Chapter 5

Binary CNN Characterization via Boolean Functions

5.1. Binary and universal CNN truth table

Our objective in this section is to show that every space-invariant *binary* (black-and-white) *CNN* belonging to the *uncoupled* class $\mathscr{C}(A^0, B, z)$ with a 3x3 neighborhood (r=1) which maps any static *binary* 3x3 input pattern U into a static binary 3x3 output pattern $Y(\infty)$ can be *uniquely* defined by a *Boolean function* $\mathscr{C}$ of 9 binary input variables[1]

$$\mathbf{u}_{ij} = [u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9]^T \qquad (5.1)$$

where $u_i \in \{0, 1\}$ denotes one of the 9 pixels within the sphere of influence of cell C_{ij} as shown in Fig.1(a). Note that we have opted for a "single" rather than a "double" subscript notation to avoid clutter. Note also that $\mathbf{u}_{ij}$ has a *subscript* (ij) and is set in a *bold face* type in order to distinguish it from the input u_{ij} (set in light-face type) of cell C_{ij}. Although we can code the 9 pixels u_{kl}, $kl \in \{-1, 0, 1\}$ by any combination of u_i, we have chosen the coding scheme shown in Fig.1(b) for pedagogical reasons that will be obvious later. A simple mnemonic to reconstruct this code is to remember the subscript "5" always refers to the input u_{00}, corresponding to the *center* cell C_{ij}, whereas the subscripts {1, 2, 3, 4} refer to the surround cells in the N→E→S→W clockwise compass directions, and the remaining subscripts {6, 7, 8, 9} refer to the surround cells in the NE→NW→SE→SW clockwise compass directions.

$u_{-1,-1}$	$u_{-1,0}$	$u_{-1,1}$
$u_{0,-1}$	$u_{0,0}$	$u_{0,1}$
$u_{1,-1}$	$u_{1,0}$	$u_{1,1}$

(a)

⇒

u_9	u_8	u_7
u_6	u_5	u_4
u_3	u_2	u_1

(b)

Fig.1 Every 3x3 binary pattern from (a) will henceforth be coded by the standard scheme in (b).

Now given any static binary input pattern U, the color (black or white, since the CNN is *assumed* to be binary) of *any* output pixel is determined uniquely by only a small part of U exposed to a 3x3 transparent window centered at cell C_{ij}, because the sphere of influence $S_1(i,j)$ is *assumed* to be a 3x3 neighborhood. Hence the color {0, 1} of the output pixel $y_{ij}(\infty)$ is uniquely determined by the binary value (0 or 1) of the 9 pixels u_1,

---

[1] We have chosen here {0,1} instead of {-1,1} as our binary codes in order to exploit directly the immense literature and theory on Boolean functions, which are almost always couched in terms of "zeros" and "ones".

u_2, ..., u_9 exposed by the 3x3 window. This unique answer is obtained by solving the system of MxN ODE having the *prescribed* CNN templates (A^0, B, z), and prescribed initial state **x**(0). Now even though there are *infinitely many* distinct templates (recall the coefficients of A^0, B, and z can be any *real* number, which is uncountable), there will be only a finite (albeit very large) number of distinct combinations of 3x3 "checkerboard" patterns of black and white cells, namely, $2^9 = 512$.

Figure 1c shows how a single binary input is represented.

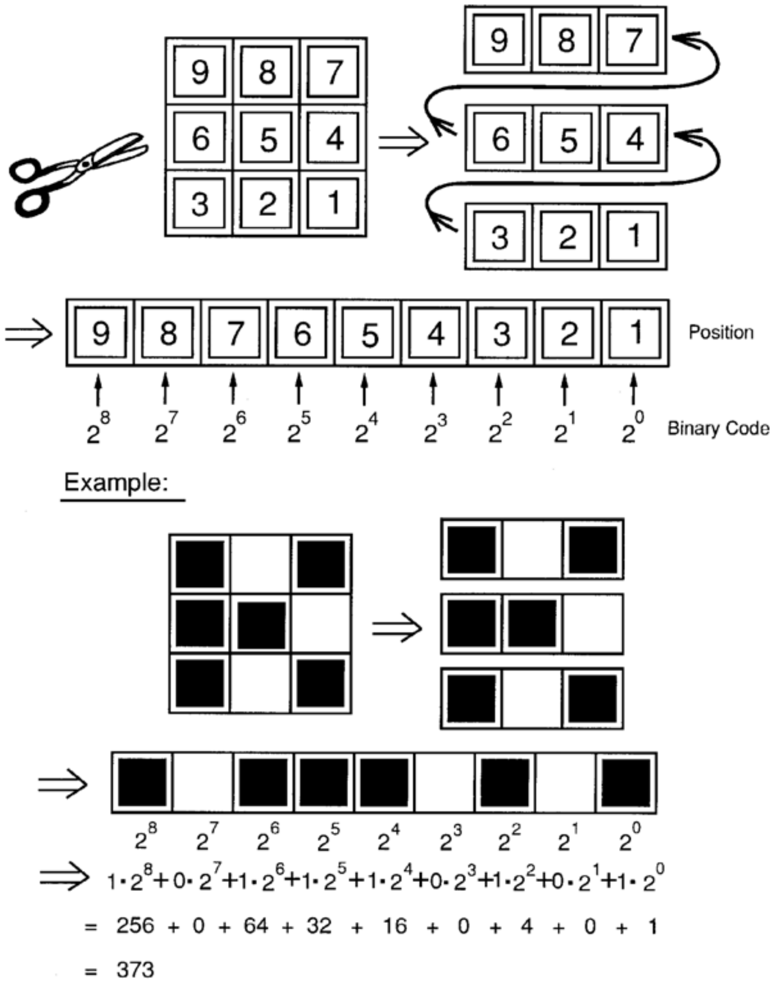


Fig. 1c Representing a single binary input

Since each such pattern can map to either a "0", or a "1", there are exactly[2]

[2] In order to appreciate how large the number Ω is, compare it to the following universal benchmarks:

- *Age of the universe* = 10^{30} picoseconds.
- Mass of the universe (calculated in units of mass of a hydrogen atom) = 10^{80}
- *Volume of the universe* (calculated as a sphere with a diameter of 10 thousand million light-years) $=10^{84}$ cm^3

$$\Omega \triangleq 2^{2^9} = 2^{512} \approx 1.3408\text{x}10^{154} > 10^{154} \tag{5.2}$$

distinct Boolean maps of 9 binary variables. This maps can be ordered in a table shown in Fig. 1d. Each row shows a different binary 9 input 1 output map.

CNN Program NAME	CNN Program Code													
	512	511	510	509	508	507	506	7	6	5	4	3	2	1
Fill White	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	0	0	0	0	0	1	1
4	0	0	0	0	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0	0	0	0	0	1	0	1
6	0	0	0	0	0	0	0	0	0	0	0	1	1	0
Game of Life	0	0	0	1	0	1	0	0	0	0	0	0	0	1
Edge Detector	1	0	1	1	1	1	1	0	0	0	0	0	0	0
$2^{512}-3$	1	1	1	1	1	1	1	1	1	1	1	1	0	1
$2^{512}-2$	1	1	1	1	1	1	1	1	1	1	1	1	1	0
Fill Black	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Total Number of Distinct CNN Programs = $2^{512} \approx 1.34078\text{X}10^{154}$

Fig. 1d CNN Program Code of 9 variable binary input

Let $\mathcal{C}_\Omega$ denote the universe of *all* such maps. Now since $\mathcal{C}_\Omega$ is the maximal set, by definition, the Boolean map generated by each member of the standard CNN universe $\mathcal{C}(A^0, B, z)$ must be a member of $\mathcal{C}_\Omega$[3]. Hence

$$\mathcal{C}(A^0, B, z) \subset \mathcal{C}_\Omega \tag{5.3}$$

We have just proved the following fundamental result:

Theorem 1 Binary CNN Truth Table

Every binary standard CNN with template (A^0, B, z) and prescribed initial state $\mathbf{x}(0)$ is a member of the universe $\mathcal{C}_\Omega$ of all Boolean functions of 9 variables and is therefore uniquely characterized by the *CNN truth table* shown in Fig.2, consisting of 512 *rows* (one for each distinct 3x3 checkerboard pattern), 9 *input columns* (one for each binary input variables u_i), and 1 *output* column whose value (0 or 1) corresponds to $y_{ij}(\infty)$.

binary pattern number	input variables									output $y_{ij}(\infty)$
	u_9	u_8	u_7	u_6	u_5	u_4	u_3	u_2	u_1	
0										
1										
. . .										
510										
511										

Fig. 2 Truth table for defining *any* Boolean functions of 9 variables.

Theorem 1 gives us *the most rigorous* method for characterizing a space-invariant binary CNN, and is therefore of fundamental importance. Since this table will in general exceed the length of a typical page, let us divide it into 16 component truth tables each one containing 32 rows. For example, the 16 component truth tables which characterize the *Edge* templates are given in Fig.3(a)-3(p). To clarify our notations, in the first component table shown in Fig.3(a), each entry for the input variables is coded by a “0” or a “1”, instead of our earlier notation of “-1” and “1”, in order for us to exploit the extensive theory and literature on Boolean functions, which are almost universally couched in “zeros” and “ones”. Observe that we have ordered the binary values in the truth table in the same order for enumerating the *binary number* 0, 1, 2, 3, ..., 511, consecutively. Since it is usually more pleasing for the eyes to decode a table of black-

[3] Note that $\mathcal{C}(A^0, B, z)$ may generate non-Boolean maps as well.

and-white cells than a table of "zeros" and "ones", we will henceforth code our CNN truth tables by black-and-white cells.

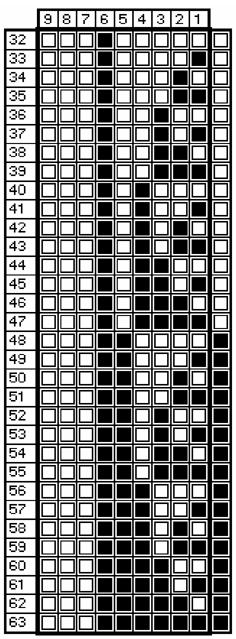
To construct the truth table for any binary CNN $\mathcal{C}(A^0, B, z)$ with prescribed initial state **x**(0), simply solve the associate system of differential equations for each input of 512 distinct binary patterns listed in Fig. 2 and fill in the corresponding calculated output, either black (1) or (0). Since the 512 binary patterns are fixed, each corresponding to a 9-bit binary expression of an integer N, N = 0,1, 2, ..., 511, it is easy to write a computer program to generate the truth table automatically, given any templates (A^0, B, z) and the prescribed initial condition **x**(0). In particular, simply assume a 3x3 CNN array (M=N=3) and find the solution of the *center cell* C_{00}.

Example 1 Edge CNN

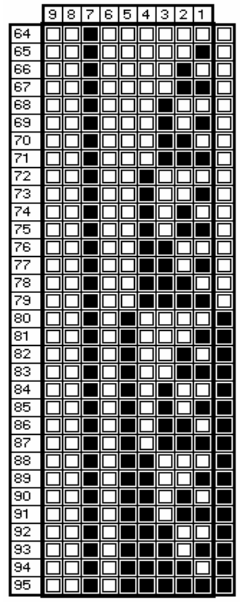
1

	10	9	8	7	6	5	4	3	2	1	
0	D	0	0	0	0	0	0	0	0	0	0
1	D	0	0	0	0	0	0	0	0	1	0
2	D	0	0	0	0	0	0	0	1	0	0
3	D	0	0	0	0	0	0	0	1	1	0
4	D	0	0	0	0	0	0	1	0	0	0
5	D	0	0	0	0	0	0	1	0	1	0
6	D	0	0	0	0	0	0	1	1	0	0
7	D	0	0	0	0	0	0	1	1	1	0
8	D	0	0	0	0	0	1	0	0	0	0
9	D	0	0	0	0	0	1	0	0	1	0
10	D	0	0	0	0	0	1	0	1	0	0
11	D	0	0	0	0	0	1	0	1	1	0
12	D	0	0	0	0	0	1	1	0	0	0
13	D	0	0	0	0	0	1	1	0	1	0
14	D	0	0	0	0	0	1	1	1	0	0
15	D	0	0	0	0	0	1	1	1	1	0
16	D	0	0	0	0	1	0	0	0	0	1
17	D	0	0	0	0	1	0	0	0	1	1
18	D	0	0	0	0	1	0	0	1	0	1
19	D	0	0	0	0	1	0	0	1	1	1
20	D	0	0	0	0	1	0	1	0	0	1
21	D	0	0	0	0	1	0	1	0	1	1
22	D	0	0	0	0	1	0	1	1	0	1
23	D	0	0	0	0	1	0	1	1	1	1
24	D	0	0	0	0	1	1	0	0	0	1
25	D	0	0	0	0	1	1	0	0	1	1
26	D	0	0	0	0	1	1	0	1	0	1
27	D	0	0	0	0	1	1	0	1	1	1
28	D	0	0	0	0	1	1	1	0	0	1
29	D	0	0	0	0	1	1	1	0	1	1
30	D	0	0	0	0	1	1	1	1	0	1
31	D	0	0	0	0	1	1	1	1	1	1

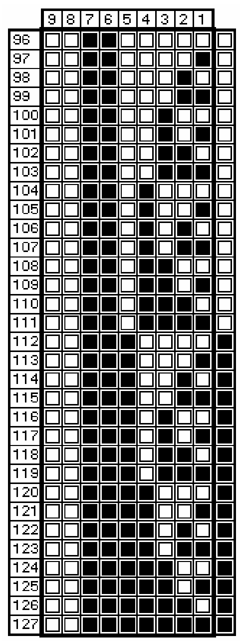
2



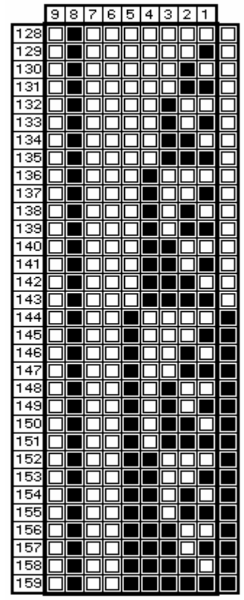
3



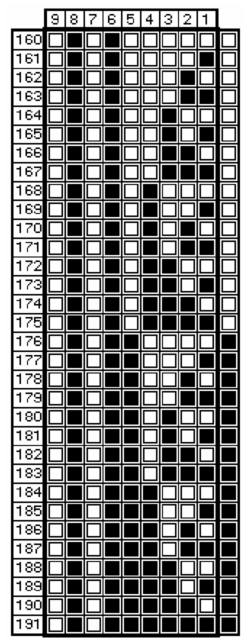
4



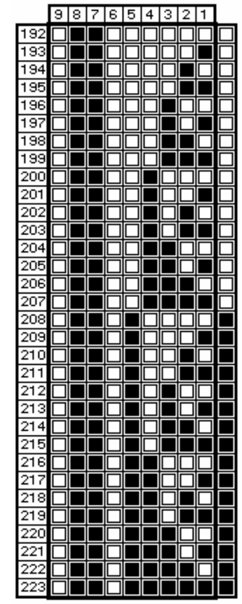
5



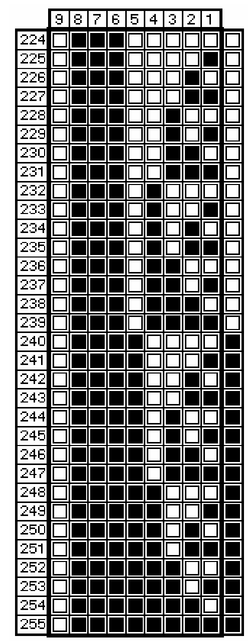
6



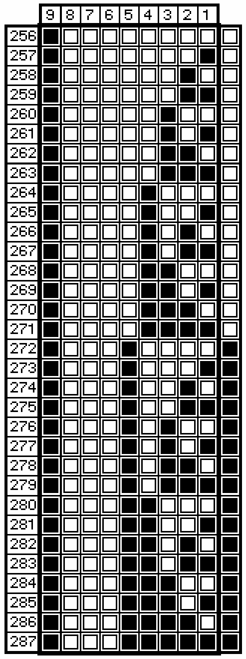
7



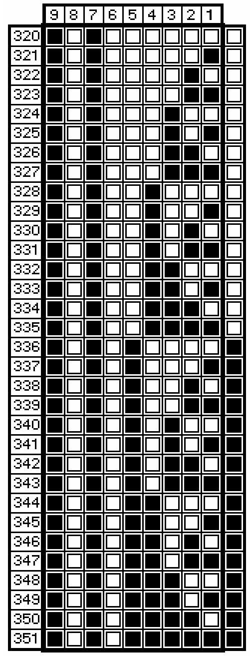
8



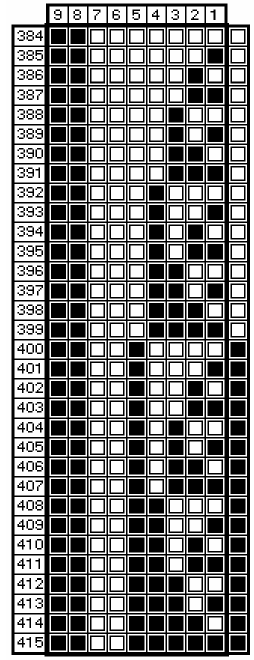
9



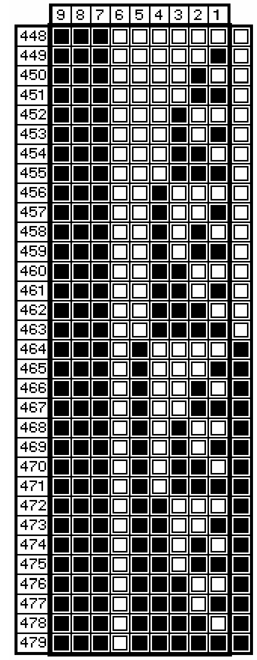
11



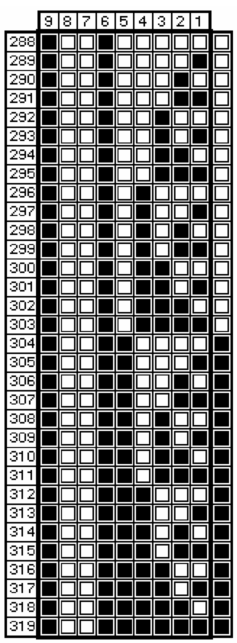
13



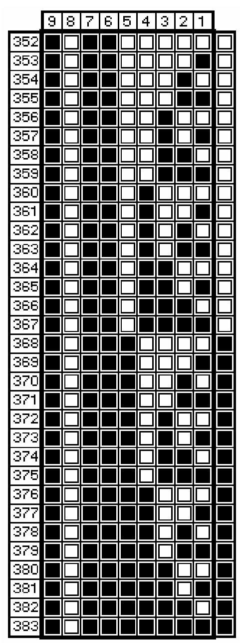
15



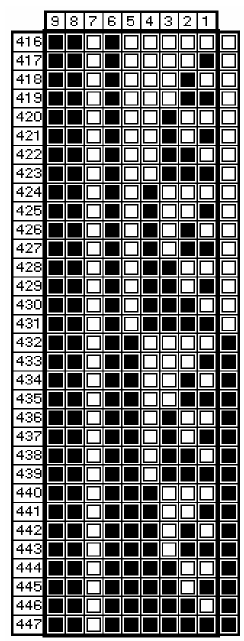
10



12



14



16

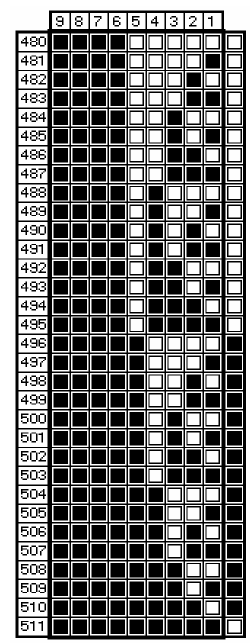


Fig 3(a)-3(p)

The truth table for the edge CNN calculated by the above procedure is shown in Fig.3, decomposed into 16 components, clearly, except for displaying a few of these truth tables for analysis and pedagogical purposes, it is impractical to list the truth table of all useful CNN's. They can, however, easily be stored on a diskette, to be retrieved only when needed. Displaying the truth table on a computer screen has the advantage of showing a continuous table when any part of the table can be scrolled into the entire view.

The alert reader will have already realized that the truth table format of Fig.1 contains a great deal of redundancy. Indeed, in each of the 16 components shown in Fig.3, the domain of the binary input variables u_1, u_2, ..., u_9, which constitutes the bulk of the space of each table, remains unchanged. Hence, we only need to record *the last column* of each of these 16 component tables. Since each column has 32 cells, we need only store 16x32=512 pixel values (0 or 1) for each binary CNN $\mathcal{C}(A^0, B, z)$ with prescribed initial conditions and will be able to reconstruct these 16 component tables. For maximum space efficiency, we can pack all 16 columns from Fig.2, each with 32 entries, into 16 rows, next to each other to form a grid containing exactly 16x32=512 cells, as shown in Fig.4. Since this table contains the same information as those of Fig.2, we have achieved an immerse amount *of data compression*. Indeed, since this table contains only 512 entries, one for each input pattern, it is a *minimal representation*. We will henceforth refer to Fig. 4 *as minimal CNN truth table*.

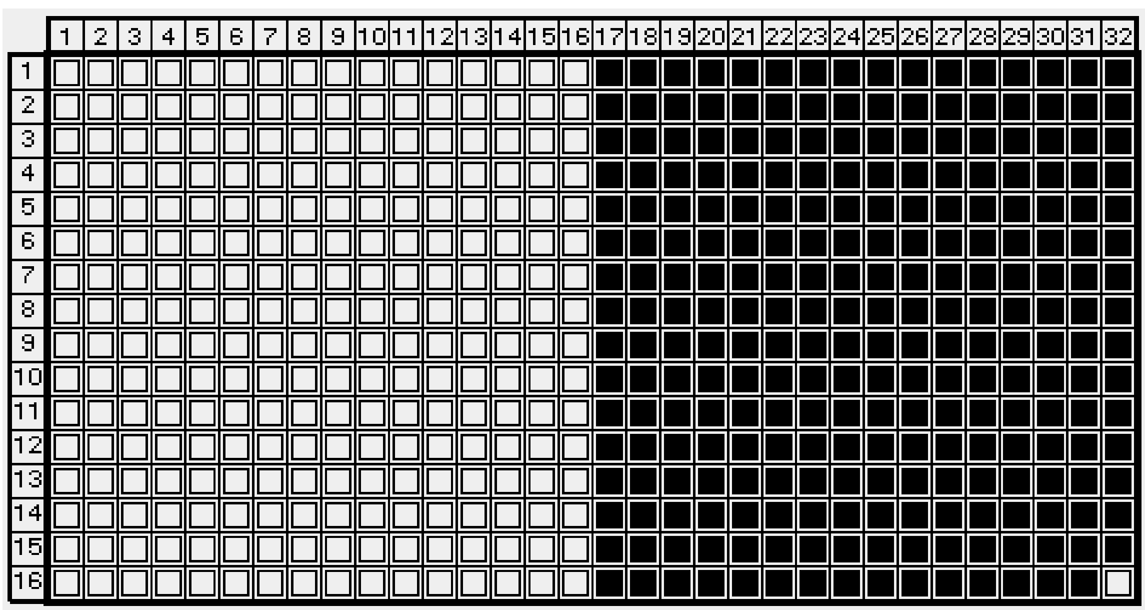


Fig. 4 Minimal CNN truth table

Corollary to Theorem 1

Every space-invariant binary CNN with a 3x3 neighborhood and specified by templates (A^0, B, z) and a prescribed initial state **X**(0) is associated with a unique minimal CNN truth table.

Remarks

1. The *uniqueness* assertion in the above corollary is with respect to a given template (A^0, B, z) *and* initial state **X**(0). It is not unique with respect to a given "global task" since a given task in general can be implemented by many distinct CNN templates (infinitely many indeed).

2. The above corollary only asserts that for every CNN template (A, B, z) and initial state, there corresponds, a minimal truth table, or equivalently, a Boolean function of 9 variables. However, the converse is not true, i.e., given a Boolean function $\mathscr{B} \in \mathscr{C}_\Omega$, or its associated minimal truth table, there may *not* exist a CNN template and an initial state **X**(0) which yields this truth table. However, we will prove later that every member of $\mathscr{C}_\Omega$, i.e., every Boolean function of 9 variables, can be realized by a *CNN universal machine* to be studied in depth later. We will prove later that there are more than 10^{154} such Boolean functions of 9 variables that can be programmed by a single CNN universal machine. This immensely large number is greater than the volume of the universe (10^{84} cm^3, calculated as a sphere with a diameter of 10 thousand million light years) !

5.2. Boolean and Compressed Local Rules

Every CNN with 3x3 neighborhood or its generalization, the *CNN universal machine*, to be presented later which maps a static *binary input* image into a static *binary output* image has a unique CNN truth table representation consisting of 512 rows, each one mapping a Boolean expression involving 9 Boolean variables, into a "0" or a "1" digit:

$$(d_1, d_2 \ldots, d_9) \rightarrow \{0,1\} \tag{5.4}$$

where $d_i \in \{0,1\}$. We can now define rigorously our earlier heuristic notation of a local rule:

Definition 1*: Complete Set of CNN Boolean Local Rules*

Each row of the CNN truth table is called a Boolean *CNN local rule*. Every CNN with 3x3 neighbors is rigorously defined by a complete set of 512 Boolean Local Rules.

Definition 2: *Compressed Boolean Local Rules*

Any other rule which can be used to derive one or more Boolean local rules is called a *compressed local rule*, or simple local rule if the usage is clear.

The motivation for devising compressed local rules is simply to reduce the large number (512) of Boolean local rules to a smaller number. They are usually derived by heuristic methods and may not be adequate in view of the following reasons:

(a) While some local rule may correctly reproduce a large subset of the Boolean local rules, it may contradict some others. In this case, we say it is an *inconsistent* local rule. If the inconsistency occurs only for a few rare input patterns, it may still be useful for pedagogical purpose especially if the local rule Compression Ratio $\gamma_{LR} \triangleq$ number of correctly reproduced Boolean local rules divided by 512.

is sufficiently large, $\gamma_{LR} \leq 1$. In this case the errors may be acceptable for pedagogical reasons especially if this local rule makes it possible to visualize or identify the main features of the input image that are to be extracted, modified, or transformed.

(b) The set of compressed local rules are *incomplete* in the sense that some Boolean local rules can *not* be deduced from them.

Definition 3: *Complete Set of Compressed Local Rules*.

A set of compressed local rules is said to be *complete* if and only if no member of this set is inconsistent and if all 512 Boolean local rules can be deduced from this set.

Definition 4: *Minimal Set of Compressed Local Rules*.

A complete set of compressed local rules is said to be *minimal* if no member of this set can be eliminated and still achieve completeness.

Computer-Aided Method for Proving Local Rules

Given a CNN template (A^0, B, z) and initial state $\mathbf{X}(0)$, there is presently no systematic algorithm to derive a *complete* set

$$\mathcal{L}_{LR} = \{ \mathcal{L}_1, \mathcal{L}_2, ..., \mathcal{L}_p \} \tag{5.5}$$

of local rules which are sufficient to map any binary input patterns into the prescribed output patterns obtained by solving the associated system of ODE's. In most cases, only a subset $\mathcal{L}^-_{LR} \subset \mathcal{L}_{LR}$ may be found. On rare occasions, a superset $\mathcal{L}^+_{LR} \supset \mathcal{L}_{LR}$ may be found. On few occasions, some local rules may be *redundant* in the sense that for *some* input patterns, they predict the same output. It is also quite possible that two or more local rules may contradict each other's prediction and hence are said to be *inconsistent*. Finally, given a *complete* set of local rules, does there exist a proper subset which is also complete? If so, is it possible to find a complete set of local rules which are minimal in the sense that no other complete set exists which contains a fewer number of elementary local rules? We will now show that all of these questions, except the last one, can be easily resolved with the help of the CNN truth table, or equivalently, its associated minimal truth table. We will give a constructive solution to each question (except the last one) raised above in the form of an algorithm.

Algorithm 1: Checking whether a local rule candidate $\mathcal{L}_i$ is consistent

1. Use the prescribed template (A^0, B, z) and initial state **X**(0) to derive the associated CNN truth table $\mathcal{T}$.
2. Apply the local rule $\mathcal{L}_1$ to *each* of the 512 input patterns. In general, $\mathcal{L}_1$ may *not* be *applicable* (NA) for some patterns (due to inadequate or overly simplistic assumptions). In this case, the output cell will be denoted by NA, or simply coded in gray scale. For those input patterns where $\mathcal{L}_1$ is applicable, there are 3 possibilities for the output cell: (i) output is black (coded by Boolean number 1) and agrees with the corresponding output in the truth table. In this case, the output will be printed "black". (ii) output is white (coded by Boolean number 0) and agrees with the corresponding output in the truth table. In this case, the output cell will be printed "white". (iii) the output is black (resp. white) but the corresponding cell in the truth table is white (resp., black). In this case, the output cell will be denoted by a cross ⊠, thereby indicating $\mathcal{L}_1$ is inconsistent and is not a valid local rule.
1. The local rule $\mathcal{L}_1$ is proved to be valid if and only if it is *not* inconsistent.

Algorithm 2: Checking whether a set $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k\}$ *is complete.*

1. Derive the CNN truth table $\mathcal{T}$, as in Algorithm 1.
2. Apply *Algorithm 1* to each $\mathcal{L}_1 \in \mathcal{L}$. If any $\mathcal{L}_i$ is inconsistent, stop. Otherwise, go to 3.
3. If *each* output cell is predicted to be either black or white by at least one local rule $\mathcal{L}_i \in \mathcal{L}$, then $\mathcal{L}$ is complete. In this case, we have a rigorous proof of the validity *and* completeness of the set of local rules.

Algorithm 3: Given a complete set $\mathcal{L}_{LR}$ *of local rules, find a smallest proper subset which is also complete.*

1. Delete $\mathcal{L}_1$ from $\mathcal{L}_{LR}$ and apply Algorithm 2 to the remaining set. If it is complete, delete the first 2 elements $\mathcal{L}_1$ and $\mathcal{L}_2$ from $\mathcal{L}_{LR}$ and repeat Algorithm 2. Continue the same "pruning" procedure until the remaining set is no longer complete. In this case, the immediately preceding remaining set of local rules constitutes the *smallest* complete set with respect to the order where the elements of $\mathcal{L}_{LR}$ are deleted.
2. Repeat step 1 to all permutations of the ordering of the members of $\mathcal{L}_{LR}$.
3. Any complete set resulting from steps 1 and 2 having the smallest number of elements is a *minimal complete* set, relative to $\mathcal{L}_{LR}$.

Remarks:

1: The above choice of minimal complete set may not be unique. Since there may exist several complete sets all containing the same smallest number of elements.

2: The “minimality” derived from Algorithm 3 may not be global in the sense that there may exist an entirely different set $\mathcal{L}_{LR}$ of complete rules in which Algorithm 3 would yield a minimal complete set having fewer elements than that determined from $\mathcal{L}_{LR}$. The difficulty in deriving a global minimal complete set is that there is no obvious algorithm to guarantee all distinct sets of complete local rules have been exhausted. A further difficulty lies in the criterion to be used for certifying which local rule is qualified as *elementary*. For otherwise, one could combine several local rules into a single but more complex local rule. Hence it is necessary to define “elementary” in the sense that no decomposition into two or more simpler local rules is possible. The algorithms are contained in the TEMPO program (Appendix C)

5.2. Optimizing the Truth Table

Recall that once a CNN template is specified a unique truth table can be easily generated by a simple computer program, say by solving a system of 9 ODE’s a total of 512 times, one for each distinct Boolean pattern of nine input variables, or by some explicit formula that applies only to some specific subclass of CNN’s, e.g., the *uncoupled* class. One can examine each of the 512 3x3 binary input patterns and determine whether the output (black or white) of this CNN is “correct” from the user’s perspective. The next tables (Table 1-11) show Minimal Truth Table, the Truth Tables and the Window Truth Tables of the CORNER template. However, for example, among the 32 input patterns shown in Table 5 (corresponding to the Boolean local rules no. 96-127) and the 32 input patterns shown in Table 6 (corresponding to the Boolean local rules no. 160-191) for the CORNER CNN, we found the “black” output of this CNN for input patterns no. 114, 116, 176, 177, 178, 180, and 184 to be “incorrect” in the sense that the center black pixel in each of these 7 input patterns do *not look* like “corners”, from the perspective of the human visual system. Similarly ,we also disagree with this CNN’s classification (white; i.e., *not* corner) of input patterns no.115 and 121 because the black center pixel in this two patterns really look like a “corner”. Hence, we would consider these 9 classifications made by the CORNER CNN to be “incorrect”. It is important that this does *not* mean the CORNER truth table is incorrect, as *every truth table* is an exact and hence correct representation of the CNN having the prescribed template. Indeed, from the perspective of a *robot*, or some creatures having a different visual system, the above classifications may be completely acceptable.

From the human perspective, however, it would be desirable to reclassify the above 9 Boolean local rules to obtain an optimized CNN truth table[4]. Once this is done, our next task is to design a CNN template (which may not exist) having this optimized

[4] This reclassification task is a *subjective* exercise since not everyone may agree on whether a particular pixel in fuzzy cases is a corner, or not a corner.

truth table. If no such template exists, we will show later that a CNN universal machine can always be used to realize this optimized truth table, or any other truth table.

As an example, all misclassified input patterns by the CORNER CNN are designated in the minimal truth table shown in Table 12 by a *light-gray* pixel, if this pixel should be reclassified as white, and by a *dark-gray* pixel, if this pixel should be reclassified as black. The resulting *optimized* CORNER CNN is shown in Table 13. We leave this as a challenge to the reader invent a CNN template having this optimized CORNER truth table. Tables 14 and 15 show the binary and decimal code for the CORNER and optimized CORNER templates, respectively.

Minimal Truth Table of CORNER Template

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	□
2	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	□	■	■	■	□	■	□	□	□
3	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	□	■	■	■	□	■	□	□	□
4	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	□	■	□	□	□	■	□	□	□	□	□	□	□
5	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	□	■	■	■	□	■	□	□	□
6	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	□	■	□	□	□	■	□	□	□	□	□	□	□
7	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	□	■	□	□	□	■	□	□	□	□	□	□	□
8	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	■	■	■	■	□	■	■	■	□	■	□	□	□
10	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	□	■	□	□	□	■	□	□	□	□	□	□	□
11	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	□	■	□	□	□	■	□	□	□	□	□	□	□
12	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
13	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	■	■	□	■	□	□	□	■	□	□	□	□	□	□	□
14	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
15	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
16	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

Table 1

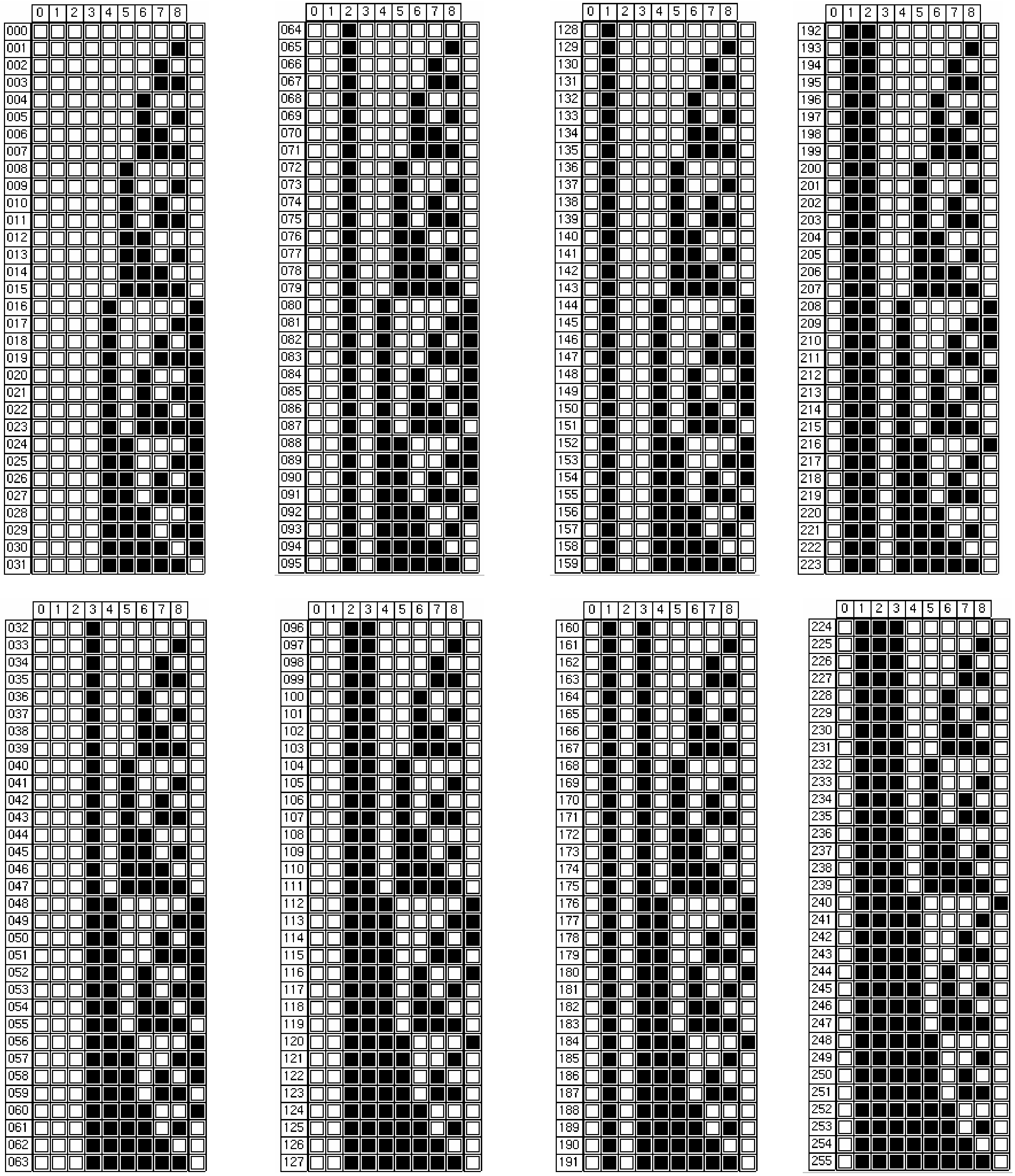


Table 2

	0	1	2	3	4	5	6	7	8	
256	■	□	□	□	□	□	□	□	□	□
257	■	□	□	□	□	□	□	□	■	□
258	■	□	□	□	□	□	□	■	□	□
259	■	□	□	□	□	□	□	■	■	□
260	■	□	□	□	□	□	■	□	□	□
261	■	□	□	□	□	□	■	□	■	□
262	■	□	□	□	□	□	■	■	□	□
263	■	□	□	□	□	□	■	■	■	□
264	■	□	□	□	□	■	□	□	□	□
265	■	□	□	□	□	■	□	□	■	□
266	■	□	□	□	□	■	□	■	□	□
267	■	□	□	□	□	■	□	■	■	□
268	■	□	□	□	□	■	■	□	□	□
269	■	□	□	□	□	■	■	□	■	□
270	■	□	□	□	□	■	■	■	□	□
271	■	□	□	□	□	■	■	■	■	□
272	■	□	□	□	■	□	□	□	□	■
273	■	□	□	□	■	□	□	□	■	■
274	■	□	□	□	■	□	□	■	□	■
275	■	□	□	□	■	□	□	■	■	■
276	■	□	□	□	■	□	■	□	□	■
277	■	□	□	□	■	□	■	□	■	■
278	■	□	□	□	■	□	■	■	□	■
279	■	□	□	□	■	□	■	■	■	□
280	■	□	□	□	■	■	□	□	□	■
281	■	□	□	□	■	■	□	□	■	■
282	■	□	□	□	■	■	□	■	□	■
283	■	□	□	□	■	■	□	■	■	□
284	■	□	□	□	■	■	■	□	□	■
285	■	□	□	□	■	■	■	□	■	□
286	■	□	□	□	■	■	■	■	□	□
287	■	□	□	□	■	■	■	■	■	□
288	■	□	□	■	□	□	□	□	□	□
289	■	□	□	■	□	□	□	□	■	□
290	■	□	□	■	□	□	□	■	□	□
291	■	□	□	■	□	□	□	■	■	□
292	■	□	□	■	□	□	■	□	□	□
293	■	□	□	■	□	□	■	□	■	□
294	■	□	□	■	□	□	■	■	□	□
295	■	□	□	■	□	□	■	■	■	□
296	■	□	□	■	□	■	□	□	□	□
297	■	□	□	■	□	■	□	□	■	□
298	■	□	□	■	□	■	□	■	□	□
299	■	□	□	■	□	■	□	■	■	□
300	■	□	□	■	□	■	■	□	□	□
301	■	□	□	■	□	■	■	□	■	□
302	■	□	□	■	□	■	■	■	□	□
303	■	□	□	■	□	■	■	■	■	□
304	■	□	□	■	■	□	□	□	□	■
305	■	□	□	■	■	□	□	□	■	■
306	■	□	□	■	■	□	□	■	□	■
307	■	□	□	■	■	□	□	■	■	□
308	■	□	□	■	■	□	■	□	□	■
309	■	□	□	■	■	□	■	□	■	□
310	■	□	□	■	■	□	■	■	□	□
311	■	□	□	■	■	□	■	■	■	□
312	■	□	□	■	■	■	□	□	□	■
313	■	□	□	■	■	■	□	□	■	□
314	■	□	□	■	■	■	□	■	□	□
315	■	□	□	■	■	■	□	■	■	□
316	■	□	□	■	■	■	■	□	□	□
317	■	□	□	■	■	■	■	□	■	□
318	■	□	□	■	■	■	■	■	□	□
319	■	□	□	■	■	■	■	■	■	□
320	■	□	■	□	□	□	□	□	□	□
321	■	□	■	□	□	□	□	□	■	□
322	■	□	■	□	□	□	□	■	□	□
323	■	□	■	□	□	□	□	■	■	□
324	■	□	■	□	□	□	■	□	□	□
325	■	□	■	□	□	□	■	□	■	□
326	■	□	■	□	□	□	■	■	□	□
327	■	□	■	□	□	□	■	■	■	□
328	■	□	■	□	□	■	□	□	□	□
329	■	□	■	□	□	■	□	□	■	□
330	■	□	■	□	□	■	□	■	□	□
331	■	□	■	□	□	■	□	■	■	□
332	■	□	■	□	□	■	■	□	□	□
333	■	□	■	□	□	■	■	□	■	□
334	■	□	■	□	□	■	■	■	□	□
335	■	□	■	□	□	■	■	■	■	□
336	■	□	■	□	■	□	□	□	□	■
337	■	□	■	□	■	□	□	□	■	■
338	■	□	■	□	■	□	□	■	□	■
339	■	□	■	□	■	□	□	■	■	□
340	■	□	■	□	■	□	■	□	□	■
341	■	□	■	□	■	□	■	□	■	□
342	■	□	■	□	■	□	■	■	□	□
343	■	□	■	□	■	□	■	■	■	□
344	■	□	■	□	■	■	□	□	□	■
345	■	□	■	□	■	■	□	□	■	□
346	■	□	■	□	■	■	□	■	□	□
347	■	□	■	□	■	■	□	■	■	□
348	■	□	■	□	■	■	■	□	□	□
349	■	□	■	□	■	■	■	□	■	□
350	■	□	■	□	■	■	■	■	□	□
351	■	□	■	□	■	■	■	■	■	□
352	■	□	■	■	□	□	□	□	□	□
353	■	□	■	■	□	□	□	□	■	□
354	■	□	■	■	□	□	□	■	□	□
355	■	□	■	■	□	□	□	■	■	□
356	■	□	■	■	□	□	■	□	□	□
357	■	□	■	■	□	□	■	□	■	□
358	■	□	■	■	□	□	■	■	□	□
359	■	□	■	■	□	□	■	■	■	□
360	■	□	■	■	□	■	□	□	□	□
361	■	□	■	■	□	■	□	□	■	□
362	■	□	■	■	□	■	□	■	□	□
363	■	□	■	■	□	■	□	■	■	□
364	■	□	■	■	□	■	■	□	□	□
365	■	□	■	■	□	■	■	□	■	□
366	■	□	■	■	□	■	■	■	□	□
367	■	□	■	■	□	■	■	■	■	□
368	■	□	■	■	■	□	□	□	□	■
369	■	□	■	■	■	□	□	□	■	□
370	■	□	■	■	■	□	□	■	□	□
371	■	□	■	■	■	□	□	■	■	□
372	■	□	■	■	■	□	■	□	□	□
373	■	□	■	■	■	□	■	□	■	□
374	■	□	■	■	■	□	■	■	□	□
375	■	□	■	■	■	□	■	■	■	□
376	■	□	■	■	■	■	□	□	□	□
377	■	□	■	■	■	■	□	□	■	□
378	■	□	■	■	■	■	□	■	□	□
379	■	□	■	■	■	■	□	■	■	□
380	■	□	■	■	■	■	■	□	□	□
381	■	□	■	■	■	■	■	□	■	□
382	■	□	■	■	■	■	■	■	□	□
383	■	□	■	■	■	■	■	■	■	□
384	■	■	□	□	□	□	□	□	□	□
385	■	■	□	□	□	□	□	□	■	□
386	■	■	□	□	□	□	□	■	□	□
387	■	■	□	□	□	□	□	■	■	□
388	■	■	□	□	□	□	■	□	□	□
389	■	■	□	□	□	□	■	□	■	□
390	■	■	□	□	□	□	■	■	□	□
391	■	■	□	□	□	□	■	■	■	□
392	■	■	□	□	□	■	□	□	□	□
393	■	■	□	□	□	■	□	□	■	□
394	■	■	□	□	□	■	□	■	□	□
395	■	■	□	□	□	■	□	■	■	□
396	■	■	□	□	□	■	■	□	□	□
397	■	■	□	□	□	■	■	□	■	□
398	■	■	□	□	□	■	■	■	□	□
399	■	■	□	□	□	■	■	■	■	□
400	■	■	□	□	■	□	□	□	□	■
401	■	■	□	□	■	□	□	□	■	■
402	■	■	□	□	■	□	□	■	□	■
403	■	■	□	□	■	□	□	■	■	□
404	■	■	□	□	■	□	■	□	□	■
405	■	■	□	□	■	□	■	□	■	□
406	■	■	□	□	■	□	■	■	□	□
407	■	■	□	□	■	□	■	■	■	□
408	■	■	□	□	■	■	□	□	□	■
409	■	■	□	□	■	■	□	□	■	□
410	■	■	□	□	■	■	□	■	□	□
411	■	■	□	□	■	■	□	■	■	□
412	■	■	□	□	■	■	■	□	□	□
413	■	■	□	□	■	■	■	□	■	□
414	■	■	□	□	■	■	■	■	□	□
415	■	■	□	□	■	■	■	■	■	□
416	■	■	□	■	□	□	□	□	□	□
417	■	■	□	■	□	□	□	□	■	□
418	■	■	□	■	□	□	□	■	□	□
419	■	■	□	■	□	□	□	■	■	□
420	■	■	□	■	□	□	■	□	□	□
421	■	■	□	■	□	□	■	□	■	□
422	■	■	□	■	□	□	■	■	□	□
423	■	■	□	■	□	□	■	■	■	□
424	■	■	□	■	□	■	□	□	□	□
425	■	■	□	■	□	■	□	□	■	□
426	■	■	□	■	□	■	□	■	□	□
427	■	■	□	■	□	■	□	■	■	□
428	■	■	□	■	□	■	■	□	□	□
429	■	■	□	■	□	■	■	□	■	□
430	■	■	□	■	□	■	■	■	□	□
431	■	■	□	■	□	■	■	■	■	□
432	■	■	□	■	■	□	□	□	□	■
433	■	■	□	■	■	□	□	□	■	□
434	■	■	□	■	■	□	□	■	□	□
435	■	■	□	■	■	□	□	■	■	□
436	■	■	□	■	■	□	■	□	□	□
437	■	■	□	■	■	□	■	□	■	□
438	■	■	□	■	■	□	■	■	□	□
439	■	■	□	■	■	□	■	■	■	□
440	■	■	□	■	■	■	□	□	□	□
441	■	■	□	■	■	■	□	□	■	□
442	■	■	□	■	■	■	□	■	□	□
443	■	■	□	■	■	■	□	■	■	□
444	■	■	□	■	■	■	■	□	□	□
445	■	■	□	■	■	■	■	□	■	□
446	■	■	□	■	■	■	■	■	□	□
447	■	■	□	■	■	■	■	■	■	□
448	■	■	■	□	□	□	□	□	□	□
449	■	■	■	□	□	□	□	□	■	□
450	■	■	■	□	□	□	□	■	□	□
451	■	■	■	□	□	□	□	■	■	□
452	■	■	■	□	□	□	■	□	□	□
453	■	■	■	□	□	□	■	□	■	□
454	■	■	■	□	□	□	■	■	□	□
455	■	■	■	□	□	□	■	■	■	□
456	■	■	■	□	□	■	□	□	□	□
457	■	■	■	□	□	■	□	□	■	□
458	■	■	■	□	□	■	□	■	□	□
459	■	■	■	□	□	■	□	■	■	□
460	■	■	■	□	□	■	■	□	□	□
461	■	■	■	□	□	■	■	□	■	□
462	■	■	■	□	□	■	■	■	□	□
463	■	■	■	□	□	■	■	■	■	□
464	■	■	■	□	■	□	□	□	□	■
465	■	■	■	□	■	□	□	□	■	□
466	■	■	■	□	■	□	□	■	□	□
467	■	■	■	□	■	□	□	■	■	□
468	■	■	■	□	■	□	■	□	□	□
469	■	■	■	□	■	□	■	□	■	□
470	■	■	■	□	■	□	■	■	□	□
471	■	■	■	□	■	□	■	■	■	□
472	■	■	■	□	■	■	□	□	□	□
473	■	■	■	□	■	■	□	□	■	□
474	■	■	■	□	■	■	□	■	□	□
475	■	■	■	□	■	■	□	■	■	□
476	■	■	■	□	■	■	■	□	□	□
477	■	■	■	□	■	■	■	□	■	□
478	■	■	■	□	■	■	■	■	□	□
479	■	■	■	□	■	■	■	■	■	□
480	■	■	■	■	□	□	□	□	□	□
481	■	■	■	■	□	□	□	□	■	□
482	■	■	■	■	□	□	□	■	□	□
483	■	■	■	■	□	□	□	■	■	□
484	■	■	■	■	□	□	■	□	□	□
485	■	■	■	■	□	□	■	□	■	□
486	■	■	■	■	□	□	■	■	□	□
487	■	■	■	■	□	□	■	■	■	□
488	■	■	■	■	□	■	□	□	□	□
489	■	■	■	■	□	■	□	□	■	□
490	■	■	■	■	□	■	□	■	□	□
491	■	■	■	■	□	■	□	■	■	□
492	■	■	■	■	□	■	■	□	□	□
493	■	■	■	■	□	■	■	□	■	□
494	■	■	■	■	□	■	■	■	□	□
495	■	■	■	■	□	■	■	■	■	□
496	■	■	■	■	■	□	□	□	□	□
497	■	■	■	■	■	□	□	□	■	□
498	■	■	■	■	■	□	□	■	□	□
499	■	■	■	■	■	□	□	■	■	□
500	■	■	■	■	■	□	■	□	□	□
501	■	■	■	■	■	□	■	□	■	□
502	■	■	■	■	■	□	■	■	□	□
503	■	■	■	■	■	□	■	■	■	□
504	■	■	■	■	■	■	□	□	□	□
505	■	■	■	■	■	■	□	□	■	□
506	■	■	■	■	■	■	□	■	□	□
507	■	■	■	■	■	■	□	■	■	□
508	■	■	■	■	■	■	■	□	□	□
509	■	■	■	■	■	■	■	□	■	□
510	■	■	■	■	■	■	■	■	□	□
511	■	■	■	■	■	■	■	■	■	□

Table 3

Tables of Input-Output patterns for CORNER template (1, 2)

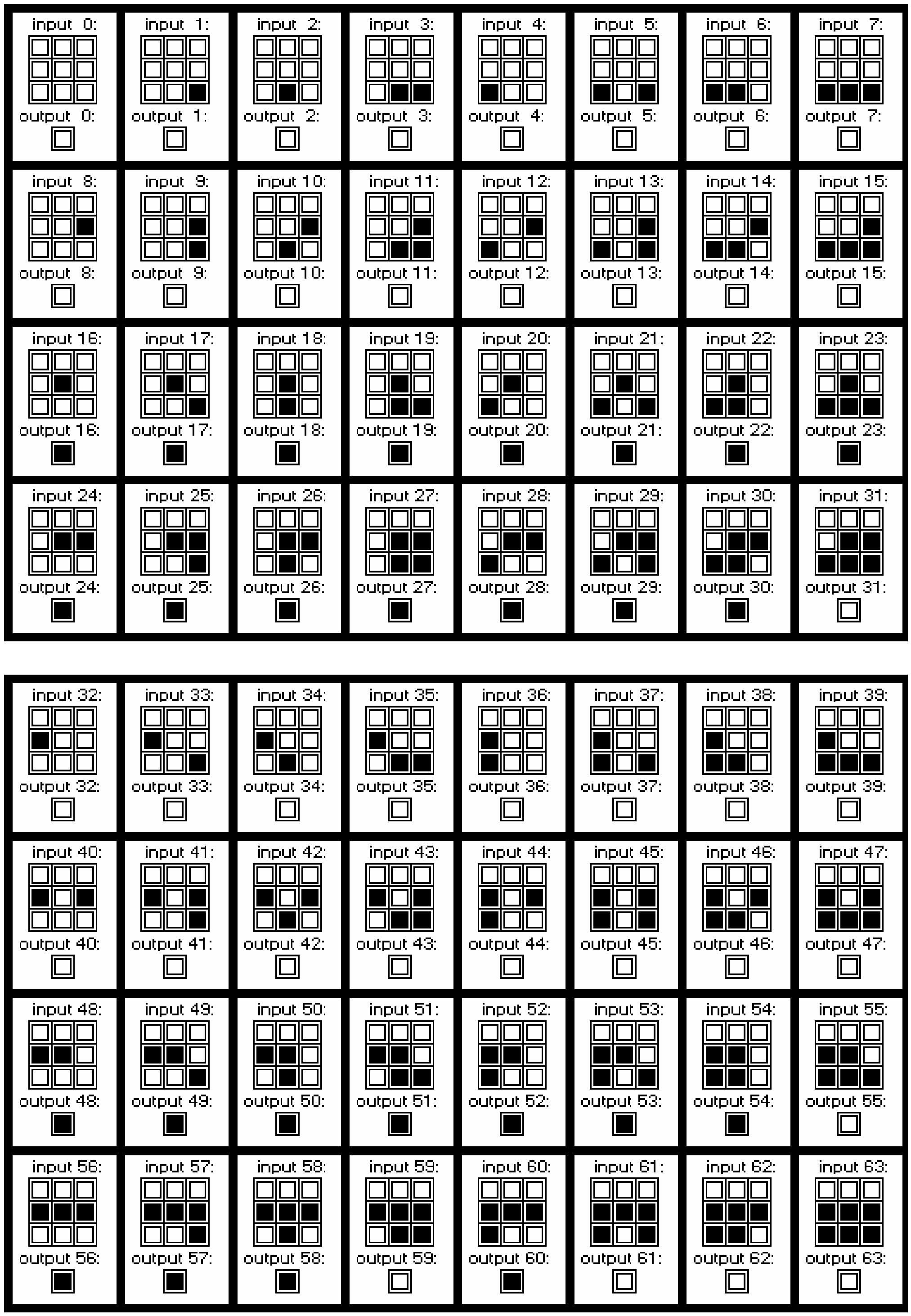
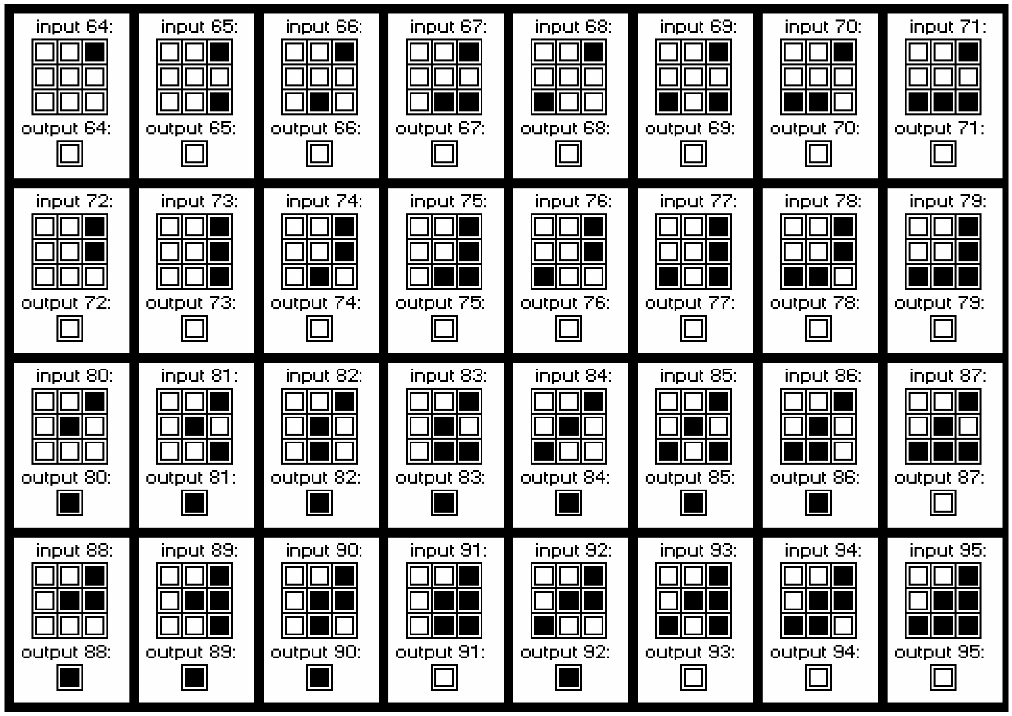


Table 4

Tables of Input-Output patterns for CORNER template (3, 4)



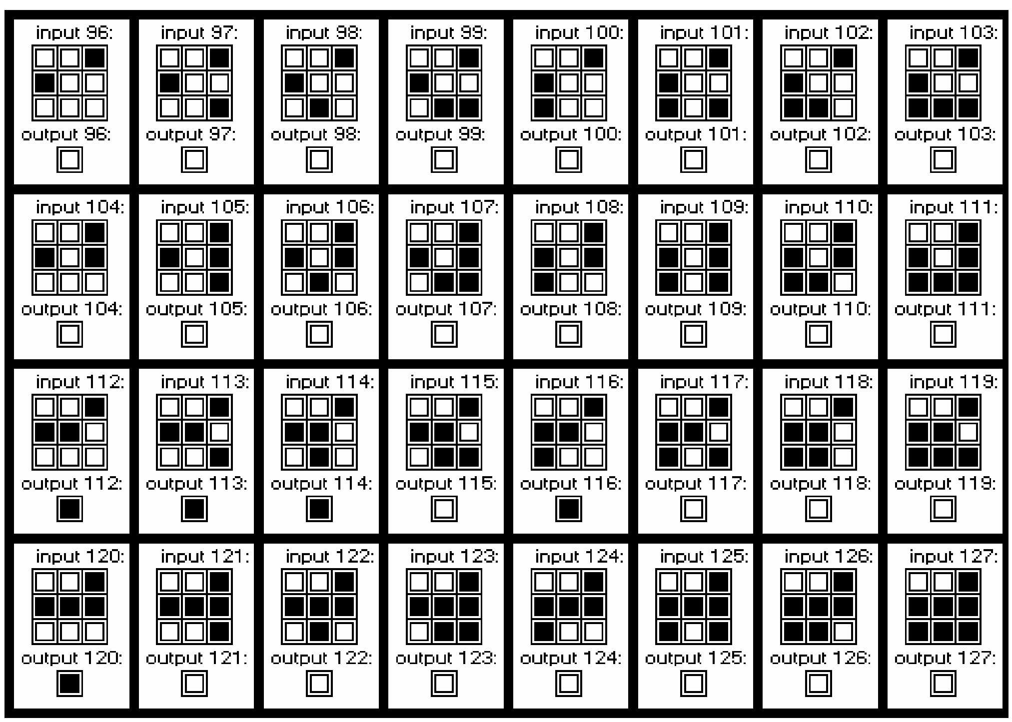
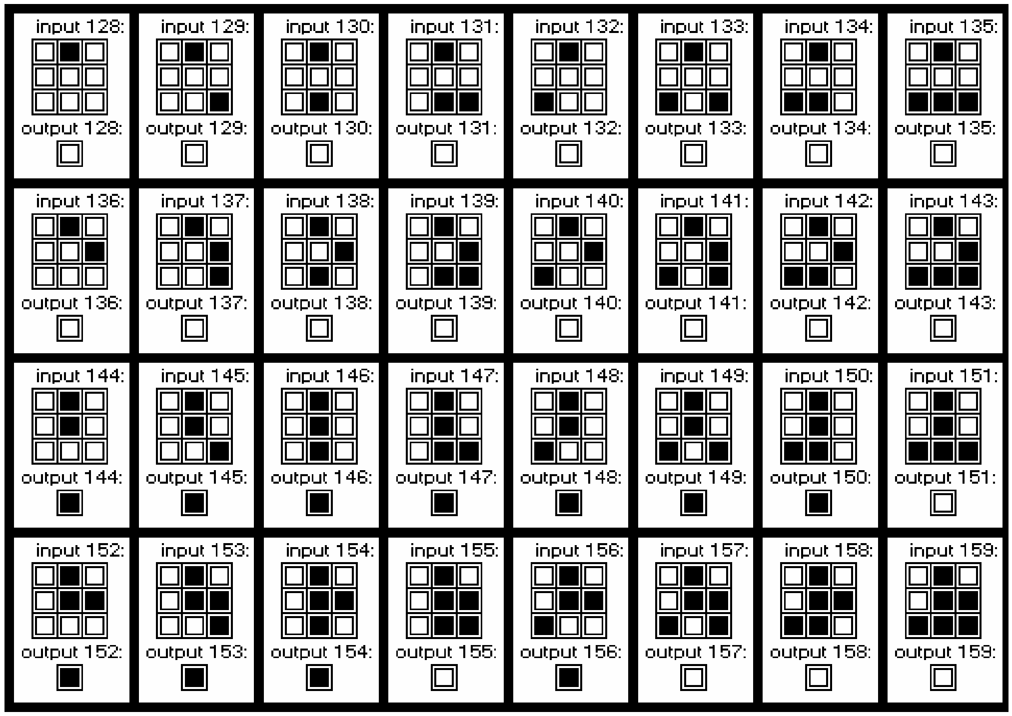


Table 5

Tables of Input-Output patterns for CORNER template (5, 6)



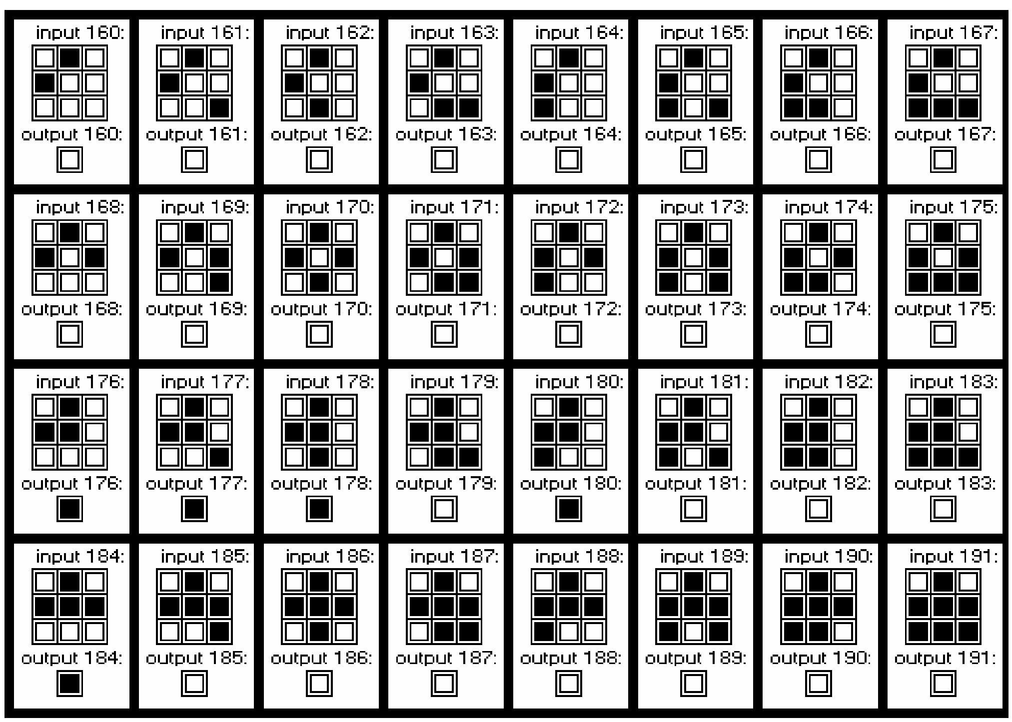
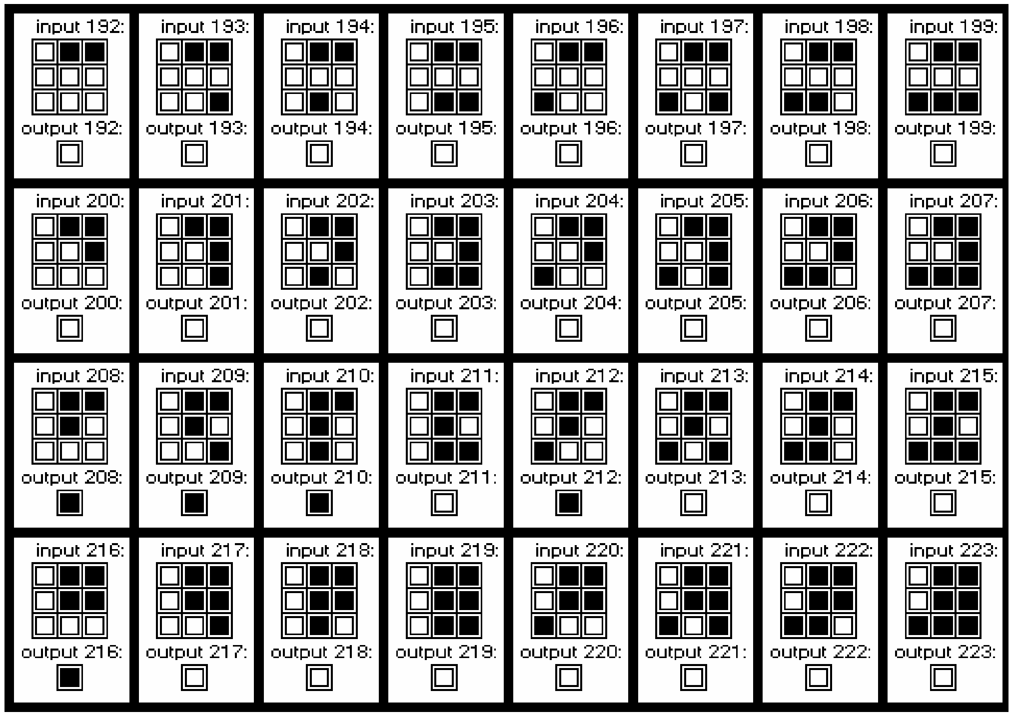


Table 6

Tables of Input-Output patterns for CORNER template (7, 8)



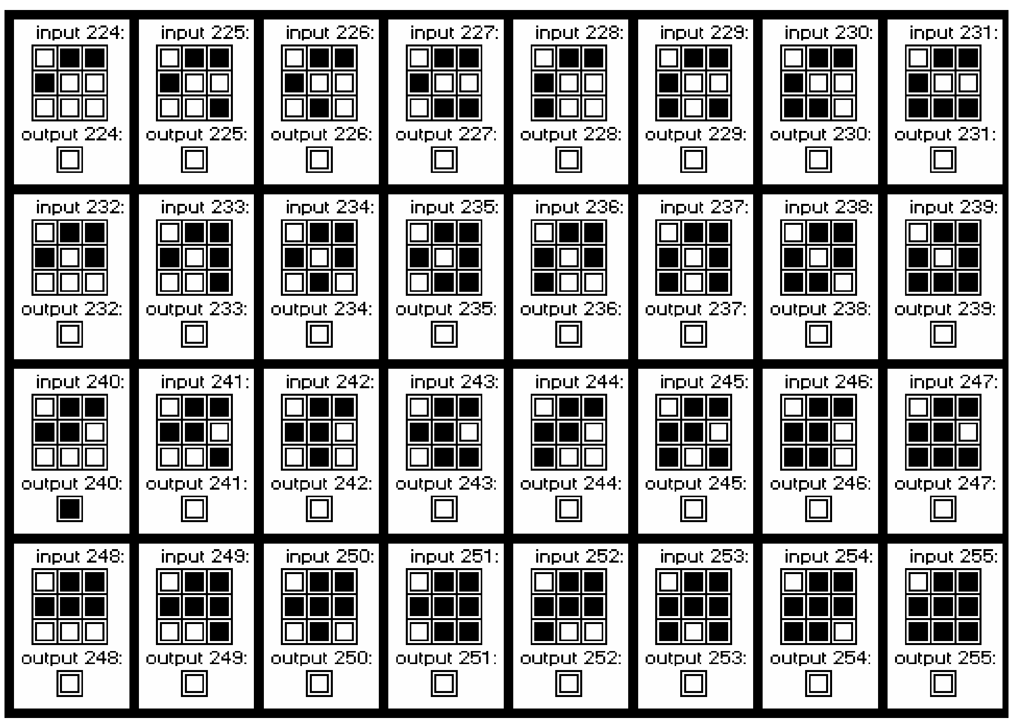
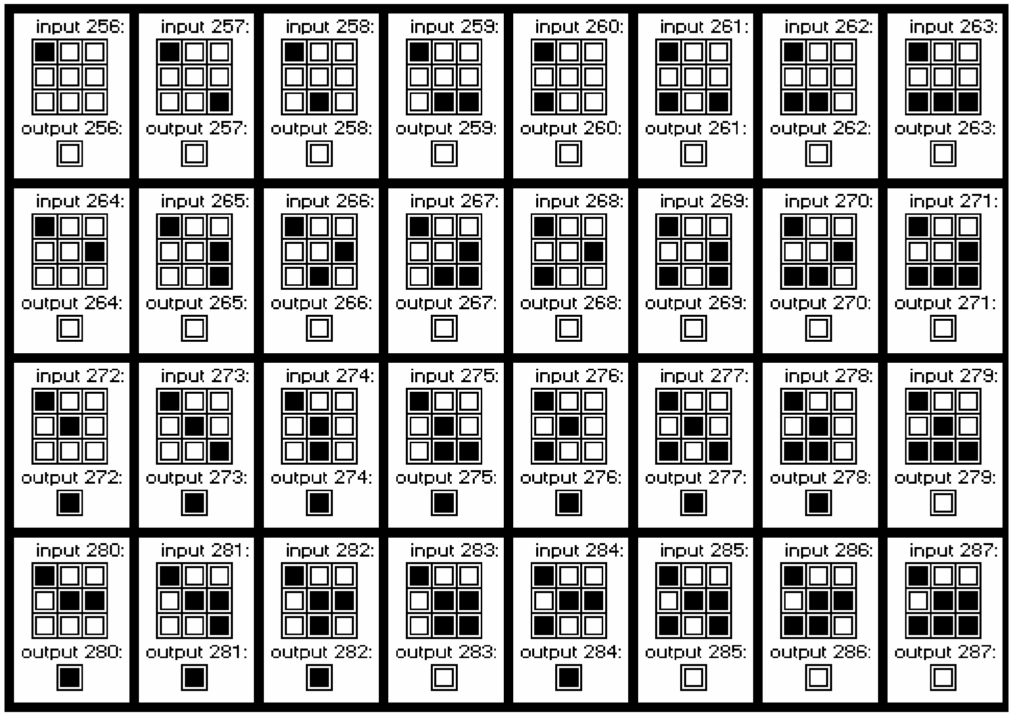


Table 7

Tables of Input-Output patterns for CORNER template (9, 10)



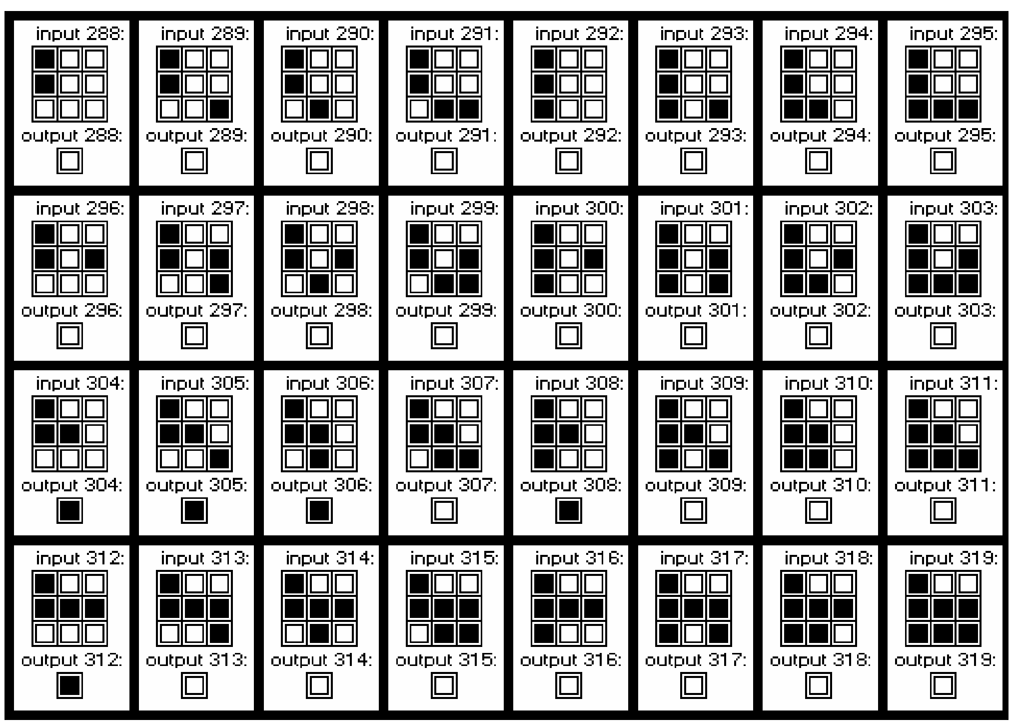
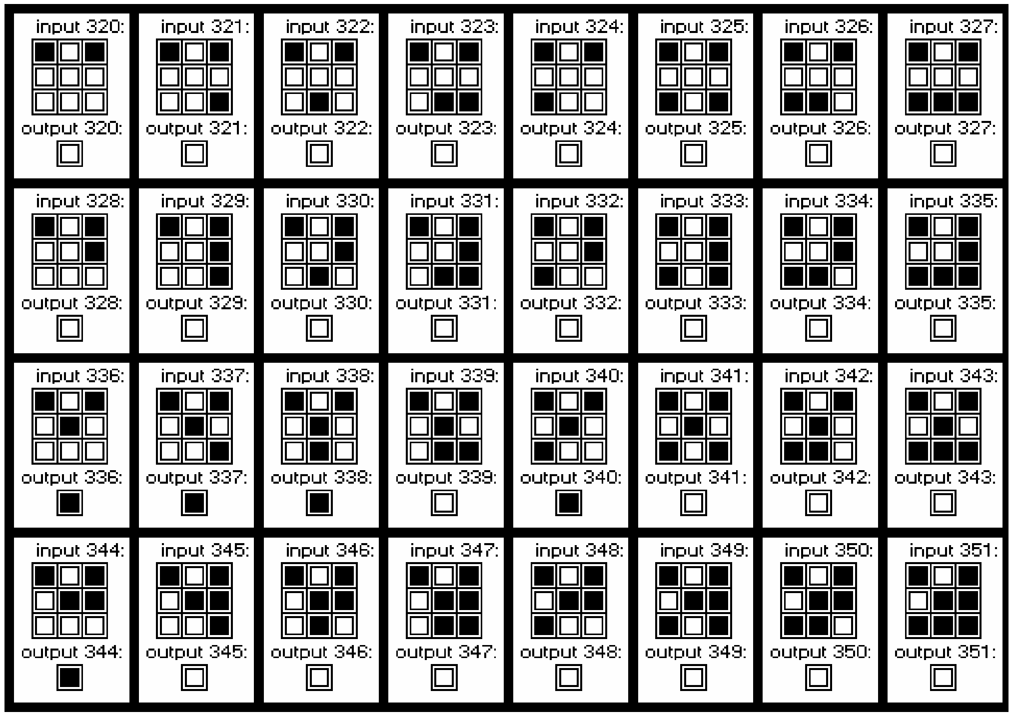


Table 8

Tables of Input-Output patterns for CORNER template (11, 12)



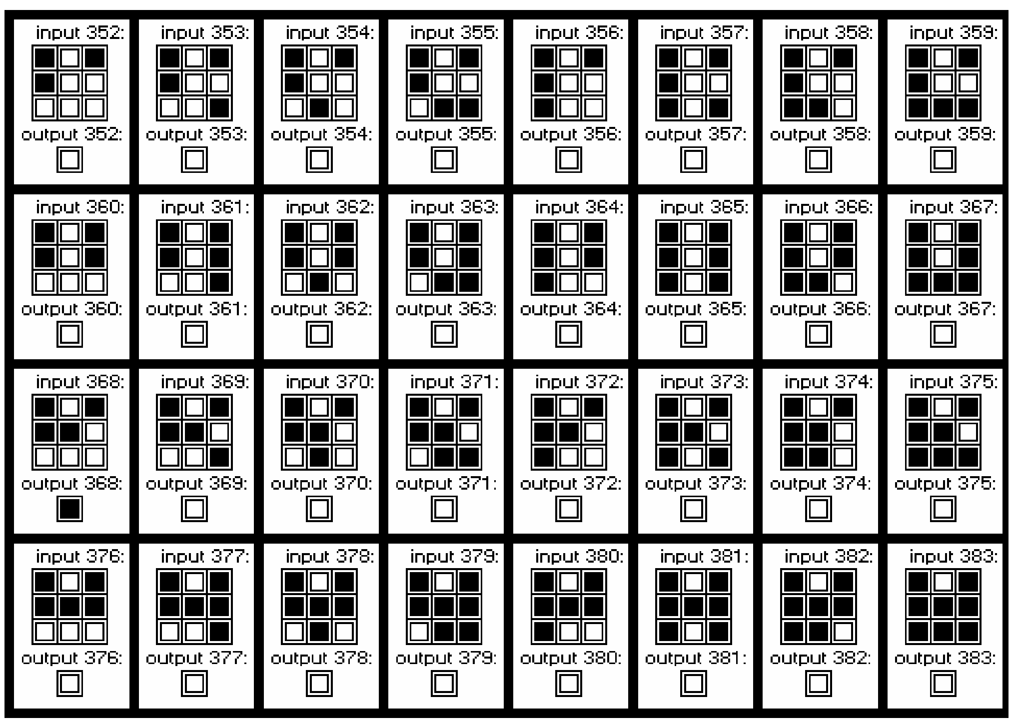
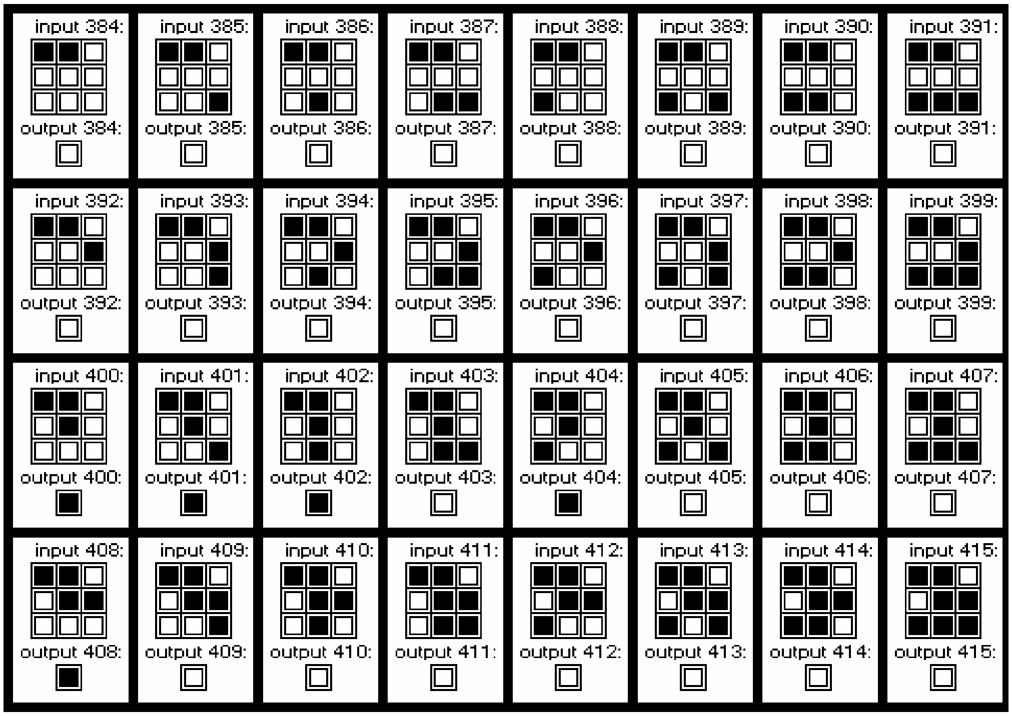


Table 9

Tables of Input-Output patterns for CORNER template (13, 14)



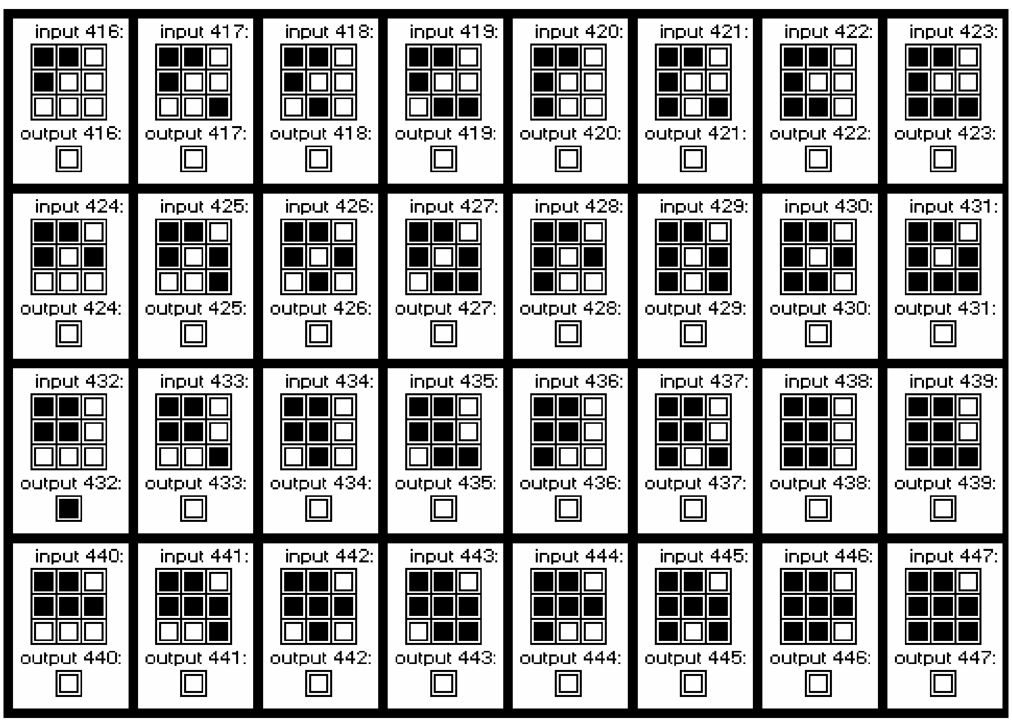
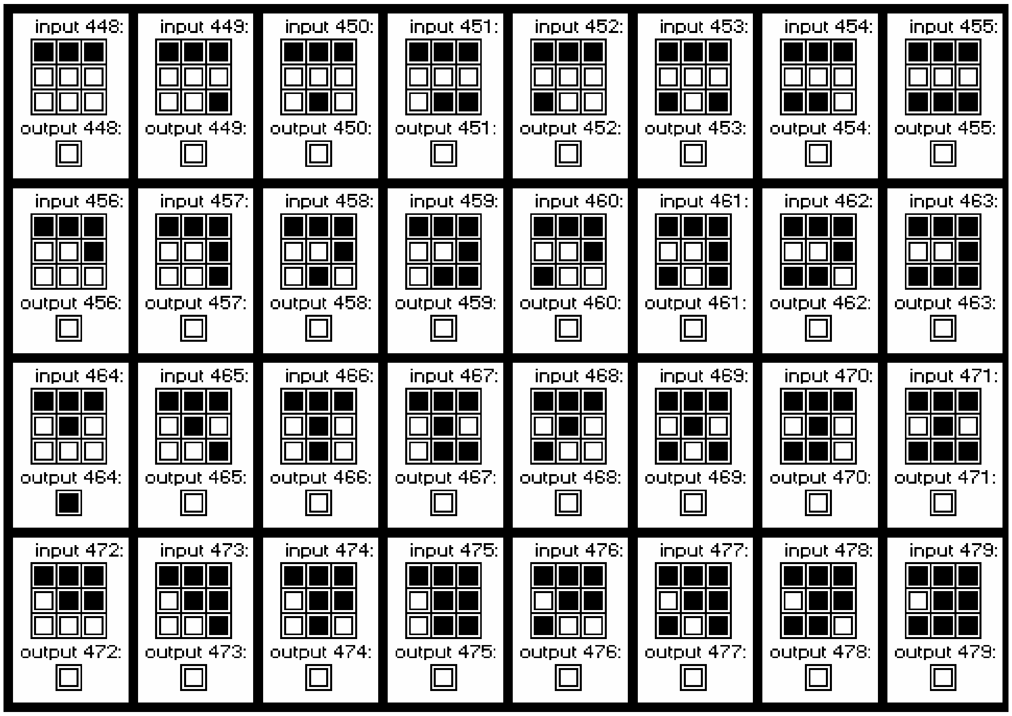


Table 10

Tables of Input-Output patterns for CORNER template (15, 16)



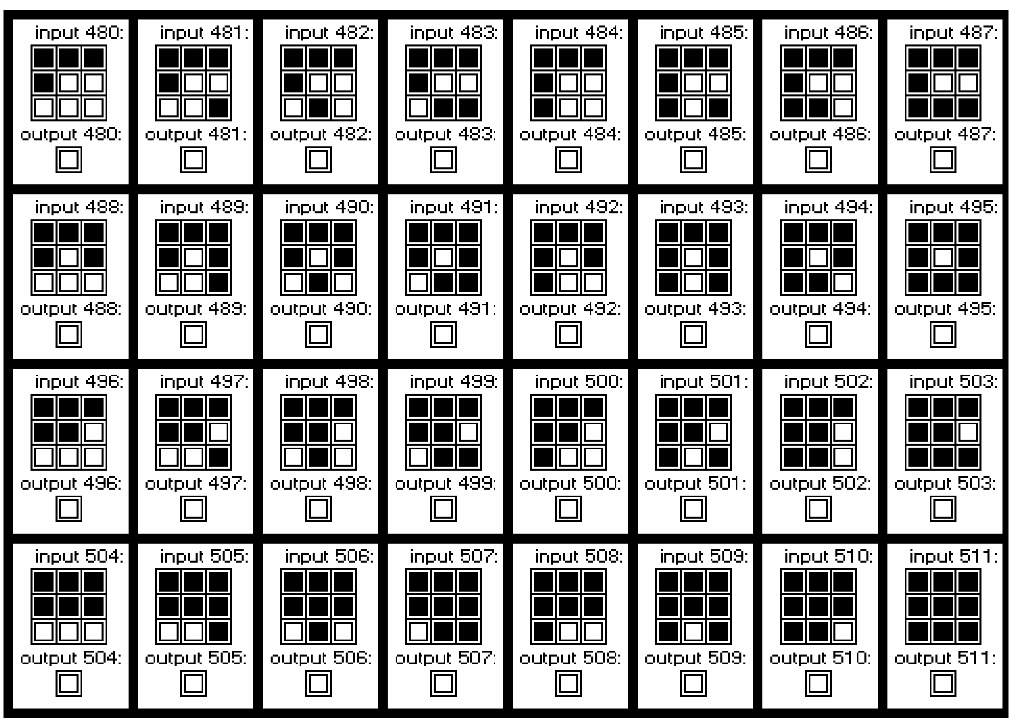


Table 11

Corrected Minimal Truth Table of CORNER template

squares correspond to corner misclassification (they should be white)

squares correspond to non-corner misclassification (they should be black)

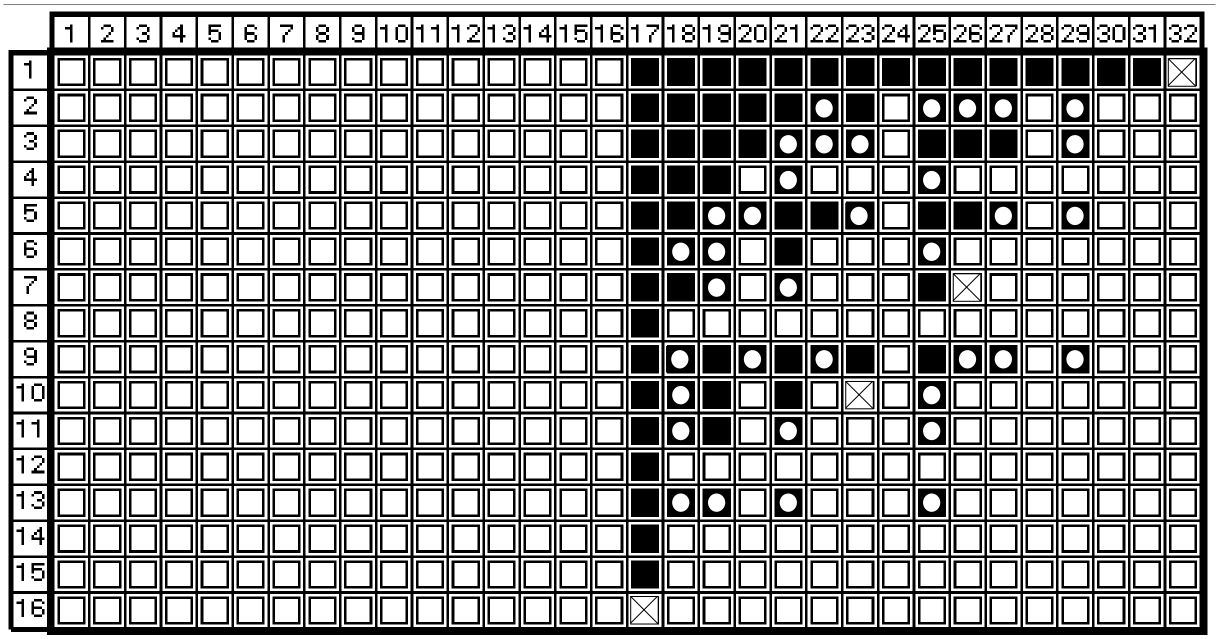


Table 12

Optimized Minimal Truth Table of CORNER Template

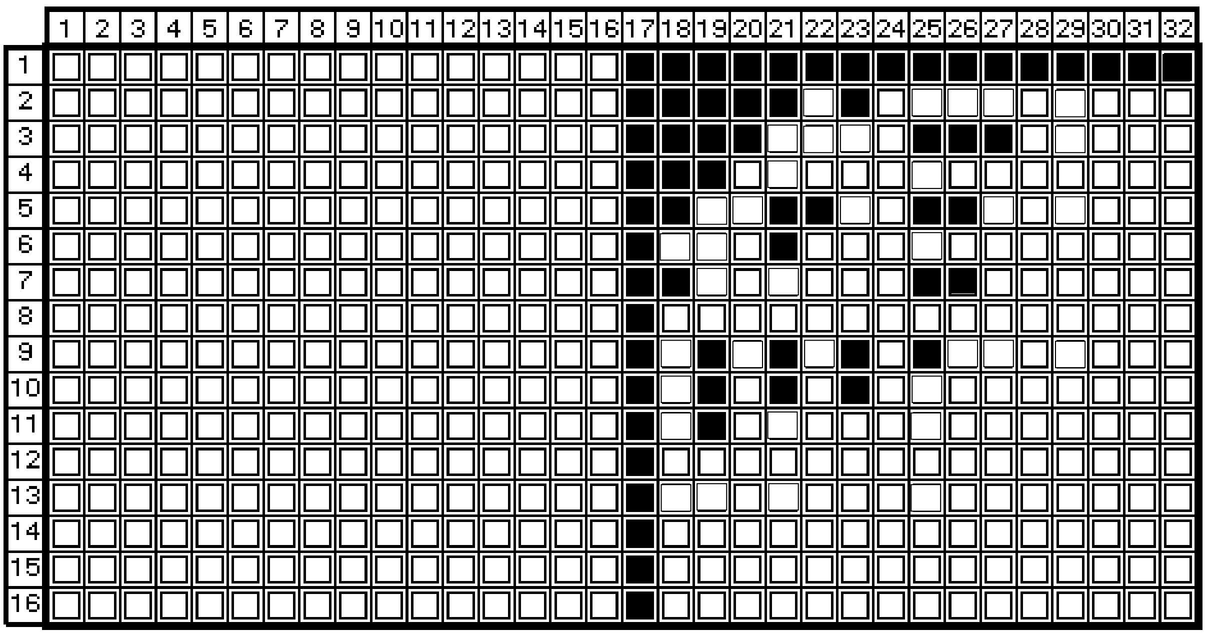


Table 13

Binary Code for ***CORNER*** Template
(512 bits)

00000000,00000000,00000000,00000000,00000000,00000001,00000000,00000000
,00000000,00000001,00000000,00000000,00000001,00010111,00000000,00000000
,00000000,00000001,00000000,00000000,00000001,00010111,00000000,00000000
,00000001,00010111,00000000,00000000,00010111,01111111,00000000,00000000
,00000000,00000001,00000000,00000000,00000001,00010111,00000000,00000000
,00000001,00010111,00000000,00000000,00010111,01111111,00000000,00000000
,00000001,00010111,00000000,00000000,00010111,01111111,00000000,00000000
,00010111,01111111,00000000,00000000,01111111,11111111,00000000,00000000

Decimal Code for ***CORNER*** Template
(140 digits)

47,634,102,646,527,572,675,971
,460,498,910,645,354,219,674,273,748,634,236,474,670
,546,006,561,432,941,907,354,541,093,642,727,873,594
,350,604,011,030,198,552,062,948,695,326,343,495,680

Table 14

Binary Code for ***Optimized CORNER*** Template
(512 bits)

,00000000,00000001,00000000,00000000,00000000,00000001,00000000,00000000
,00000000,00000001,00000000,00000000,00000000,00000001,00000000,00000000
,00000000,00000001,00000000,00000000,00000000,00000101,00000000,00000000
,00000000,01010101,00000000,00000000,00000001,01010101,00000000,00000000
,00000000,00000001,00000000,00000000,00000011,00000011,00000000,00000000
,00000000,00010001,00000000,00000000,00000011,00110011,00000000,00000000
,00000000,00000111,00000000,00000000,00000111,00001111,00000000,00000000
,00000000,01011111,00000000,00000000,11111111,11111111,00000000,00000000

Decimal Code for ***Optimized CORNER*** Template
(150 digits)

204,586,913,041,142,969,522,351,928,009,830
,941,404,290,185,269,210,065,083,499,186,859,428,943
,804,165,897,630,843,608,945,882,697,576,708,597,045
,469,082,137,675,717,688,639,024,082,912,326,647,808

Table 15