Regional maxima

- 局所的最大値の定義
  - 局所的な最大値となる条件
    - 同じピクセルの値を持つ連結要素ピクセルである.
    - そのピクセルの値より小さい値を持つピクセルに囲まれている.
  - 例として6×6の画像について局所的な最大値をとる.

0	2	3	3	1	0
0	0	0	0	0	0
0	0	5	0	0	0
0	0	0	0	0	0
4	0	0	0	0	0
4	5	0	0	0	0

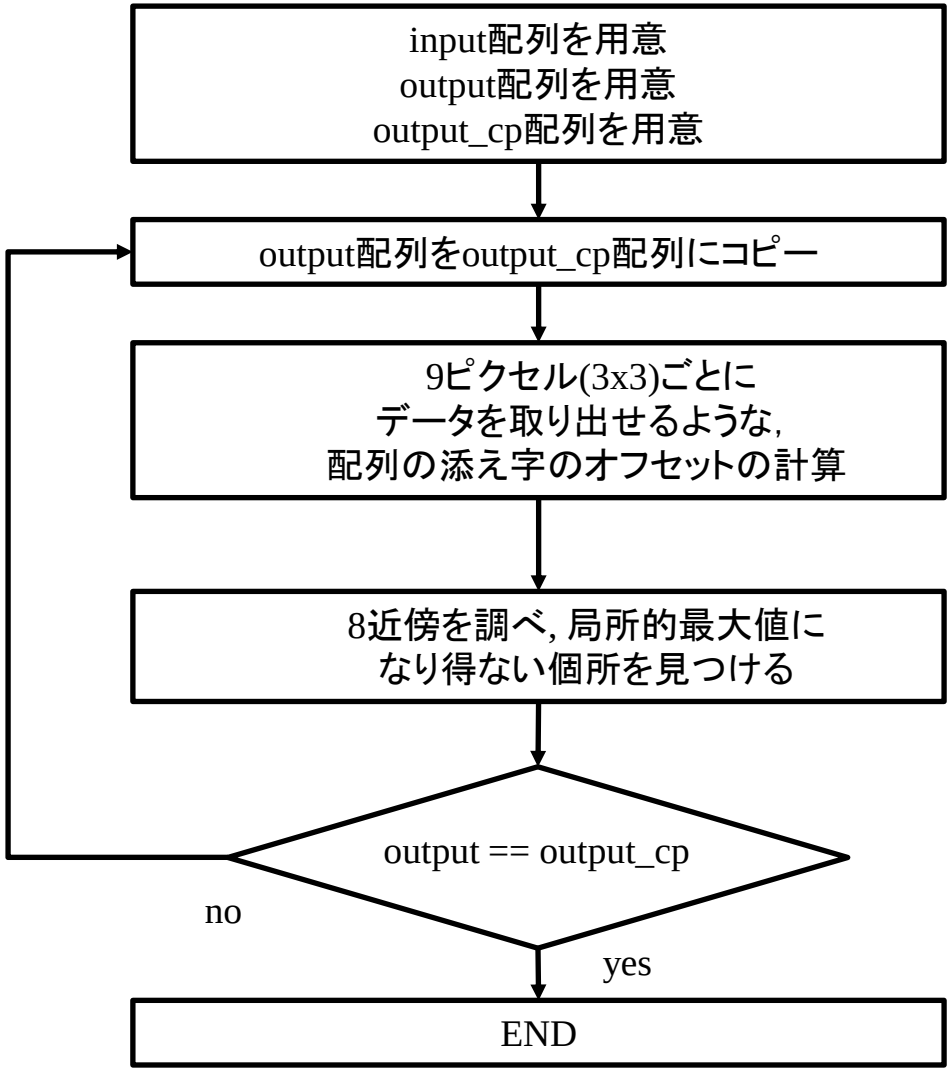
Input[36]

0	0	1	1	0	0
0	0	0	0	0	0
0	0	1	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	1	0	0	0	0

output[36]

Algorithm

フローチャート



Algorithm

処理の流れ(1/6)

- 入力画像を格納する配列 input と true(1) で初期化された配列 output, 一時的に利用する配列 output_cp を用意する

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

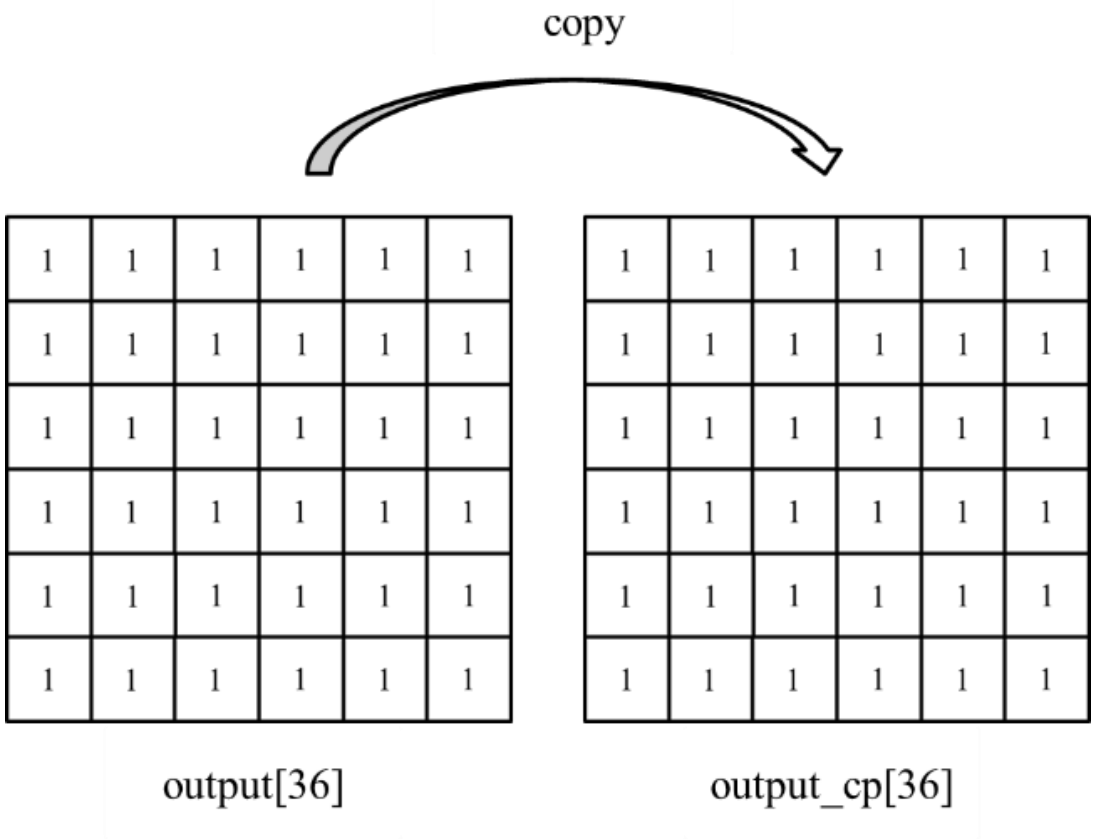
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

Algorithm

処理の流れ(2/6)

- output を output_cp にコピーする



Algorithm

処理の流れ(3/6)

- 9ピクセル(3x3)ごとにデータを取り出せるような，配列の添え字のオフセットの計算を行う.(赤枠部分)

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

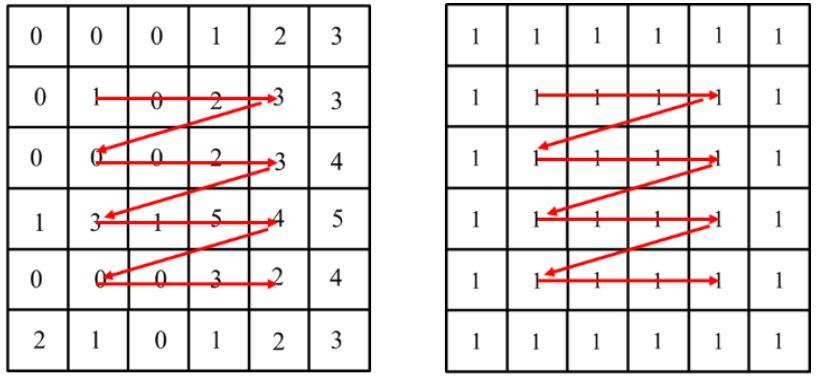
output[36]

Algorithm

処理の流れ(4/6)

- 入力画像の m 行 n 列成分(m = [2,3, …5], n = [2,3, …5])に着目し, 8近傍と順に比較して, outputのm 行 n 列成分がtrueの場合に, 最大になる可能性があるか調べる.

- 調べた結果, 中央のピクセルが最大値になる可能性がない場合にoutputのm行n列成分の値をtrueからfalseに更新.

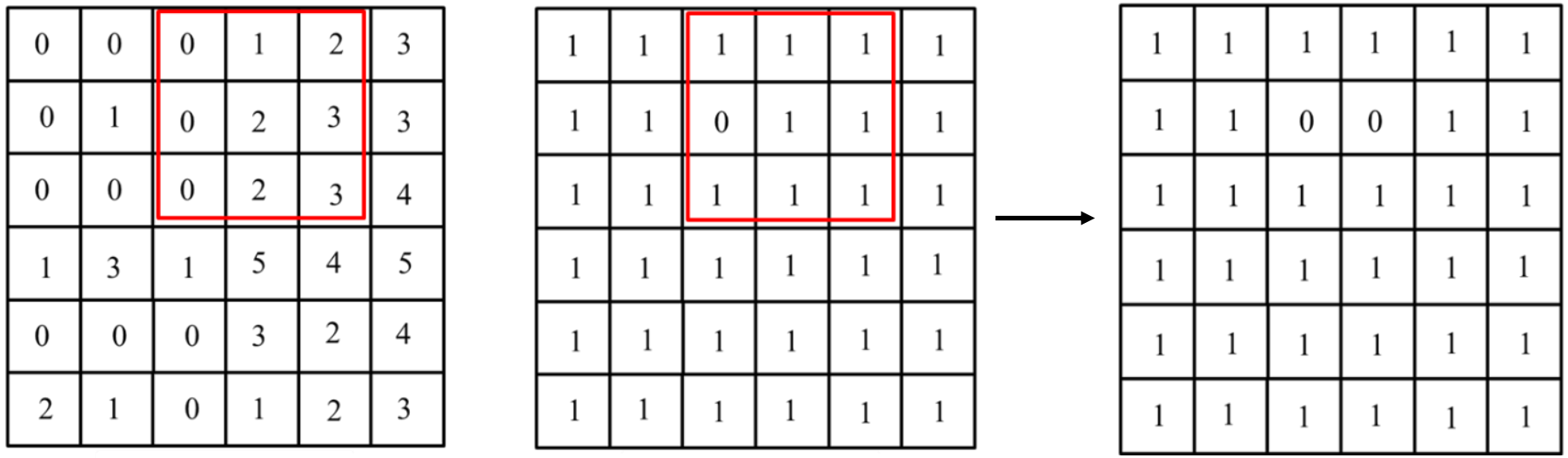


input[36] output[36]

Algorithm

- 局所的最大値を取るか否かの判定
  - 以下のいずれかの条件を満たした場合，最大値になる可能性がなくなる
    - 条件(1)

(任意の8近傍にあるピクセルAの値) > (中央のピクセルの値)が真



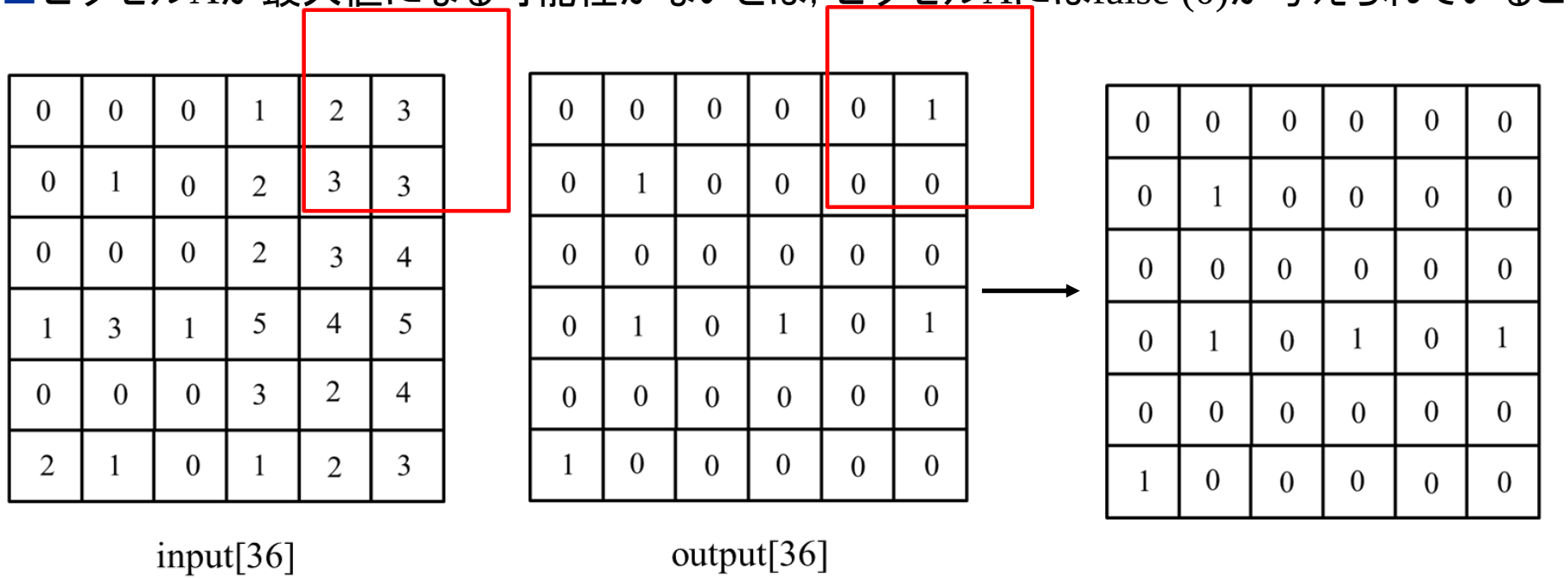
Algorithm

■ 局所的最大値を取るか否かの判定

■ 条件(2)

(任意の8近傍になるピクセルAの値) = (中央のピクセルの値) が真であり, かつピクセルAが最大値になる可能性がない.

- ピクセルAが最大値になる可能性がないとは, ピクセルAにはfalse (0)が与えられていることである.



Algorithm

処理の流れ(5/6)

- 同様に以下の成分に対しても8近傍と順に比較して, 着目しているピクセルの値がtrueの場合に, 最大になる可能性があるか調べる.
  - 入力画像の m 行 n 列成分(m=1, n=[1, 2, …, 6])
  - 入力画像の m 行 n 列成分(m=6, n=[1, 2, …, 6])
  - 入力画像の m 行 n 列成分(m=[2, …, 5], n=1)
  - 入力画像の m 行 n 列成分(m=[2, …, 5], n=6)

Algorithm

- 処理の流れ(6/6)
  - output[36]とoutput_cp[36]を比較する
  - 比較した結果, outputとoutput_cpが等しければ処理を終了.異なっていれば(2/6)の処理へ戻る.

Example

- 例
  - 6×6の画像について一つずつ確認を行った

Example

- 入力画像 input[36]= [0,0,0,1,2,3; 0,1,0,2,3,3; 0,0,0,2,3,4; 1,3,1,5,4,5; 0,0,0,3,2,4; 2,1,0,1,2,3]
- 出力画像 output[36]

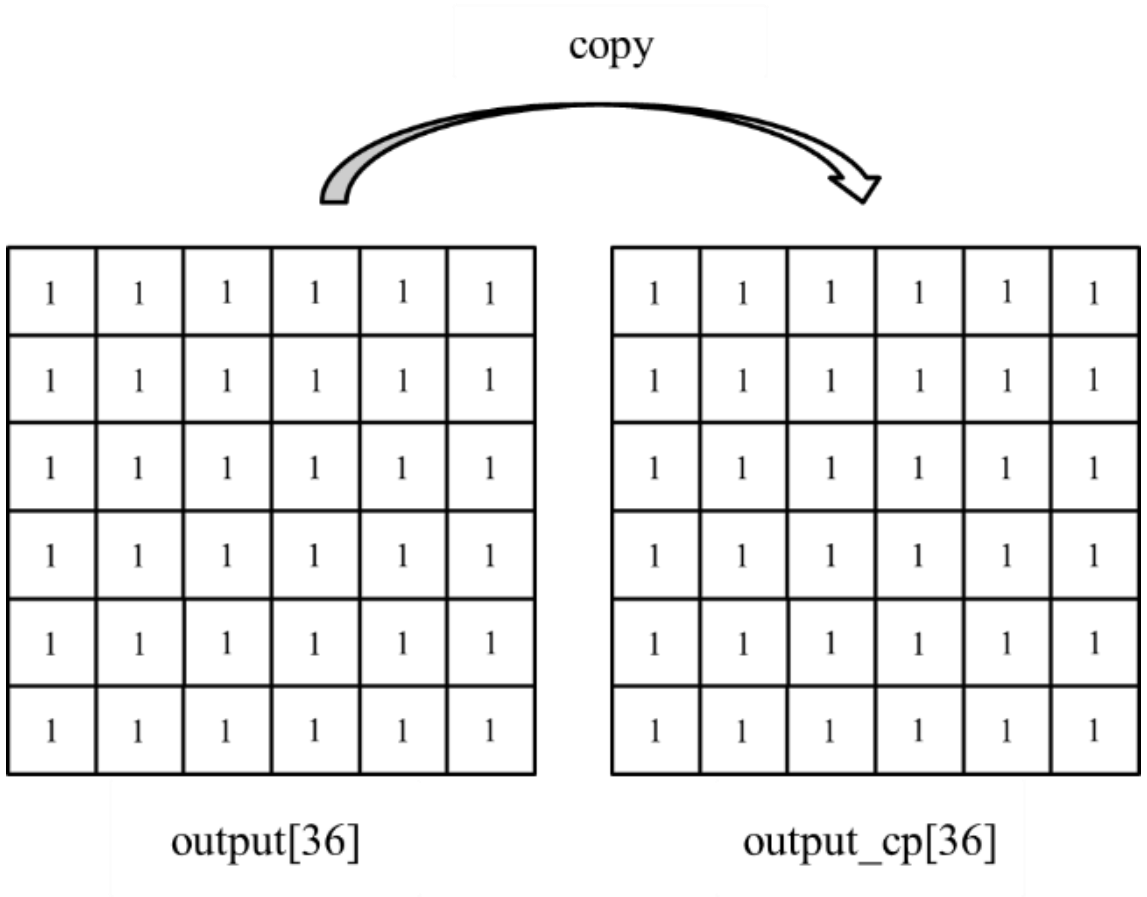
0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

- outputをcp_outputにコピーする



- 9ピクセル(3x3)ごとにデータを取り出せるような，配列の添え字のオフセットの計算を行う.(赤枠部分)

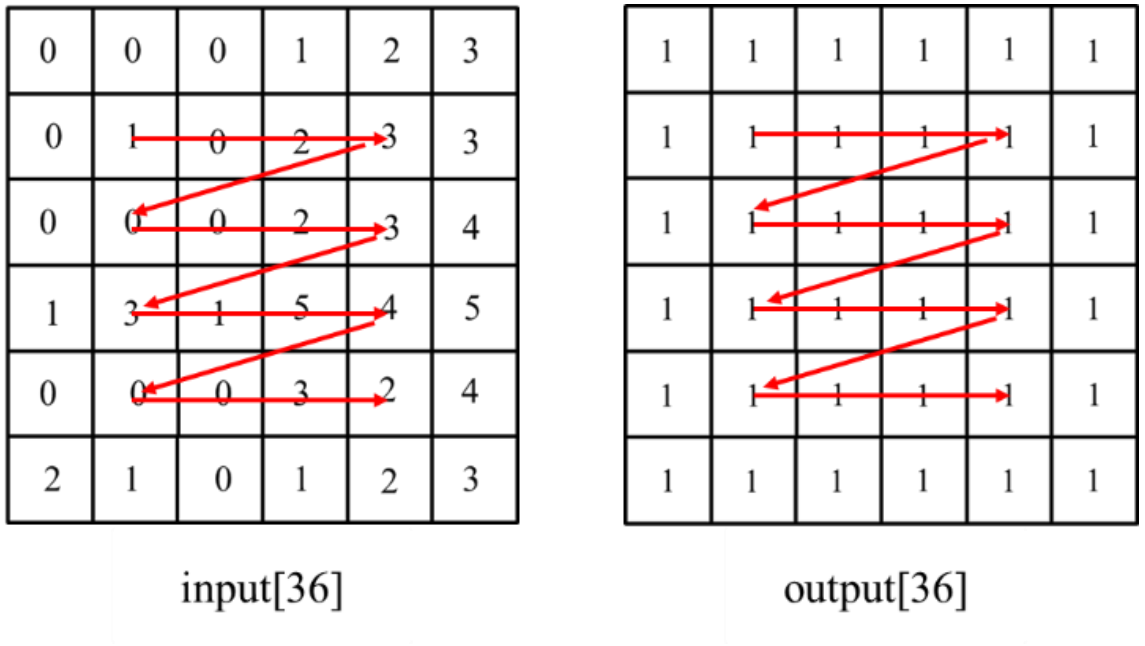
0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

- 入力画像の m 行 n 列成分(m=[2:5], n=[2:5])に着目し,　8近傍と順に比較して, 着目しているピクセルの値がtrueの場合に, 最大になる可能性があるか調べ, outputの値を更新する.



- 中央のピクセルは8近傍のいずれのピクセルよりも値が大きいため，outputの中央のピクセルは”true(=1)”のまま更新されない.（条件(1)も(2)も該当しない）

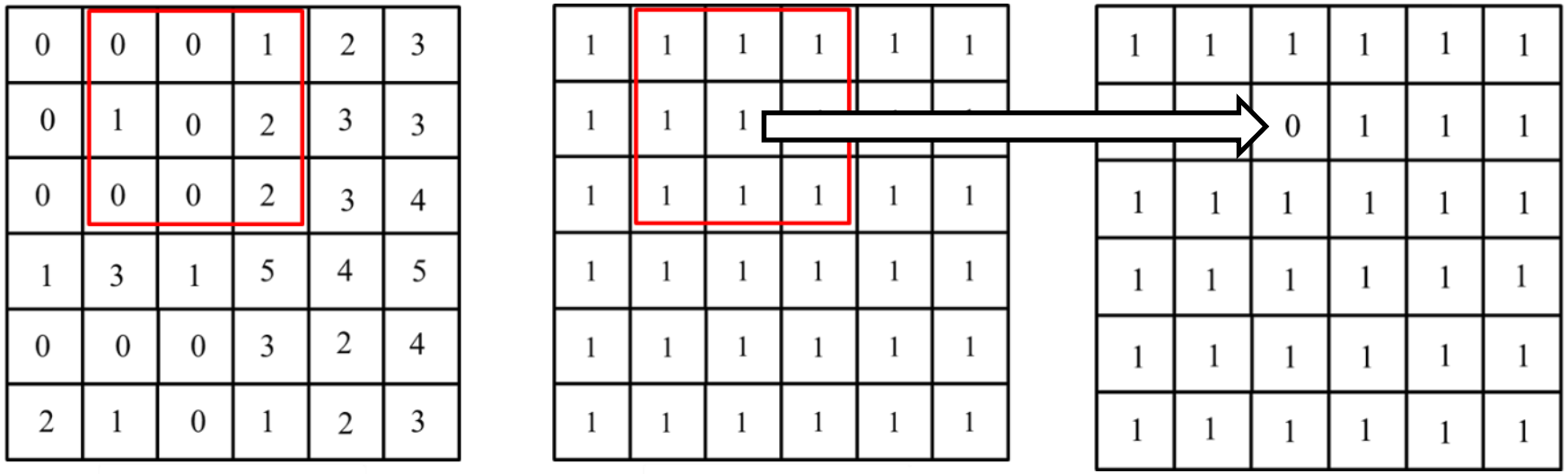
0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

- 以下の図では，条件(1) が該当し，outputが更新される．同様の更新処理を続けていく．



- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

1	1	1	1	1	1
1	1	0	0	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	0	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

- 中央のピクセルは8近傍のいずれのピクセルよりも値が大きいため，outputの中央のピクセルは”true(=1)”のまま更新されない.（条件(1)も(2)も該当しない）

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

- 中央のピクセルは8近傍のいずれのピクセルよりも値が大きいため，outputの2行2列成分は”true(=1)”のまま更新されない.（条件(1)も(2)も該当しない）

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	1	1	1	1	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	1	1	1	1	1
1	1	1	1	1	1

output[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	1	1	1	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

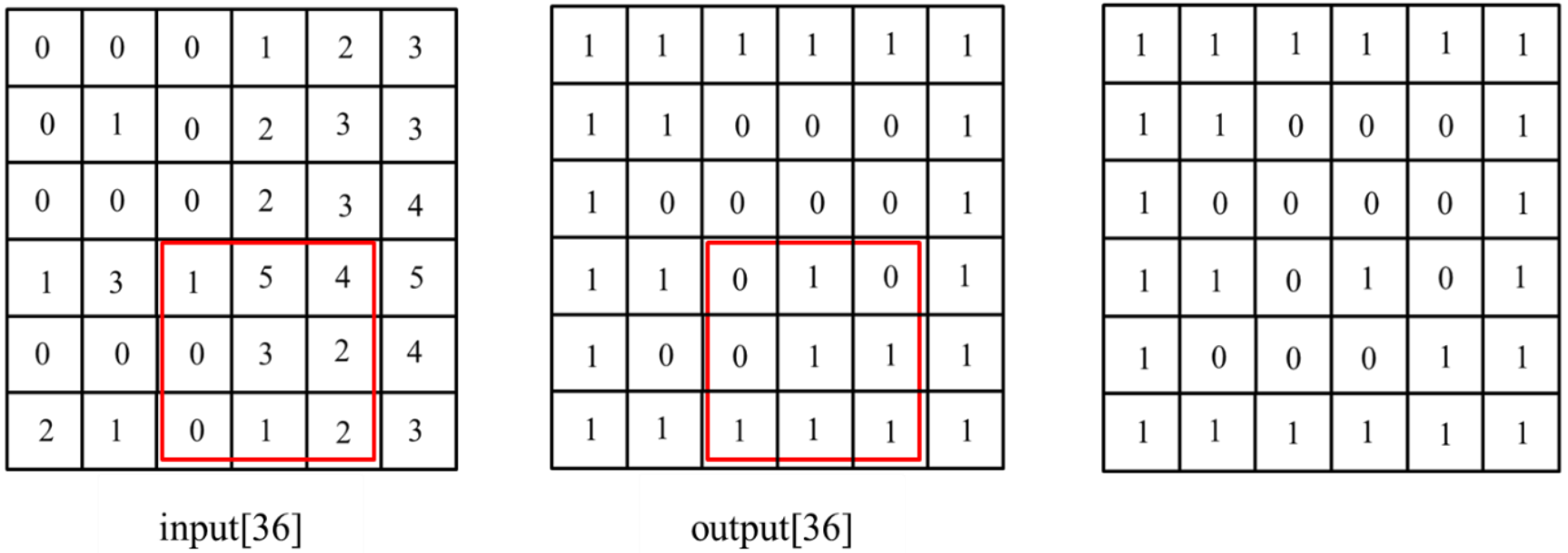
input[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	1	1	1	1
1	1	1	1	1	1

output[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	1	1	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.



0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	1	1	1
1	1	1	1	1	1

output[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	1	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	1	1
1	1	1	1	1	1

output[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

上辺の処理

- 入力画像の m 行 n 列成分(m=1, n=[1:6])に着目し,　8近傍と順に比較して, 着目しているピクセルの値がtrueの場合に, 最大になる可能性があるか調べ, outputの値を更新する.

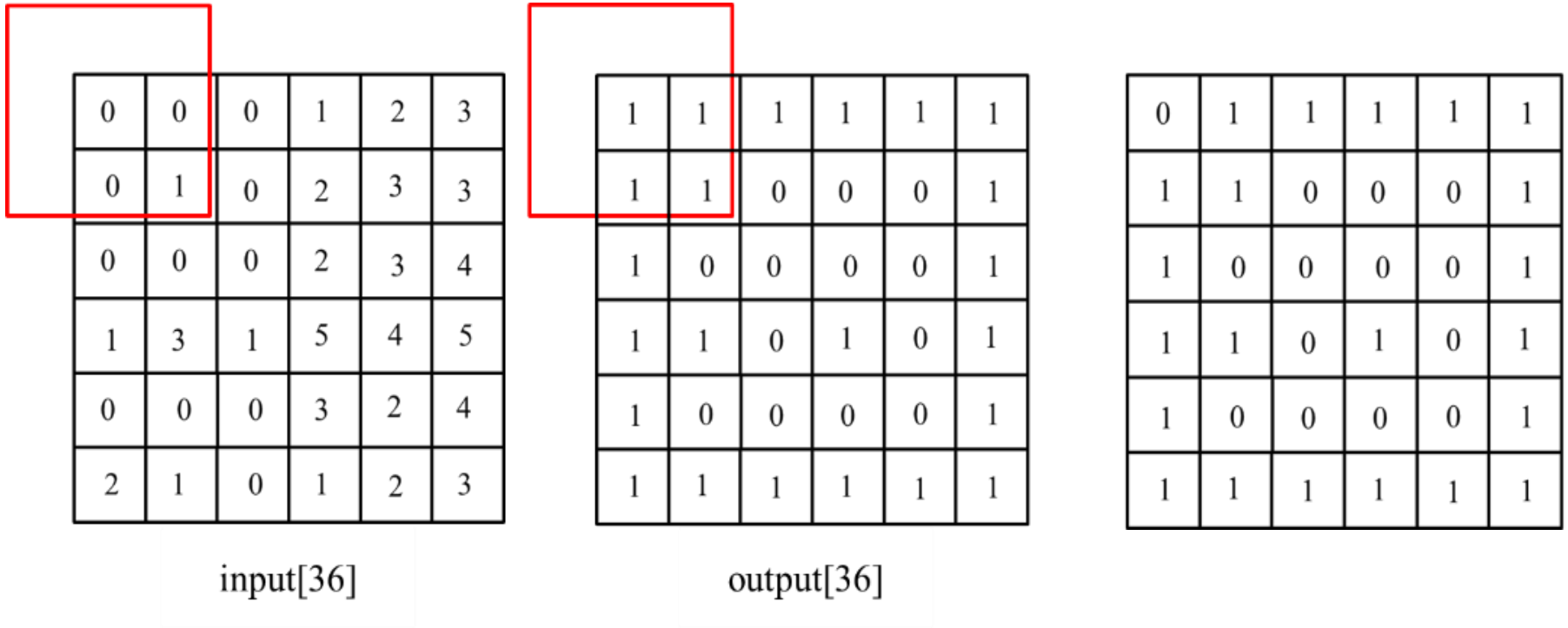
0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	1	1	2	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

output[36]

- 以下の図では，条件(1) が該当し，outputが更新される.
  - 領域外の値は利用しない.



0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

1	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

output[36]

0	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

0	1	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

output[36]

0	0	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

0	0	1	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

output[36]

0	0	0	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

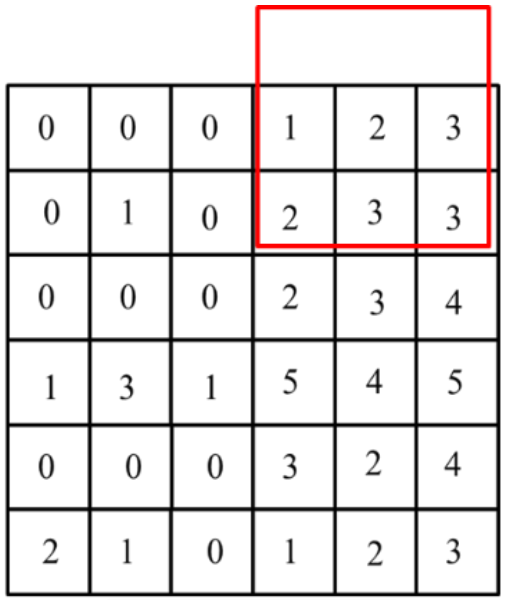
input[36]

0	0	0	1	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

output[36]

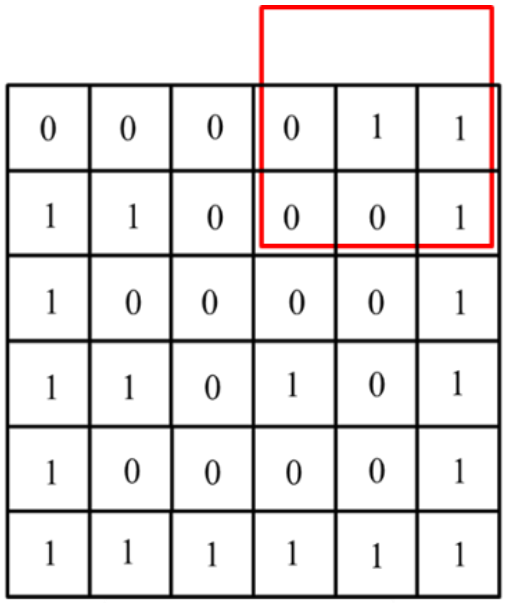
0	0	0	0	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.



0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

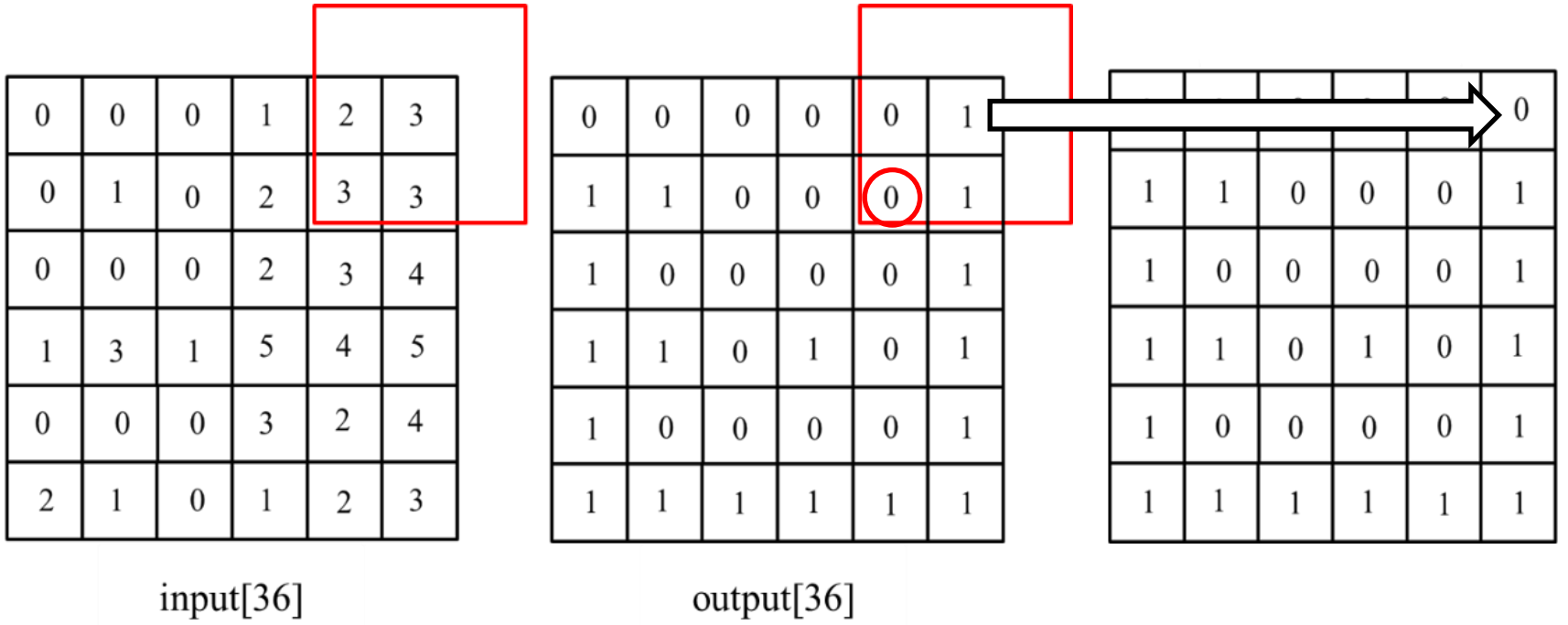


0	0	0	0	1	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

output[36]

0	0	0	0	0	1
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

- (任意の8近傍になるピクセルAの値) = (中央のピクセルの値) が真であり, かつピクセルAが最大値になる可能性がないため，outputが更新される.（条件(2)に該当）



下辺の処理

- 入力画像の m 行 n 列成分(m=6, n=[1:6])に着目し,　8近傍と順に比較して, 着目しているピクセルの値がtrueの場合に, 最大になる可能性があるか調べ, outputの値を更新する.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	1	1	2	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

0	0	0	0	0	0
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

output[36]

- 中央のピクセルは8近傍のいずれのピクセルよりも値が大きいため，outputの中央のピクセルは”true(=1)”のまま更新されない.（条件(1)も(2)も該当しない）
  - 領域外の値は利用しない.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

0	0	0	0	0	0
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

output[36]

0	0	0	0	0	0
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

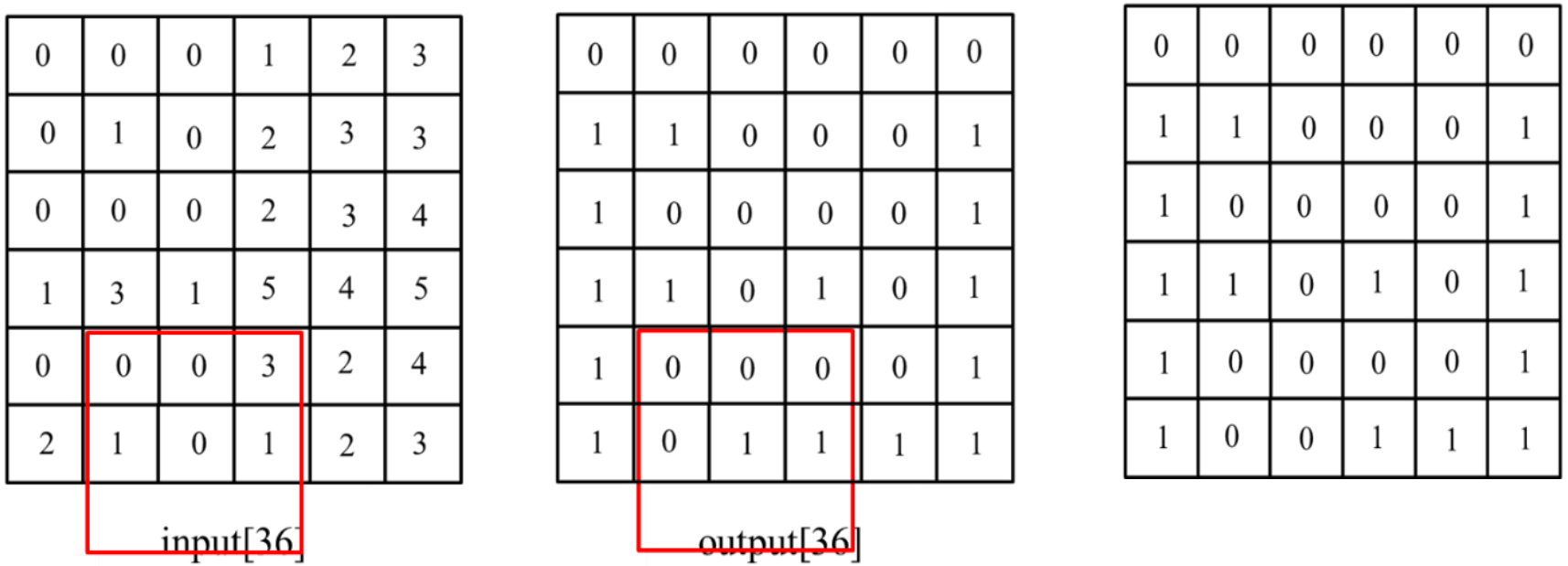
input[36]

0	0	0	0	0	0
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	1	1	1	1	1

output[36]

0	0	0	0	0	0
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	0	1	1	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.



- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

nput[36]

0	0	0	0	0	0
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	0	0	1	1	1

output[36]

0	0	0	0	0	0
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	0	0	0	1	1

- 以下の図では，条件(1) が該当し，outputが更新される.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

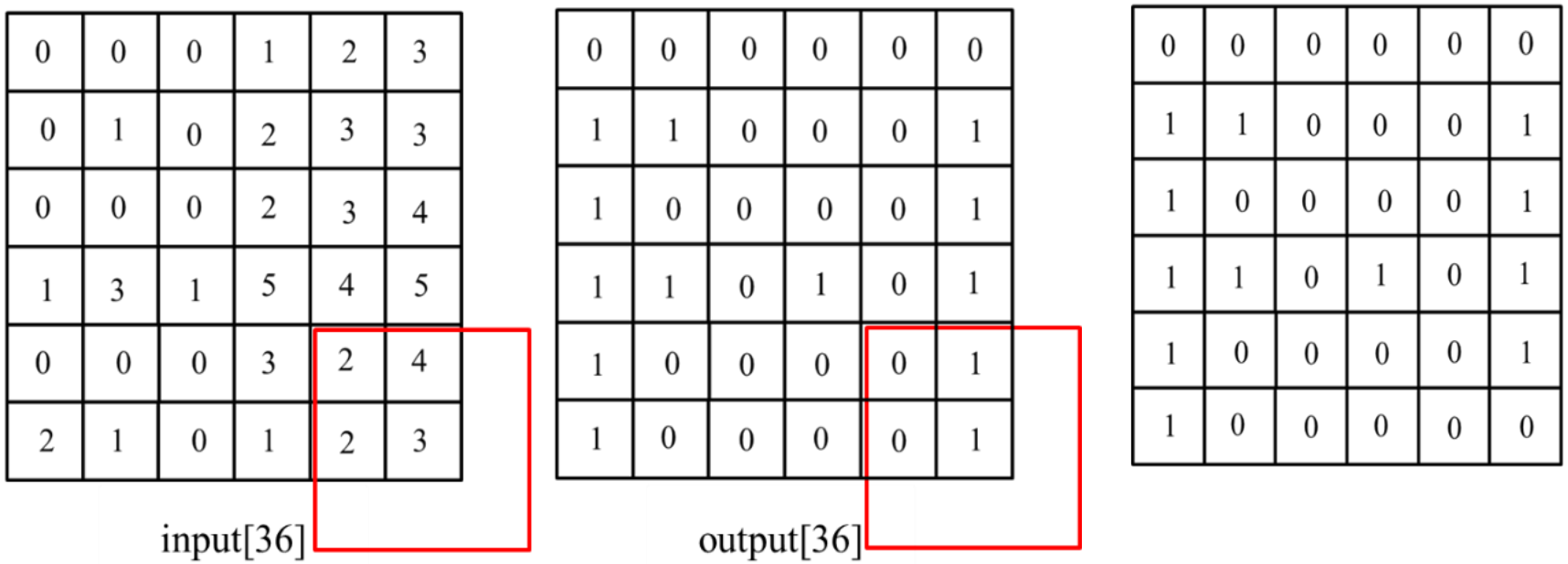
input[36]

0	0	0	0	0	0
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	0	0	0	1	1

output[36]

0	0	0	0	0	0
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	0	0	0	0	1

- 以下の図では，条件(1) が該当し，outputが更新される.



左辺の処理

- 入力画像の m 行 n 列成分(m=[2:5], n=1)に着目し,　8近傍と順に比較して, 着目しているピクセルの値がtrueの場合に, 最大になる可能性があるか調べ, outputの値を更新する.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	1	1	2	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

0	0	0	0	0	0
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	0	0	0	0	0

output[36]

- 以下の図では，条件(1) が該当し，outputが更新される.
  - 領域外の値は利用しない.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

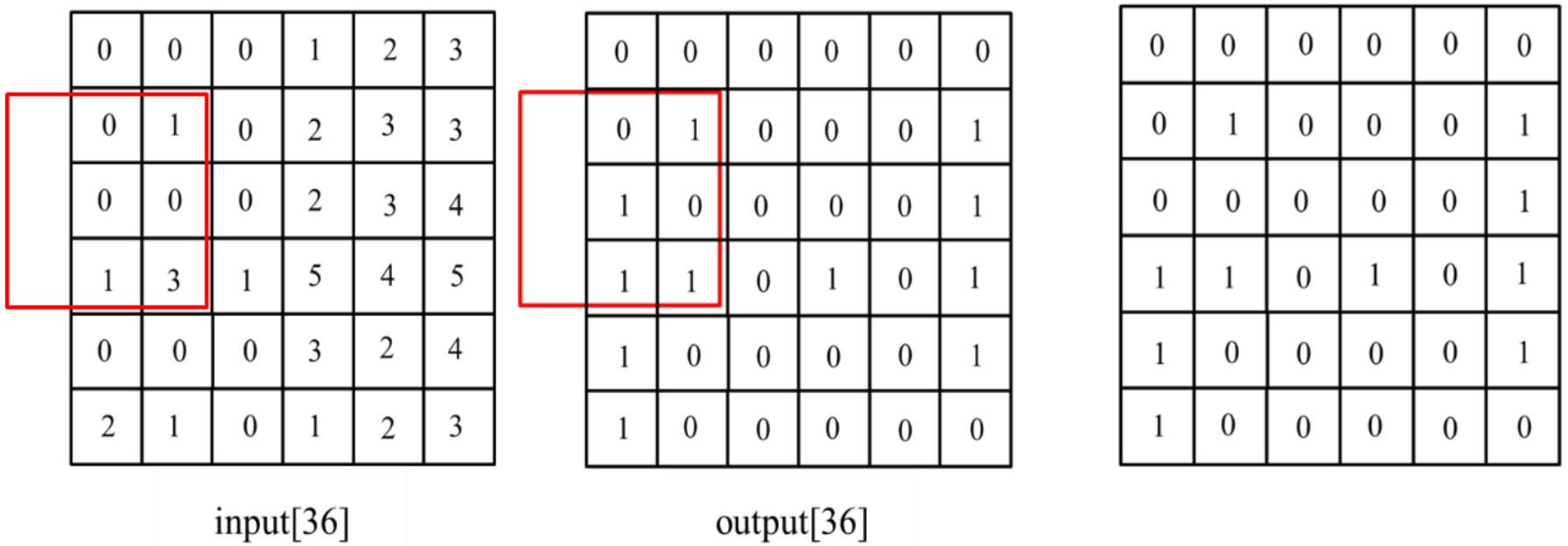
input[36]

0	0	0	0	0	0
1	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	0	0	0	0	0

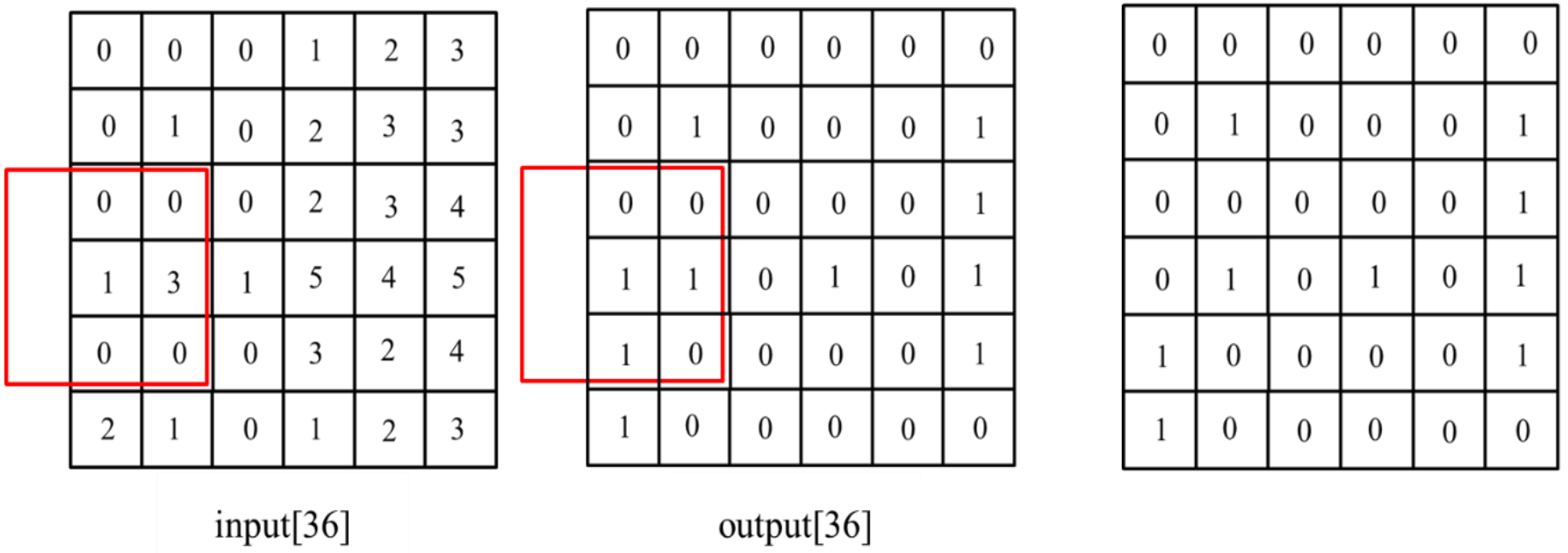
output[36]

0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	0	0	1
1	1	0	1	0	1
1	0	0	0	0	1
1	0	0	0	0	0

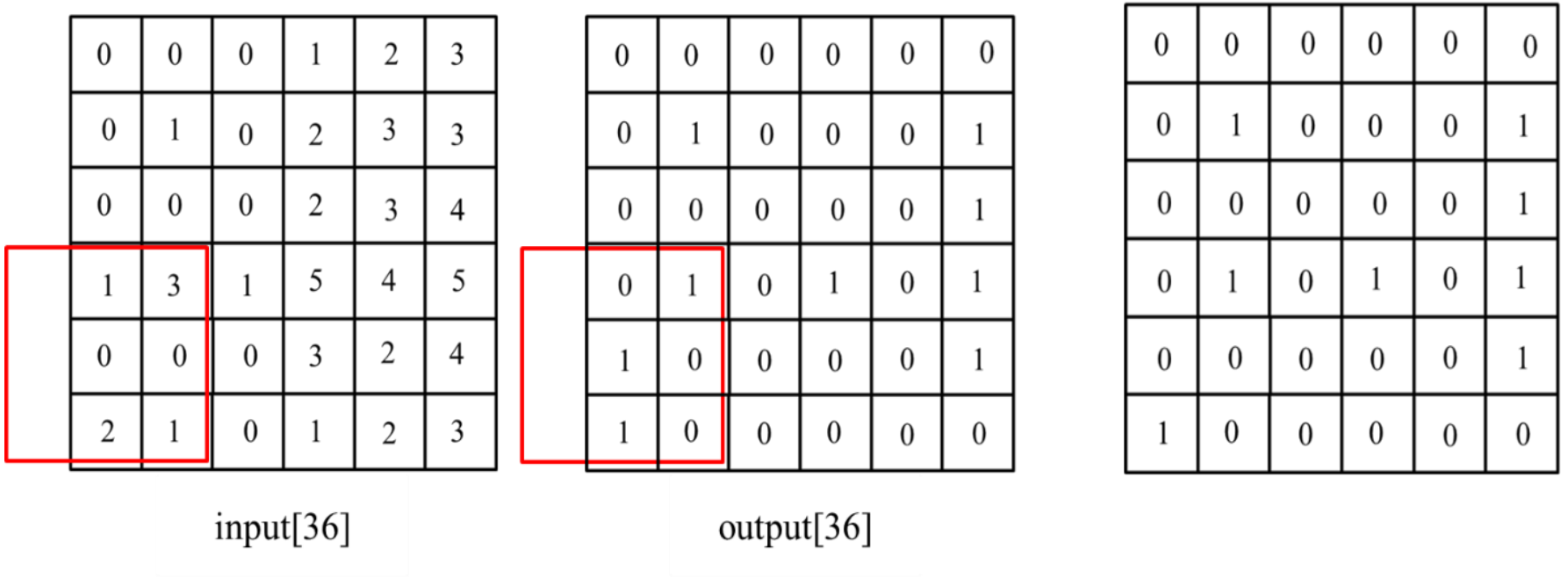
- 以下の図では，条件(1) が該当し，outputが更新される.



- 以下の図では，条件(1) が該当し，outputが更新される.



- 以下の図では，条件(1) が該当し，outputが更新される.



右辺の処理

- 入力画像の m 行 n 列成分(m=[1:6], n=6)に着目し,　8近傍と順に比較して, 着目しているピクセルの値がtrueの場合に, 最大になる可能性があるか調べ, outputの値を更新する.

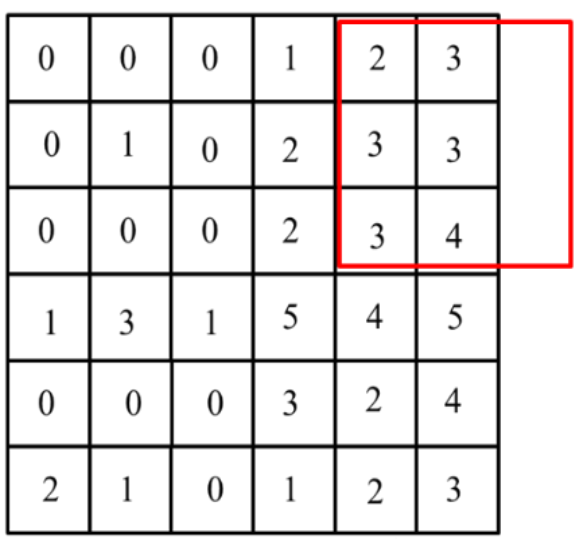
0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	1	1	2	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

0	0	0	0	0	0
0	1	0	0	0	1
0	0	0	0	0	1
0	1	0	1	0	1
0	0	0	0	0	1
1	0	0	0	0	0

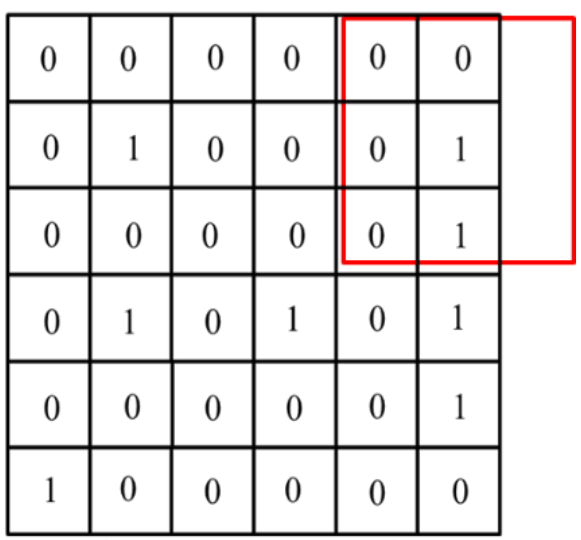
output[36]

- 以下の図では，条件(1) が該当し，outputが更新される
  - 領域外の値は利用しない.



0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

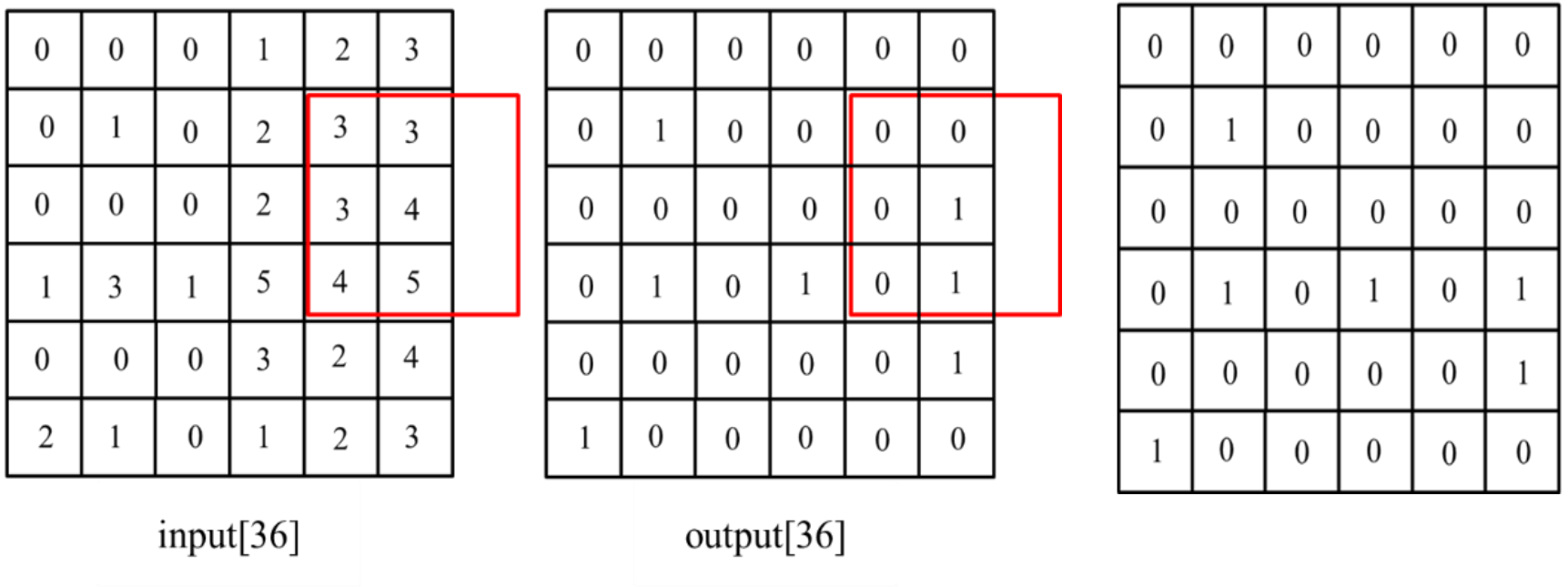


0	0	0	0	0	0
0	1	0	0	0	1
0	0	0	0	0	1
0	1	0	1	0	1
0	0	0	0	0	1
1	0	0	0	0	0

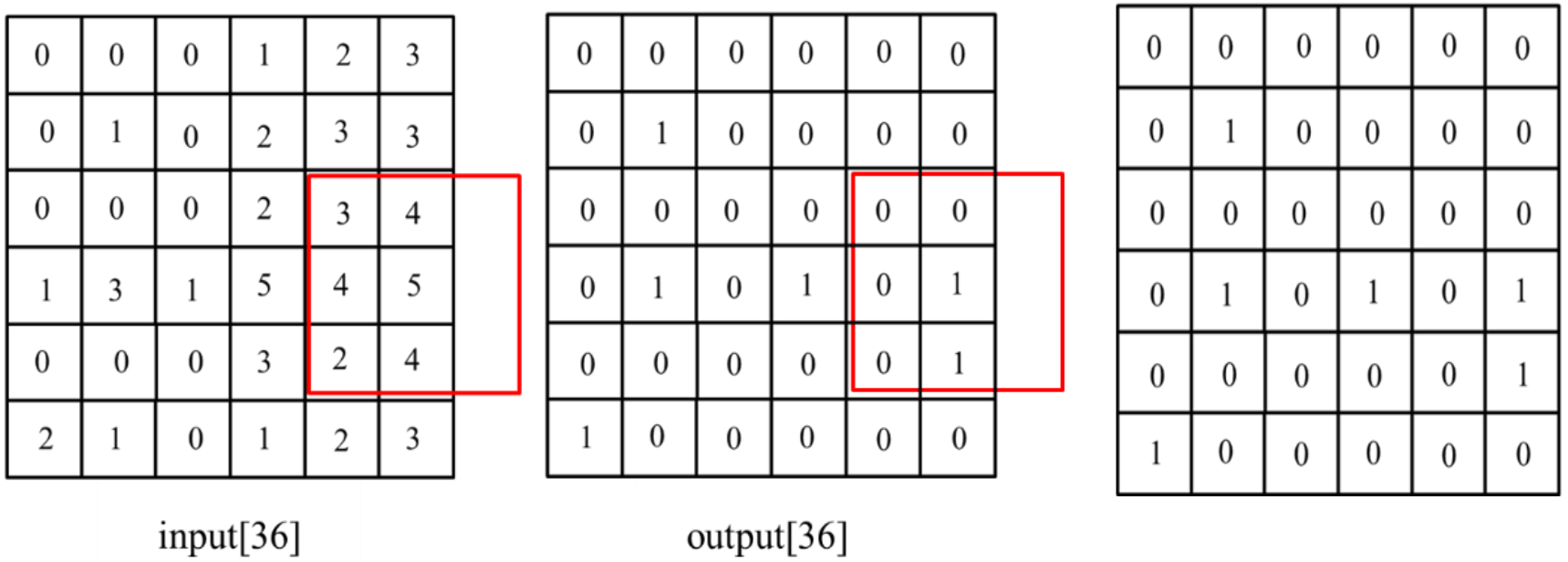
output[36]

0	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	1
0	1	0	1	0	1
0	0	0	0	0	1
1	0	0	0	0	0

- 以下の図では，条件(1) が該当し，outputが更新される



- 中央のピクセルは8近傍のいずれのピクセルよりも値が大きいため，outputの中央のピクセルは”true(=1)”のまま更新されない.（条件(1)も(2)も該当しない）



- 以下の図では，条件(1) が該当し，outputが更新される

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

0	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	1
1	0	0	0	0	0

output[36]

0	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	0
1	0	0	0	0	0

- output[36]とoutput_cp[36]を比較する.
- 比較した結果, outputとoutput_cpが異なっているので, 2回目の処理を行う.

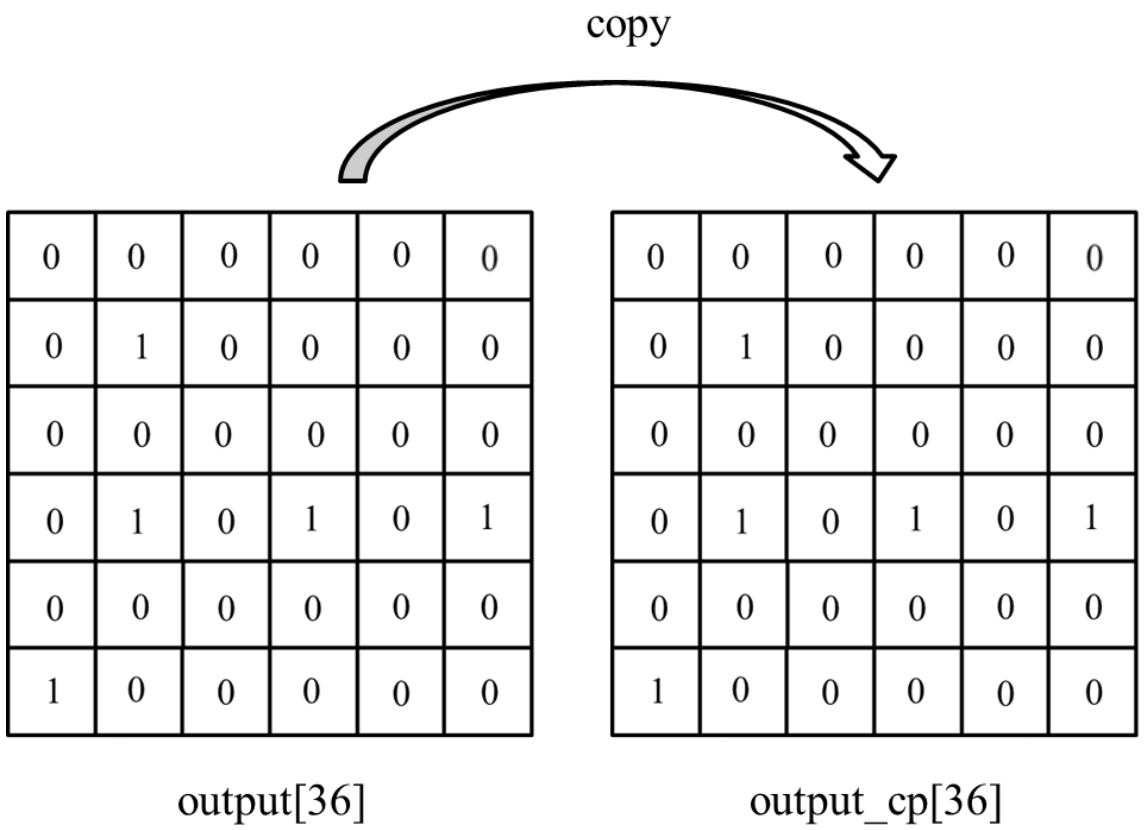
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

output_cp[36]

0	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	0
1	0	0	0	0	0

output[36]

outputをcp_outputにコピーする



0	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	0
1	0	0	0	0	0

0	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	0
1	0	0	0	0	0

- 処理を繰り返す. Output成分がfalseの場合は, 調査する必要がないのでここでは省略する.
- 中央のピクセルは8近傍のいずれのピクセルよりも値が大きいため，outputの中央のピクセルは”true(=1)”のまま更新されない.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

0	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	0
1	0	0	0	0	0

output[36]

- 中央のピクセルは8近傍のいずれのピクセルよりも値が大きいため，outputの中央のピクセルは”true(=1)”のまま更新されない.

0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

0	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	0
1	0	0	0	0	0

output[36]

- 中央のピクセルは8近傍のいずれのピクセルよりも値が大きいため，outputの中央のピクセルは”true(=1)”のまま更新されない.

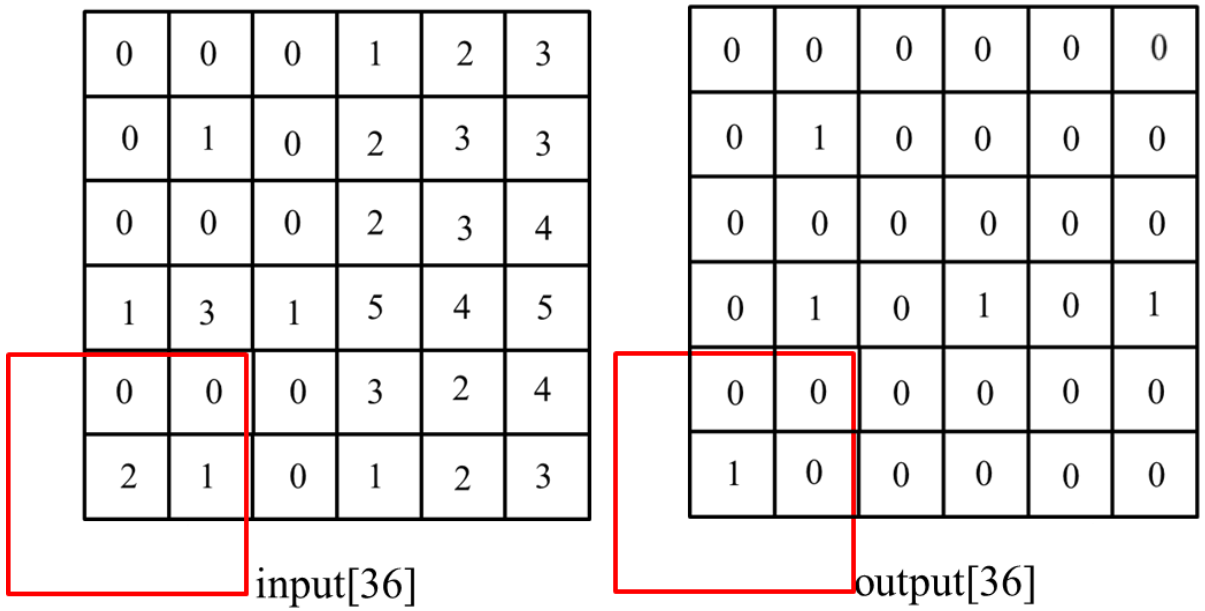
0	0	0	1	2	3
0	1	0	2	3	3
0	0	0	2	3	4
1	3	1	5	4	5
0	0	0	3	2	4
2	1	0	1	2	3

input[36]

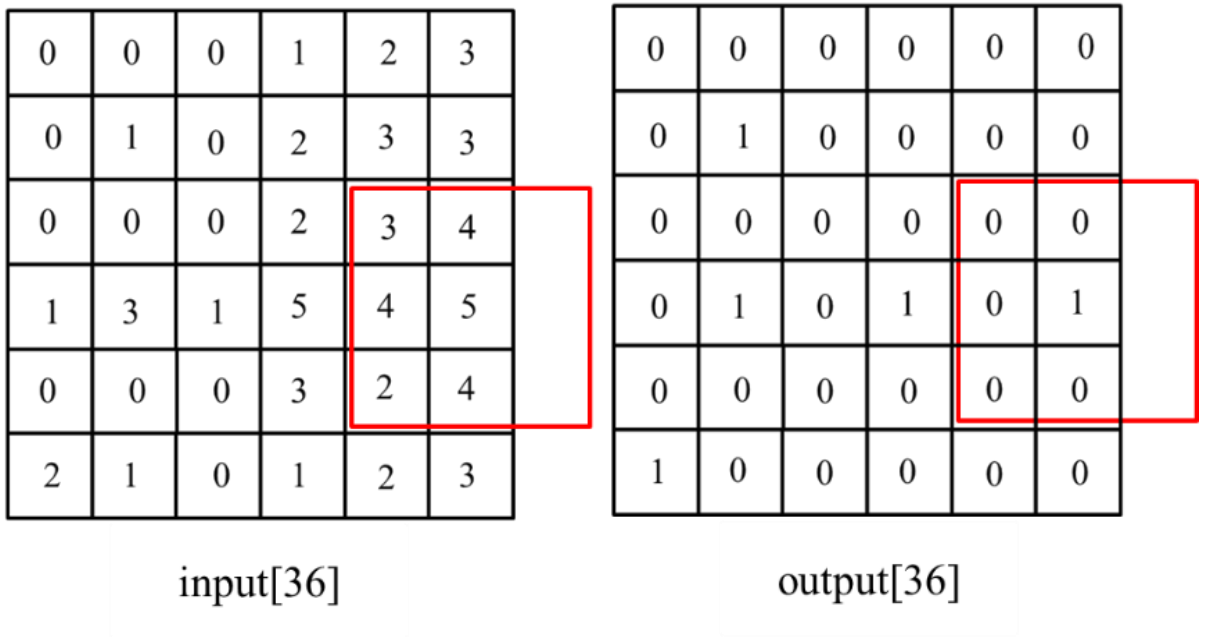
0	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	0
1	0	0	0	0	0

output[36]

- 中央のピクセルは8近傍のいずれのピクセルよりも値が大きいため，outputの中央のピクセルは”true(=1)”のまま更新されない.



- 中央のピクセルは8近傍のいずれのピクセルよりも値が大きいため，outputの中央のピクセルは”true(=1)”のまま更新されない.



- output[36]とoutput_cp[36]を比較する.
- 比較した結果, outputとoutput_cpが同じであるので終了.

0	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	0
1	0	0	0	0	0

output_cp[36]

0	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	0
1	0	0	0	0	0

output[36]

- 最終的に以下の配列outputが得られる.

0	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0
0	1	0	1	0	1
0	0	0	0	0	0
1	0	0	0	0	0

output[36]