テレビ映像からの特定動作シーンの自動検出

小林　隼人† 　柳井　啓司†

† 電気通信大学情報理工学部総合情報学科　〒 182–8585 東京都調布市調布ケ丘 1-5-1
E-mail: †kobaya-ha@mm.inf.uec.ac.jp, ††yanai@cs.uec.ac.jp

あらまし　テレビのデジタル化による多チャンネル化で同時に多くの番組が放送されているがそれらをすべて見ることは困難である. 見たい番組をあらかじめ選択し録画しても, その番組の多くは 30 分から 60 分のものばかりである. よって大量の録画映像の中から自分の見たいシーンを探すのにも多くの時間と手間をかけることになる. そこで本研究ではあらかじめ自分の見たい動作の種類を決めておくことで, 動作分類を用いて録画したテレビ映像から自動で特定の動作のシーンを検出し, 自由に閲覧できるようにすることを目的とする. 今回は「食べる」動作の検出を行った. 動作の検出には物体認識, 動作認識, 顔認識を組み合わせた. 物体認識には食事画像認識を使用し, 動画中の食事, 非食事クラスの分類率について実験をした. 上位 5 クラスに非食事クラスが分類された場合を非食事とした場合の分類率は 90.0%となった. 動作認識については学習データを作成し, 分類率は 92.5%となった. 最終的にそれらの組み合わせを試した結果, 2 つのテストデータにおいて動作認識と物体認識を組み合わせた場合, 適合率は最大 55.6%, 再現率は最大 66.7%, F 値は最大 57.2%という結果を示した.

キーワード　動作認識, 動画像認識, テレビ映像認識, 食事画像認識

1. はじめに

テレビでは多くの放送局が作成した番組をそれぞれのチャンネルで放送している. 放送される内容はニュース, バラエティ, スポーツなど多種多様であり, 視聴者はそれぞれチャンネルを切り替えることで各番組を視聴しているが, 視聴者が実際に視聴したいと思っている内容はそれぞれで異なっている. しかしある特定のシーンのみを視聴したいと思っていても, その瞬間のみを探すのは困難である. 現在放送されている番組の多くは 30 分から 60 分であり, 視聴者はその番組をすべて視聴しなければならない. さらに, 近年はキーワードを入力することで番組表とキーワードがマッチしたものを録画するシステムが存在する. このシステムのおかげで番組表を見なくともある程度キーワードにマッチする番組を複数録画することができるようになった. しかし, それによって録画される番組数は非常に多くなり, 結局その中から特定のシーンを探すのには多くの時間がかかってしまう.

地上デジタル放送の場合, 番組に関する情報（メタデータ）が簡単に取得できるが, それだけでは番組中のどこに特定の動作シーンが含まれているかを正確に推定することは難しい. また字幕情報を手がかりとして利用することもできるが, 全ての番組に付いているわけではない. 特定動作シーンの検出には, 実際に放送されている動画の内容を認識することが必要である. 実際の放送内容を認識するための方法には物体認識や動作認識がある. これは静止画や動画から物体や動作を検出, 分類するための技術である. これらの技術を複合的に用いることで, テレビ番組中の特定シーンの検出が期待できる.

本研究ではあらかじめ自分の見たい動作の種類を決めておくことで, 動作分類を用いて録画したテレビ映像から自動で特定の動作シーンを検出し, 自由に閲覧できるようにすることを目的とする.

実際に録画した映像を閲覧するためにはテレビや録画機器に本来備わっている機能を利用して閲覧することが多いが, 本研究では一度録画した内容を録画サーバー内で複数の認識を行いシーンの検出をする. さらに, 検出されたシーンをブラウザ上で閲覧可能とする GUI インターフェースも併せて実現する.

2. 関連研究

今回の研究は, テレビ映像から動作を認識するというものである. 過去にもテレビ映像を利用した研究や, 動作を認識するための手法の提案などが多くされてきた.

2.1 テレビ映像の研究

テレビ映像の研究は数多く行われてきた. テレビドラマからのシーン検出の研究として, Liang らの TVParser がある [1]. この研究ではドラマでの台本と字幕データを利用することで, 登場人物の顔と名前を関連付け, シーン検出を行った. システムの概要は図 1 に示す.

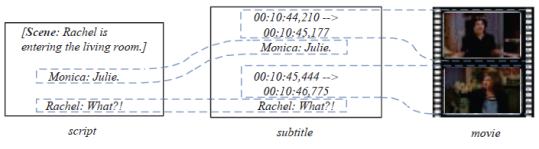

図 1　TVParser([1] より引用)

テレビドラマ以外の研究では向井らの研究がある [2]. この研究では位置情報と字幕情報を利用することで, 旅行番組内での紹介されている場所を特定し, 場所ごとに分割, 地図上に配置することで視覚的にも紹介位置を探すことができるというものである. 番組内で地名を紹介されているもの限定ではあるが, 一定

時間のシーンを場所毎に検出することが可能となっている. システムの概要を図 2 に示す.

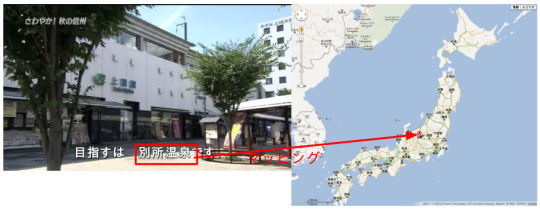

図 2　位置情報を利用したシーン検出の研究 ([2] より引用)

2.2　動 作 認 識

動作認識は非常に難しい研究である. 画像認識とは違い, 複数の並んだフレーム間でどのような動きをとっているのかを考慮する必要があり, 動作時間も各動作で変わってくる. 現在動作認識には多くの手法が存在する. どの手法にも利点や欠点があり, 大きく分けると手法には以下のようなものがある.

（1） 動きを元に認識する方法
（2） 物体やシーン認識を使って認識する方法
（3） 人間の姿勢 (ポーズ) を元に認識する方法

2.2.1　動きを元に認識する方法

この方法は動画中の動きに注目し, その動画の中で物体がどのような動きをしているのかを利用して認識をする手法である. 代表的な手法としては Karen らによる Two-stream convolutional networks [3] や Hang らによる Dense Trajectories [4], Improved Dense Trajectories [5] がある. どちらも動きを元にして認識を行うが, 特徴量を自動的に設計する “Deep learning” を利用するものと時空間特徴を利用し手動で設計するものとがあり, 実際に行っている手法には違いがある.

Two-stream convolutional networks [3] は “Deep learning” を使用し認識を行うものである. Deep Neural Network(DCNN) を利用することで動画から動き特徴となる optical-flow を利用し学習を行うものである. 学習するのに多くの時間を要するが, 通常の画像認識とは違い, 動画中の連続的に動いているものの特徴のみを使用することができる. optical-flow の例は図 3 に示す.

Dense Trajectories [4] は, 人物の行動認識のために提案された特徴点追跡と特徴記述を利用した動作認識の手法である. Improved Dense Trajectories [5] は [4] を改良したもので, 背景などの余計なフローを排除することで精度をあげることに成功した.

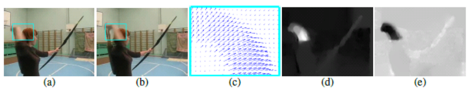

図 3　動きを元に認識する方法 ([3] より引用)

2.2.2　物体やシーン認識を使って認識する方法

物体認識やシーン認識の手法は先ほどのような手法とは違い, 各時間単位でのキャプチャから物体やシーンを認識することでどのような動作をしているかを推定する方法である. この手法は動作が単純ではない場合や, 体の動きが同じでも実際の動作に違いがある場合に, 人間以外の物体を認識することで動作を推定することができる.

最新の研究では Jain らの研究がある [6].　認識の例を図 4 に示す.　通常の動作分類では慎重に選ばれた物体を利用し動作を分類するが, この研究では 200 時間以上の 6 つのデータセットを用いることで, 動画中に出現する 1500 種類の物体から 180 種類の動作を分類する. このような認識方法は動画中に存在しているだけの物体なども特徴として使用することができる. また岡元らの研究としてモバイルでの認識がある [7,8]. これはモバイル端末を使ってアジアの食事を認識したもので, 非常に短い時間で認識することができる.

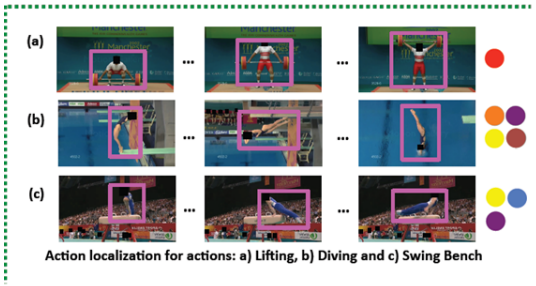

図 4　物体やシーン認識を使って認識する方法 ([6] より引用)

2.2.3　人間の姿勢 (ポーズ) を元に認識する方法

位置推定の研究としては DeepPose という Toshev らの研究がある [9]. この研究は DeepNeuralNetworks(DNN) を使った姿勢推定の論文である. 通常動作認識には特徴量を設計するが, この研究では DCNN を用いることで画像を与えると, 頭や腕など体の各関節の位置を推定することができる. これによって特徴量などを設計せずに人の動きを推測することができる. DeepPose を利用した例を図 5 に示す.

図 5　人間の姿勢 (ポーズ) を元に認識する方法 ([9] より引用)

今回我々は 2.2.1 節で述べた動作認識と 2.2.2 節で述べた動作認識を組み合わせることで, テレビ映像から特定動作シーンを検出する.

3. 手法概要

本研究では, テレビ映像から特定の動作を認識し, 検出するシステムを作成する. 今回は動作の 1 つとして「食べる」動作を検出することを目的とする. ここでは提案手法の全体的な流れを説明する. システムの概要は以下のようになる.

（1） 録画映像を単位時間ごとに静止画像に変換
（2） 顔認識, 物体認識, 動作認識による分類
（3） 分類結果をもとに指定した動作のシーンを検出

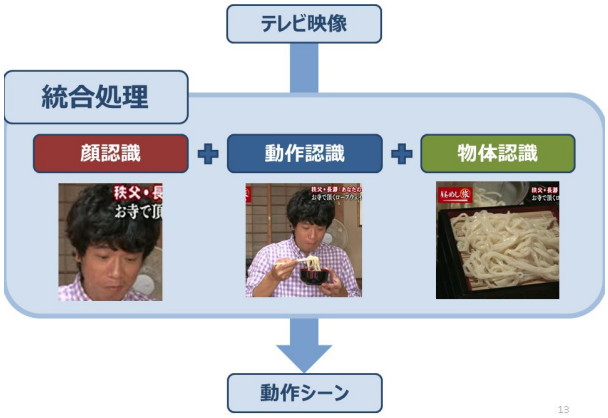

図 6 システムの概要

図 4 にシステムの概要を示した.

4. 手法詳細

4.1 テレビ映像の準備

あらかじめ録画したテレビ映像を入力とする. 映像の録画に関しては, キーワード検索を用いてある程度認識対象となる番組を絞った上で録画する. 今回物体認識と顔認識については静止画像に対して認識し, 動作認識に関しては, 動画像に対して認識をする. それぞれのデータを用意する必要があるため, 入力のテレビ映像を静止画像, 動画像に変換する.

4.1.1 静止画像の変換

物体認識, 顔認識のためにテレビ映像を静止画像へ変換する. 0.5 秒おきに一枚ずつ画像に変換する. 画像サイズについてはテレビ映像のまま使用すると解像度が大きすぎるため, 320 * 240 のサイズに変換する. これによって 30 分の番組から約 3600 枚の画像が出力される.

4.1.2 動画像への変換

動作認識のために入力のテレビ映像をショットに変換する. ここでも画像サイズに関しては 320 * 240 とする. ショットの時間は 1 つのショットあたり 2 秒とする. 動作認識においては 2 秒のショットでも開始時間が 1 秒ずれるだけで別の動作特徴になってしまうため, 0.5 秒おきにショットを生成した. ここでも 30 分の番組から約 3600 のショットを出力される.

4.2 顔認識による分類

動作検出するにあたって, 「食べる」などの人の顔が画面内に入りやすい動作に関しては, あらかじめ顔検出をおこなうことで, 検出精度が向上することが期待できる. 4.1.1 節で変換した各静止画像に対して顔検出を行う使用するのは画像認識ライブラリ OpenCV の顔検出を使用し, 顔検出できた画像のリストを生成する. 顔検出された画像の例を図 7 に示す.

図 7 顔検出された画像の例

4.3 物体認識による分類

動画すべてに対して動作認識を行ってしまうと人以外が動いてしまった場合などの動作を誤認識してしまう可能性がある. よって, 一つの動画に検出するべき動作に関連した物体があるのかどうかを認識することで, 動作に関連性のない動画を対象から除くことができる. 「食べる」動作の場合, 動画中に食べ物が存在するかを調べるため, 切り出したすべての画像に対して食事認識を行う. 食べ物が出てくるシーンの前後に食事シーンの確率が高いものとすることで, 食事に類似した動作「書く」等の動作との混同を防ぐ. 実際には分類器として [7] で用いられている食事・非食事 101 種類認識エンジン FoodCNN を使用した. この分類器の非食事クラスが上位 5 位以内に入る結果を非食事候補とすることで, 食事, 非食事の 2 クラスに分類する.

図 8 に FoodCNN が認識可能な UEC-FOOD100 食事画像データセット [10] の 100 種類の食事画像一覧を示す.

図 8 UEC-FOOD100 データセットの 100 種類の食事一覧 ([10] より引用)

4.4 動作特徴量による動作認識

動作認識には, improved Dense Trajectories [5] と呼ばれる時空間特徴を利用する. これは Dense Trajectories [11] に動き補正を追加したものである. 図 9 で Dense Trajectories の例を示す. 図 10 で示されるように, 426 次元の特徴を持ち, 30d Trajectory, 96d HOG, 108d HOF(Histgram of Flow), 192d MBH(Motion Boundary

Histgram) による組み合わせとなっている. 今回は 30d Trajectory を除いた 396 次元を特徴量として抽出した後, GMM によるコーディングをおこなって Fisher vector にした. 主成分分析 (PCA) をすることでそれぞれを 64 次元とし, 最終的にコーディングされた Fisher vector の動画一本あたりの次元数は 16384 となった.

抽出した Fisher vector を識別するために, Support vector machine(SVM) を用いた.

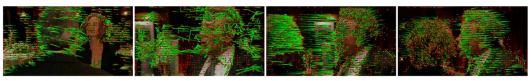

図 9　Dense Trajectories の例 ([11] より引用)

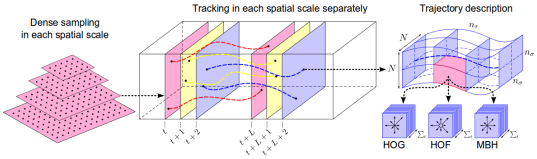

図 10　Dense Trajectories の構造 ([11] より引用)

4.5　認識の組み合わせ

これまで顔認識, 物体認識, 動作認識それぞれについて述べてきたが, ここでは, その 3 つの認識手法をどのようにして組み合わせていくのかについて述べる.

4.5.1　顔認識と動作認識

組み合わせ方についての図を図 11 に示す. 動作認識が正しい動作だと判断した場合, その動作が実際には人以外の物体や生物が行った動作の可能性が存在する. そこで, 顔認識を組み合わせることで仮に動作認識が誤認識してしまった場合でも, 非食事クラスであると認識できる可能性が向上することが期待できる. 具体的には, 顔認識と動作認識それぞれの正解リストを比較し, 互いのリストで正しく認識できていないものを正解リストから外すこととする.

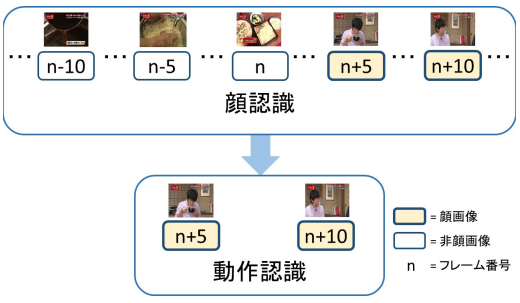

図 11　顔認識と動作認識の組み合わせ

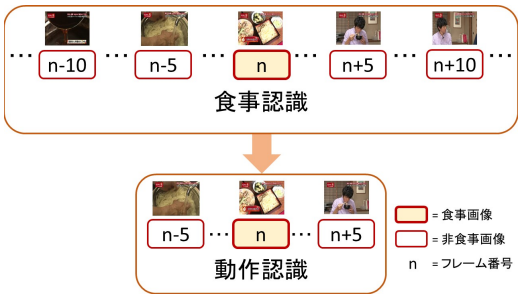

図 12　食事認識と動作認識の組み合わせ

図 13　類似した動作でも意味的に関連性のない例

4.5.2　物体認識と動作認識

組み合わせ方の図を図 12 に示す. 動作認識で正解だと思われるような動作が行われた場合でも実際は動作が類似しているだけで全く関連性のない動作が存在する.

このような動作を誤認識しないために, 物体認識を組み合わせる. 具体的な手法としては, 動作に関連する物体が検出された場合, その前後 5 フレーム内の動作に対してのみ認識を行うようにする. 前後に時間を設けた理由としては, 顔認識と違い, 物体認識で物体を認識した場合, 人がフレーム内に写ってなく, その動作をしていない可能性が高いからである. 動作周辺に関連する物体が存在するかどうかを調べることで誤認識を減らすことが期待できる.

4.5.3　顔認識と物体認識と動作認識

3 つの手法の組み合わせ方法は 4.5.1 節で述べた条件を満たし, かつ 4.5.2 節で述べた条件を満たす動作にのみ検出を行う. これによって認識する無駄な動作を最も減らすことができる.

4.6　シーンの検出

ここでは最終的に出力する食事シーンの検出方法について示す. 動作認識での出力として食事動作についての確率値が得られる. 前後の時間 50 ショット分の確率値の平均値を取り, 分布することで食事可能性のグラフとする. さらにこの中から確率値の高いものだけを選ぶため, 確率値全体の平均値を取り, しきい値とすることで, しきい値以上となった部分を今回検出する食事シーンの動作位置とする.

図 14 に動作認識と食事認識を組み合わせた場合のグラフを示す.

5.　実　　験

今回は「食べる」という動作についてに限定して物体認識と

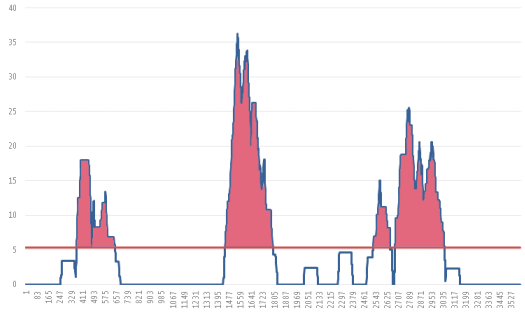

図 14　確立値のグラフの例：赤い位置が実際に検出するシーン

動作認識が正しく行われているかどうかを実験する.

5.1　データセットの準備

5.1.1　学習用データセット

SVM で認識実験を行うためにはデータセットを作成することが必要である. 食事動作データセットのためにはポジティブショットとネガティブショットが必要である. ポジティブショットには YouTube から食事をしている動画を収集した. しかしそのまま利用すると, 複数の人が写っていたり, 動作を行う人以外の物体が動作してしまっている可能性があるため, さらにその中から実際に「食べる」動作のみが写っているシーンを手動で探し, 1 ～ 5 秒程度で解像度が 320*240 のサイズのショットに変換した. なお今回「食べる」動作に関しては,「一人の人がカメラ正面で箸やスプーンを使用して皿から口に食事を運び, 咀嚼する.」までを一つの動作として定義する. 今回はポジティブショットを 100 本用意した. ポジティブショットの例を図 15 に示す.

図 15　ポジティブショットの例

ネガティブショットについてだが, 動作認識のみを使用する場合は人が写っていないショットも加えるべきだが, 今回はあらかじめ顔認識を使用して実際に人が写っていた場合の周辺フレームに対して動作認識を行うので, 人が写っているショットのみを対象とした. 使用したショットは UCF-101 [12] と呼ばれる 101 種類の動作のショットで構成されたデータセットを利用した. 各動作から 3～4 種類ずつ収集した. ネガティブショットの数は 320 本となった. ネガティブショットの例を図 16 に示す.

図 16　UCF-101 の例 ([12] より引用)

5.1.2　テスト用データ

テレビ映像から食事動作を検出できるか実験するために必要なテレビ映像のテストデータについて示す.

映像 1

番組名：昼飯旅　～あなたのご飯見せてください!～
放送局：テレビ東京
放送時期：2015 年 7 月 2 日 (木)
放送時間：約 45 分
ショット数：5600 個

映像 2

番組名：雨上がり食楽部
放送局：東京 MX
放送時期：2015 年 11 月 11 日 (水)
放送時間：約 30 分
ショット数：3600 個

5.2　食事画像認識による分類の評価

ここでは動画から切り出した静止画像から食事画像認識を行った場合の分類率について実験を行った. 本来 FoodCNN ではどのような食事なのかを分類しているが, 本実験では食事と非食事クラスの 2 クラス分類で考える. 今回は上位 5 クラス以内に非食事クラスがあった場合と, 上位 10 クラス以内に非食事クラスがあった場合それぞれを非食事画像と判定した場合について実験する.「食事画像」として認識された食事画像数を true positive,「非食事画像」として認識された非食事画像数を true negative, 全動画像数を all images とした場合, 分類率 (classification rate) は式 1 のように示される. all images の枚数は一時間番組で約 7200 枚である.

$$\text{classification rate} = \frac{\text{true positive} \ + \ \text{true negative}}{\text{all images}} \qquad (1)$$

実際に分類した結果は表 1 に示す.

表 1　FoodCNN による実験結果

読み込むクラス上限	true positive(枚)	true negative(枚)	分類率
5	360	2241	**90.0%**
10	252	2340	89.7%

図 17 と図 18 に正しく食事と分類できたものと正しく食事に分類されなかったものの例を示した.

図 17　正しく食事クラスに分類された例

図 18　正しく食事クラスに分類されなかった例

食事クラスに分類されなかったものはほとんどが人がメインで写っているものや, 人間でも判断するのが難しいものばかりであった. 分類率はどちらもほぼ同じ結果となった.

5.3　動作特徴量による食事動作認識の評価

「食べる」動作について, SVM による分類実験を行った. 各カーネルによる実験結果は以下に示す. ポジティブショット, ネガティブショットすべてのショット 420 本からそれぞれ Dense Trajectories を抽出し, Fisher Vector にコーディングした. SVM による結果として分類率を示す. これは使用した動画の中で正しく認識されたショットの数の割合である. 「食事シーン」として認識された食事ショット数を true positive, 「非食事シーン」として認識された非食事ショット数を true negative, 全動画像数を all shots とした. SVM による分類には 5-fold cross validation による評価を用いた. 分類率 (classification rate) は表 2 のように示される.

$$\text{classification rate} = \frac{\text{true positive} \ + \ \text{true negative}}{\text{all shots}} \tag{2}$$

表 2　SVM による実験結果

	カーネル	分類率
libsvm	liner	76.3%
	polynomial	**92.5%**
	RBF	84.2%
	sigmoid	76.3%
	RBF-chi2	83.7%
libliner	liner	79.4%

表 2 では SVM による実験の結果である. カーネルを polynomial にした場合, 最も分類率がよいという結果となり, かなり高い精度で分類できていることがわかる. よって実際の分類ではこの polynomial カーネルの非線形 SVM を使用する.

5.4　物体認識, 動作認識による特定動作検出の評価

これまでに行ってきた物体認識と動作認識の実験の結果を利用して実際にテレビ映像から「食べる」動作の検出を行う. 今回は複数のテスト映像を用意し, さらに複数の認識手法を組み合わせを実験した. 認識の組み合わせの一覧は以下の様である.

（1）「動き」
（2）「動き」+「食事」
（3）「動き」+「顔」
（4）「動き」+「食事」+「顔」

5.3 節で述べた結果を踏まえ, 最もよかったカーネルの “polynomial” で実験をおこなう. SVM にはカーネル多くのパラメータがある. さらに学習データの精度を向上させるため, パラメータチューニングを行った.

5.4.1　各組み合わせによる結果

映像 1 と映像 2 でのそれぞれの組み合わせによる結果を示す. 結果には適合率 (precision) と再現率 (recall) を使用する. また適合率と再現率から F 値を使用する. 正しく認識された食事シーンを true positive, 正しく認識されなかった食事シーンを false negative, 正しく認識された非食事シーンを true negative, 正しく認識されなかった食事シーンを false positive とした場合, それぞれの式は以下のようになる.

$$\text{precision} = \frac{\text{true positive}}{\text{true positive} \ + \ \text{false positive}} \tag{3}$$

$$\text{recall} = \frac{\text{true positive}}{\text{truepositive} \ + \ \text{false negative}} \tag{4}$$

$$\text{F} = \frac{2 \ * \ \text{precision} \ * \ \text{recall}}{\text{precision} \ + \ \text{recall}} \tag{5}$$

検出されたシーンの中に食事シーンが含まれていた場合を正解とした場合, 映像 1 と 2 による適合率, 再現率を表 5.4.1, 5.4.1, 図 19, 20 に示す.

表 3　映像 1 における適合率, 再現率, F 値

	検出数	正解数	適合率	再現率	F 値
動作	22	4	18.2	44.4	25.8
動作+顔	8	4	50	44.4	47.0
動作+食事	9	5	**55.6**	**55.6**	**55.6**
動作+顔+食事	8	3	37.5	33.3	35.3

表 4　映像 2 における適合率, 再現率, F 値

	検出数	正解数	適合率	再現率	F 値
動作	11	2	18.2	**66.7**	28.6
動作+顔	3	1	33.3	33.3	33.3
動作+食事	4	2	**50**	**66.7**	**57.2**
動作+顔+食事	3	1	33.3	33.3	33.3

正しく検出できた各テストデータ毎の動作シーンの例を以下の図 21, 22 に示す.

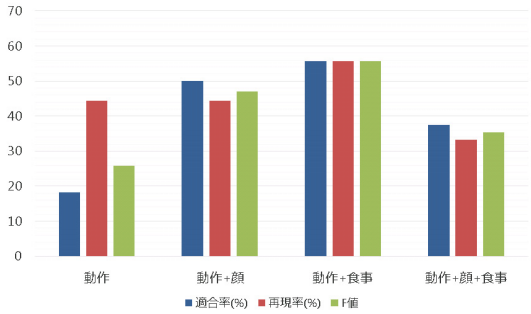

図 19　映像 1 における適合率, 再現率

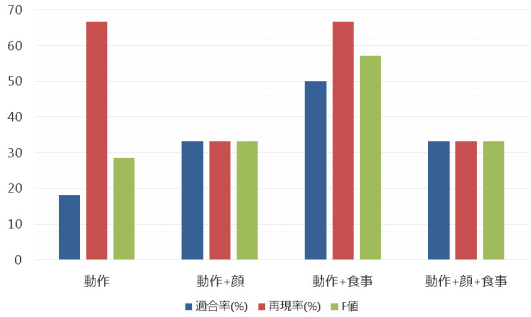

図 20　映像 2 における適合率, 再現率

図 21　映像 1 の正解動作の例

図 22　映像 2 の正解動作の例

6. 考　　察

6.1　食事画像認識の評価

食事画像認識を行った結果, ほとんどの食事と思われる画像を認識することができた. しかし一部の画像では食事画像と思われるものでも食事クラスに分類することができなかった. この要因はいくつかある. まず食事と顔が同時に写っていた場合である. 顔が先に認識された場合, 優先して非食事クラスに分類されてしまうことによって食事クラスに分類されなかった. 2 つ目は単純に FoodCNN に存在しない食事だったものや, 食品ではあるが, 素材であったりするものである. このようなものを認識するためには学習クラスを増やすことで対応することができるが, 本研究においては食材等の認識に関してはできなくても「食べる」という動作への影響は少なかった.

6.2　食事動作認識の評価

ここではテレビ映像に対して食事動作分類を行った結果について述べる.

6.2.1　誤って分類された非食事動作

学習データの SVM での結果だけを見ると, 非常に高精度の分類率となっているが, 実際にテスト動画で実験を行った場合, 食事動作以外にも食事と関係のない動作が上位に分類されていた. このような動画はいくつかのパターンが見られた. 1 つは全く動画中だが音声で説明しているだけで静画が出力されているだけのものや, ほとんど動きのないショットである. 今回認識する「食べる」動きは非常に小さな動きのため, 他の非食事の動きに比べ, 「食べる」動作に近い動きだと判断された可能性がある. もうひとつは「食べる」動作に非常に類似した動きである. 「食べる」動作は手を上げて皿の食事を口に運ぶまでの動きを学習用データとして使用しているため, 人が驚いた際に手を口に上げる動作や, 食事を皿に盛り付ける動作などが非常に類似したショットとして検出されてしまった. 図 23 に実際の例を示す.

図 23　誤って認識された非食事動作の例

6.2.2　正しく分類されなかった食事動作

また食事シーンであっても正しく分類されなかった動作も存在した. このような原因となった原因の 1 つとして挙げられるのが, カメラの動きについてである. 今回利用した Improved Dense Trajectories [5] は、通常の Dense Trajectories [4] と比較するとカメラ動作の補正が行われているが, これは連続したシーン中でのカメラの動きについてであり, シーンそのものがショット内で変化してしまった場合には対応することができない. テレビ映像の多くは予め用意された 30 分から 60 分という限られた番組の時間内で放送すべき内容をまとめる必要性がある. このために行われるのがシーンの無駄な部分のカットである. 咀嚼する動きなどに見られる連続した同じ動きは一部は放送するが, これが番組中複数人分放送する場合や, 同じ料理を食べ続ける場合, 省略されるものが多かった.

またテレビ映像特有の動きとして実験する上で非常に多く見られた動きがカメラのズームであった. カメラのズームという動きには見せたい部分を限定的に見せることで視聴者に見せたい部分に注視させるような特性がある. しかし動作認識という部分においてはこのズームという動きは画面全体の特徴に大きな変化を持たせることになってしまう. テスト映像では 1 秒おきに 2 秒間の動画を取得するようなシステムになっているが,

同じ「食べる」動作を行っているシーンでもカメラのズームが行われているものに関しては, 正しく分類されなかった.

図 24　同じ食事動作でも結果が変わった例：左図ではショット内でカメラのズームの動きがあったため, 非食事動作となった. 右図はズーム後のショットのため食事動作として認識された.

6.3　認識手法の評価

6.3.1　ショットの枚数について

各認識方法を組み合わせた結果, 動作認識のみで行うより, 顔認識や物体認識を組み合わせた場合のほうが誤認識したショットの数が大幅に減ることが確認できた. 特に顔認識を利用することで非常に多くの関係ないショットを非食事動作として認識することができた.

6.3.2　適合率, 再現率, F 値について

すべての認識手法を組み合わせた場合の適合率と再現率, F 値があまりよい結果とはならなかった. 要因として考えられるのが認識の統合手法の問題である. 顔認識に成功していて且つ物体認識に成功している周辺の動作に対して認識を行ったのだが, これによって, 顔認識+動作認識のみで検出成功したシーンと食事認識+動作認識のみで成功したシーンが検出対象から外されてしまった. 互いの認識が成功した場合ではなく, どちらかが成功した場合の結果を反映させることで精度の向上が期待できる.

今回は 3 つの手法を組み合わせたが, 他の情報を組み合わせることも精度を向上させるのに役立つ. 具体的には姿勢推定, 字幕情報, シーン遷移を利用することである. 姿勢推定は, 人間の姿勢の座標を取得することで人の関節の動きを見ることができる. 動作の大きい動きなどを対象にした場合において精度を向上させることが期待できる. 字幕情報では, 使用することで文字を利用した手掛かりを使うことができる. シーンの遷移を認識として手掛かりとする方法は単純なシーンを認識することで実際に検出すべきシーンの位置を推定するというものである. 例をあげると, 食事の前の入店するという動作を認識することで, 食事シーンの手掛かりとすることである. これらの手法から多くの組み合わせを試し, 最適な検出方法を発見することが必要である.

7.　お わ り に

7.1　ま　と　め

本研究ではテレビ映像中から「食べる」動作を認識し検出した. 顔認識, 食事画像認識, 食事動作認識を組み合わせることで精度の向上を図った. 顔認識は人間が写っていないショットを取り除くために使用した. 食事画像認識では FoodCNN を用いることで食事, 非食事クラスでの分類率を 90.0 %とすることができた. 動作認識では SVM による 5-fold cross validation による評価を用いて 92.5 %となった. 2 つのテストデータにおいて複数の認識手法の組み合わせで実験をしたところ, 食事認識と動作認識を組み合わせた場合, 適合率は最大で 55.6%, 再現率は最大 66.7%, F 値は最大で 57.2%となった.

7.2　今後の課題

現在の認識手法では動作検出の精度を向上させなければカメラのズーム中などに目的の動作が行われている場合認識することが難しい. 原因として考えられるのは今回の学習用データにはテレビ映像のデータが入っていないことである. しかし本システムを利用することで, 動作のショットを簡単に集めることができるようになった. 今後は本システムで集めたテレビ映像のデータを学習用データとして加えていくことで全体の精度の向上を図ることができる.

また今回は実験として「食べる」動作に対してのみに絞って検出をした. 今後は別の動作も認識, 検出できるように拡張させる必要がある. 顔認識, 物体認識, 動作認識を組み合わせて実験を行ったが, 今回は姿勢の検出や, 字幕情報などを利用していない. このような他の手法を加えていくことでさらなる精度向上を目指す.

文　　献

[1] C. Liang, C. Xu, J. Cheng, and H. Lu. Tvparser: An automatic tv video parsing method. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pp. 3377–3384. IEEE, 2011.

[2] 向井康貴, 柳井啓司. テレビ番組からの位置情報付き旅行映像データベースの自動構築. 電子情報通信学会論文誌 D, Vol. J98-D, No. 1, pp. 269–274, 2015.

[3] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pp. 568–576, 2014.

[4] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin. Action recognition by dense trajectories. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2011.

[5] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proc.of IEEE International Conference on Computer Vision*, pp. 3551–3558, 2013.

[6] M. Jain, J. C. Van, and C. G. M. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *Proc.of IEEE Computer Vision and Pattern Recognition*, pp. 46–55, 2015.

[7] 岡元晃一, 柳井啓司. DeepFoodCam: DCNN による 101 種類食事認識アプリ. 画像の認識・理解シンポジウム (MIRU), 2015.

[8] K. Yanai and Y. Kawano. Food image recognition using deep convolutional network with pre-training and fine-tuining. In *Proc. of ICME Workshop on Multimedia for Cooking and eating Activities (CEA)*, 2015.

[9] A. Toshev and C. Szegedy. DeepPose: Human pose estimation via deep neural networks. In *Proc.of IEEE Computer Vision and Pattern Recognition*, pp. 1653–1660, 2014.

[10] Y. Matsuda, H. Hoashi, and K. Yanai. Recognition of multiple-food images by detecting candidate regions. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, 2012.

[11] H. Wang, A. Kläser, C. Schmid, and C. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, Vol. 103, No. 1, pp. 60–79, 2013.

[12] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.