

11 画像認識 2 ～顔の認識～

11.1 色特徴を用いた顔検出

近年、さまざまな場面で顔検出や顔認識が利用されるようになってきました。例えば、デジタルカメラの中には、顔を検出することで、ピントを調整する機能を持ったものがあります。また、スマホアプリでは、顔を検出し、耳を付けたり、顔を入れ替えたりといったものもあります。また、個人認証や犯罪捜査などでも、顔認識の利用が期待されています。このような顔検出や顔認識は、どのように行われるのでしょうか。

例えば、私の顔を検出するには、どうすれば良いのでしょうか？ 顔を検出する最も簡単な方法の1つが色情報を利用する方法です。私の顔の領域を見てみると、おでこの辺りも、ほおの辺りも、あごの辺りも同じような色であることが分かります。色で言うとペールオレンジといった感じでしょうか。また、髪の毛の領域に注目すると、私の場合、全体的に黒いことが分かります。このような色の情報は、顔を検出する際の手がかりとして利用できます。

では、顔領域がどのような色分布を持つかが詳しく見てみましょう。図11-1は、画像上で顔の領域を人が指定し、その色分布をとったものです。RGBと書いてあるのは、赤（R）、緑（G）、青（B）のヒストグラムです。HVCと書いてあるのは、色相（H）、明度（V）、彩度（C）のヒストグラムです。RGBのヒストグラムでは、左端が0、右端が255に対応します。このヒストグラムを見ると、RGBの分布が広がっていることが分かります。これは、同じ色のように見える顔の領域でも、光が当たる部分は明るい、すなわちRGBの値が大きく、光が当たらない部分は暗い、すなわちRGBの値が小さくなるためです。このように分布が広い場合、範囲を指定することで顔領域を抽出することは難しくなります。一方、HVCは、RGBを色相、明度、彩度に変換してから、ヒストグラムをとったものです。明度（V）について見てみると、明るい部分と暗い部分である程度幅があることが分かります。一方、彩度（C）を見ると、分布が、彩度が低い部分に集まっていることが分かります。色相（H）は、角度で表現されます。左端が0度、右端が360度になります。色相の分布を見ると、赤い領域にピークが立っています。これは、顔領域の色は赤に近いということで、色相の分布は幅が狭いことから、顔領域の色を指定するのに有効であることが分かります。

一方、図11-2は、髪の毛の色分布をとったものです。明度（V）について見てみると、ある程度、明るさに幅があることが分かります。これは、髪の毛が黒いと言っても、光が当たる部分は明るくなっているためです。色相については、幅広く広がっていることが分かります。黒は無彩色の色であり、色相の抽出が難しくなることが原因と考えられます。一方、RGBについては、値が小さい方に分布が偏っていることが分かります。この結果から、髪の毛領域の抽出には、むしろRGBの値をそのまま使う方が良いことが分かります。

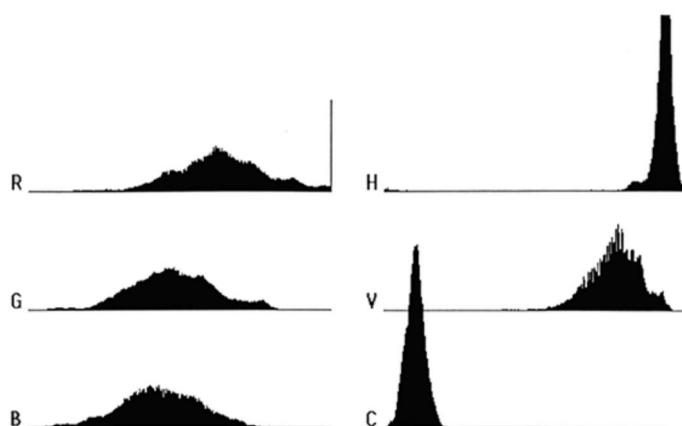


図11-1 顔領域の色分布

図11-2 髪の毛領域の色分布

顔の抽出には、まず、顔の色の範囲を指定し、その色が含まれる領域を抽出します。その中で、ある程度の大きさがあり、正方形に近ければ顔の候補とします。あまりに小さいものや、細長いものは、顔の候補から外してやります。また、髪の毛に対応する領域（今回は黒い領域）を抽出します。そして、その領域が顔の候補領域の上であれば、それらの領域を顔として検出することができます。

11.2 Haar-like特徴を用いた顔検出

顔検出の手法として広く使われる手法の1つとして、特徴量としてHaar-like特徴を用いた手法があります。Haar-like特徴は、単純な白と黒のパターンを使って特徴量の抽出を行います。具体的には、Haar-like特徴は、図11-3のような、白黒で表される矩形領域を画像に重ね合わせることで、特徴量の抽出を行います。ここでは、代表的なパターンのみを示していますが、左上から、横に白黒が並ぶパターン、上下に白黒が並ぶパターン、白黒が斜めに並ぶパターン、白黒白が並ぶパターン、黒を白が囲むパターンなどになります。

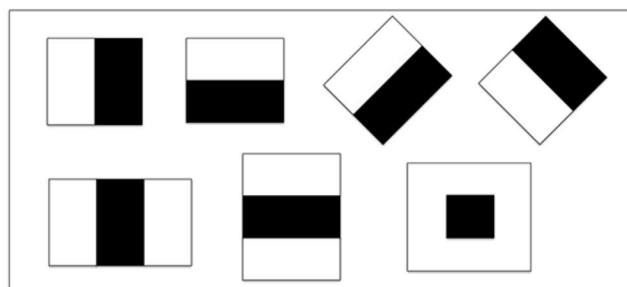


図11-3 Haar-like特徴のパターン

具体的な特徴量抽出は、図11-4のように行います。まず、検出窓を設定します。これは、画像から切り出した認識の対象となる領域です。今、私の顔が映っています。では、実際にどのように特徴量の抽出を行うかですが、先ほど述べた白黒のパターンを画像上に配置します。そして、画像の黒領域に対応する部分の輝度と、白領域に対応する部分の輝度を求めてやります。そしてその差を求めることで、その値を特徴量として利用します。これにより、1つの矩形領域から1つの特徴量を抽出することができます。

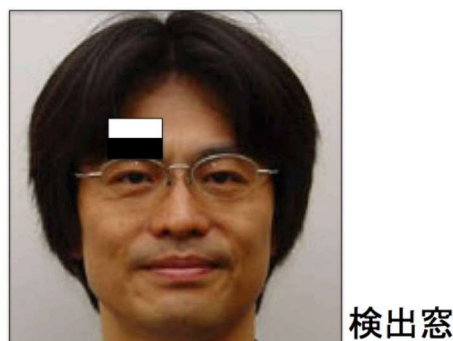


図11-4 Haar-like特徴量の抽出

また、別のパターンを使って特徴量を抽出することができます。もし、パターンの数が10あったとすれば、10個の特徴量を抽出できるわけです。また、同じパターンでも、位置を変えることで異なる特徴量を抽出することができます。例えば、横方向に10段階、縦方向に10段階、位置を変えるとすれば、10×10通りの特徴量を抽出できることになります。大きさを变えることで、また違った特徴量の値を得ることができます。例えば、横方向に10段階、縦方向に10段階大きさを变えるとすれば、10×10通りの特徴量を抽出できることになります。全体では、オーダーとして、約10万通りの特徴量を抽出できることになります。

このようにして抽出した特徴量を使って、どのように顔検出を行うかですが、通常、このような特徴量を用いて、弱検出器と呼ばれるものを構成します。弱検出器は、例えば、顔の検出などに使えるものの、必ずしも精度は高い検出器です。具体的には、得られた特徴量がある閾値以上であれば1、以下であれば-1を返すようにします。顔の検出であれば、1の場合は顔であるということで、-1であれば顔ではないということです。ただし、この時、符号を反転させても構いません。これにより、白黒を反転したパターンも利用できるわけです。

しかし、このような弱検出器は必ずしも精度は高くはありません。このような精度の低い検出器を組み合わせ、精度のより高い検出器を構成するための手法の1つとして、AdaBoostがあります。

AdaBoostでは、まず、データにデフォルトの重みを付けます。そして、検出器ごとのエラーの大きさを計算することで、最も検出性能が良い検出器を選択します。また、そのとき、その検出器の信頼度も合わせて計算します。次に、その検出器で検出できたデータの重みを小さくし、検出できなかったデータの重みを大きくします。そして、再び、検出器ごとのエラーの大きさを計算します。そして、残りの検出器の中から、最も検出性能が良い検出器を選択するという処理を繰り返します。データの重みを変えることで、既に選ばれた検出器が得意ではないデータに対して検出性能が高い検出器を選んでいくということが可能になります。

では、このように求められた検出器によって、どのように検出を行うかですが、検出の際は、求めた検出器の出力に、信頼度を掛けて足し合わせます。そして、その符号がプラスであれば顔、マイナスであれば顔ではないと判断します。

ただし、このようにして得られた検出器は、例えば、顔でないものを顔としてしまうなど、精度が必ずしも十分ではない場合もあります。このような場合、検出器を多段で組み合わせることで、精度の向上を図ることができます。図11-5で、1と書いてあるのは1段目の検出器、2と書いてあるのは2段目の検出器という意味です。まず左から顔画像が入力されるとします。それぞれの検出器は、顔画像をほぼすべて顔画像と検出できるものを選んだとします。

まず、1段目の検出器に顔画像を入力します。顔ではないと判断されると、Fの矢印の方に進み、以降の処理は行われません。しかし、検出器として、顔画像をほぼすべて顔画像と検出するものを選んでいるので、ほとんどすべての顔画像は2段目の検出器の方に進みます。同様に、ほとんどすべての顔画像は、最後の検出器まで通ります。その場合、「顔画像である」と判断します。例えば、顔を99.9%で検出できる検出器を10段重ねたとすると、0.999の10乗が、およそ0.99なので、理屈の上では、顔画像が最後まで残る確率は、99%になるわけです。

次に顔ではない画像を入力した場合を考えてみましょう。左から顔ではない画像を入力していきます。検出器としては、顔ではない画像の半分を顔ではないとする検出器を選んだとします。この場合、検出器を通るたびに、データが半分、半分となっていくので、顔ではない画像が最後まで残る確率は下がり、検出精度の向上が期待できます。例えば、10段、検出器を重ねたとすると、1/2の10乗がおよそ1/1000なので、理屈の上では、顔ではない画像が最後まで残る確率は、1/1000になるわけです。実際には、検出器の独立性が高くないとこのような結果にはなりません、精度の低い検出器を重ねることで、精度の高い検出器を実現できるわけです。

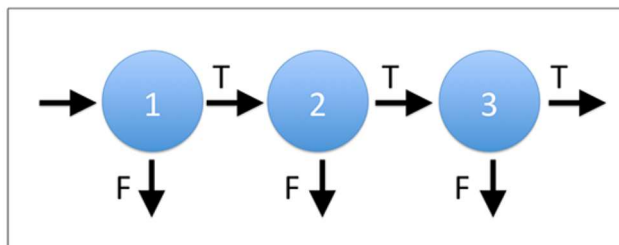


図11-5 カスケード型検出器

11.3 顔の識別

顔を識別する方法として、固有顔についてお話ししたいと思います。これは、平均顔からのずれをそれぞれの顔の特徴とする方法です。

まず、平均顔の方から見ていきましょう。平均顔とは、複数の顔画像を平均化したものです。顔を平均化する場合には、顔の位置や大きさをそろえておく必要があります。例えば、黒目の位置や、鼻の下の点などを基準に、顔の位置や大きさ等をそろえます。そして、画素ごとに色を平均化してやります。

この平均顔からのずれは、それぞれの個人に特有の特徴として利用することができます。このため、それぞれの顔画像から平均顔の画素値を引いてやります。その結果を、画像の左上から順に並べることで、もし、もとの画像が横N画素、縦N画素であったとすれば、 $N \times N$ 次元のベクトルを得ることができます。このベクトルに対して主成分分析を行うことで、固有値と固有ベクトルを求めることができます。この固有ベクトルをもとの横N画素、縦N画素の画像として表現したものが固有顔になります。

固有値が小さい軸を省略することで、それぞれの顔画像は、より次元数が低いベクトルとして表現することができます。このベクトル間の距離により、入力された顔画像が、どの顔画像に近いかを判断することができます。

