Computer Vision

Project part 2

Team 4

M. Meurisse, F. Rozet, O. Rumfels and V. Vermeylen

December 20, 2019

University of Liège

Introduction

Divided into 2 parts :

1. Performance assessment of our line detection algorithms
2. Digits recognition in sudoku grids

Plan of the presentation

1. Performance assessment of our line detection algorithms
   - Line segments recovery
   - Performance assessment
2. Digits recognition in sudoku grids
3. Neural networks
4. Performance evaluation

Performance assessment of our line detection algorithms

Line segments recovery

6 information to store per image :

- the 4 coordinates of the 2 end points of the segments
- the width and height

Line segments recovery - Final database

3 `.dat` files per image :

line_sudoku_00001.dat

```
554 1037 575 446
425 1030 446 446
116 755 655 774
172 1020 192 439
361 1027 382 444
366 845 380 444
236 1023 256 441
107 946 695 977
235 1022 255 441
122 507 711 517
360 1027 380 445
693 990 711 462
```

line_sudoku_00001.dat

```
690 335 711 925
642 925 622 339
556 341 575 924
510 925 491 345
445 928 427 347
363 350 381 928
319 928 297 352
235 356 254 930
190 932 171 359
108 361 127 934
688 335 108 361
110 426 690 400
```

line_sudoku_00001.dat

```
1376 774
```

Figure 1: Cytomine (left), personal (center) and dimensions (right)

Performance assessment

4 methods :

- a naive method : visual comparison
- distance calculation
- adjacent pixels
- confusion matrix

Performance assessment

4 methods :

- a naive method : visual comparison
- distance calculation
- adjacent pixels
- confusion matrix

Confusion matrix

The complete evaluation process for an image :

1. Create a binary matrix
2. Thicken the segments of our algorithm
3. Calculate the confusion matrix
4. Calculate several measurements

Binary matrix

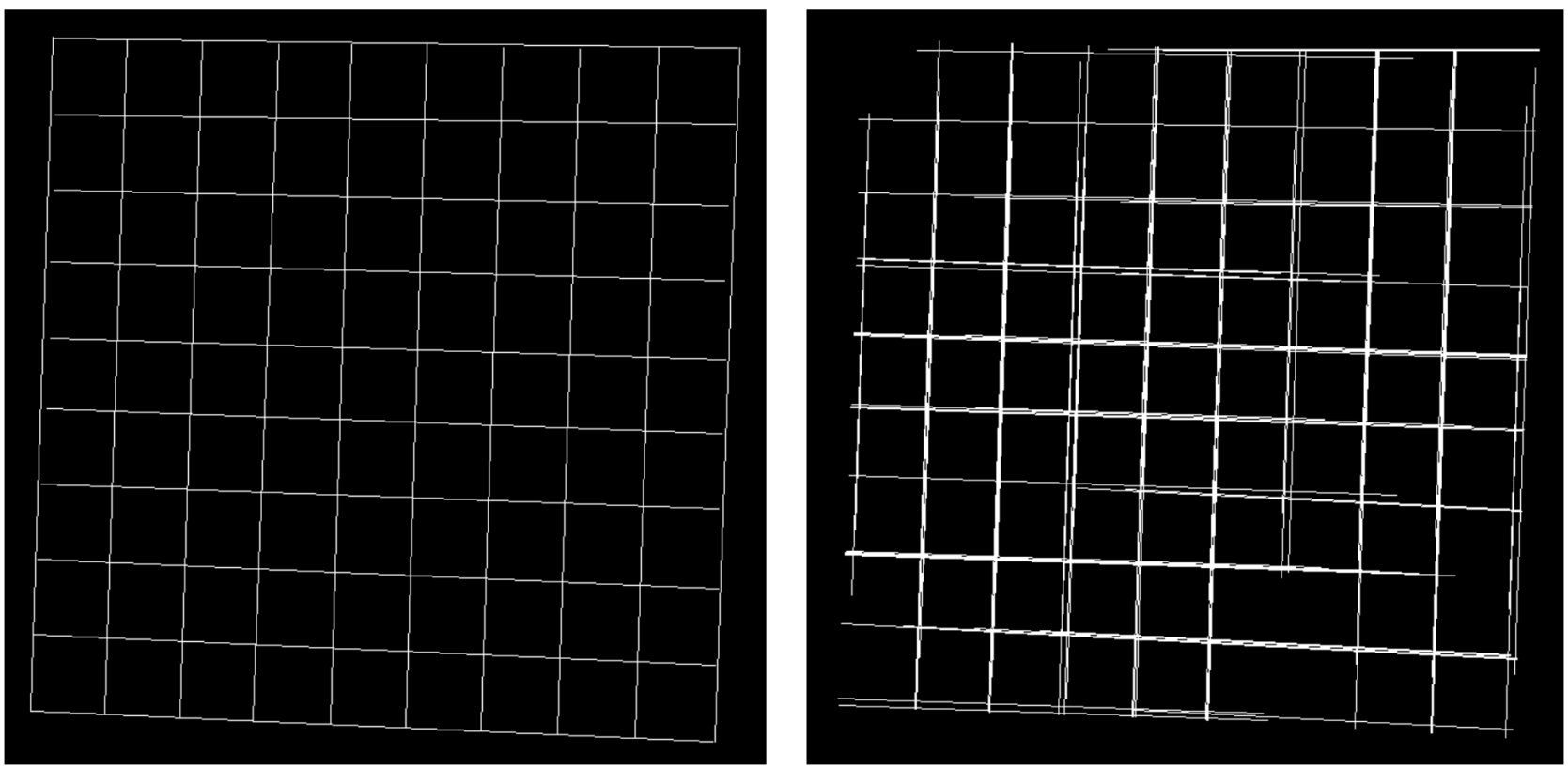


Figure 2: Cytomine (left) and personal (right)

Thicken the segments

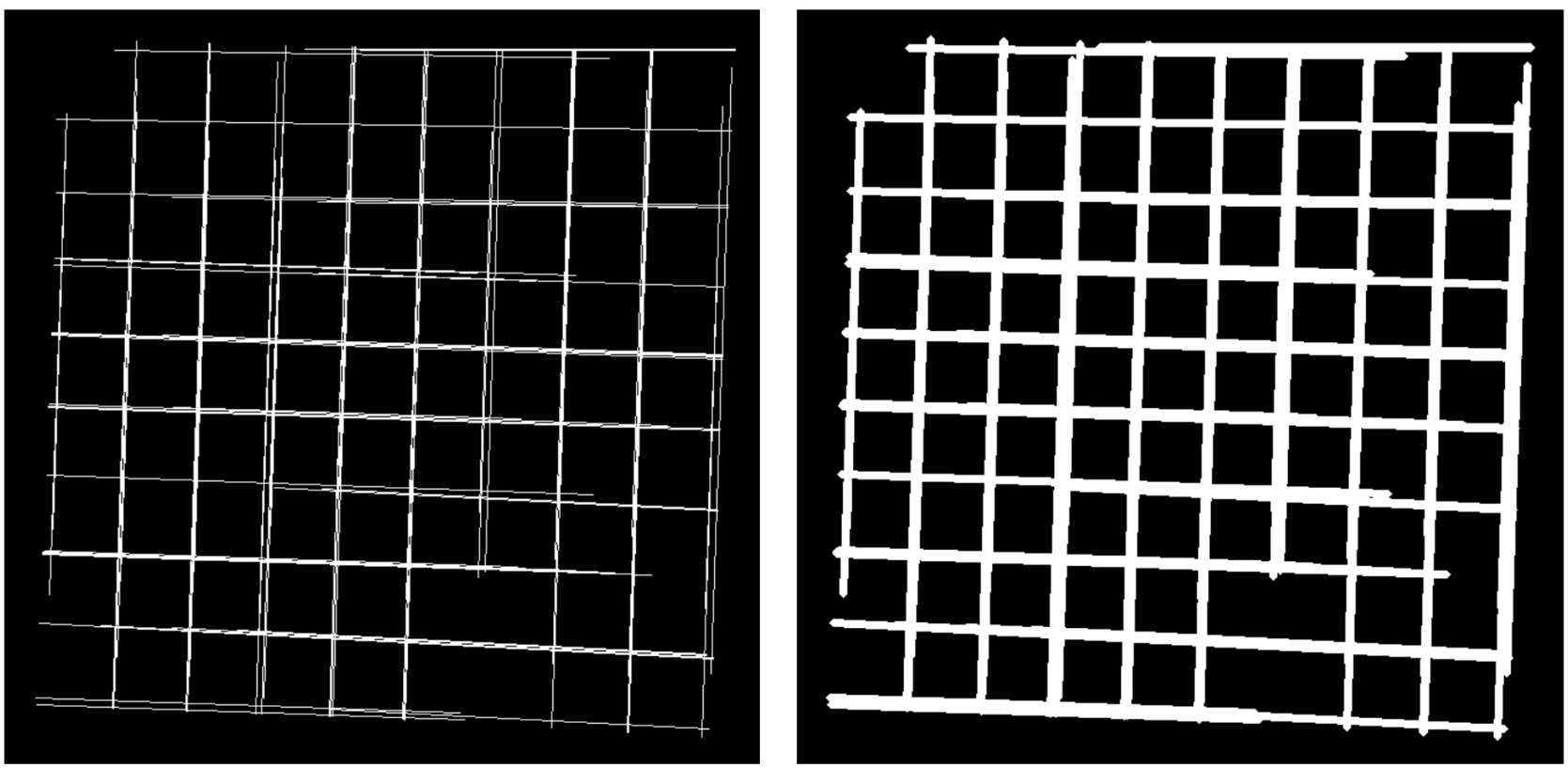


Figure 3: Original (left) and thickened (right)

Confusion matrix

The confusion matrix quantities :

	Our algorithm	Reference
True Positive (TP)	1	1
True Negative (TN)	0	0
False Positive (FP)	1	0
False Negative (FN)	0	1

0 : do not belong to a segment

1 : belong to a segment

Measurement and performance evaluation

- **recall** :

$$recall = \frac{TP}{TP + FN}$$

- **specificity** :

$$specificity = \frac{TN}{TN + FP}$$

- **precision** :

$$precision = \frac{TP}{TP + FP}$$

Performance evaluation

Mean value, per image class, for each measurement :

	recall	**specificity**	**precision**
sudoku	0.9252	0.9749	0.3023
road	0.4471	0.9940	0.2383
soccer	0.7064	0.9944	0.3126

Sudoku comparison

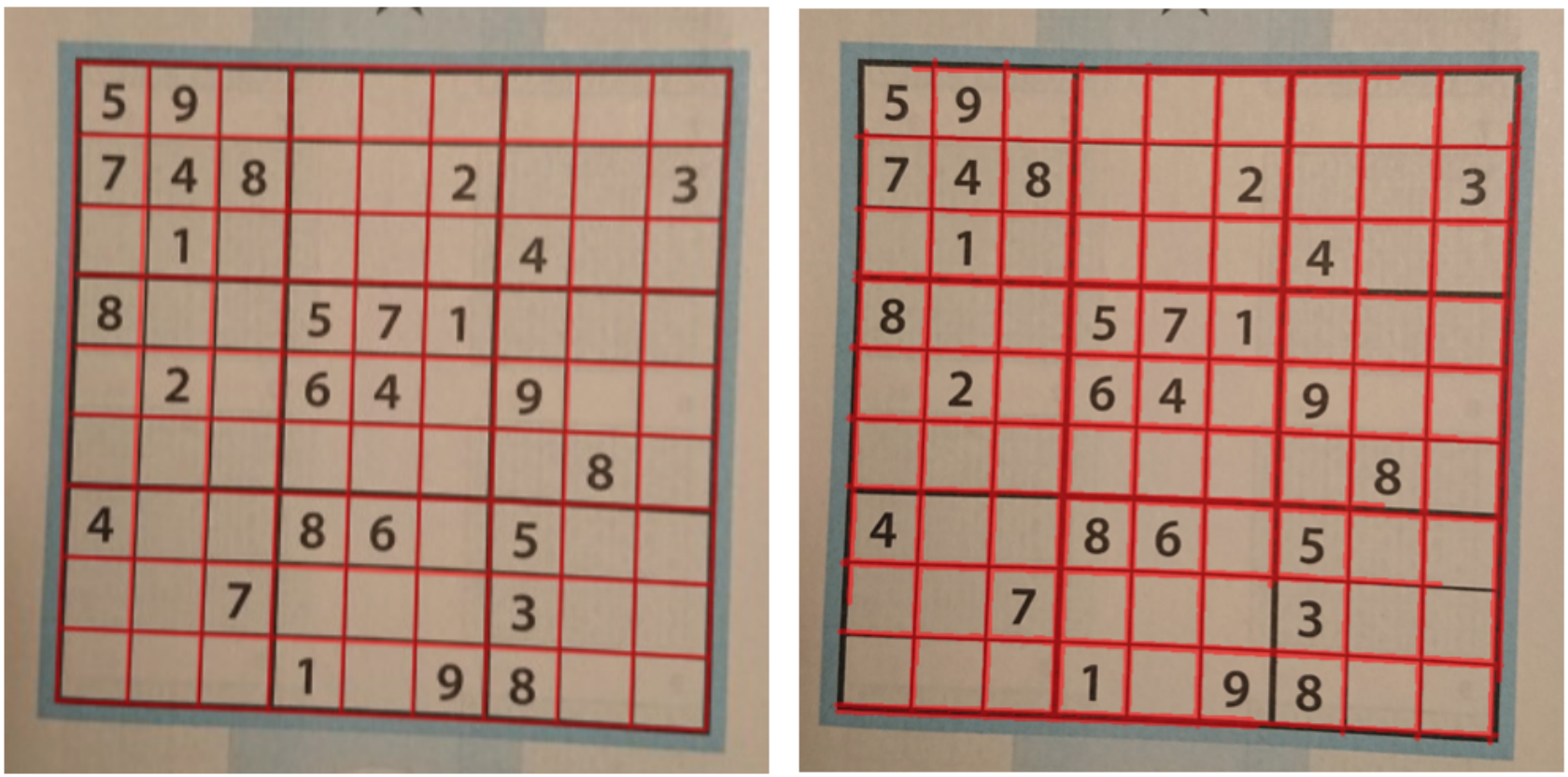


Figure 4: Original (left) and personal (right)

Road comparison

Figure 5: Original (left) and personal (right)

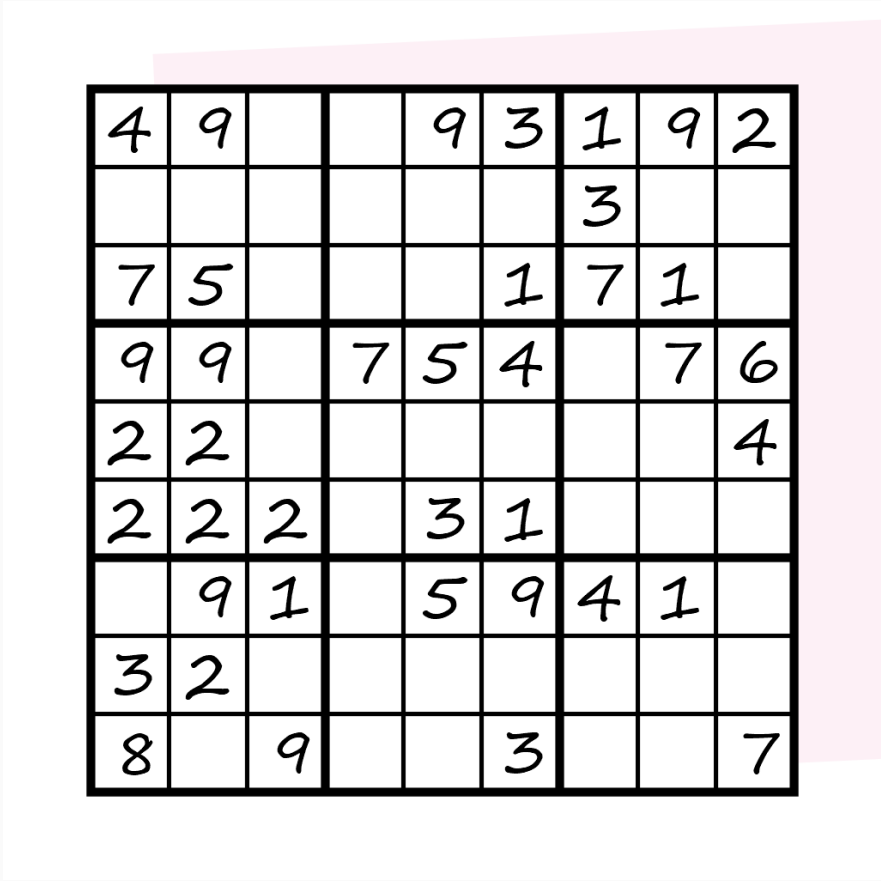
Soccer comparison

Figure 6: Original (top) and personal (bottom)

Digits recognition in sudoku grids

Creation of the database

Python script using the graphical library `pycairo` (400 images)



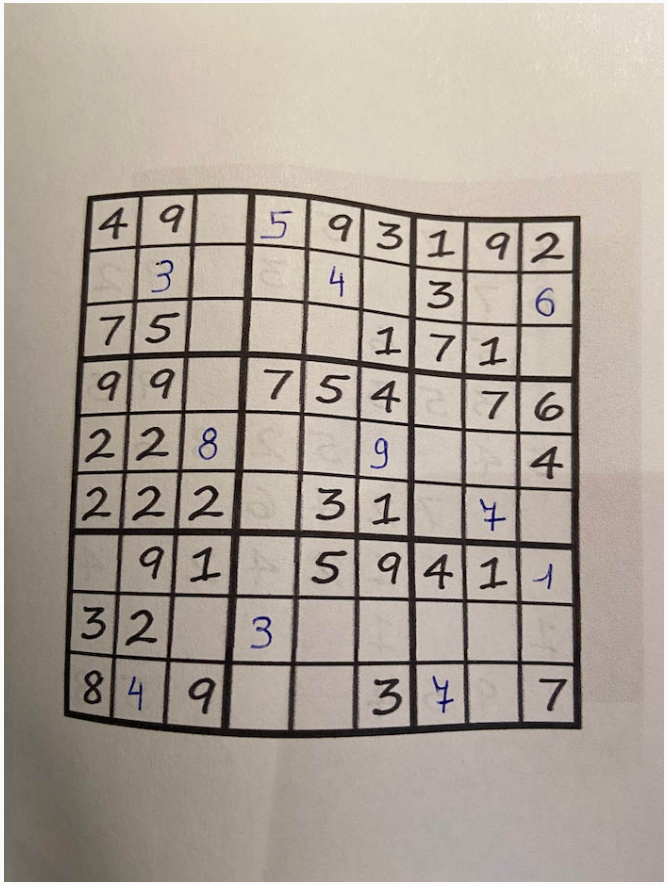

```
4 9 0 0 9 3 1 9 2
0 0 0 0 0 0 3 0 0
7 5 0 0 0 1 7 1 0
9 9 0 7 5 4 0 7 6
2 2 0 0 0 0 0 0 4
2 2 2 0 3 1 0 0 0
0 9 1 0 5 9 4 1 0
3 2 0 0 0 0 0 0 0
8 0 9 0 0 3 0 0 7
```

Associated `.dat` file.

Figure 7: Digitally generated Sudoku grid.

Creation of the database

Printing, addition of handwritten digits and photograph.



```
4 9 0 5 9 3 1 9 2
0 3 0 0 4 0 3 0 6
7 5 0 0 0 1 7 1 0
9 9 0 7 5 4 0 7 6
2 2 8 0 0 9 0 0 4
2 2 2 0 3 1 0 7 0
0 9 1 0 5 9 4 1 1
3 2 0 3 0 0 0 0 0
8 4 9 0 0 3 7 0 7
```

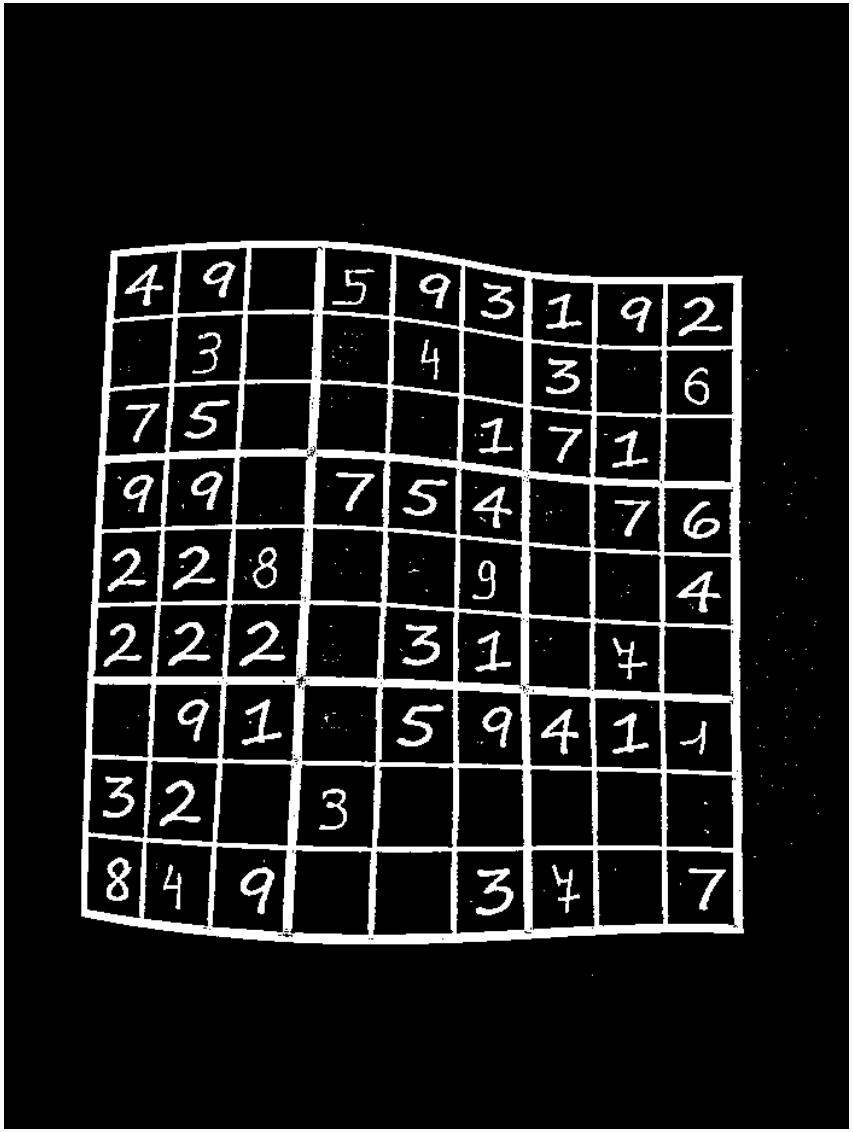
Associated `.dat` file.

Figure 8: Sudoku with handwritten digits.

Grid detection

Adaptive threshold : threshold value for each pixel based on surrounding.

Better management of lighting conditions (darkness, shadows, ...) than overall threshold.

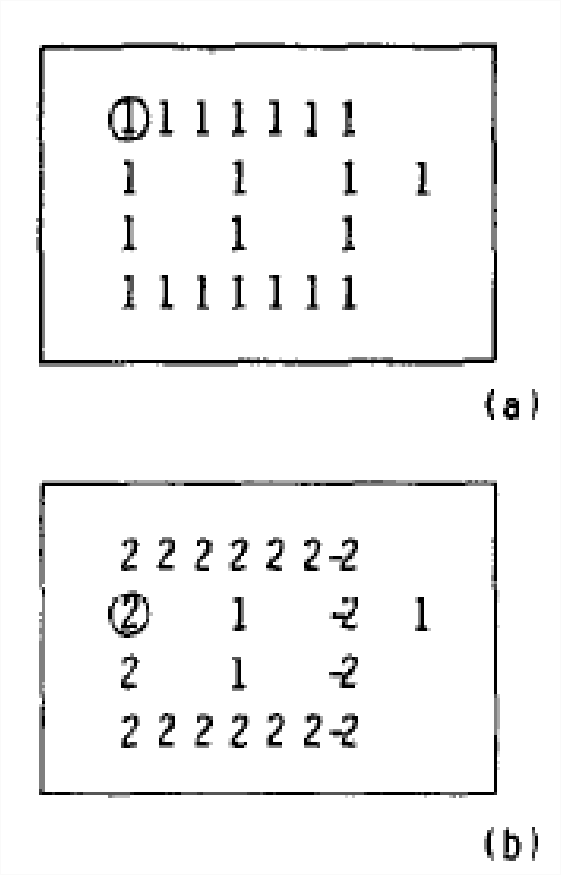


Contour finding procedure [1]

Raster scan until pixel change (border).

Follow border and give label to pixels.

Continue raster scan.



[1] Topological Structural Analysis of Digitized Binary Images by Border Following, S. Suzuki and K. Abe

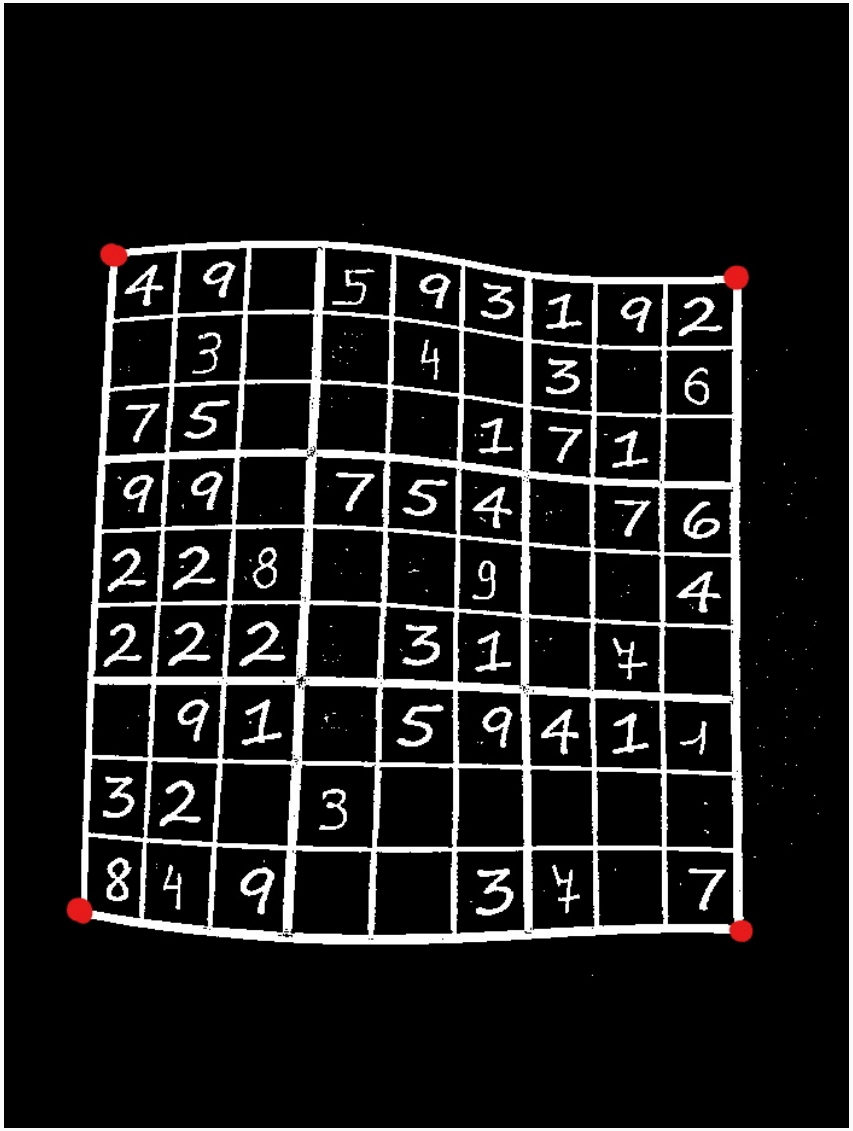
Grid detection

Transformation matrix such that

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{pmatrix} x_i' \\ y_i' \\ 1 \end{pmatrix} t_i$$

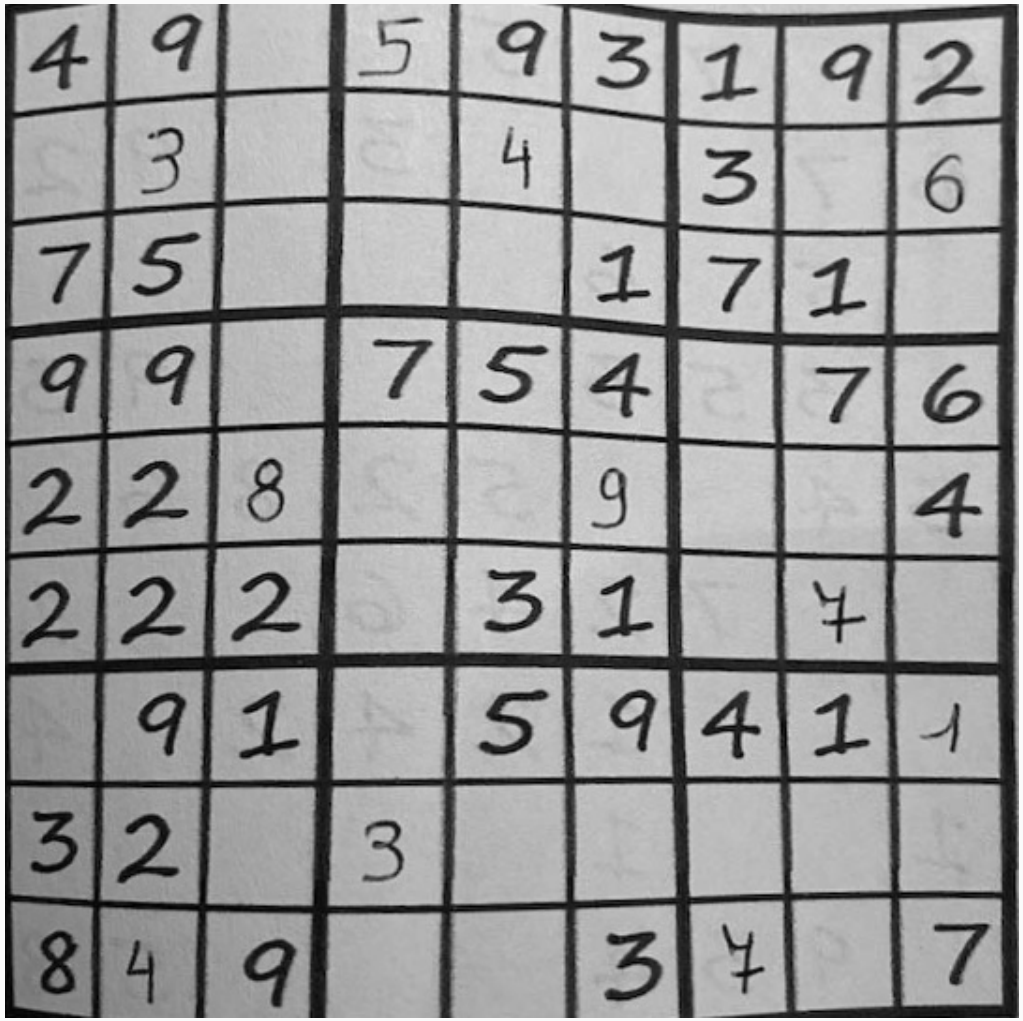
$$\forall i = 0, 1, 2, 3$$

12 unknowns, 12 equations



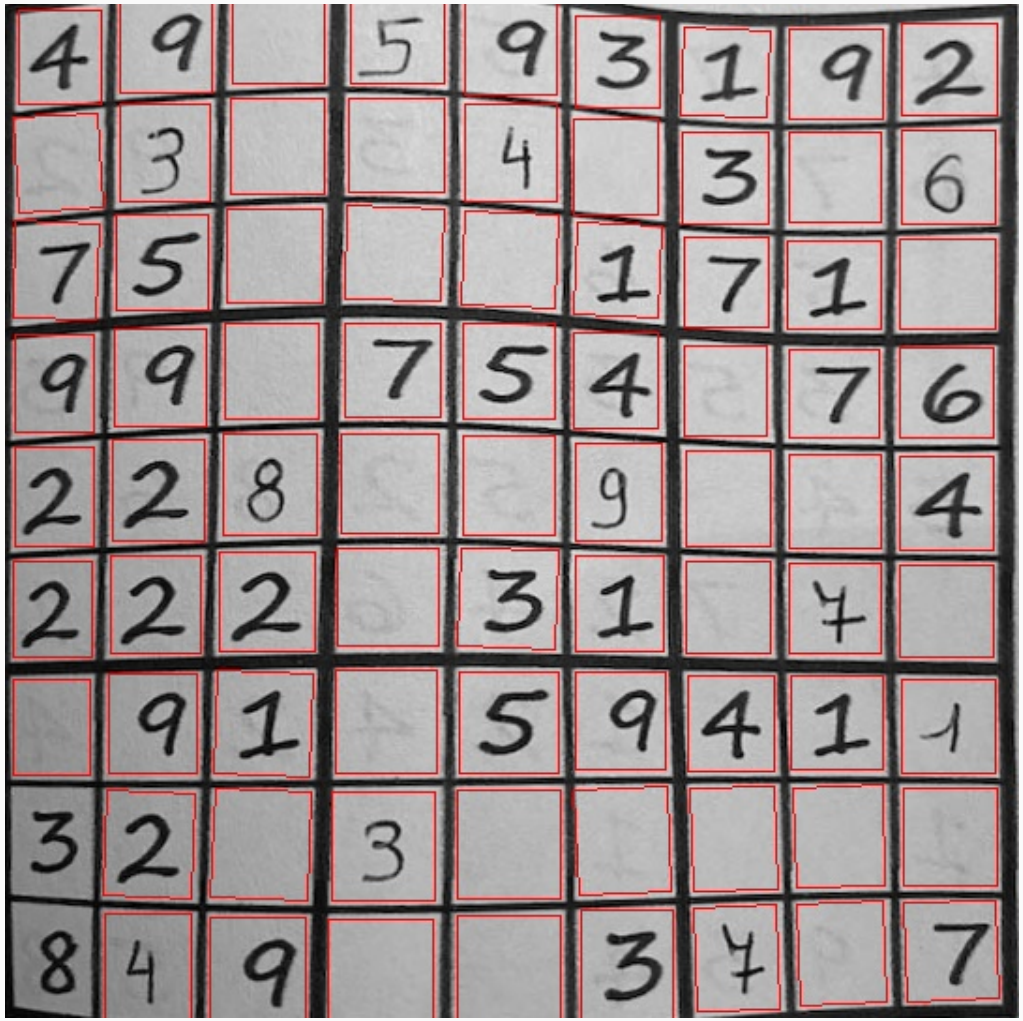
Grid detection

Apply transformation matrix to all pixels to correct perspective.



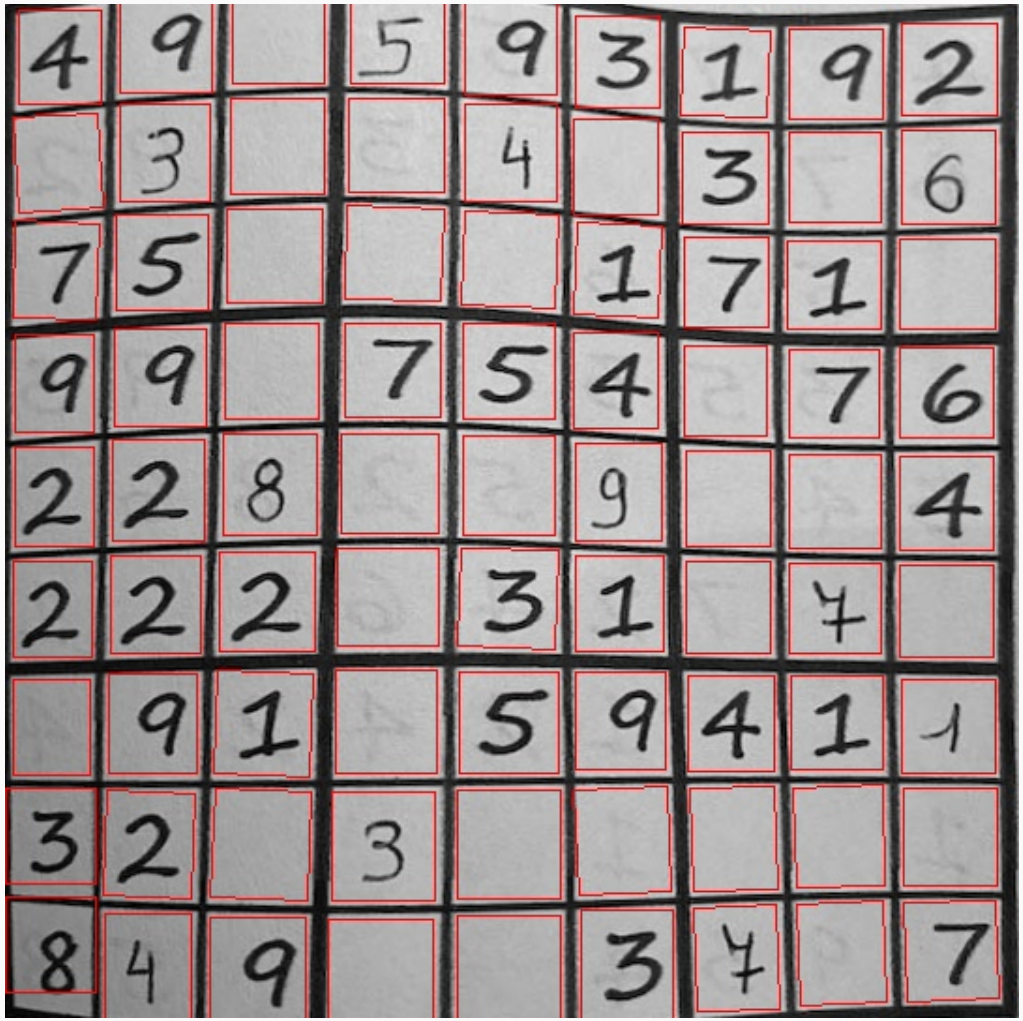
Cell detection

Contour finding procedure(s) aiming for cells.



Cell detection

Guessing position of missing cells.



Digits recognition

Neural network(s) for digit recognition.

Digits recognition

4	9	7	5	9	3	1	9	2
0	3	0	0	4	0	3	0	6
7	5	0	0	0	1	7	1	0
9	9	0	7	5	4	0	7	6
2	2	8	0	0	9	0	0	4
2	2	2	0	3	1	0	4	0
0	9	1	0	5	9	4	1	1
3	2	0	3	0	0	0	0	0
8	4	9	2	2	3	7	0	7

Constructed `.dat` file.

Neural networks

Network design - Ghouzam

MNIST Data Preparation

- Normalization.
- Reshaping.
- One-hot encoding.
- Data augmentation.

Network design - Ghouzam

Neural Network Model

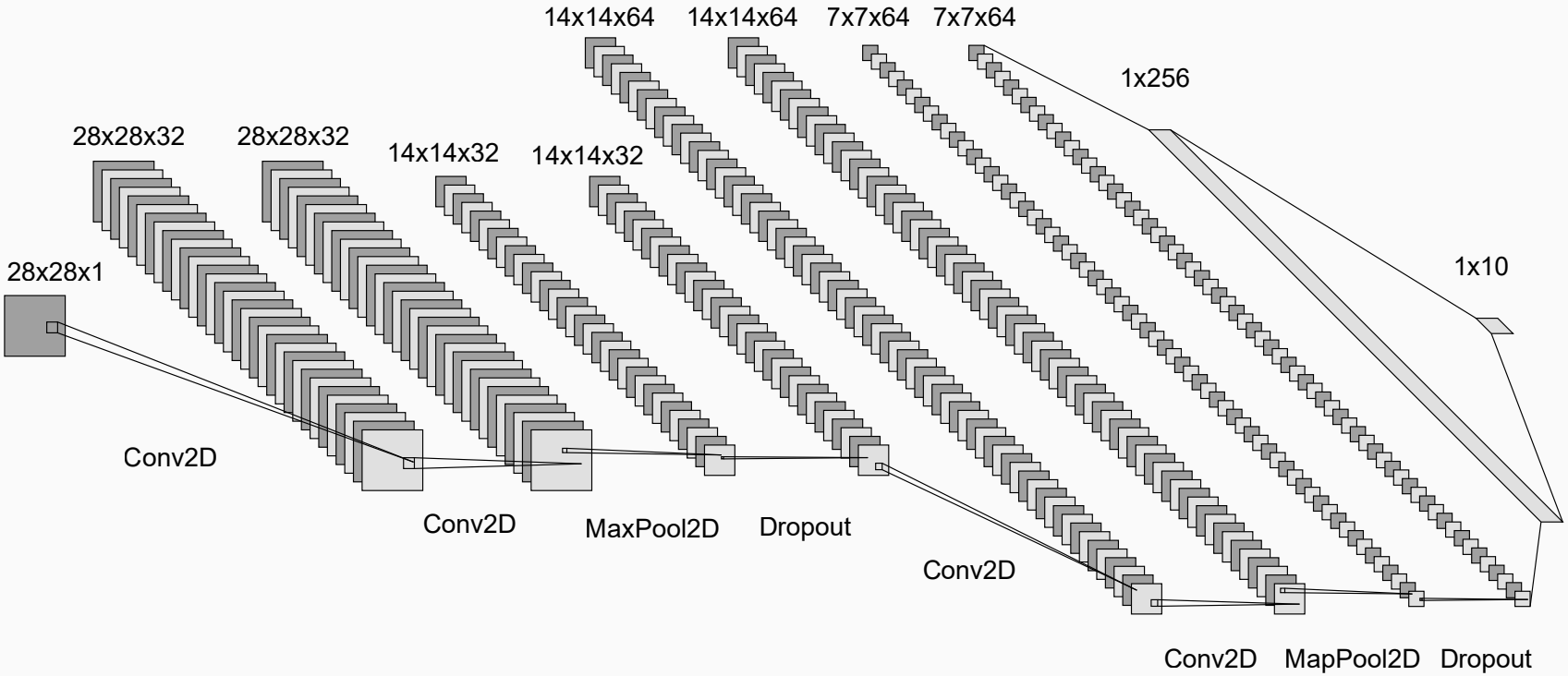


Figure 9: Ghouzam neural network model.

Pre-Trained Models

- VGG16
- MobileNet
- ResNet164

VGG16

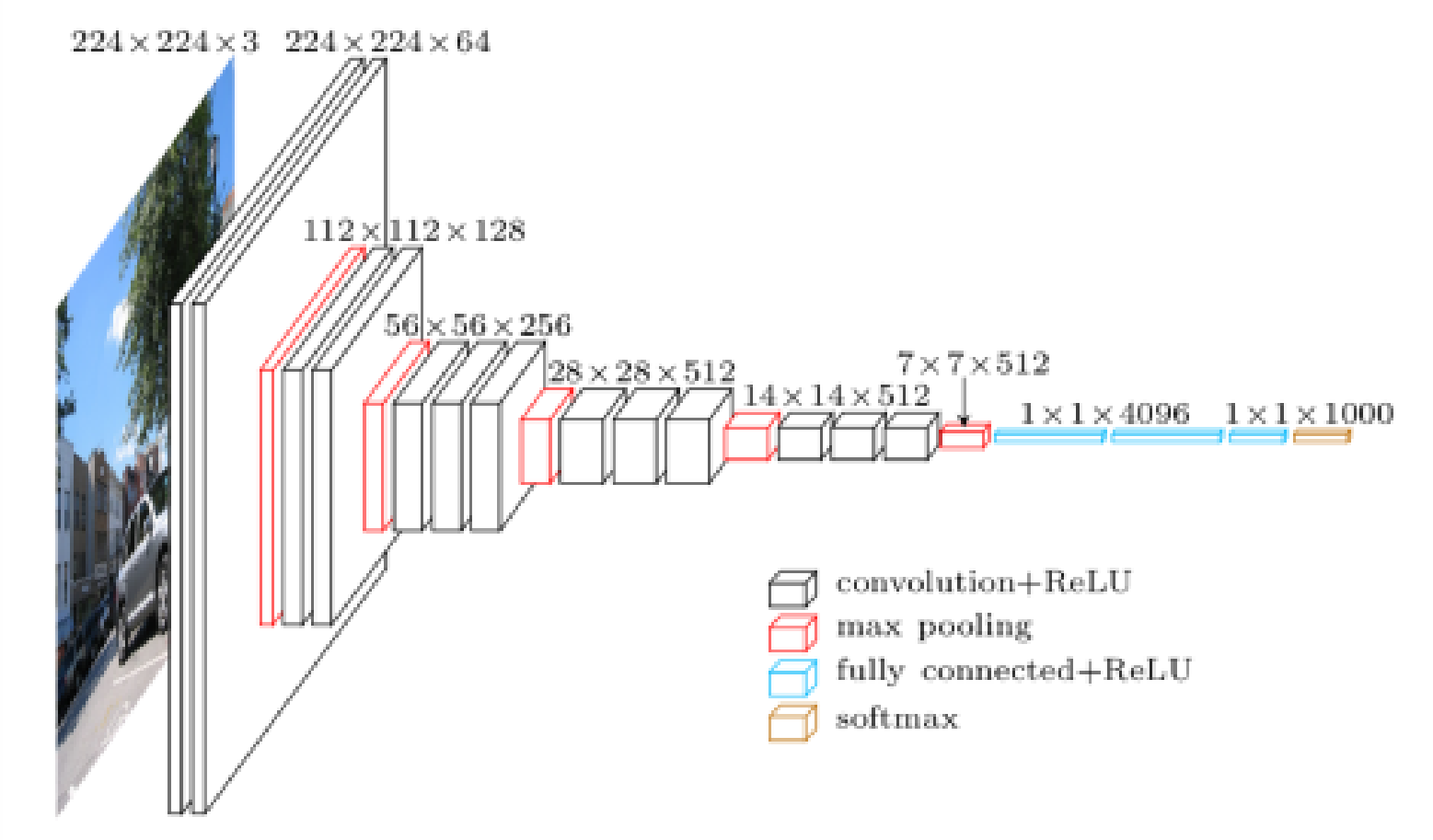


Figure 10: VGG16 structure. [2]

[2] https://www.quora.com/How-does-VGG-network-architecture-work

MobileNet

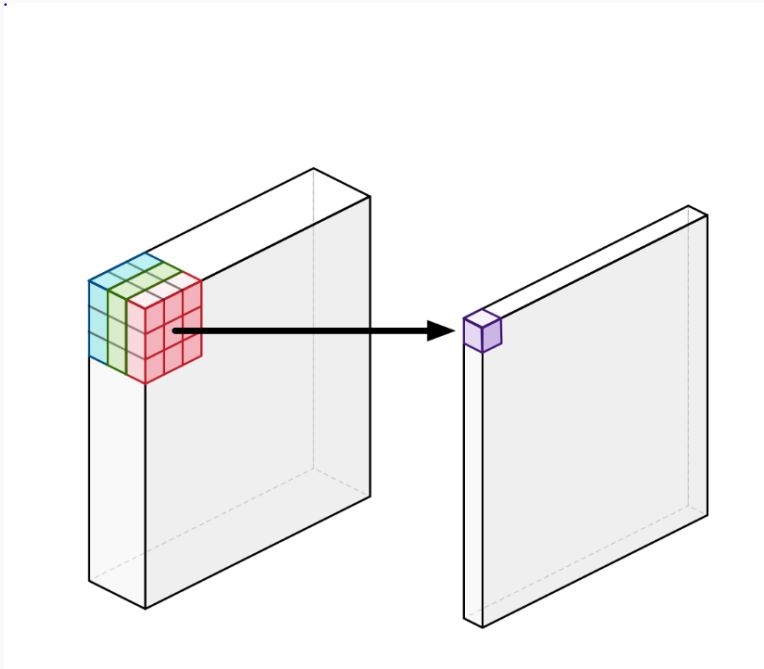


Figure 11: Normal convolution.[3]

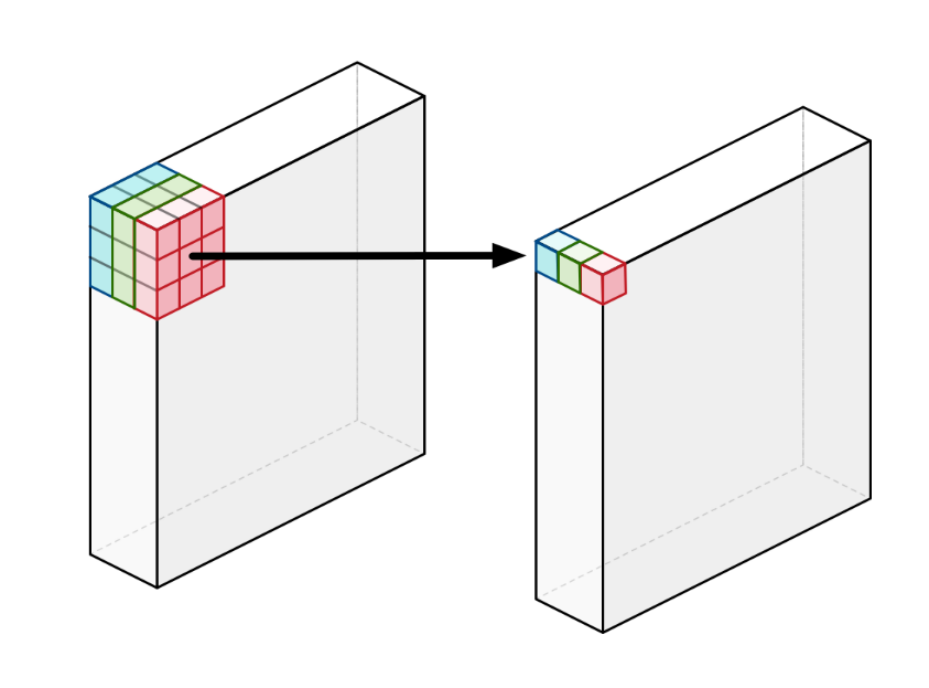


Figure 12: Depth-wise convolution.[2]

[3] https://machinethink.net/blog/googles-mobile-net-architecture-on-iphone/

ResNet164

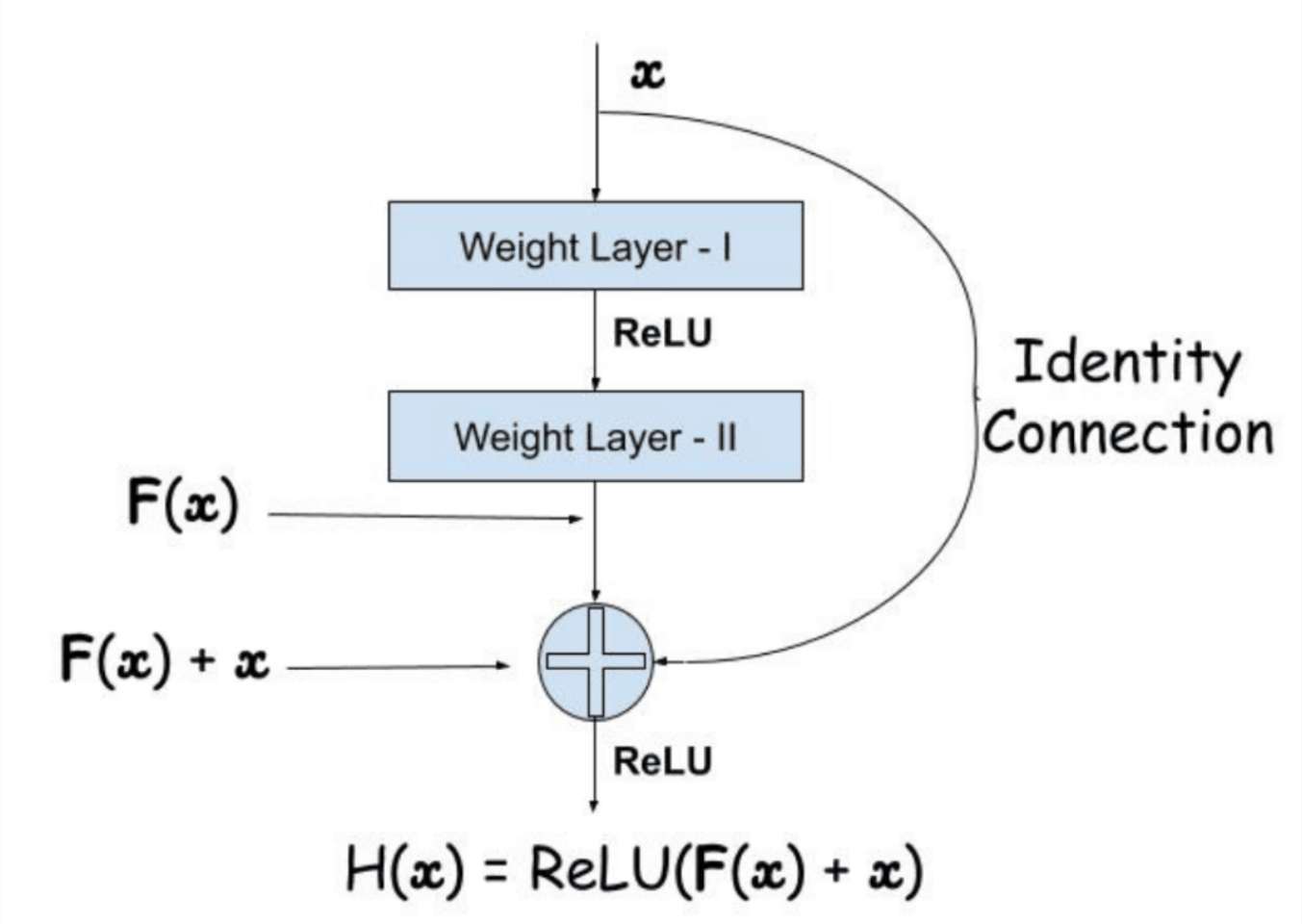


Figure 13: Skip connections in ResNet models.[4]

[4] https://cv-tricks.com/keras/understand-implement-resnets/

Performance evaluation

Performance Criteria

- **Accuracy** : Proportion of cells that were correctly identified.
- **Non-zero Accuracy** : Proportion of non-empty cells that were correctly identified.
- **Confusion matrix** : For the overall accuracy.

Ghouzam

Accuracy : 0.976

Non-zero accuracy : 0.957

	0	1	2	3	4	5	6	7	8	9
0	13923	146	18	9	7	49	8	35	5	16
1	40	1773	10	3	36	0	0	87	1	10
2	6	1	1929	2	4	1	0	2	11	4
3	6	1	1	1914	3	3	2	1	1	0
4	21	2	2	0	1895	3	3	3	6	27
5	7	1	0	2	3	1882	0	1	1	2
6	10	3	2	2	5	5	1892	0	2	1
7	6	1	9	3	26	1	1	1857	2	2
8	4	3	1	1	3	2	1	3	1945	1
9	12	0	4	1	6	12	0	3	3	1826

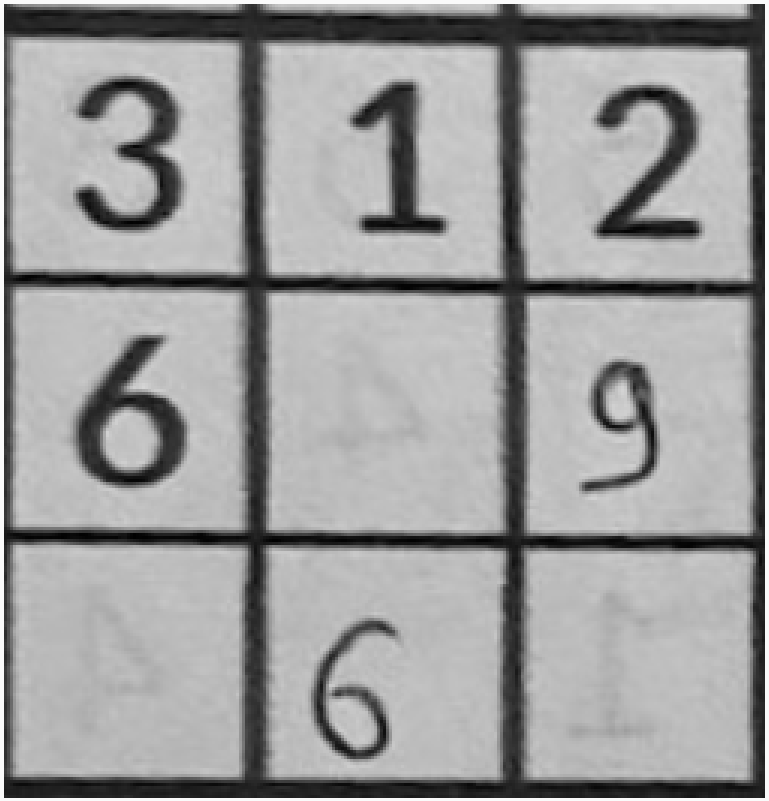
Ghouzam

Empty cells are not always detected as such.

	0	1	2	3	4	5	6	7	8	9
0	13923	146	18	9	7	49	8	35	5	16
1	40	1773	10	3	36	0	0	87	1	10
2	6	1	1929	2	4	1	0	2	11	4
3	6	1	1	1914	3	3	2	1	1	0
4	21	2	2	0	1895	3	3	3	6	27
5	7	1	0	2	3	1882	0	1	1	2
6	10	3	2	2	5	5	1892	0	2	1
7	6	1	9	3	26	1	1	1857	2	2
8	4	3	1	1	3	2	1	3	1945	1
9	12	0	4	1	6	12	0	3	3	1826

Ghouzam

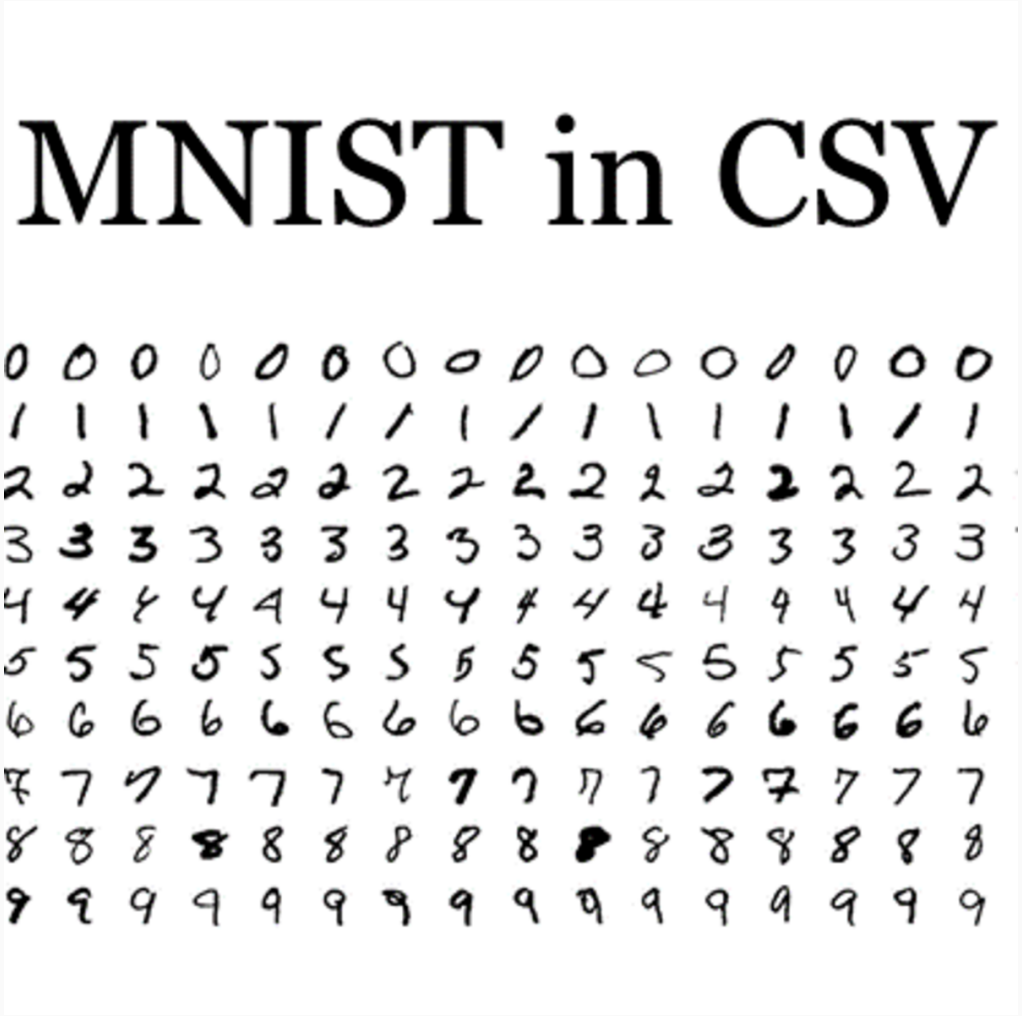
Empty cells are not always detected as such.



Confusion between 1 and 7.

	0	1	2	3	4	5	6	7	8	9
0	13923	146	18	9	7	49	8	35	5	16
1	40	1773	10	3	36	0	0	87	1	10
2	6	1	1929	2	4	1	0	2	11	4
3	6	1	1	1914	3	3	2	1	1	0
4	21	2	2	0	1895	3	3	3	6	27
5	7	1	0	2	3	1882	0	1	1	2
6	10	3	2	2	5	5	1892	0	2	1
7	6	1	9	3	26	1	1	1857	2	2
8	4	3	1	1	3	2	1	3	1945	1
9	12	0	4	1	6	12	0	3	3	1826

Confusion between 1 and 7.



[4] https://www.kaggle.com/oddrationale/mnist-in-csv

Handwritten digits (1877 digits) : accuracy of 0.882.

	0	1	2	3	4	5	6	7	8	9
1	24	151	6	1	25	0	0	27	1	9
2	3	0	194	1	5	0	0	0	6	4
3	2	1	0	211	3	2	0	0	0	0
4	11	0	0	0	204	0	0	2	1	19
5	1	0	0	1	0	152	0	0	0	1
6	2	3	0	0	1	1	175	0	1	0
7	5	1	6	2	20	0	0	168	0	1
8	0	0	0	0	0	1	0	1	208	1
9	4	0	1	2	3	11	0	1	1	189

1	2	3	4	5	6	7	8	9
0.619	0.911	0.963	0.861	0.981	0.956	0.828	0.986	0.892

Ghouzam

Typewritten digits (15497 digits) : accuracy of 0.984.

	0	1	2	3	4	5	6	7	8	9
1	19	1649	13	4	1	0	0	29	0	1
2	2	1	1738	1	0	1	0	2	1	1
3	5	0	1	1703	0	0	2	1	1	0
4	15	1	7	0	1678	3	3	1	3	14
5	6	1	0	0	3	1730	0	1	1	2
6	5	1	4	2	5	10	1711	0	1	0
7	6	2	21	8	1	0	0	1664	2	1
8	4	3	1	2	3	1	0	2	1737	0
9	7	0	4	1	2	7	0	2	1	1631

1	2	3	4	5	6	7	8	9
0.961	0.995	0.994	0.973	0.992	0.984	0.976	0.991	0.985

Pre-trained models

	0	1	2	3	4	5	6	7	8	9
0	14139	4	8	2	14	39	0	2	8	0
1	1142	1	623	0	192	1	0	0	1	0
2	257	0	1691	0	6	2	0	0	4	0
3	1109	0	2	590	1	17	0	0	213	0
4	301	0	0	16	57	3	0	0	1	0
5	19	0	0	0	5	1871	0	0	4	0
6	1536	0	2	1	5	9	162	0	207	0
7	1118	0	5	0	88	2	0	694	1	0
8	1354	0	2	1	4	2	0	0	601	0
9	438	0	1	0	152	20	0	0	802	454

Figure 14: Confusion matrix for the ResNet164 model

- Confusion between all numbers and empty cells
- **Accuracy :** 0.692 (0.327 for VGG16, and 0.424 for MobileNet)
- **Non-zero Accuracy :** 0.442 (0.323 for VGG16, and 0.420 for MobileNet)

Pre-Trained models : Example

Pre-trained models

Main issues :

- No empty cells in mnist
- Shape of the ones in mnist
- Augmented training set
- probability threshold

Demonstration