

# イラスト描画補助ツールのための キャラクターの顔領域抽出および特徴点推定手法

樋口 叡<sup>1</sup> 新見道治<sup>2</sup>

**概要:** イラストの練習では、描きはじめでの各パーツのバランスを確認することや、描き終えた絵に対する評価を行う行為は大切なことであり、そのためには自分の絵にある修正すべき点を理解することが必要である。しかしイラストの評価は感覚的な話になることが多く、また描画者のみでは、描いた本人ということもあり各パーツのサイズ感やズレといったものに気づけない場合もある。そこで本研究では、描画者自身で修正すべき箇所や現状を判断できるようにするため、イラスト内に存在する特徴を数値化し、それらを利用してイラスト描画の補助を行うツールを提案する。特に本稿ではこのツールで使用するための、イラストから顔領域を自動で抽出し、特徴点を推定する手法について検証する。

**キーワード:** 顔イラスト, 物体検出, イラスト描画補助ツール, 顔領域抽出, 特徴点推定

## Detecting the face region of a character and Estimating feature points for Drawing Assistance Tools

SATOSHI HIGUCHI<sup>1</sup> and MICHIHARU NIIMI<sup>2</sup>

**Abstract:** In the training of drawing illustration, it is important to evaluate the balance of each part at the beginning of the drawing and to evaluate the finished drawing, and for this purpose, it is necessary to understand what needs to be corrected in one's own drawing. However, the evaluation of illustrations is often based on the senses, and the drawer alone may not be able to notice the size of each part or discrepancies in the drawing. In this study, we propose a tool that quantifies the features in an illustration and uses them to assist in drawing the illustration so that the drawer can judge the current state and the parts that need to be corrected by oneself. In this paper, we propose a method for automatically extracting face regions from illustrations and extracting feature points for use in this tool.

**Keywords:** Face illustration, Object detection, Illustration drawing assistance tools, Face detection, Feature points estimation

### 1. はじめに

キャラクターイラストの描画練習をする際には描画者自身の現状と目標を明確にすることが重要である。イラストの描きはじめのアタリを描く段階では各パーツのバランスやサイズ感といったものが決まるため、この段階での描画確認や、描き終えた絵に対してのバランスの評価は大変重要である。こうした評価確認作業では、描画者本人が評価をする場合と第三者に評価を依頼する場合が考えられる。自身で評価をする際は、目指しているイラストの雰囲気などを良く理解をしているため、各部の変更の有無を確認することは比較的容易である。しかし描画者本人ということもあり、描き終えたイラストでのパーツのアンバランス感に案外気づくことができないことも多い。その場合には、第三者に絵を見てもらい評価をしてもらう方法が効果的で

ある。友人や SNS などを活用し描き終えた絵に対して評価してもらうことで、自分では気づけない部分を指摘してもらえる可能性がある。しかしこの場合でも評価者たちの趣味嗜好に評価が左右されてしまわないように描画者がどうなりたいたのか・どうしたいのかを明確に伝えなければならず、また評価者によっては気づいた点を指摘してくれないといった問題も存在している。さらに描画者側の問題として、練習中の絵を他人に見せることに抵抗を感じることも考えられる。そのためイラストのアタリ描画や評価をより簡単に、描画者自身で使用可能なツールが必要であると考えられる。文献[1]のように 3D モデルと用いて顔のアタリと呼ばれる描画作業の補助を行うツールや、文献[2]のように模写をしたイラストに対して、本絵と比較をして模写結果を点数で表してくれるツールが提案されている。しかし [1]は描画補助のみであり評価方法は変わらず、[2]では定め

1 九州工業大学大学院情報工学府情報創成工学専攻  
Department of Creative Informatics,  
Graduate school of Computer Science and Systems Engineering,  
Kyushu Institute of Technology  
2 九州工業大学大学院情報工学研究院  
Faculty of Computer Science and Systems Engineering,  
Kyushu Institute of Technology

られたパーツのみ評価が行われるため、描画者が重きを置く基準などを無視した点数結果になる可能性がある。

そこでキャラクタの顔を対象を絞り、図1のように顔を特徴付ける18個の特徴点を定義して、アタリ描画段階でのズレや描き終えた後のイラストに対して特徴点を計測することで、描画者自身で描画物のズレを判断できる描画補助ツールを提案する。特に本稿では、この特徴点をイラスト内から自動で特定するために、キャラクタイラストの顔領域抽出および特徴点推定手法について検討する。なお、本研究では Manga109 [3][4][5]のデータセットを利用する。

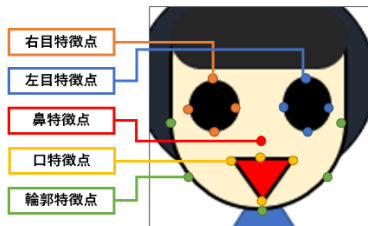


図1 特徴点の位置

Figure 1 Position of the feature points.

## 2. 物体検出のための既存技術の利用

顔領域の推定を行うために、物体検出が可能な技術のいくつかを「顔識別器」や「目識別器」として利用する。

### (1) OpenCV のカスケード型分類器[6]

カスケード型分類器は正解画像と不正解画像を用意して、それぞれの画像内の輝度勾配や局所領域での明暗差といった特徴を抜き出し正解画像を学習する。

### (2) YOLOv3[7]

YOLOは深層学習を用いて物体検出を行うもので、デフォルトで人や犬、冷蔵庫など80種類の物体識別が可能となっており、また検出したいクラスを追加で学習させることが可能である。

### (3) FaceRecognition[8]

2017年に発表された現実の人物の顔検出技術で、HOG特徴量かCNNを選んで顔検出を行うことができ、その人物の特定や、顔の輪郭や目の輪郭といった顔にある特徴を検出する機能を有する物体検出技術である。

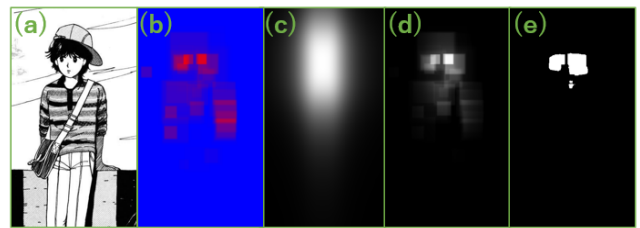
## 3. 顔領域の推定方法と実験

顔は丸みを帯びている物体で描画されるという点に着目し、顔領域を推定したい入力画像に対してHough変換を利用した円検出を行い、顔領域のたまかな位置を抜き出し、抜き出した円領域に対して顔領域の推定を行う手法を提案する。なお提案手法の記載に当たり、以下の定義を行う。

- ・入力画像(図2の(a))は、人物イラストが描かれた横ビ

クセル $w$ 、縦ピクセル $h$ である入力データ $f(x,y)(x=0,1,\dots,w-1,y=0,1,\dots,h-1)$ とする。

- ・画像上に存在する一つの点位置を $P=(x,y)$ として表記をし、 $P+c=(x+c,y+c)$ と定義する。
- ・矩形領域を左上の点 $P^{TL}$ と右下の点 $P^{BR}$ を用いて、 $(P^{TL},P^{BR})$ と表記する。
- ・顔領域を検出することは、 $f(x,y)$ から一つの $(P^{TL},P^{BR})$ を特定することである。
- ・ $count((P^{TL},P^{BR}))$ を、矩形 $(P^{TL},P^{BR})$ 領域の画素値が1、その他の領域が0となる、 $f$ と同サイズの画像データを生成する関数であると定義する。



©よしまさこ『愛さずにはいられない』2巻P.3より

図2 提案手法での顔推定の例

Figure 2 Example of face estimation.

### 3.1 提案する顔領域の推定方法

$f(x,y)$ から円を検出すると、円の中心 $P^C$ と半径 $r$ が得られる。検出結果は複数個存在する可能性を考慮し、その個数を $M$ とした場合、円領域は $P_i^C, r_i(i=0,1,\dots,M-1)$ を用いて $P_i^{CTL}=P_i^C-r_i-m, P_i^{CBB}=P_i^C+r_i+m$ で表される円領域 $(P_i^{CTL}, P_i^{CBB})$ が検出される。このときの $m$ は検出された円領域に持たせるマージンとした。また示す領域が画像範囲を外れる場合は適切な値に調整をする。

得られた円領域を一つずつ顔識別器にかけることで、顔領域の検出結果を得る。一つの円領域から検出される顔領域も複数個存在する可能性があるため、その検出個数を $N_i$ とした場合、顔領域の矩形候補として式(1)が求まる。

$$(P_j^{TL}, P_j^{BR}), \quad (j=0,1,\dots,\sum(N_i)) \quad (1)$$

今、 $count$ と $N_i$ を用いて、 $F_{count}$ を式(2)と定義される画像データとする(図2の(b))、検出された範囲に多く含まれる画素ほど赤色)。

$$F_{count} = \sum_{j=0}^{\sum N_i} count((P_j^{TL}, P_j^{BR})) \quad (2)$$

この画像とFace Mapを使用して顔領域の位置を一つの領域に定める。Face Mapは「画像内の顔の位置は上部の中心に、縦長に位置する」という考えのもと、 $f$ と同じ横幅 $w$ 、縦幅 $h$ として、式(3)で表される画像データである(図2の(c))。なお $g_x, g_y$ は各軸における倍率であり、今回の実験では $g_x=5.0e^{-6}, g_y=5.0e^{-7}$ とした。

$$FM(x, y) = \frac{1}{\left(g_x \left| \left(x - \frac{w}{2}\right)^3 \right| + 1\right) \left(g_y \left| \left(y - \frac{h}{4}\right)^3 \right| + 1\right)} \quad (3)$$

この $FM$ と $F_{count}$ を利用して、顔領域密度(Face Density)を以下の式 (4) で定義をする。

$$FD = F_{count} \cdot FM \quad (4)$$

この $FD$ を閾値  $T$  より値が大きい画素を前景画素として二値化した画像を式 (5) と表記する。

$$FD_{binary}^T \quad (5)$$

$FD_{binary}^T$ の前景画素を不足なく囲む最小矩形において、 $x, y$ ともに最も小さい点を  $P_{min}^{TL}$ ,  $P_{min}^{TL}$ , 最も大きい点を  $P_{max}^{BR}$  としたとき、 $f(x, y)$ から求めたい顔矩形領域は  $(P_{min}^{TL}, P_{max}^{BR})$  に定まる。

### 3.2 特徴点の推定方法

入力画像から特徴点である 18 点の  $x, y$ 座標を推定するため、深層学習を用いる。推定を行うモデルは図 3 の構成とする。Manga109 のデータから、各目の中心座標間の中心座標を画像の中心となるように 1090 枚用意し、1 枚につき縦横比率の拡大縮小や中心座標をランダムに上下左右数ピクセルずらしてデータ数の拡張を行ったデータセットを用意して学習を行った。また画像の縦横サイズは  $128 \times 128$  として、イラストの線部分を前景画素として学習できるようにグレースケール化を施し、ラプラシアンフィルタを用いて画像処理を行う。入力画像に使用する顔領域は  $128 \times 128$  にリサイズをして特徴点座標を推定する。

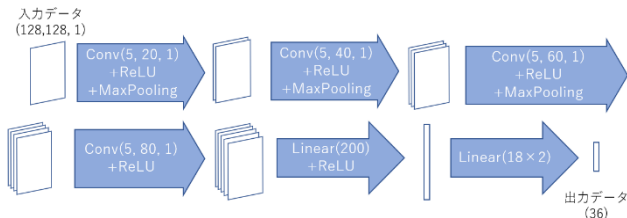


図 3 特徴点推定を行う深層学習のモデル

Figure 3 CNN model for feature point estimation.

### 3.3 実験方法

実験では「どの顔識別器」が、「どんな顔領域推定」をして、「閾値がいくつの時」に良い結果が出るのかを確認する。

顔識別器に関しては、第 2 節で挙げた 3 つの既存技術を用いて、以下のような 6 通りを用意した。なお学習において記述されていないパラメータはすべてデフォルト値を利用した。

- YOLO : Manga109 のデータから顔領域などを抜き出して、YOLOv3 に追加で学習させる。学習対象はキャラクター画像の顔を含めた 4 クラスで、学習画像は縦横サイズを  $256 \times 256$ 、画像枚数は 5450 枚とした。
- FaceRecognition(hog) : FaceRecognition の hog 特徴での

顔検出を行うモードを利用

- FaceRecognition(cnn) : FaceRecognition の CNN での顔検出を行うモードを利用
- Cascade(def) : OpenCV が配布をしている現実の人物の顔を検出するためのファイルを利用
- Cascade(haar) : Manga109 のデータを利用し、Haar-like 特徴を学習させたファイルを使用。正解画像数が 7000 枚、不正解画像が 4000 枚として、画像の縦横サイズに制約などは設けなかった。
- Cascade(lbp) : Manga109 のデータを利用し、LBP 特徴を学習させたファイルを使用。学習させる各画像は Cascade(haar)と同様。

顔領域の推定に関しては、提案手法を使わない場合と使う場合、使う場合での  $m$  の違いを確認するために 5 通りでの実験を行う。提案手法を使わない処理名を「PF1」として、入力画像をそのまま顔識別器に入れた結果を式 (1) として  $FM$  との計算に利用して顔矩形領域を一つに定める。提案手法を使う処理名を「PF2」としマージン  $m$  が 0,  $r \times 1/10$ ,  $r \times 1/8$ ,  $r \times 1/4$  としたときのデータをそれぞれ「PF2,  $m=0$ », 「PF2,  $m=1/10$ », 「PF2,  $m=1/8$ », 「PF2,  $m=1/4$ 」とする。

二値化の際の閾値に関しては可変値と固定値を合わせて 7 通り用意をする。可変値では、FD の 0 以外のピクセルの総計  $pval$  としたときに 0 以外の累積ヒストグラムがそれぞれ  $pval \times 1/4$  以上,  $pval \times 2/4$  以上,  $pval \times 3/4$  以上となる閾値を  $T$  とする。固定値では  $T=50, T=100, T=150, T=200$  とした。

### 3.4 計測方法

計測に使用するテスト画像は 100 枚で、90 枚は Manga109 内から抜き出した画像で 10 枚は素人が描いた画像を、許可を得て使用しており、なるべくキャラクタの全身が描いてあること、一画像につき一人のキャラクタのみが映っていることなどを推奨される条件として画像を集めた。それぞれの画像について顔領域と特徴点をあらかじめ定め、顔領域については一致度合いを、IoU (Intersection over Union) を用いて比較する。IoU とは「正解の矩形領域と推定した矩形領域との重なっている画素数」を、「両矩形領域を一つの領域としたときの領域内の総画素数」で割った値であり、最大値が 1 (パーセント表記では 100%が最大値) となる。その後抽出した顔領域を、特徴点の位置推定学習を終えた深層学習に入力して特徴点の位置推定結果を得て、正解位置からどの程度ずれているのかを計算し比較する。特徴点の推定結果をはかる尺度には、値が小さいほど推定された特徴点位置が正解位置に近いことを示す平均二乗誤差 (MSE) を用いた。

### 3.5 実験結果

表 1 は提案手法を使用する場合としない場合の結果の違いを確認するために、閾値  $T$  を  $pval \times 1/4$  として、「PF1」と「PF2,  $m=0$ », 「PF2,  $m=1/8$ 」の結果を示している。検出率

は 100 枚のテスト画像のうち、顔領域の推定を行った際に顔領域を一つでも検出している画像の割合である。IoU の平均値は、各テスト画像での IoU の平均をとった値であり、顔領域が検出できなかった場合のテスト画像の IoU は 0 としている。

表 1 T=pval×1/4 における各識別器の結果

Table 1 Results table for each discriminator.

識別器	顔検出処理	検出率	IoU の平均値
YOLO	PF1	1%	0.31%
	PF2, $m=0$	98%	22.13%
	PF2, $m=1/8$	97%	20.25%
FaceRecognition (hog)	PF1	25%	6.70%
	PF2, $m=0$	29%	9.46%
	PF2, $m=1/8$	32%	11.11%
FaceRecognition (cnn)	PF1	35%	11.25%
	PF2, $m=0$	49%	18.37%
	PF2, $m=1/8$	52%	20.22%
Cascade(def)	PF1	51%	14.33%
	PF2, $m=0$	71%	17.18%
	PF2, $m=1/8$	82%	21.12%
Cascade(haar)	PF1	100%	18.42%
	PF2, $m=0$	100%	19.66%
	PF2, $m=1/8$	100%	19.05%
Cascade(lbp)	PF1	100%	24.81%
	PF2, $m=0$	100%	25.17%
	PF2, $m=1/8$	100%	26.04%

また各顔識別器における、 $T=pval \times 1/4$  組み合わせごとの検出率・IoU 平均値は図 4、図 5 の通りである。これらの結果から提案手法を使った場合では、使わなかった場合と比べて検出率や IoU の平均値がわずかに向上した。特に識別器が YOLO の場合では、他と比べても遜色のないほどに顔領域を検出できるようになっていた。

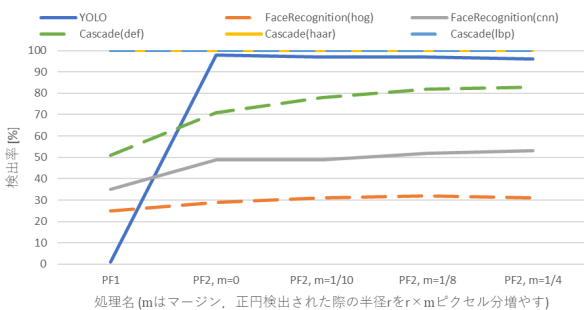


図 4 各識別器における検出率の結果

Figure 4 Results of detection rate for each discriminator.

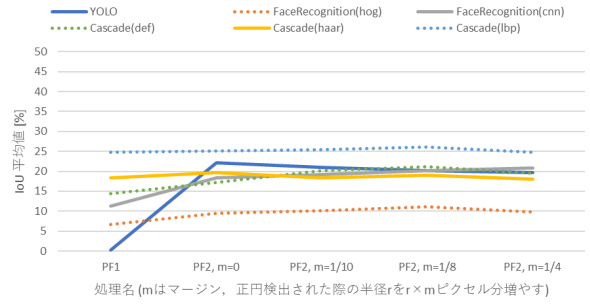


図 5 各識別器における IoU の平均値の結果

Figure 5 Results of average IoU for each discriminator.

また手法「PF2,  $m=0$ 」での各顔識別器における、閾値  $T$  の違いによる IoU の平均値の推移を図 6、手法「PF2,  $m=1/8$ 」での各顔識別器における、閾値  $T$  の違いによる IoU の平均値の推移を図 7 に示す。ここから実験を行った値の中では  $T$  を 50 に固定している場合が、どの識別器を使った場合でも他の閾値よりも結果が良かった。

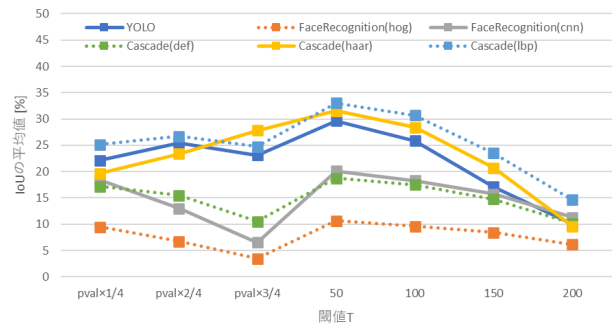


図 6 「PF2,  $m=0$ 」における各識別器の IoU の平均値

Figure 6 Results of average IoU for each discriminator.

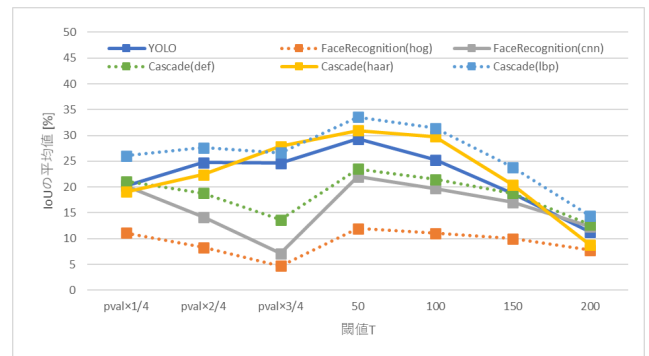


図 7 「PF2,  $m=1/8$ 」における各識別器の IoU の平均値

Figure 7 Results of average IoU for each discriminator.

そこで「 $T=50$ 」のときの「PF2,  $m=0$ 」と「PF2,  $m=1/8$ 」の場合の検出率や IoU の平均値, MSE の平均値と中央値をまとめたものを表 2 に示す。MSE に関しては、顔を抜き出すことができ特徴点の推定を行うことのできたテスト画像群でのみ、平均値や中央値を計算している。そのため MSE の総データ数は 100 個でない場合がある。ここから

FaceRecognition は検出率自体は低いが、顔検出ができた際の MSE 値は低く、顔検出の精度は良い。また FaceRecognition のみの結果としては hog よりも cnn を使用した場合のほうが全体的な値は良かった。検出率や IoU の平均値から、カスケード分類器を使用した場合の顔検出結果が良かった。

また FaceRecognition や Cascade(def)といった現実の人物の顔を抜き出すものと、Manga109 のデータを学習させたものとを比べた場合では、後者のほうがより値が良かった。

しかし検出率が高い各識別器では MSE の平均値や中央値が高くなってしまっている。そのため次に MSE の値を低くするための追加実験を行う。

表 2 T=50 における、「PF2,  $m=0$ 」と「PF2,  $m=1/8$ 」での各識別器の結果

Table 2 Results table for each discriminator.

識別器	顔検出処理	検出率	IoU 平均値	MSE 平均値	MSE 中央値
YOLO	PF2, $m=0$	98%	29.68%	3982.66	946.71
	PF2, $m=1/8$	97%	29.32%	5855.16	1050.10
FaceRecognition (hog)	PF2, $m=0$	29%	10.64%	2165.90	85.29
	PF2, $m=1/8$	32%	11.97%	3410.96	103.33
FaceRecognition (cnn)	PF2, $m=0$	49%	20.11%	1120.40	70.66
	PF2, $m=1/8$	52%	22.03%	1134.86	71.88
Cascade(def)	PF2, $m=0$	71%	18.73%	9039.42	1183.70
	PF2, $m=1/8$	82%	23.54%	9410.64	1078.90
Cascade(haar)	PF2, $m=0$	100%	31.57%	3410.55	2758.20
	PF2, $m=1/8$	100%	30.97%	3575.59	2908.05
Cascade(lbp)	PF2, $m=0$	100%	33.06%	2156.68	840.42
	PF2, $m=1/8$	100%	33.55%	2260.36	864.69

#### 4. 特徴点の位置推定向上のための追加実験

MSE が下がらない原因の一つとして、特徴点推定の機械学習で利用した学習時の顔領域と、検出された顔領域のズレが考えられる。そこで顔領域を学習時に近づけることを目的に、まず検出された顔領域から目を検出し、二つの目矩形の中心座標を計算し、左右中心点の midpoint から一定距離を顔領域として抽出し、新たな顔領域とする。

- ・ 入力画像は第 3 節にて推定された顔領域( $P^{TL}, P^{BR}$ )とする。
- ・ 画像上に存在する一つの点位置を  $P = (x, y)$  として表記をし、 $P + c = (x + c, y + c)$  と定義する。
- ・ 矩形領域を左上の点  $P^{TL}$  と右下の点  $P^{BR}$  を用いて、( $P^{TL}, P^{BR}$ ) と表記する。
- ・  $\max(a, b)$  と表記した場合、 $a$  と  $b$  のうち大きいほうの値を採用する。

- ・  $\min(a, b)$  と表記した場合、 $a$  と  $b$  のうち小さいほうの値を採用する。

#### 4.1 提案する顔領域の調整手法

入力画像を目識別器に入力することで目領域を複数個検出する。検出個数を  $K$  個とした場合、検出結果内の本当のペアは「同程度の高さにあり、横幅や縦幅は同じくらいになっている」と考えられるため、検出された目領域の  $i$  番目の領域  $r_i(P_i^{TL}, P_i^{BR})$  と  $j$  番目の領域  $r_j(P_j^{TL}, P_j^{BR})$  ( $i < j$ ) において、図 8 を例として以下の 3 つのパラメータを定義する。

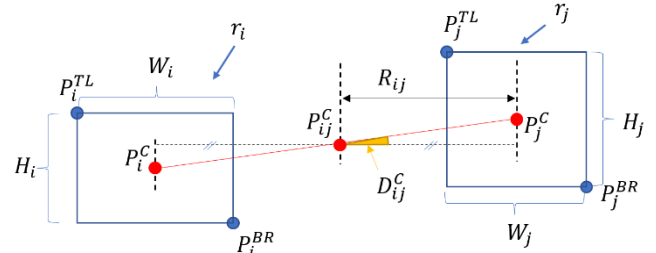


図 8 目のペア検出に使用される値の位置や距離  
Figure 8 Location and distance of values used for eye pair detection.

- ・ それぞれの矩形領域の中心座標  $P_i^C$  と  $P_j^C$  として、その二点の中心点が  $P_{ij}^C$  で表されるときにその角度を  $D_{ij}^C$  ( $-90^\circ \leq D_{ij}^C \leq 90^\circ$ ) として、以下の (6) 式で表されるパラメータ  $D_{ij}$ 。またパラメータ  $D_{ij}$  については一定以上の角度  $D_{th}$  になる場合は目のペアではないとして他のパラメータをすべて 0 とする。

$$D_{ij} = \begin{cases} \text{if } (|D_{ij}^C| \leq D_{th}) & 1 - \frac{D_{ij}^C}{D_{th}} \\ \text{otherwise} & 0 \end{cases} \quad (6)$$

- ・ 各矩形領域の横幅  $W_i$  と  $W_j$  とした場合に式 (7) で表せる比率パラメータ  $W_{ij}$ 。

$$W_{ij} = \frac{\min(W_i, W_j)}{\max(W_i, W_j)} \quad (7)$$

- ・ 各矩形領域の縦幅  $H_i$  と  $H_j$  とした場合に式 (8) で表せる比率パラメータ  $H_{ij}$ 。

$$H_{ij} = \frac{\min(H_i, H_j)}{\max(H_i, H_j)} \quad (8)$$

上記の三つのパラメータを、それぞれ最大値 1、最小値 0 で線形変換を行った  $D_{ij}^L, W_{ij}^L, H_{ij}^L$  を用いて式 (9) で表される。

$$DR(r_i, r_j) = B_1 D_{ij}^L + B_2 W_{ij}^L + B_3 H_{ij}^L \quad (9)$$

$DR$  が一番大きい  $i, j$  の組を両目のペアとして抜き出す。なお  $B$  は各パラメータにおけるバイアスである。 $P_{ij}^C$  と  $P_j^C$  との  $x$  軸での距離を  $R_{ij}$  として、 $P_{ij}^{CTL} = P_{ij}^C - 3.2R_{ij}$ ,  $P_{ij}^{CBR} = P_{ij}^C + 3.2R_{ij}$  で表される矩形領域 ( $P_{ij}^{CTL}, P_{ij}^{CBR}$ ) を目領域の検出を用いた顔領域推定結果とする。

#### 4.2 実験方法

実験では「どの目識別器」が、「 $DR$  の各バイアス値」がど

の場合で良い結果が出るのかを確認する。

目識別器に関しては、第2節で挙げた2つの既存技術を用いて、以下のような3通りを用意した。なお学習において記述されていないパラメータはすべてデフォルト値を利用した。

- ・YOLOE:Manga109のデータから目領域を抜き出した画像を学習させる。なお学習画像は縦横サイズを96×96、画像枚数は3440枚とした。
- ・CascadeE(haar):Manga109のデータを利用し、Haar-like特徴を学習させたファイルを使用。正解画像数が3500枚、不正解画像が2000枚とする。
- ・CascadeE(lbp):Manga109のデータを利用し、LBP特徴を学習させたファイルを使用。学習させる各画像はCascadeE(Haar-like)と同様。

DRの各バイアスの値に関しては、今回は $B = [B_1, B_2, B_3]$ に対して $[1, 1, 1]$ ,  $[2, 1, 1]$ ,  $[1, 2, 1]$ ,  $[1, 1, 2]$ ,  $[1, 2.5, 1.5]$ の組での実験を行った。また、相対角度の閾値 $D_{th}$ は40に定めた。なお、実験の計測方法は第3節と同様とし、目検出に投入する顔領域は、検出率が100%でありIoUの値なども高い「T=50」、顔識別器は「Cascade(lbp)」の場合で、「PF2,  $m=0$ 」と「PF2,  $m=1/8$ 」の2つの手法での顔検出処理を利用した。特徴点の推定には3.2の方法を利用した。

### 4.3 実験結果

各目識別器での検出率を除く3データの結果を表3に載せる。また表3にある識別手順ごとのMSE値の結果を箱ひげ図でまとめたものを図8に示す。なおDRのバイアスに関しては5つのパターンで実験を行ったが目立った差異は見られなかったため、 $[1,1,1]$ の状態での実験結果となっている。ここから目識別器を利用した場合にはMSEの値の低下だけではなく、IoUの平均値も向上した。識別器に関してはYOLOを利用した場合よりもカスケード分類器をしようした場合のほうが良い結果が得られており、Haar-like特徴を学習させたもののほうが、LBP特徴を学習させたものよりも良い結果が得られた。また中央値や平均値からは確認できなかったが図9の箱ひげ図から、目識別器を「CascadeE(haar)」とした場合には顔検出処理が「PF2,  $m=0$ 」の場合(薄い青色棒)よりも「PF2,  $m=1/8$ 」の場合(緑色棒)のほうが、MSEが狭い範囲で分布していることが確認できる。

また、各テスト画像での、顔識別器「Cascade(lbp)」, 検出手法「PF2,  $m=1/8$ 」, 閾値「T=50」における、目識別の有無による特徴点推定結果の違いを図10に示す。図10-(a)-1, 図10-(c)-1, 図10-(c)-1は目識別器を使用していない場合, 図10-(a)-2, 図10-(b)-2, 図10-(c)-2はHaar-like特徴を学習させた目検出器「CascadeE(haar)」を使用している場合である。図中の各丸は特徴点の推定位置を表している。図10-(a)と図10-(b)の4枚では目識別器を利用すると顔を正確に捉えられ、顔の形や色の有無に対しても影響なく特徴点

を抜き出すことに成功していることが見られた。しかし図

表3 目識別器の性能

Table 3 Results table for each eye discriminator.

顔識別器	目識別器	顔検出処理	IoU 平均値	MSE 平均値	MSE 中央値
Cascade (lbp)	なし	PF2, $m=0$	33.06%	2156.68	840.42
		PF2, $m=1/8$	33.55%	2260.36	864.69
	YOLOE	PF2, $m=0$	35.19%	2217.77	814.54
		PF2, $m=1/8$	34.48%	2350.80	880.30
	CascadeE (haar)	PF2, $m=0$	41.76%	1585.25	191.19
		PF2, $m=1/8$	38.71%	1412.50	268.39
CascadeE (lbp)	PF2, $m=0$	37.96%	1784.19	494.71	
	PF2, $m=1/8$	36.17%	1430.90	328.59	

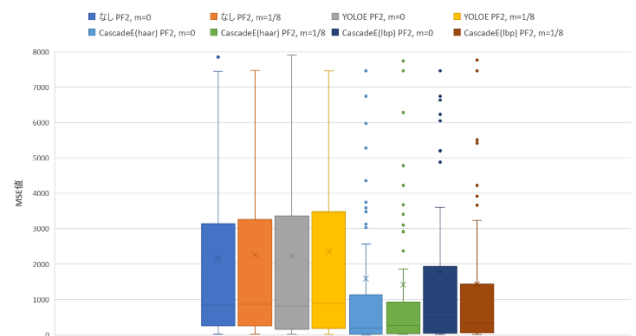


図9 箱ひげ図を用いた各手順でのMSE

Figure 9 MSE for each procedure represented by a box-and-whisker diagram.



図10 テスト画像における特徴点推定の例

Figure 10 Examples of feature point estimation in test images.

10-(c)のように、顔領域をより特定できていたとしても特徴点の推定がずれてしまう結果も見られた。

## 5. おわりに

デジタルイラストの描画補助ツールに使用するため、イラスト内の顔領域を抽出し特徴点座標を推定する一連の処理に関する実験を行った。顔領域を推定する手法としてHough変換を利用した円の検出を行うことで、実験に使用した識別器の多くにおいて検出率やIoUの値が向上することが確認できた。また目検出による顔領域の調整を行うことで、MSEの値が低くなることを確認することができた。また顔識別器に関しては、LBP特徴でManga109のデータ

を学習させたカスケード分類器が、目識別器では Haar-like 特徴で Manga109 のデータを学習させたカスケード分類器が、それぞれ良い結果を出すことがわかった。今後の課題として、入力する対象イラストレーションの種類を増やして提案手法の頑健性調査等が挙げられる。

**謝辞** 実験に使用するテスト画像への使用を快諾して頂いた皆様に、謹んで感謝の意を表する。

## 参考文献

- [1] 戒直哉. 顔のアタリ描画支援システム. 映像情報メディア学会技術報告, 2013, vol. 31, No. 17, p. 27- 30.
- [2] 山田太雅. 人物キャラクターの模写における絵の評価システムの提案. エンタテインメントコンピューティングシンポジウム, 2015, p. 574- 579.
- [3] 相澤清晴. 学術漫画データセットの構築～Manga109～. 映像情報メディア学会誌, 2018, vol. 72, No. 3, p. 358- 362.
- [4] Yusuke Matsui and Kota Ito and Yuji Aramaki and Azuma Fujimoto and Toru Ogawa and Toshihiko Yamasaki and Kiyoharu Aizawa, “Sketch-based Manga Retrieval using Manga109 Dataset”, Multimedia Tools and Applications, 2017, Vol. 76, No. 20, pp.21811-21838
- [5] Kiyoharu Aizawa and Azuma Fujimoto and Atsushi Otsubo and Toru Ogawa and Yusuke Matsui and Koki Tsubota and Hikaru Ikuta, “Building a Manga Dataset “Manga109” with Annotations for Multimedia Applications”, IEEE MultiMedia, 2020, Vol. 27, No. 2, pp.8-18
- [6] Opencv 2.2 documentation objects. 2021/2/3 時点  
URL: [http://opencv.jp/opencv-2svn/py/objdetect\\_cascade\\_classification.html](http://opencv.jp/opencv-2svn/py/objdetect_cascade_classification.html)
- [7] YOLO: Real-Time Object Detection 2021/2/3 時点  
URL : <https://pjreddie.com/darknet/yolo/>
- [8] face-recognition 1.3.0 Project description 2021/2/3 時点  
URL: <https://pypi.org/project/face-recognition/>