報である RGB 値の関連度を学習する Cross Attention をキャプション生成に組み込む手法を提案する．提案手法は、Transformer 型 Encoder と、Cross Attention を追

[1] 宮崎大学 工学研究科
Graduate School of Engineering, University of Miyazaki, Japan
[2] 宮崎大学 農学工学総合研究科
Interdisciplinary Graduate School of Agriculture and Engineering, University of Miyazaki, Japan
[3] 宮崎大学工学教育研究部
Faculty of Engineering, University of Miyazaki, Japan
[a)] hm18010@student.miyazaki-u.ac.jp
[b)] hm14018@student.miyazaki-u.ac.jp
[c)] yamamori@cs.miyazaki-u.ac.jp

加した拡張 Encoder の 2 つの Encoder を用いて、Optical Flow 特徴と RGB 特徴の関連度を計算し、Transformer 型 Decoder、同 Generator を通してキャプションを生成する．

MSVD と MSR-VTT の 2 つのデータセットを用いて実験を行い、BLEU と METEOR を用いて提案手法を評価する．さらに、Optical Flow を用いる事の有用性についても確認する．

2. Transformer

Transformer は Vaswani ら [1] によって提案された、機械翻訳を目的とした人工知能モデルであり、RNN や CNN の代わりに Attention 機構 [2] を持つ．Attention 機構を持つことにより、計算コストの低下や学習の並列化が可能になっている．Transformer は Encoder と Decoder の 2 つのモジュールから構成される．本研究では、Transformer を基礎のモデルとして採用しており、以下で Attention 機構を発展させた Multi-Head Attention のほか、Transformer を構成する主要なモジュールについて説明する．

2.1 Multi-Head Attention

Encoder、Decoder 双方で用いられている Multi-Head Attention は、入力の特定箇所に注目する Attention 機構を基にしている．Attention 機構は、画像や文章が入力されたとき、画像の領域や単語同士の関係などの特定部分に注目する．

Multi-Head Attention のベースとなっている Attention 機構での処理の流れを図 1 に示す．図 1 の matmul、scale、softmax はそれぞれ、行列積、スケーリング、softmax 関数を表す．Attention 機構は、モデルが記憶するデータと入力との関連度を計算する．

モデルが記憶するデータを Memory、入力を Input とするとき、Input から Query、Memory から Key と Value の 3 つの行列を生成する．Attention 機構は、Query と Key の関連度を計算し、関連度に基づいた Value を出力する．具体的には、Query と Key の内積に softmax 関数を適用して確率とし、その値と Value との行列積を計算する．Query を $\boldsymbol{Q}$、Key を $\boldsymbol{K}$、Value を $\boldsymbol{V}$ としたとき、Attention は式 (1) で定義される．

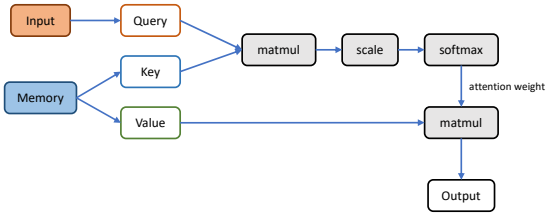

図 1 Attention 機構の概要

$$\mathrm{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}}\right)\boldsymbol{V} = \boldsymbol{A}\boldsymbol{V} \tag{1}$$

ここで、d_k は Key の次元数、 $\boldsymbol{A}$ は attention weight と呼ばれ、Query が Memory から情報をどの程度引き出すのかを示す度合いを表す．attention weight が大きければ大きいほど、Query と Key の関連度が高いということになる．また、$\sqrt{d_k}$ はスケーリングファクターであり、次元数が大きくなると内積の値が大きくなるのを防ぐ役割を持ち、図 1 の scale に対応する．スケーリングした Attention を Scaled Dot-Product Attention と呼ぶ．また、Input と Memory が全く同じモードデータの時は Self Attention、異なるモードデータの時は Cross Attention と呼ばれる．Self Attention は Encoder と Decoder のそれぞれで、Cross Attention は Decoder のみで使用されている．

Attention 機構は学習パラメータを持たないため、学習することができない．この問題を解決するために、Multi-Head Attention が提案された．

Multi-Head Attention は各特徴量をより小さい次元の行列に分割し、それぞれの特徴量ごとに Attention を計算し重みを乗じて統合することで、複数の Attention を学習する機構である．$\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ をそれぞれ H 個に分け、それぞれ head と呼ばれる H 個の Attention 機構に入力する．シークエンス長 S、特徴次元 d_{model} が与えられた時、Multi-Head Attention への入力 $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ は、$\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V} \in \mathbf{R}^{S\times d_{model}}$ であり、出力は各 head で計算した Scaled Dot-Product Attention の結果を結合し、線形変換を行った行列である．線形変換の係数をパラメータとすることで学習可能とし、Attention 機構の問題を解決している．線形変換では、係数行列 $\boldsymbol{W}_h^Q, \boldsymbol{W}_h^K \in \mathbf{R}^{d_{model}\times d_k}$、$\boldsymbol{W}_h^V \in \mathbf{R}^{d_{model}\times d_v}$ を用いて、入力行列 $\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ を低次元空間に射影する．なお、d_k、d_v は $d_k = d_v = d_{model}/H$ である．h 番目の head における Attention である $\mathrm{head}_h(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})$ は式 (2) で定義される．

$$\mathrm{head}_h(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \mathrm{Attention}(\boldsymbol{Q}\boldsymbol{W}_h^Q, \boldsymbol{K}\boldsymbol{W}_h^K, \boldsymbol{V}\boldsymbol{W}_h^V) \tag{2}$$

H 個の head を結合し、$W^O \in \mathbf{R}^{h\cdot d_v\times d_{model}}$ を右からかけることによって、最終的な Multi-Head Attention の出力 $\mathrm{MA}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})$ である式 (3) を得る．

$$\mathrm{MA}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \begin{bmatrix} \mathrm{head}_1(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) \\ \mathrm{head}_2(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) \\ \vdots \\ \mathrm{head}_H(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) \end{bmatrix} \boldsymbol{W}^O \tag{3}$$

2.2 Encoder

Encoder は L 層のレイヤーから構成されており、各レ

イヤーは 2.1 節で述べた Multi-head Self Attention と、Position-wise fully connected feed-forward network(FCN) の 2 つのサブレイヤーからなる．FCN は入力層を除き、単純な 2 層のニューラルネットワークであり、2 回の積和演算と隠れニューロンの活性化関数に ReLU 関数を用いている．ReLU 関数は入力が 0 以下であれば 0、0 より大きければ入力をそのまま出力する．各サブレイヤーの出力は残差接続 [3] を層正規化 [4] したものとなる．残差接続はサブレイヤーへの入力行列と同一サブレイヤーからの出力行列を足し合わせることであり、層正規化はレイヤーごとに特徴量を正規化することである．

入力行列を $\boldsymbol{X}$、1 層目の Encoder の出力行列を $\boldsymbol{z}_1$ とすると、1 層目の出力 $\boldsymbol{z}_1$ は式 (4) で表される．

$$\boldsymbol{z}_1 = \mathrm{LayerNorm}(\bar{\boldsymbol{z}}_1 + \mathrm{FCN}(\bar{\boldsymbol{z}}_1)) \tag{4}$$

$$\bar{\boldsymbol{z}}_1 = \mathrm{LayerNorm}(\boldsymbol{X} + \mathrm{MA}(\boldsymbol{X}, \boldsymbol{X}, \boldsymbol{X})) \tag{5}$$

ここで、LayerNorm は層正規化を表す．2 層目以降は前層の出力を入力とし、最終 L 層目の出力が Encoder の出力となる．

2.3 Decoder

Decoder は、Encoder と同様に L 層のレイヤーからなる．各レイヤーは Multi-Head Self Attention と Multi-Head Cross Attention、Position-wise fully connected feed-forward network の 3 つのサブレイヤーからなる．Encoder と異なる箇所は、Multi-Head Self Attention の出力と Encoder からの出力を入力とする Multi-Head Cross Attention の存在である．

Decoder への入力を $\boldsymbol{Y}$、Encoder からの出力を $\boldsymbol{z}$、Decoder の 1 層目の出力行列を $\boldsymbol{g}_1$ とすると、1 層目の出力 $\boldsymbol{g}_1$ は式 (6) で表される．

$$\boldsymbol{g}_1 = \mathrm{LayerNorm}(\boldsymbol{\alpha}_1 + \mathrm{FCN}(\boldsymbol{\alpha}_1)) \tag{6}$$

$$\boldsymbol{\alpha}_1 = \mathrm{LayerNorm}(\bar{\boldsymbol{g}}_1 + \mathrm{MA}(\bar{\boldsymbol{g}}_1, \boldsymbol{z}, \boldsymbol{z})) \tag{7}$$

$$\bar{\boldsymbol{g}}_1 = \mathrm{LayerNorm}(\boldsymbol{Y} + \mathrm{MA}(\boldsymbol{Y}, \boldsymbol{Y}, \boldsymbol{Y})) \tag{8}$$

ここで、$\boldsymbol{\alpha}_1$ は Multi-Head Cross Attention の出力を層正規化した行列である．Encoder 同様、2 層目以降は前層の出力が次層への入力となり、最終 L 層目の出力が Decoder の出力となる．式 (7) により、全ての層において、Encoder の出力を Key と Value、Decoder の入力を Query として、Encoder の出力と Decoder の入力の関連度を計算している．

2.4 Transformer ベースの関連研究

ビデオキャプショニングの分野において、Transformer を用いた研究の初出は 2018 年の Zhou ら [5] によるものである．Zhou らは、動画内のイベントに応じてシーンを区切り、区切ったシーンごとにキャプションを生成する「密」なビデオキャプショニングというタスクにおいて、Transformer を用いた．提案された手法は、動画の特徴を抽出するエンコーダモジュール、イベントの区切りを提案するデコーダモジュール、キャプションを生成するデコーダモジュールから構成されている．Zhou らの貢献は、RNN ベースのモデルを用いず、Transformer を用いて「密」なビデオキャプショニングを実現したことである．

Iashin ら [6] は、映像と音声のようなマルチモーダルな情報を、Transformer を用いて学習するモデルを発表している．映像のみを学習した場合に比べて、音声を併用して学習することで評価指標が向上することを示した．

これらの研究は、LSTM や Attention を用いて視覚情報とオブジェクトの動きをマルチモーダルな関係として学習しているが、オブジェクトの動きを Transformer を用いて学習した研究ではない．

3. 提案手法

本章では、Optical Flow 特徴と RGB 特徴を扱うための Transformer ベースのアーキテクチャについて説明する．図 2 は提案するモデルの全体図である．図 2 に示すように、提案手法は主に Encoder、Decoder、Generator の 3 つのモジュールから構成されている．

前処理として、データセットに付随するキャプションのトークン化を行う．トークン化は、キャプションに出現する単語に一意の整数値を割り当てベクトル化する作業である．トークン化の過程で、語彙数である語彙サイズを取得する．また、対象とする動画から特徴量を抽出するために、CNN ベースのモデルである Two-stream Infrated 3D ConvNet(以降、I3D)[7] に、動画のフレーム毎の RGB 値と、Teed ら [8] が提案した RAFT を用いて得た Optical Flow を入力し、dim 次元の特徴を持つ RGB 特徴量と Optical Flow 特徴を得る．次に、Optical Flow 特徴を Transformer と同様の Encoder へ入力し、自身の入力に対してどの部分に注目するべきかを求める．この Encoder を以下「Pure Encoder」と呼ぶ．

Pure Encoder からの出力と RGB 特徴を Cross Attention を備えた Encoder に入力する．この Encoder を「Cross Encoder」と呼ぶ．Cross Encoder の出力と特徴空間に埋め込まれたテキストを Decoder へ入力し、2 つの入力の関係を学習したベクトルを出力する．Generator では Decoder の出力から線形変換を行い、softmax 関数によって語彙ごとの出現確率が出力される．ここで最も確率が高い単語を選択することを繰り返すことにより文章を生成する．

3.1 Pure Encoder

Optical Flow 特徴を入力する Pure Encoder は、Multi-Head Self Attention により、Optical Flow 特徴の重要な部分を強調する役割を持つ．Pure Encoder は $L(\geq 1)$ か

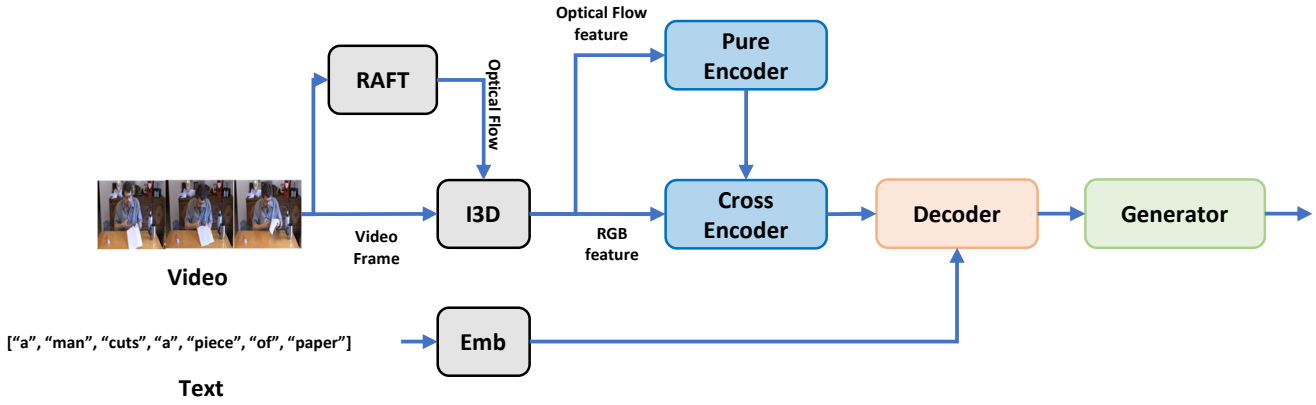

図 2 提案モデルの全体図

らなる Encoder であり、その構造は Vaswani らが提案した Transformer の Encoder とほぼ同じだが、層正規化をする場所が異なる．Vaswani らが提案した Transformer の Encoder は Attention や FCN の後に層正規化を行っているが、提案手法では Attention や FCN の前に正規化を行う．これは、Wang ら [9] によってサブレイヤーの入力に層正規化を適用した方が学習が安定すると示されているからである．l 層目の Pure Encoder の動作は式 (9) から式 (12) で定義される．

$$(\boldsymbol{zp})^i_{l+1} = \boldsymbol{\gamma}^i_l + \mathrm{FCN}(\bar{\boldsymbol{\gamma}}^i_l) \tag{9}$$
$$\bar{\boldsymbol{\gamma}}^i_l = \mathrm{LayerNorm}(\boldsymbol{\gamma}^i_l) \tag{10}$$
$$\boldsymbol{\gamma}^i_l = (\boldsymbol{zp})^i_l + \mathrm{MA}((\bar{\boldsymbol{zp}})^i_l, (\bar{\boldsymbol{zp}})^i_l, (\bar{\boldsymbol{zp}})^i_l) \tag{11}$$
$$(\bar{\boldsymbol{zp}})^i_l = \mathrm{LayerNorm}((\boldsymbol{zp})^i_l) \tag{12}$$

ここで、S を入力の最大長とすると、$(\boldsymbol{zp})^i \in \mathbf{R}^{S\times dim}$ は Encoder 内の内部表現、MA は式 (3) で示した Multi-Head Self Attention による演算、FCN は Position-wise fully connected feed-forward network を表す．Vaswani ら [1] が提案した Transfromer と同様に、$l = 1$ のときは Optical Flow 特徴量 $\boldsymbol{v}^i_{flow}$ が入力、$l \geq 2$ からは前層の内部表現 $\boldsymbol{zp}^i_{l-1}$ が入力となる．$l = L$ のときの内部表現 $\boldsymbol{zp}^i_L$ が Pure Encoder の最終出力となる．

3.2 Cross Encoder

Cross Encoder は、Optical Flow 特徴と RGB 特徴の関連度を計算し学習する役割を持つ．Cross Encoder も Pure Encoder 同様、$L(\geq 1)$ 層からなる．Cross Encoder の入力は、Pure Encoder からの出力と RGB 特徴であり、出力は Cross Encoder の内部表現 $\boldsymbol{zc}^i$ となる．

図 3 に Cross Encoder の内部構造を示す．Cross Encoder は Decoder と同様の構造を持つ．まず、Multi-Head Self Attention により、入力である RGB 特徴において注目する部分を出力する．Multi-Head Self Attention からの出力を正規化したベクトルと Pure Encoder の出力が、Multi-Head Cross Attention への入力となる．提案手法では、Query は RGB 特徴、Key と Value は Optical Flow 特徴である．これにより、RGB 特徴と Optical Flow 特徴の関連度が大きい特徴を抽出することができる．Cross Encoder の l 層目の動作を定式化すると、式 (13) から式 (18) で表される．

$$(\bar{\boldsymbol{zc}})^i_l = \mathrm{LayerNorm}((\boldsymbol{zc})^i_l) \tag{13}$$
$$\boldsymbol{\delta}^i_l = (\boldsymbol{zc})^i_l + \mathrm{MA}((\bar{\boldsymbol{zc}})^i_l, (\bar{\boldsymbol{zc}})^i_l, (\bar{\boldsymbol{zc}})^i_l) \tag{14}$$
$$\bar{\boldsymbol{\delta}}^i_l = \mathrm{LayerNorm}(\boldsymbol{\delta}^i_l) \tag{15}$$
$$\boldsymbol{\epsilon}^i_l = \bar{\boldsymbol{\delta}}^i_l + \mathrm{MA}(\bar{\boldsymbol{\delta}}^i_l, (\boldsymbol{zp})^i, (\boldsymbol{zp})^i) \tag{16}$$
$$\bar{\boldsymbol{\epsilon}}^i_l = \mathrm{LayerNorm}(\boldsymbol{\epsilon}^i_l) \tag{17}$$
$$(\boldsymbol{zc})^i_{l+1} = \boldsymbol{\epsilon}^i_l + \mathrm{FCN}(\bar{\boldsymbol{\epsilon}}^i_l) \tag{18}$$

ここで、$(\boldsymbol{zc})^i \in \mathbf{R}^{S\times dim}$ は Cross Encoder の内部表現を表す．

Pure Encoder と同様に、$l = 1$ のとき RGB 特徴量 $\boldsymbol{v}^i_{rgb}$ を入力する．$l \geq 2$ からは前層の内部表現 $\boldsymbol{zc}^i_{l-1}$ が入力となる．$l = L$ のときの内部表現 $\boldsymbol{zc}^i_L$ が、Cross Encoder の最終出力となる．最後に、式 (16) で Pure Encoder の出力と式 (15) の出力 $\bar{\boldsymbol{\delta}}$ の Cross Attention を求める．

3.3 Decoder と Generator

Decoder は Cross Encoder と同様の構造を持ち、$L(\geq 1)$ 層のレイヤーからなる．Decoder への入力は、テキストを dim 次元の特徴空間に埋め込んだベクトルと Cross Encoder の出力 $\boldsymbol{zc}$ であり、出力は Cross Encoder の出力とテキストの関連度を表す内部表現である．

各レイヤーは Multi-Head Self Attention と Multi-Head Cross Attention、FCN の 3 つのサブレイヤーからなる．Multi-Head Cross Attention では、Cross Encoder の出力とテキストの関連度を求めており、テキストと Cross Encoder の出力間で関係が大きい部分が分かるようになる．Decoder は学習時、サブレイヤーの Multi-Head Self Attention への入力には入力ベクトルの全次元、すなわち将来現れる単語の情報まで含まれているので、これらをマスキングする必要がある．マスキングとはテキストを表す特徴ベクトルの

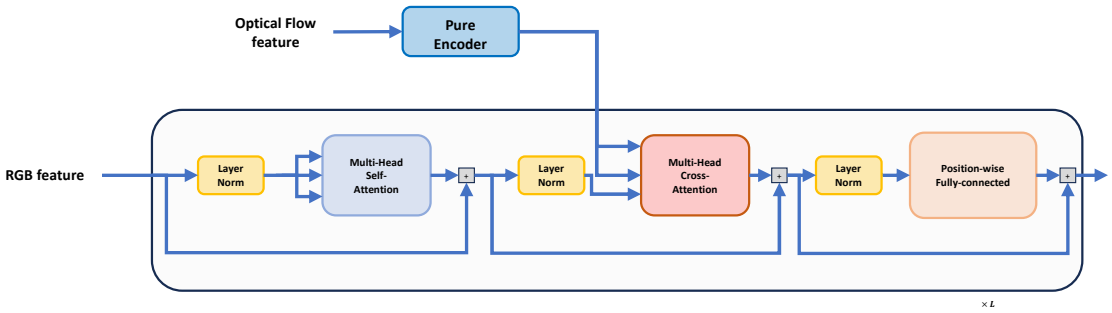

図 3 Cross Encoder 内の構造

該当部の値を意図的に 0 にすることで、当該特徴の影響を無くすことである．具体的には、i 番目の単語を予測するとき、マスキングにより $i-1$ 番目までの情報のみを用いて予測する．

推論時はテキスト入力として、まず < start > トークンが与えられ、Decoder の各層と Generator を経て、次に来る単語を予測する．予測した単語は次ステップの入力に加えられる．このプロセスを < end > トークンが出現するまで繰り返す．

Generator は Decoder の出力を基に単語を予測する役割を持つ．Decoder の出力を語彙サイズ d_{voc} 次元に写像するため、重み行列 $\boldsymbol{W}_G \in \mathbf{R}^{dim \times d_{voc}}$ を持つ全結合層に入力する．この出力に softmax 関数を適用することで、単語に割り当てられた ID ごとに確率が求められる．このうち、最も確率の高い ID が選択され、対応する単語に置換する．

4. 実験と考察

4.1 実験環境

本研究では、MSVD データセット [10] と MSR-VTT データセット [11] を学習データとして用いる．MSVD データセットは、2011 年に発表されたデータセットであり、2010 年の 7 月から 9 月の 2 ヶ月で収集された 2,089 の動画と、それらに対応付けられた 85,550 の英語キャプションのほか、15 言語のキャプションから構成される．現在は、収集元のリンクが機能しないなどにより 1,970 の動画が利用可能になっている．本研究で使用した言語は英語のキャプションのみであり、他言語のキャプションは利用しない．MSR-VTT データセットは、20 カテゴリごとに各 1,000 のビデオクリップからなる．1 つのビデオクリップには英語で記述された 20 のキャプションがあり、29,000 の固有の単語から構成される．本研究ではどちらのデータセットについても約 70%をトレーニングに使用し、20%が検証用、残りをテストに使用する．

評価指標は、機械翻訳や文章生成などのタスクにおいて一般的に使用されている BLEU と METEOR を使用する．BLEU は機械翻訳を評価するために考案された指標であり、複数の職業翻訳者が生成したキャプションと一致すればするほどスコアが高くなるように設計されている．BLEU@3 は 3-*gram* における結果を表し、BLEU@4 は 4-*gram* における結果を表す．METEOR は、BLEU よりも人の評価と相関が高くなるように設計されている．具体的には、BLEU のように生成したキャプションと参照キャプションとの単語の一致率のみに基づくのではなく、類義語を生成した場合も正解にすることで、幅広い表現のキャプションを評価することが出来る指標である．これらの評価指標は、0〜1 の範囲で表され、値が高ければ高いほど良い評価となる．

実験で用いた主なハードウェアについて表 1 に示す．実験を行った環境がデータセットで異なるため、表 1 はデータセットごとに分けて示している．提案手法は、Python3.9.17 で記述し、Pytorch のバージョン 2.0.1 を用いて実装する．本研究で設定したハイパーパラメータを表 2 に示す．全ての学習パラメータは、ノード数 n に対して平均 0、標準偏差 $\sqrt{\frac{2}{n}}$ である正規分布に従って設定され、フルチューニングで行っている．

表 1 使用したハードウェアと OS

	MSVD	MSR-VTT
CPU	Intel Core i9-1270	Intel Xeon 6326
GPU	NVIDIA RTX A5000 24GB	NVIDIA A30 24GB
Memory	128GB	128GB
OS	Alma Linux(WSL2)	Ubuntu 22.04 LTS

表 2 ハイパーパラメータ

ハイパーパラメータ	値
エポック数	50
バッチサイズ	64
Embedding の次元数	1,024
RGB 特徴の次元数	1,024
Optical Flow 特徴の次元数	1,024

表 3　各評価指標における先行研究と提案モデルの比較

Method	MSVD		MSR-VTT	
	BLEU@4	METEOR	BLEU@4	METEOR
PickNet[12]	22.3	33.3	41.3	27.7
OA-BTG[13]	56.9	36.2	41.4	28.3
POS+CG[14]	52.5	34.1	42.0	28.2
STG-KD[15]	52.2	36.9	40.5	28.3
ORG-TRL[16]	54.3	36.4	43.6	28.8
SwinBERT[17]	58.2	41.3	41.9	29.9
CLIP4Clip[18]	55.9	36.9	49.8	31.4
TextKG[19]	60.8	38.5	46.6	30.5
Our Model(RGB+Flow)	26.9	40.3	25.2	32.7

表 4　Optical Flow の有無による比較

Model	MSVD			MSR-VTT		
	BLEU@3	BLEU@4	METEOR	BLEU@3	BLEU@4	METEOR
RGB only	51.1	32.5	41.0	43.0	24.1	32.0
RGB+Flow	47.3	26.9	40.3	42.9	25.2	32.7

4.2　先行研究との比較

RGB 特徴と Optical Flow を Cross Attention に入力して学習したモデルと、先行研究による結果を比較する．表 3 に、MSVD データセットと MSR-VTT データセットにおける、提案モデルと先行研究の評価指標を示す．表 3 の中央部に引かれている線より上は CNN や RNN をベースとしたモデルであり、下は Transformer をベースとしたモデルである．表 3 に示すように、提案モデルは MSVD データセットにおいて BLEU が 26.9、METEOR が 40.3 を記録した．また、MSR-VTT データセットにおいては BLEU が 25.2、METEOR が 32.7 を示した．LSTM や GRU など RNN と CNN がベースとなった研究と比較した場合、BLEU の値は他のモデルの方が優れている．一方で、METEOR の値は他の研究と比べて高くなった．また、Transformer をベースとした研究との比較では、METEOR が他の最新研究とほぼ同程度の結果を出すことができた．

このような結果になった理由は、BLEU と METEOR における計算方法の違いがあると考えられる．モデルにより生成される文章は、参照キャプションよりも短くなることが多い．そのため、BLEU の計算方法ではペナルティが多くなり、評価指標が下がったと考えられる．また、METEOR では意味が似ていた場合、単語が参照キャプションと違っていても同一と判断される．これにより、多少単語が異なっていても、類似した意味を持つ単語が使われていれば正解と見なされ、良好な値を得たと考えられる．

4.3　Cross Encoder の有用性

RGB 特徴と Optical Flow 特徴を Cross Attention で関連付けることが有効かどうかについて調べる．表 4 は、視覚情報である RGB 特徴のみを学習しテストを行った結果 (RGB only) と、RGB 特徴と Optical Flow を Cross Attention に入力し学習した結果 (RGB+Flow) をそれぞれ示す．MSVD データセットの場合、METEOR は RGB ＋ Flow のほうが 0.7 ポイント低く、BLEU@4 も 5.6 ポイント低い結果となった．しかし、MSR-VTT データセットの場合、Cross Encoder を追加したモデルのほうが BLEU@4 は 1.1 ポイント、METEOR は 0.7 ポイント上回る結果となった．

BLEU@4 と METEOR において、MSR-VTT データセットの方が高くなった理由として、学習データが多いことが考えられる．MSR-VTT データセットはキャプション数で約 2.5 倍、動画数で約 5 倍、MSVD データセットよりサンプル数が多い．Transformer は Kaplan ら [20] によって、学習データ量とパラメータ数が増大するほど性能が向上する可能性が示唆されている．このことを考慮すると、MSVD データセットと比較してより大規模なデータセットである MSR-VTT データセットの方が結果が良くなったと考えられる．より大規模なデータセットで学習する場合、Cross Encoder を加えたモデルの方がパラメータ数が多いため結果が良くなると考えられるが、今後検証が必要である．

4.4　定性的結果

図 4 は、実験に使用した動画と参照キャプション、及び提案モデルが実際に生成したキャプションの例を示したものである．Pred は提案モデルが生成したキャプション、GT は動画に付随するキャプションの中から選んだ参照キャプションである．図 4 の上段が MSVD データセット、下段

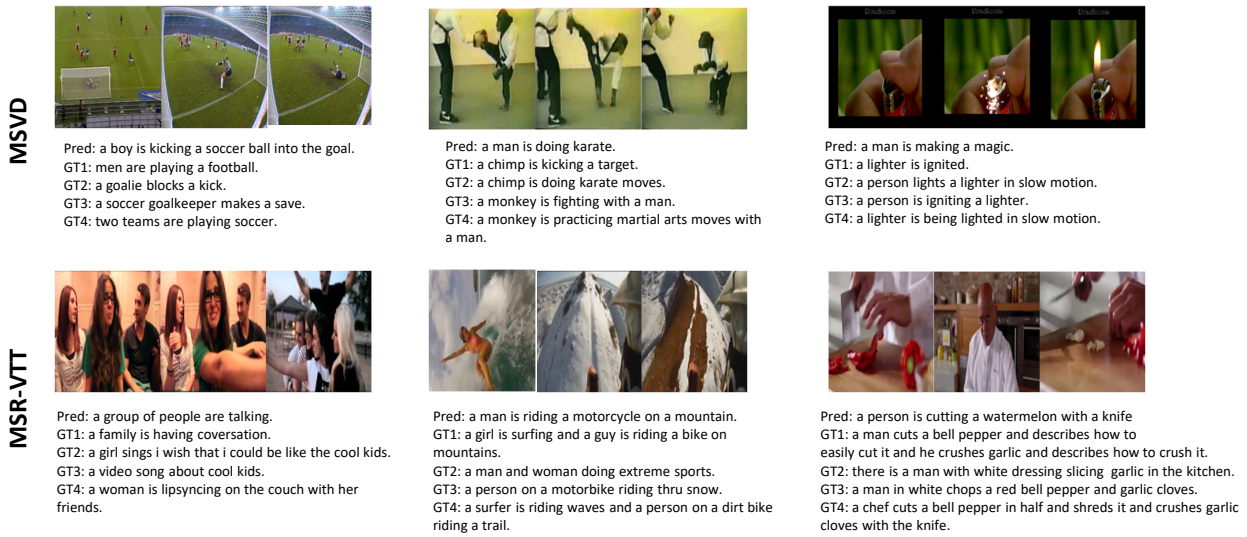

図 4 動画と生成したテキストの例

が MSR-VTT データセットでの例である．これらの例から、提案モデルは人間や動画内で行われているおおよその行動は記述できているが、動物や無生物を認識できないことがあると分かる．例えば、上段中央はサルを人間だと誤認識している．上段右の例でも、人がライターを点火する動画をマジック (手品) をしていると誤認識している．

図 4 に示した提案モデルのキャプションを、英文添削ツール「Grammarly」[21] に入力し、文法的に正しいか確認する．生成した 6 つのキャプションの内、1 つを除いて文法的に正しいことが分かった．このことから、提案モデルは文法的にほぼ正確な文章を生成することに成功しているといえる．意味的な観点からも、生成された文章は動画のシーンとマッチしていると考えられる．上段中央の生成キャプションは、人と動物の間違いはあるが、キャプション中に「doing karate」とあり、参照キャプション中にも「kicking, doing karate, fighting」とあることから、意味的には動画に適したキャプションが生成できているといえる．これらの結果から、提案手法は人間とその動きに対して適切なキャプションを生成することができたが、シーンの切り替えがある場面に対して文章を生成できないことがあることが分かった．また、人間以外の動物や無生物に対してのキャプション生成には改善の余地がある．

5. おわりに

スマートフォンやビデオカメラなどの端末や映像機器が普及に伴い、多くの人が写真や動画の撮影や鑑賞を日常的に楽しんでいる．撮影された写真や動画の一部はインターネット上で共有され、世界中の人が興味のある分野などの写真や動画を検索し、視聴している．動画を検索する際、動画に付けられているタグやキャプションを基に検索を行うが、関係ない動画が検索結果として示されることがある．原因として、タグやキャプションが付けられていない、もしくは不正確なことが考えられる．問題解決のためには正しいタグやキャプションを付ける必要があるが、インターネット上に共有されている動画は大量に存在するため、手作業でタグやキャプションを付けることは現実的ではない．

動画の内容を AI が自動で要約する研究がビデオキャプショニングである．コンピュータビジョンの分野であるビデオキャプショニングは現在盛んに研究されている．しかし、動画内で出現するオブジェクトの「動き」について情報を積極的に利用していないという課題がある．つまり、オブジェクトの動きに関する情報と動画の色情報である RGB 値の関連を学習することが出来れば、より良いキャプションが生成できる可能性がある．

本研究の目的は、AI を用いて動画内に出現するオブジェクトの動きを含めた、正確なキャプションを生成することである．そのため、深層学習における手法の一つである Transformer に注目した．Transformer は機械翻訳の分野で発表された技術であるが、現在では様々なタスクに応用されている．Transformer は、Multi-Head Attention により、入力のどの部分に注目すればよいかという情報を効率よく学習できる．本研究では、動画の色情報である RGB 値の特徴と、オブジェクトの動きを表す Optical Flow の特徴を Multi-Head Attention に入力して学習することで、オブジェクトの動きを含めた正確なキャプションを生成することを目指した．

実験では、MSVD データセットと MSR-VTT データセットを用いて評価を行い、過去の研究と比較した．また、Optical Flow を用いることの有用性を確認する実験も行った．

実験の結果、MSVD データセットにおいて BLEU が 26.9、METEOR が 40.3 を記録し、MSR-VTT データセットにおいて BLEU が 25.2、METEOR が 32.7 を示した．これは、過去の研究結果と同程度またはそれ以上であった．

また、データセットによる違いはあるものの、MSR-VTT データセットにおいては Optical Flow を利用した場合、評価指標が高くなることを示すことができた．生成したテキストに注目してみると、文法的、意味的観点からみて、許容可能と考えられる文章を生成することに成功した．

今後の課題としては、複数の GPU システムによる大規模データセットでの学習や、動物や無生物を認識するため視覚情報とテキストを学習できる Encoder、Decoder の開発、Vision Transofrmer を併用しての実験などが挙げられる．

謝辞 本研究の一部は、JSPS 科研費 JP19K12139 の助成により行われたものです．

参考文献

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u. and Polosukhin, I.: Attention is All you Need, *Advances in Neural Information Processing Systems* (Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. and Garnett, R., eds.), Vol. 30, Curran Associates, Inc. (2017).

[2] Bahdanau, D., Cho, K. and Bengio, Y.: Neural machine translation by jointly learning to align and translate, *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2015).

[3] He, K., Zhang, X., Ren, S. and Sun, J.: Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778 (2016).

[4] Ba, J. L., Kiros, J. R. and Hinton, G. E.: Layer normalization, *arXiv preprint arXiv:1607.06450* (2016).

[5] Zhou, L., Zhou, Y., Corso, J. J., Socher, R. and Xiong, C.: End-to-end dense video captioning with masked transformer, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8739–8748 (2018).

[6] Iashin, V. and Rahtu, E.: Multi-modal dense video captioning, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 958–959 (2020).

[7] Carreira, J. and Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset, *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308 (2017).

[8] Teed, Z. and Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow, *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, Springer, pp. 402–419 (2020).

[9] Wang, Q., Li, B., Xiao, T., Zhu, J., Li, C., Wong, D. F. and Chao, L. S.: Learning Deep Transformer Models for Machine Translation, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers* (Korhonen, A., Traum, D. R. and Màrquez, L., eds.), Association for Computational Linguistics, pp. 1810–1822 (online), DOI: 10.18653/V1/P19-1176 (2019).

[10] Chen, D. L. and Dolan, W. B.: Collecting Highly Parallel Data for Paraphrase Evaluation, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-2011)*, Portland, OR (2011).

[11] Xu, J., Mei, T., Yao, T. and Rui, Y.: Msr-vtt: A large video description dataset for bridging video and language, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5288–5296 (2016).

[12] Chen, Y., Wang, S., Zhang, W. and Huang, Q.: Less is more: Picking informative frames for video captioning, *Proceedings of the European conference on computer vision (ECCV)*, pp. 358–373 (2018).

[13] Zhang, J. and Peng, Y.: Object-Aware Aggregation With Bidirectional Temporal Graph for Video Captioning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

[14] Wang, B., Ma, L., Zhang, W., Jiang, W., Wang, J. and Liu, W.: Controllable video captioning with pos sequence guidance based on gated fusion network, *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2641–2650 (2019).

[15] Pan, B., Cai, H., Huang, D.-A., Lee, K.-H., Gaidon, A., Adeli, E. and Niebles, J. C.: Spatio-temporal graph for video captioning with knowledge distillation, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10870–10879 (2020).

[16] Zhang, Z., Shi, Y., Yuan, C., Li, B., Wang, P., Hu, W. and Zha, Z.-J.: Object Relational Graph With Teacher-Recommended Learning for Video Captioning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).

[17] Lin, K., Li, L., Lin, C.-C., Ahmed, F., Gan, Z., Liu, Z., Lu, Y. and Wang, L.: Swinbert: End-to-end transformers with sparse attention for video captioning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17949–17958 (2022).

[18] Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N. and Li, T.: Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning, *Neurocomputing*, Vol. 508, pp. 293–304 (2022).

[19] Gu, X., Chen, G., Wang, Y., Zhang, L., Luo, T. and Wen, L.: Text with Knowledge Graph Augmented Transformer for Video Captioning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18941–18951 (2023).

[20] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J. and Amodei, D.: Scaling Laws for Neural Language Models, *CoRR*, Vol. abs/2001.08361 (online), available from ⟨https://arxiv.org/abs/2001.08361⟩ (2020).

[21] grammarly: (online), available from ⟨https://www.grammarly.com/⟩.