画像処理適用のための検証作業支援ツールの開発

Development of support tool for validation to apply image processing

電子・機械技術部 電子・情報科　鈴木健司　三瓶史花

画像処理に関する各種アルゴリズムを簡易に実行でき、パラメータ最適値の検証作業を支援するツールを開発した。開発した支援ツールでは、OpenCV の各種の画像処理アルゴリズムの適用の有無や順番をマウス等の簡単な操作のみで変更することができる。さらに、各アルゴリズムのパラメータ調整が可能なため、パラメータの最適値を検証することも容易である。本開発により作製したツールにより、プログラミング初学者でも画像処理の効果を検証することが可能となった。

Key words: 画像処理、OpenCV、Python、GUI アプリケーション

１．緒言

応募企業のクリナップ株式会社では、住宅設備機器向けの素材に関する研究開発を行っている。研究開発工程では、試作、または開発に用いる素材の良し悪しは、官能評価などの定性的な評価がされることが多いが、より客観的な評価をするためには定量化が重要である。応募企業においても画像処理を用いた定量化評価手法の開発に取り組もうとしているが、画像処理を適用した評価手法の確立には、画像処理アルゴリズムのパラメータを様々に変更し最適化するなど検証作業が必要となる。さらに、画像処理を使いこなすためには、プログラミングの習熟が必要となり、プログラミング初学者にとって画像処理の検証作業に至るまでには時間がかかる。

そこで本開発支援では、プログラミング初学者でも各種画像処理アルゴリズムの検証作業を可能とする、マウス操作のみで画像処理を実行できる支援ツールを開発した。支援ツールはプログラミング言語の一種である Python により実装した。一般の市販ソフトとは違いソースコードを確認できるため、適用した画像処理がプログラム上でどのように実装されているか対比することにより、プログラミングスキルの向上にも寄与するものと期待される。

２．実装する画像処理アルゴリズム

２．１．色調補正

画像処理をするにあたっては、明るさやコントラストなどの色調調整することでその後の処理が実行しやすくなる場合がある。今回は(1)式[1]に従い、パラメータを α、β として入力画像 src に対して出力画像 dst を調整できるようにした。α、β はそれぞれ明るさ、コントラストに対応する。また、別の色調補正としてガンマ補正を選択して検証できるようにした。

$$\mathrm{dst} = \alpha \mathrm{src} + \beta \qquad (1)$$

２．２．平滑化（ぼかし）

収集した画像には、ノイズが入っている場合があるが、その場合は平滑化（ぼかし）処理をすることが有効である。画像処理ライブラリである OpenCV[2] には、平滑化フィルタとして blur フィルタがある。blur フィルタは、画像内の各ピクセルを、ある一定の範囲の周囲のピクセルの平均値に置き換える処理をする。この一定の範囲はカーネルと呼ばれ、通常は正方形である。このカーネルサイズが平滑化処理のパラメータとなり、値が大きくなるほど平滑化の効果も大きくなる。

２．３．鮮鋭化

平滑化処理をした後はノイズを低減することができる一方、輪郭（エッジ）がぼやけてしまう影響がある。ぼやけているエッジに対しては鮮鋭化処理をすることで強調することができる。エッジの部分ではピクセル値変化が大きいことを利用し、元画像と平滑化画像の差分を任意倍率で元画像に重畳することで変化の大きい部分、すなわち輪郭部分が強調される。鮮鋭化処理のパラメータも平滑化処理と同様、カーネルサイズとなり、サイズが大きくなるほど鮮鋭化の効果も大きくなる。

２．４．グレースケール化と二値化

像に対してエッジ検出などをする場合は、グレースケール化や二値化をすることが有効な場合がある。グレースケール化した画像は、(2)式[3]により算出した輝度 Y の 1 チャンネル画像として得られる。

$$\mathrm{Y} = 0.299 \times \mathrm{R} + 0.587 \times \mathrm{G} + 0.114 \times \mathrm{B} \qquad (2)$$

また、二値化はグレースケール化した画像に対して閾値を設定し、各ピクセル値を閾値よりも小さいときは 0（黒色）、大きいときは 255（白色）とすることで得ることができる。

事業名「ひとつ、ひとつ、実現するものづくり企業支援事業」（応募企業　クリナップ株式会社）

２．５．エッジ検出

エッジ検出は、画像のエッジ部分を検出し、輪郭線のみの二値化画像を得ることができる。エッジ検出の手法としては、Canny 法[4]がある。Canny 法ではまずノイズを除去した後、Sobel フィルタ[5]によって輪郭を抽出する。Sobel フィルタによって抽出した輪郭は幅を持った連続値なので、最大値のみを残してその他のピクセルでは値を 0 とする。最後に大小２つの閾値を設定し、大きい閾値によって検出されたエッジと小さい閾値によって検出されたエッジを繋げて最終的な結果を得る。今回は Sobel フィルタのみの抽出結果も表示できるようにした。

３．アプリケーションの実装

３．１．操作画面

アプリケーションの実装には、Python と GUI アプリケーション作製ライブラリである tkinter を用いた。作製したアプリケーションの操作画面を図１に示す。画面左側の“Imaging View”に読み込んだ画像が表示され、画面右側の“Filter & Parameter”に画像処理アルゴリズムのチェックボックスが列挙されている。チェックボックスにチェックを入れるとその画像処理アルゴリズムが適用され、タブが有効になる。タブでは画像処理アルゴリズムのパラメータをスライドバーで調整でき、調整結果は即座に“Imaging View”に反映される。また、アルゴリズムを適用する順番は“Order”テキストボックス内で変更できる。操作画面右下には“Control”パネルがあり、“Save Image”ボタンを押すと現在の“Imaging View”が保存される。次節以降では各アルゴリズムにおけるパラメータ調整について説明する。

３．２．色調補正

色調補正は明るさ、コントラストまたはガンマ補正のどちらで調整するかをラジオボタンで選択する。図２に明るさ、コントラスト、ガンマ補正のパラメータ調整画面を示す。次に選択した対象のパラメータをスライドバーで変更し、“Imaging View”に表示される画像を見ながら最適な値を探る。

３．３．平滑化（ぼかし）と鮮鋭化

平滑化と鮮鋭化はいずれもパラメータがカーネルサイズとなるので、それぞれのタブ内のスライドバーにより調整できるようにした。通常の順番だと平滑化処理をしてノイズを消してから鮮鋭化処理によってぼやけたエッジを際立たせる。

３．４．グレースケール化と二値化

グレースケール化においてはパラメータが存在しないため、“Grayscale and Binarize”のチェックボックスを有効にするとグレースケール化処理される。グレースケール化と二値化のパラメータ画面では二値化を有効にするかのチェックボックスがあり、有効の場合はスライドバーで二値化の閾値を調整することができる。

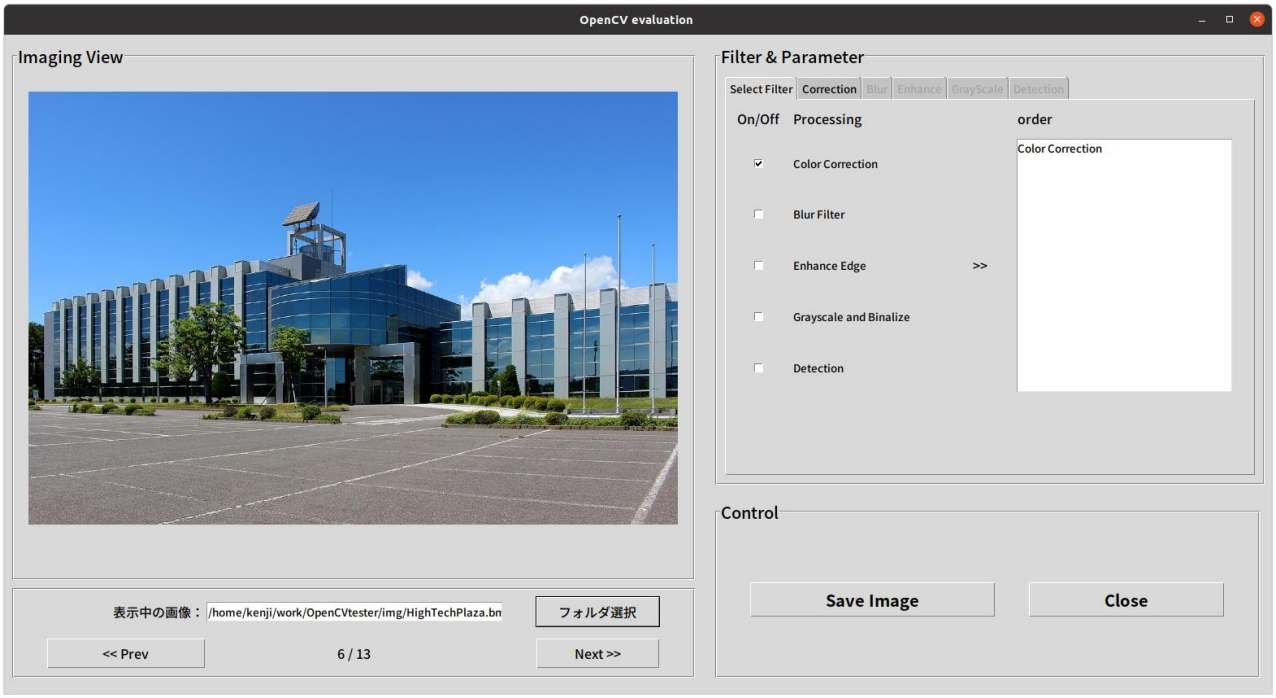

図１　作製したアプリケーションの操作画面

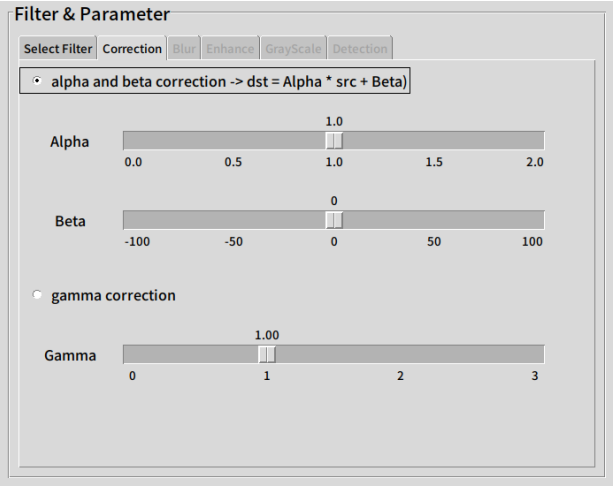

図２　パラメータ調整画面

３．５．エッジ検出

エッジ検出では、Canny 法の２つのパラメータを調整する。いずれもエッジを検出するための閾値であり、値を小さくするとよりエッジを検出しやすくなる。調整の方針としては、１つ目の小さい閾値（Min）で、エッジらしい部分を検出し、２つ目の大きい閾値（Max）で確実にエッジとなる部分を検出する。最終的な結果は２つの閾値で検出された部分がつながる線をエッジとして出力する。図３に画像処理する対象のサンプル画像を、図４に支援ツールで実装された二値化を除くすべての画像処理アルゴリズムを適用した結果を示す。

図３　画像処理対象の画像

図４　画像処理の結果

４．結言

本開発支援では、画像処理ライブラリである OpenCV の各種アルゴリズムのパラメータ調整を簡易に実行するための検証作業支援ツールを作製した。支援ツールは、Python と GUI ライブラリである tkinter を用いて作製し、マウス操作のみで画像処理のパラメータを調整できるようにした。パラメータ調整後の画像は、操作画面上に直ちに表示され、パラメータの最適値を探索しやすくした。また、画像保存機能も実装し、ボタンを押すだけの簡易な操作で結果の保存ができるようにした。本支援ツールを用いることで、画像処理パラメータの調整を直感的に行えるため、作業が効率化されるものと期待される。また、画像処理の検証作業を進めながらソースコード上の実装箇所と対比をすることにより、効率的なプログラミングスキル向上への寄与も期待される。

参考文献

１）OpenCV Documentation. “Changing the contrast and brightness of an image!”. OpenCV 公式ドキュメント. https://pystyle.info/opencv-change-contrast-and-brightness/（参照 2024-02-21）.

２）OpenCV team. “OpenCV - Open Computer Vision Library”. OpenCV 公式ホームページ. https://opencv.org/（参照 2024-02-21）.

３）OpenCV Documentation. “Color conversions”. OpenCV 公式ドキュメント. https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html（参照 2024-02-21）.

４）OpenCV Documentation. “Canny Edge Detection”. OpenCV 公式ドキュメント. https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html（参照 2024-02-21）.

５）OpenCV Documentation. “Sobel Derivatives”. OpenCV 公式ドキュメント. https://docs.opencv.org/3.4/d2/d2c/tutorial_sobel_derivatives.html（参照 2024-02-21）.