7U-08

アニメキャラクター表情識別の精度向上とシーン分類への応用

藤波 広凪† 中島 克人‡

東京電機大学未来科学研究科情報メディア学専攻†‡

1 はじめに

海外からも注目されている日本のアニメは，作品数も膨大になって来ており，作品の分類や検索ニーズも高まっている．キャラクターの表情は，アニメ視聴時に重要であるだけに，その遷移はシーン分類や作品検索のキーになり得る．

そこで我々は，まずアニメキャラクターの表情識別を行ってきたが，一定サイズの個々の顔の識別に比べ，大小の複数の顔を含むシーン画像に対する識別率が低く，シーン分類には満足できる結果を得られなかった．

今回は，データセットの改修によるシーン画像での識別率向上の結果を報告すると共に，この表情識別結果を用いたシーン分類方法を提案する．

2 先行研究

著者らはキャラクターの表情識別の可能性検証のため，データセットの作成と，それによる学習と識別精度の評価を行った[1]．顔だけを切り出した画像（以下，顔画像）では，検出率は十分に高かったが，識別精度は分類や検索キーとするには物足りない結果であった．一方で，アニメからランダムに選択したフレーム（以下，シーン画像）では，識別精度に大差が無かったものの，顔の大きさのバリエーションや顔の傾きが原因で，顔画像に比べて検出率が低く，不十分な結果であった．

3 提案手法

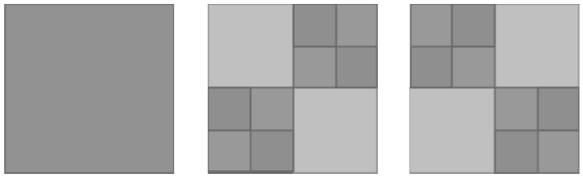
今回は，先行研究と同様のアニメ 4 作品（Charlotte，とある科学の超電磁砲 S，この素晴らしい世界に祝福を!，さくら荘のペットな彼女）を選択し，大小複数のキャラクターの表情を識別を可能にするためにデータセットの改修を行い，それによる学習，および，識別精度の評価を行った．

また，表情識別結果によるシーン分類方法も提案する．

3.1 データセットの改修

表情識別を行うため作成したデータセットを改修する．作成当初は，LBP 特徴による顔検出器[2]を用いて顔部分を切り出し，顔ではない部分の誤検出は排除した上で，それらの顔画像を表情別に手動で振り分けた．その後，顔画像の大きさを統一しており，顔の大きさが 1 種類の画像のみで構成していた．

今回は，シーン画像での検出に対応させるため，画像の位置と大きさを混ぜて学習させることにした．そこで，図 1 の改修後 a，b のように，領域を 4 分割し，その内の対角位置にある 2 つの領域を更に 4 分割したレイアウトに変更を行った．それぞれの領域には 2 種類の大きさに縮小した顔画像を配置する．同時に横顔のデータが少なかったことを考慮し，画像を左右反転したデータ増強も行った．識別種類数は Ekman らの提唱する人間の基本的な 6 表情（Angry, Disgust, Fear, Happy, Sad, Surprise）[3]に Natural（自然な表情）を加えた 7 表情から変更しない．



改修前　改修後 a　改修後 b

図 1 画像 1 枚のレイアウト例

表 1 にデータセット内の各表情の数を示す．なお，切り出した顔画像の枚数が中途半端となった場合，足りない部分は黒く塗りつぶした状態としている．

表 1 表情の内訳

表情	Angry	Disgust	Fear	Happy	Natural	Sad	Surprise
表情数	1,434	1,436	1,410	1,355	1,382	1,377	1,304

3.2 学習と検出・識別

学習には検出と識別を同時に行う検出器の中で比較的高精度で高速とされる YOLOv5[4]を使用した．画像は学習前に予め解像度を変換したものを使用する．データセット内の訓練画像は 2,038 枚，検証画像は 380 枚で，識別は前節の 7 表情で行う．

3.3 シーン分類方法

今回，次の 2 つのシーン分類方法を検討した．

(a) 表情遷移の時系列情報を用いて分類

(b) ショット毎に代表する表情を決定し，表情毎のショット数を元に分類

Improvement of facial expression discrimination of anime characters and its application to scene classification

† Hironagi Fujinami・Tokyo Denki University

‡ Katsuto Nakajima・Tokyo Denki University

(a)では各フレームでの表情の確信度をグラフ化し，その特徴からシーンを分類行う．(b)では各フレームの識別表情の確信度をショット単位で平均化し，最も高いものをそのショットを代表する表情と定める．ただし，確信度がある閾値以下の場合にはノイズと見なし，平均化のための算入を行わない．そして，シーン内の全ショットに対する代表表情の度数を用いて，シーンの分類を行う．

4　評価・考察

まず，シーン画像に対して顔の検出率と表情の識別評価を行った．識別精度は，顔として検出されたものに対してのものである．

また，この表情識別結果を利用して，3.3 節の(a), (b)それぞれによるシーン分類例を示すとともに，考察を行う．

4.1　シーン画像からの顔検出と表情識別

テストのためのシーン画像は，データセットにない 2 作品から先行研究で用いた 50 枚にもう 50 枚を追加した 100 枚で行った．各表情の出現数（真値）は表 2 に示す．顔検出率と表情識別精度を示す混同行列はそれぞれ表 3 および表 4 に示す．

表 2 シーン画像内に出現する表情の内訳

表情	Angry	Disgust	Fear	Happy	Natural	Sad	Surprise
回数	35	16	15	24	24	20	19

表 3 シーン画像での顔検出率

		推論結果	
		検出あり	検出なし
真値	顔である	92.16%	7.84%
	顔でない	0.00%	—

表 4 シーン画像での識別精度

		推論結果						
		Angry	Disgust	Fear	Happy	Natural	Sad	Surprise
正解ラベル	Angry	71.0%	3.2%	16.1%	3.2%	3.2%	3.2%	0.0%
	Disgust	10.0%	70.0%	0.0%	3.3%	0.0%	16.7%	0.0%
	Fear	15.4%	0.0%	50.0%	0.0%	7.7%	26.9%	19.2%
	Happy	0.0%	0.0%	9.5%	71.4%	4.8%	14.3%	0.0%
	Natural	0.0%	13.6%	9.1%	9.1%	59.1%	9.1%	0.0%
	Sad	5.0%	0.0%	2.5%	0.0%	5.0%	87.5%	0.0%
	Surprise	5.3%	0.0%	21.1%	0.0%	18.4%	5.3%	50.0%

顔の大小に関わらず検出できるようになった事もあって，顔の検出率は 90%以上となり，先行研究での検出率 61.6%と比較して大きく向上した．

一方で，顔の傾きに対する検出ミスや，傾きによって識別結果が変化する現象の改善はなかった．

表情識別については，Sad が 87.5%と高めの，Fear は 50%と低めの精度を示した．7 表情全体での平均識別精度は 65.6%となり，先行研究の 68.5%を若干下回る結果となった．

4.2　シーン分類例とその考察

図 2 および図 3 が 3.3 節(a), (b)の分類方法で行う際の想定出力例である．識別された表情の確信度を縦軸に，ショット遷移を横軸に取る(a)においては，シーンの特徴を詳細に表現できる形となっている．しかし，表情識別精度が不十分であると図 2 のように揺れがひどく，シーン特徴を捉えることがかえって困難になる可能性がある．一方，それぞれの表情が識別されるショット数を度数表示する(b)の場合，識別精度が不十分であっても，ショット内で一番登場している表情が真値と大きく異ならない限り，揺れが生じ難いため，図 3 のように特徴が見つけやすい形となっている．しかし，表情の時間的遷移が異なるシーン間の分別は困難となる．

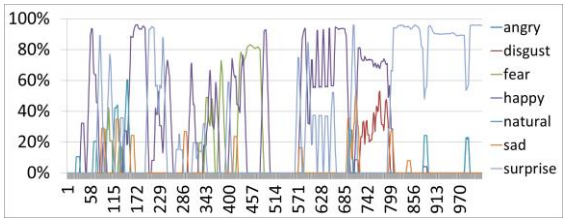


図 2 あるシーンの表情の遷移

図 3 あるシーンの表情別ショット数

5　まとめと今後の課題

アニメキャラクターの顔検出率向上を目的として，データセットの改修を行った．レイアウトの変更と左右反転で学習データの増強を行ったことから，検出率が約 90%と大幅に向上した．しかし，表情識別精度は向上せず，この精度改善のために顔の傾きなどに考慮した更なるデータ増強が必要である．

また，提案したシーン分類方法は，どのような種類と数のシーンカテゴリに分類するかによって，活用度が異なるため，具体的なシーンカテゴリを想定した上で，提案手法(a), (b)あるいはそれ以外の手法の適用を行い，表情識別の有用性を検証するのが今後の課題である．

参考文献

[1] 藤波 広凪, 他, “アニメのシーン解析のためのキャラクター表情識別,” 第 83 回情処全大, 2021.

[2] “lbpcascade_animeface,” https://github.com/nagadomi/lbpcascade_animeface, 2014, 2020/8 参照.

[3] P.Ekman, et al., “Constants across cultures in the face and emotion,” Journal of personality and social psychology 17.2, 1971.

[4] G.Jocher, “YOLOv5,” https://github.com/ultralytics/yolov5, 2020, 2021/11 参照.