レポート課題(しきい値処理によるイメージの 2 値化)

1.「グローバルしきい値を使用したイメージの 2 値化」と「局所的な適応しきい値処理を使用したイメージの 2 値化」の比較(「report01.m」参照)

ここでは,「グローバルしきい値を使用したイメージの 2 値化」と「局所的な適応しきい値処理を使用したイメージの 2 値化」を用いた画像処理を行い，その結果をもとに考察を行う.

今回は,「TDU.JPG」を原画像とする. この画像はレポート作成者が撮影したものであり, 縦 1095 画素，横 730 画素による長方形のディジタルカラー画像である.

まず,

```
ORG=imread(' TDU.JPG'); % 原画像の入力
imagesc(ORG); axis image; % 画像の表示
```

によって，原画像を読み込み，表示した結果を図 1 に示す.

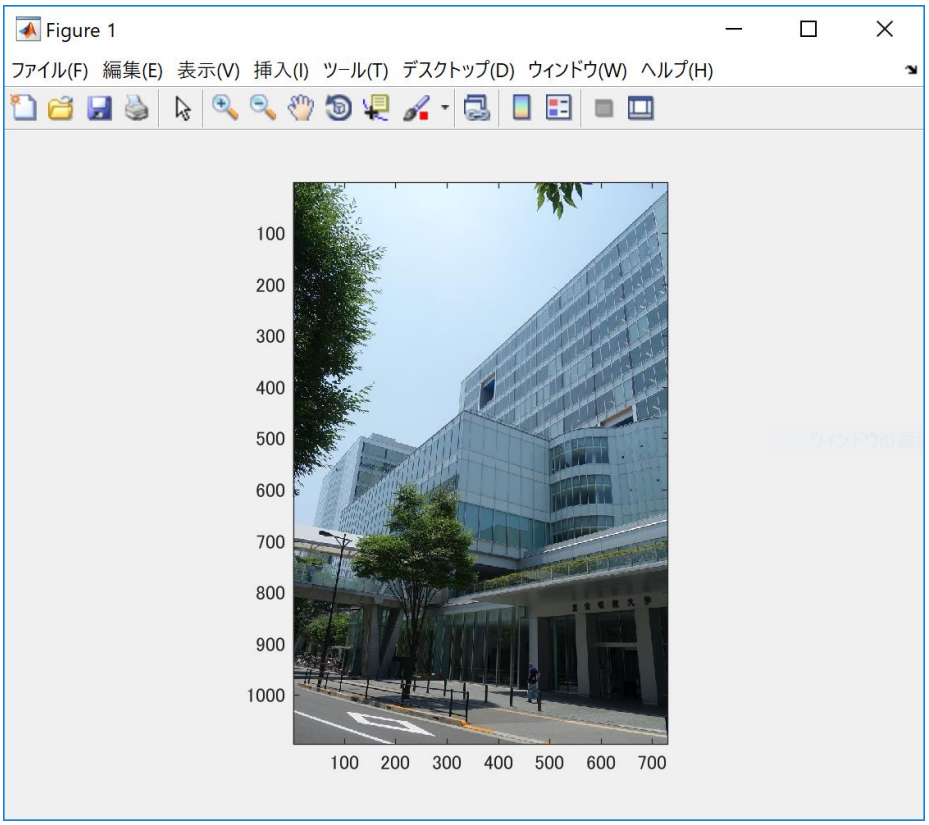

図 1　原画像

次に，

```
ORG=rgb2gray(ORG); % カラー画像をグレースケール画像へ変換
imagesc(ORG); colormap(gray); colorbar;% 変換後の画像を表示
```

によって，カラー画像である原画像をグレースケール画像へ変換し，表示した結果を図 2 に示す．

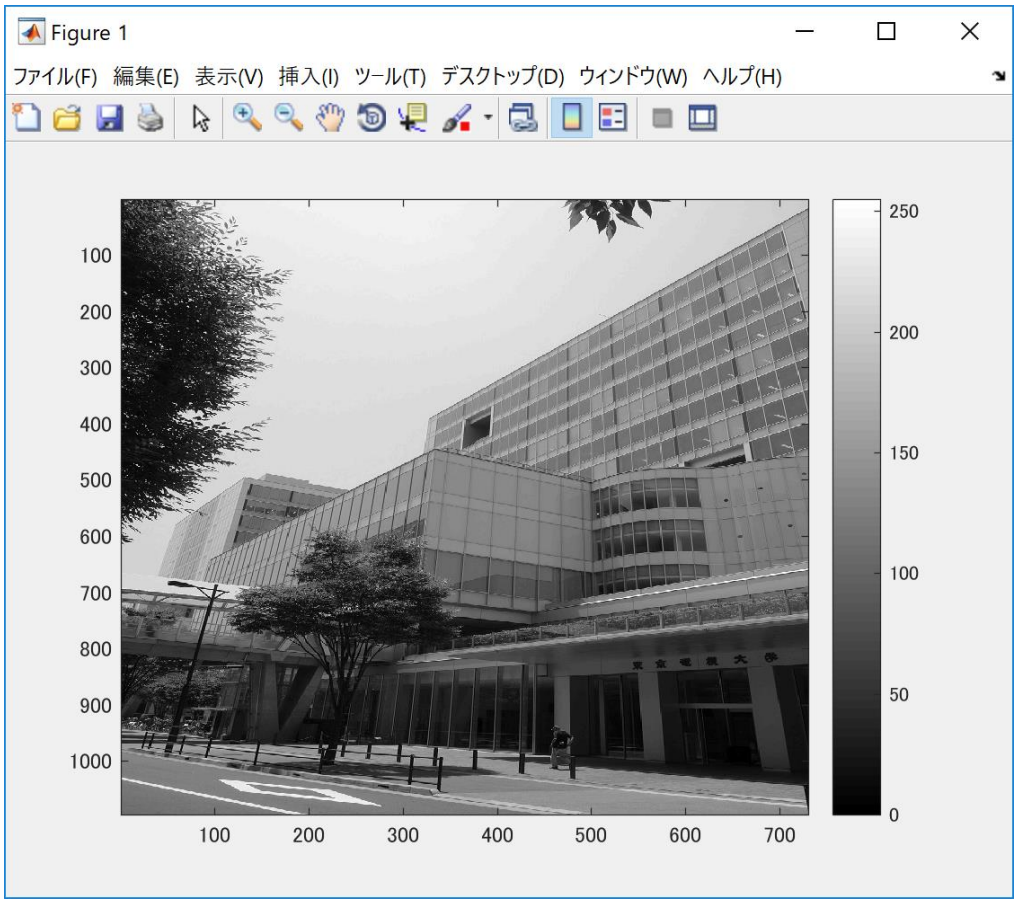

図 2　グレースケール変換を行った原画像

そして，

```
imwrite(ORG, 'TDU_gray.JPG');% 変換後の画像に名前をつけて保存
```

によって，変換後の画像に「TDU_gray.JPG」という名前をつけて保存する．

ここからは，実際に「しきい値処理によるイメージの 2 値化」を行う．はじめに「グローバルしきい値を使用したイメージの 2 値化」を行う．

まず，

```
I = imread('TDU_gray.JPG'); % グレースケール イメージをワークスペースに読み取り
```

によって，「TDU_gray.JPG」をワークスペースに読み取り，

```
BW = imbinarize(I);% イメージをバイナリ イメージに変換
figure
imshowpair(I,BW,'montage')%元のイメージの横にバイナリ バージョンを表示
```

によって，イメージをバイナリ イメージに変換し，元のイメージの横にバイナリ バージョンを表示した結果を図 3 に示す．

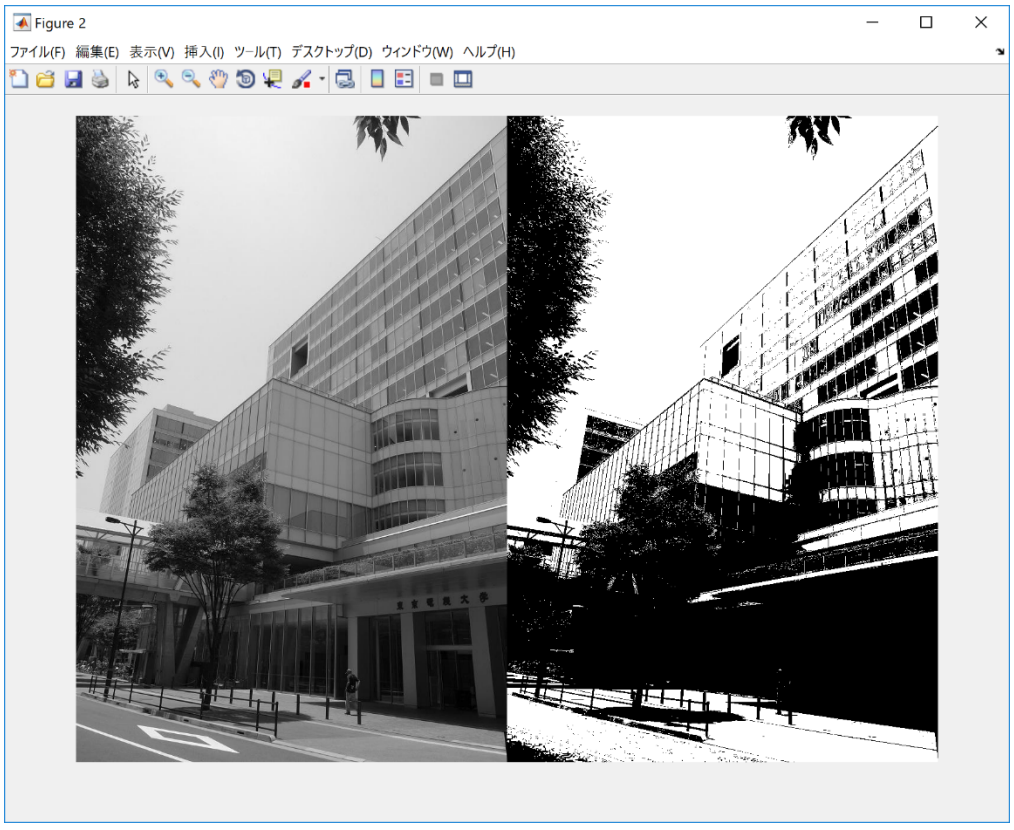

図 3　グローバルしきい値を使用したイメージの 2 値化

次に，「局所的な適応しきい値処理を使用したイメージの 2 値化」を行う．
まず，

I = imread('TDU_gray.JPG'); % グレースケール イメージをワークスペースに読み取り

によって，「TDU_gray.JPG」をワークスペースに読み取り，

BW = imbinarize(I, 'adaptive'); % グレースケール イメージをバイナリ イメージに変換
figure
imshowpair(I,BW,'montage')% 元のイメージとバイナリ バージョンを並べて表示

によって，イメージをバイナリ イメージに変換し，元のイメージの横にバイナリ バージョンを表示した結果を図 4 に示す．

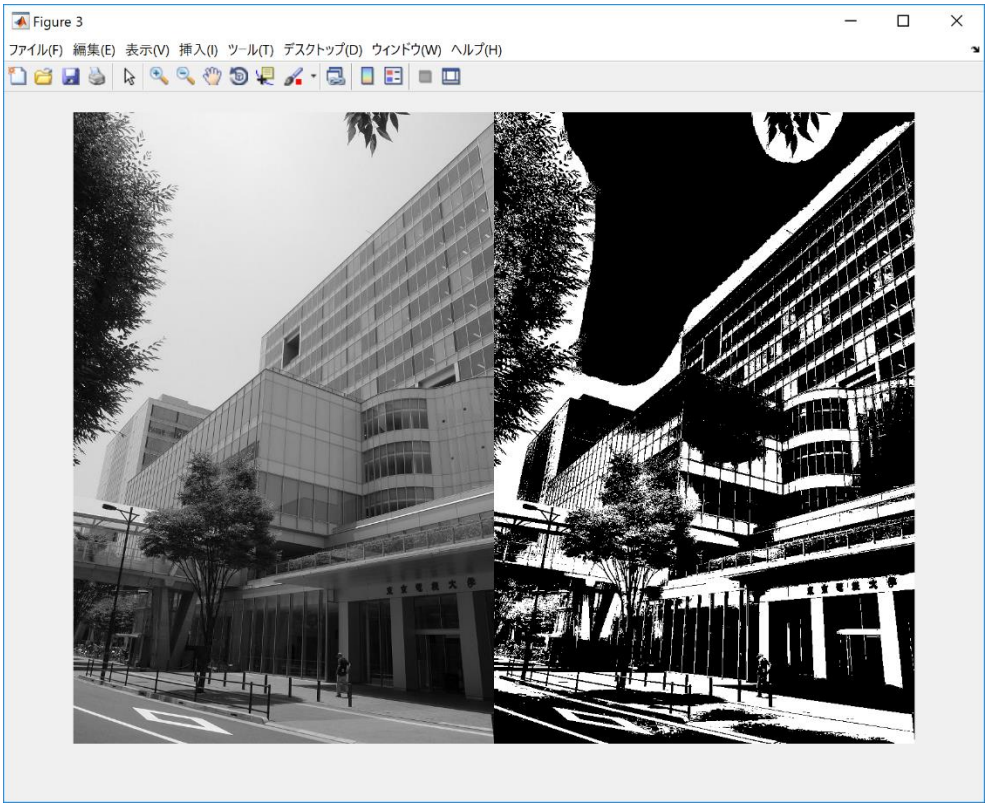

図 4　局所的な適応しきい値処理を使用したイメージの 2 値化

以上のように,「グローバルしきい値を使用したイメージの 2 値化」はそれぞれの部分を面として捉えて 2 値化を行っており，上半分は構造物の姿が大まかにわかるものの，下半分はほとんど分からない．一方,「局所的な適応しきい値処理を使用したイメージの 2 値化」はそれぞれの部分をより細かく捉えて 2 値化を行っており，構造物全体の姿が大まかに分かるものの，空の部分が一面黒くなっている．これらはそれぞれの「2 値化」が，各々目的とするものが異なっていることを示している．

このことから,「グローバルしきい値を使用したイメージの 2 値化」と「局所的な適応しきい値処理を使用したイメージの 2 値化」は各々が異なる使用目的を持っていることから,それらを考慮しつつ用いるべきであると考える．

2. 「前景が背景より暗いイメージの 2 値化」(「report02.m」参照)

ここでは,「前景が背景より暗いイメージの 2 値化」を用いた画像処理を行い，その結果をもとに考察を行う.

今回は,「Document.JPG」を原画像とする．この画像はレポート作成者が撮影したものであり，縦 1306 画素，横 980 画素による長方形のディジタルカラー画像である.

まず,

```
ORG=imread(' Document.JPG'); % 原画像の入力
imagesc(ORG); axis image; % 画像の表示
```

によって，原画像を読み込み，表示した結果を図 5 に示す.

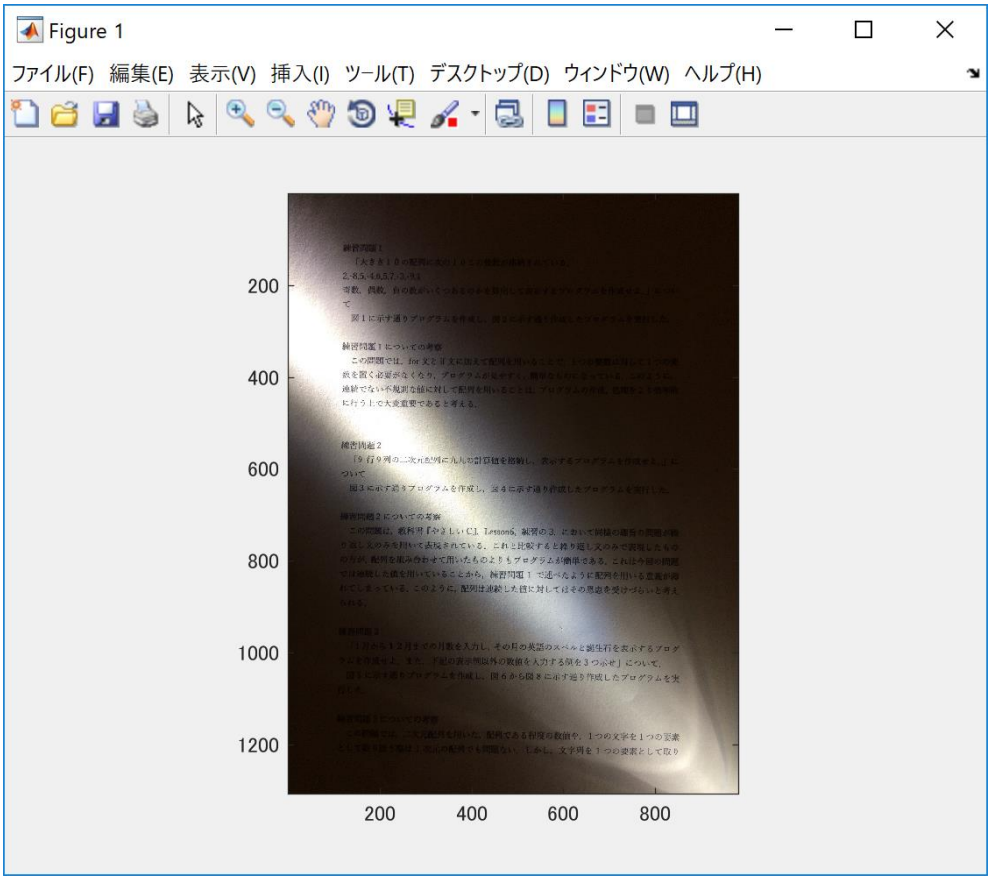

図 5　原画像

次に，

ORG=rgb2gray(ORG); % カラー画像をグレースケール画像へ変換
imagesc(ORG); colormap(gray); colorbar;% 変換後の画像を表示

によって，カラー画像である原画像をグレースケール画像へ変換し，表示した結果を図 6 に示す．

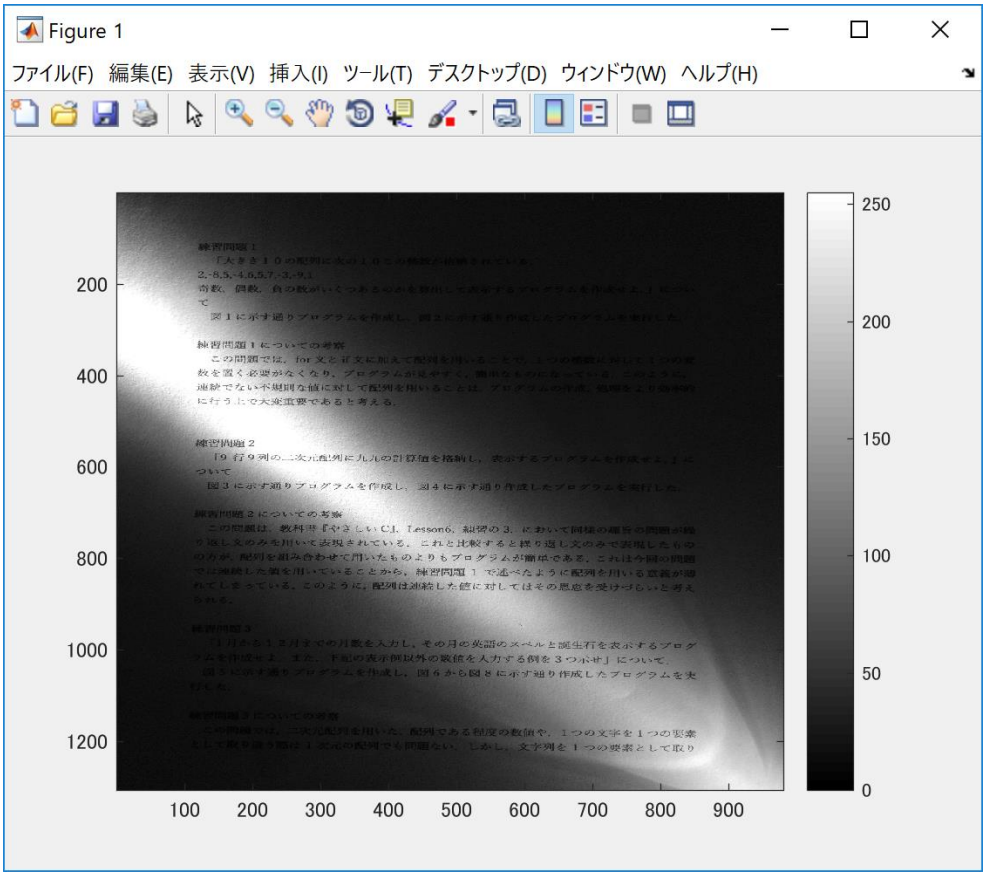

図 6　グレースケール変換を行った原画像

そして，

imwrite(ORG, ' Document_gray.JPG');% 変換後の画像に名前をつけて保存

によって，変換後の画像に「Document_gray.JPG」という名前をつけて保存する．

ここからは，実際に「前景が背景より暗いイメージの 2 値化」を行う．
まず，

```
I = imread('Document_gray.JPG'); %グレースケール イメージをワークスペースに読み取って表示
figure
imshow(I)
title('Original Image')
```

によって，グレースケール イメージをワークスペースに読み取って表示した結果を図 7 に示す．

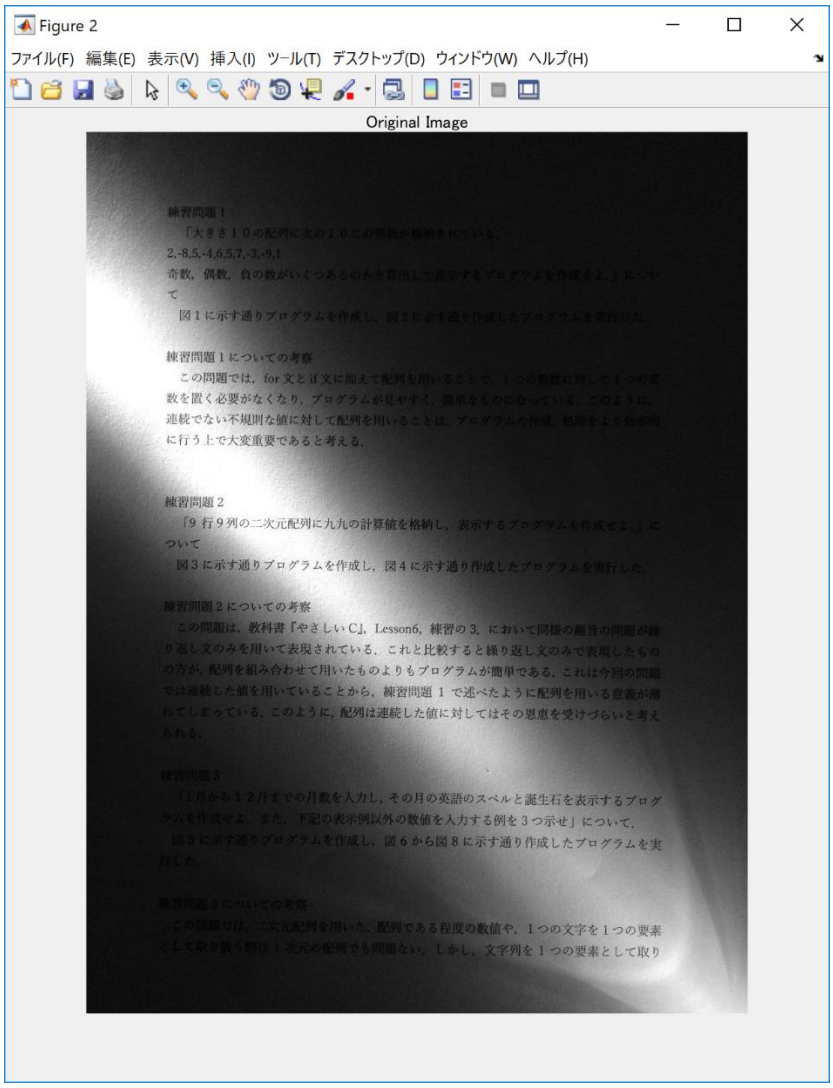

図 7　「前景が背景より暗いイメージの 2 値化」を行う画像

次に，

```
BW = imbinarize(I,'adaptive','ForegroundPolarity','dark','Sensitivity',0.4); %適応しきい値処理を使用してイメージをバイナリ イメージに変換
```

によって，適応しきい値処理を使用してイメージをバイナリ イメージに変換する．このとき，前景が背景より暗いことを示すために，ForegroundPolarity パラメーターを使用する．

そして，

```
figure
imshow(BW) %イメージのバイナリ バージョンを表示
title('Binary Version of Image')
```

によって，イメージのバイナリ バージョンを表示した結果を図 8 に示す．

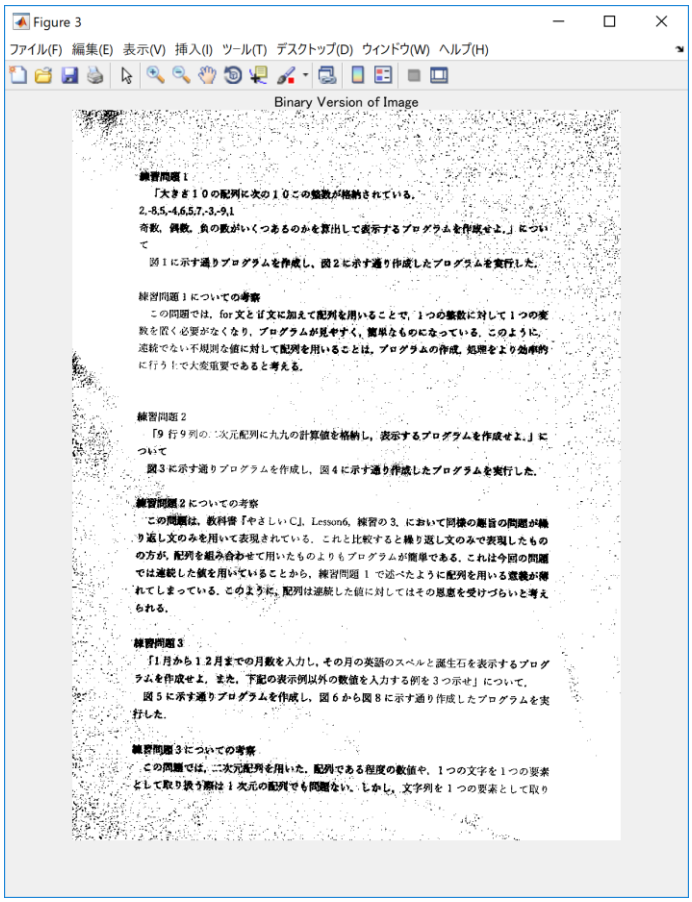

図 8　前景が背景より暗いイメージの 2 値化

以上のように，元画像では暗くて見えていなかった部分の文字が「前景が背景より暗いイメージの 2 値化」を用いることによって解読できるようになった．

このことから，「前景が背景より暗いイメージの 2 値化」は，一部が暗くなってしまい人間では解読が難しい画像を読み解く際に，大きな力を発揮するものであると考えられる．

引用文献

「しきい値処理によるイメージの 2 値化」，MathWorks，
https://jp.mathworks.com/help/images/ref/imbinarize.html