Case Study - Image Recognition

BTech/III Year CSE/V Semester

15CSE303/Theory of Computation

Roll No.	Name
CB.EN.U4CSE18331	KARTHIKEYAN. RV
CB.EN.U4CSE18332	B. KIRTHI SAGAR
CB.EN.U4CSE18362	TANMAAY KANKARIA
CB.EN.U4CSE18365	ASHOK REDDY V

Amrita School of Engineering, Coimbatore
Department of Computer Science and Engineering

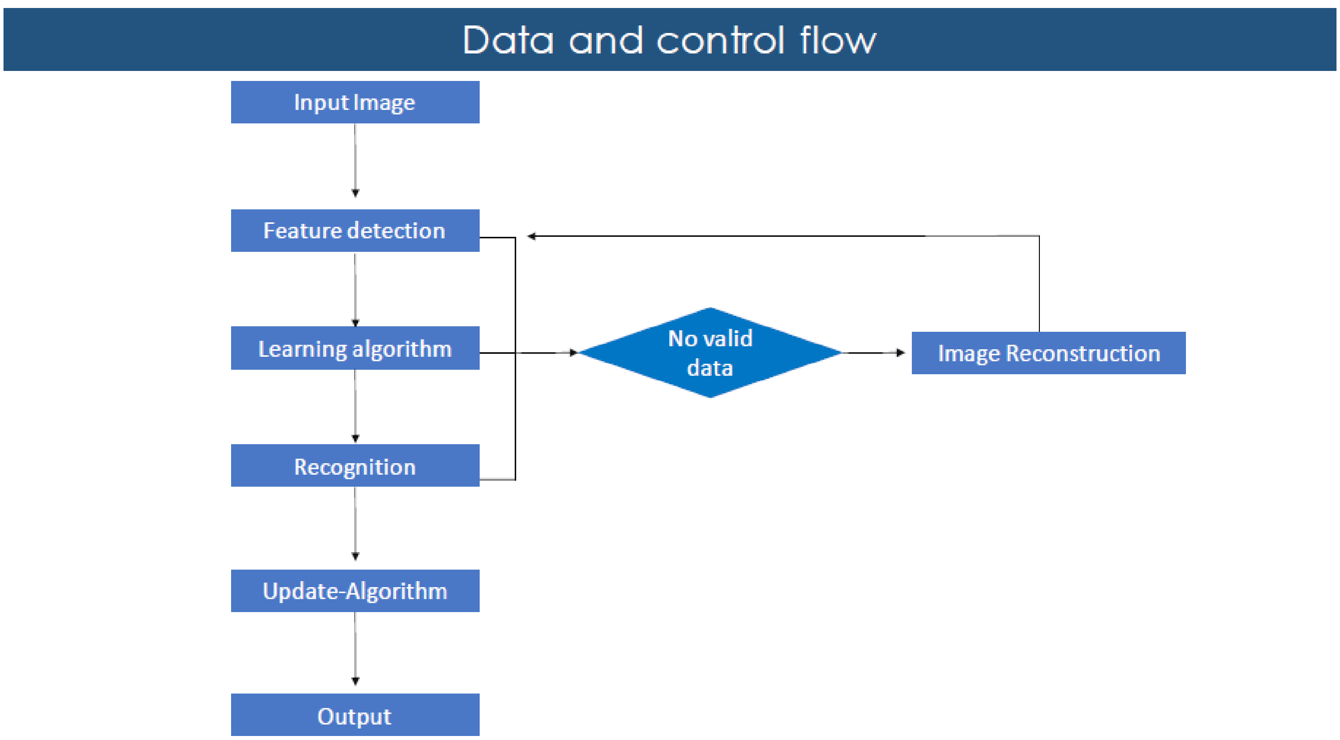
2020 -2021 Odd Semester

Problem Definition

To briefly explain image recognition using traditional computer vision techniques and automata concepts.

An image recognition algorithm (or an image classifier) takes an image as input and outputs the class to which the image belongs. We train the algorithm to learn the differences between different classes. Needless to say, this algorithm can only understand objects/classes it has learned.

Block Diagram



Approach to Image Recognition

Step 1: The input image is provided into the system. The input image is pre-processed to normalize contrast and brightness effects.

Step 2: First we simplify the image by extracting the important information contained in the image and leave out the rest. Next, proceed with algorithm for feature detection on the given input image.

Step 3: The features detected of the image is then processed by the learning algorithm.

Step 4: This data is sent to recognition module which on successful recognition, updates the learning algorithm and prints the class (category) of the image.

Exception handling: At each step there is a possibility of success (1) and failure (0). So, in the case of failure, the data is sent to the image reconstruction module where the image is reconstructed and returned.

Working of the Learning Algorithm

Before a classification algorithm can be implemented on the image, we need to train the algorithm by showing thousands of examples of the image class we want to detect. Then, if the image processed is recognized, the algorithm will be updated. If the image is not recognized it will be sent for the reconstruction process. This process occurs for multiple iterations until the algorithm recognizes the image class.

Language Description

A language is considered as a finite set of strings over some finite set of alphabets. Computer languages are considered as finite sets, and mathematically set operations can be performed on them. Finite languages can be described by means of regular expressions.

Any finite sequence of alphabets is called a string. Length of the string is the total number of occurrence of alphabets. A string having no alphabets, i.e. a string of zero length is known as an empty string and is denoted by ε (epsilon).

Operations on Languages:

The various operations on languages are:

- Union of two languages L and M is written as
  L U M = {s | s is in L or s is in M}

- Concatenation of two languages L and M is written as
  LM = {st | s is in L and t is in M}

- The Kleene Closure of a language L is written as
  L* = Zero or more occurrence of language L

Notations on Languages:

If r and s are regular expressions denoting the languages L(r) and L(s), then

- Union: (r) | (s) is a regular expression denoting L(r) U L(s)
- Concatenation: (r)(s) is a regular expression denoting L(r)L(s)
- Kleene closure: (r)* is a regular expression denoting (L(r))*
- (r) is a regular expression denoting L(r)

For our Image Classifier, the language can be derived as follows:

Input symbols: The automata has two main input symbols, 1 (True) and 0 (False)

The set of all possible strings accepted by the automata are,

Minimal string: 11111

(0)*11111

11111

1(01)1111

11(01)1111

11(01)1(01)1111

111(01)1(01)1(01)(01)(01)1111

11(01)*1111

0*1(01)*1111

0*(11111 +1(01)*1111+ 11(01)*1111+ 111(01)*1111) and so on.

Hence, using the operations and notations of a language, the generalized language accepted by the automata can be concluded as,

$$\mathbf{L(M) = \{ 0^m (1^x (01+1)^y 1^z \mid 1<=x<4, y>=0, z>=4, m>=0\}}$$

Grammar

Considering the set of possible strings that are accepted by the automata, the following production rules have been derived.

Minimal string: 11111

G: (V, T, S, P)

Production Rules – P:

{IP --> 0 IP | 1 FD

FD --> 0 IR | 1 LA

LA --> 0 IR | 1 RE

RE --> 0 IR | 1 UA

IR -->1 FD

UA -->1 OP

OP -->1 OP | ε}

Non-Terminal Symbols V: {IP, FD, LA, RE, IR, UA, OP}

Terminal Symbols T: {0, 1}

Start Symbol S: IP

Production rules Explanation

Symbol IP is start symbol and it has 0 IP because the strings can begin with infinite number of 0s (waiting for input image). It also has 1 FD in order to satisfy the minimum string condition (11111) which is obtained by concurrent substitutions of further symbols.

Symbols FD, LA, RE has 0 (respective symbol) because the automata has to reach the step of image reconstruction in case of failure.

Symbol IR has 1 FD because, every time the after the reconstruction process the automata returns to FD state.

Symbol OP has 1 OP because the automata can remain at final state (OP) for any further number of successes.

The grammar obtained is right linear and hence a **Regular Grammar**.

Finite Automata

Let us first define the states for the process based on the derived grammar and language description.

States:

IP - Input Image

FD - Feature Detection

LA - Learning Algorithm

RE - Recognition

UA - Update Algorithm

IR - Image Reconstruction

OP - Output

Start State:

IP - Input Image

Final State:

OP – Output

Transitions:

F(IP,1) --> (FD)

F(IP,0) --> (IP)

F(FD,1) --> (LA)

F(FD,0) --> (IR)

F(LA,1) --> (R)

F(LA,0) --> (IR)

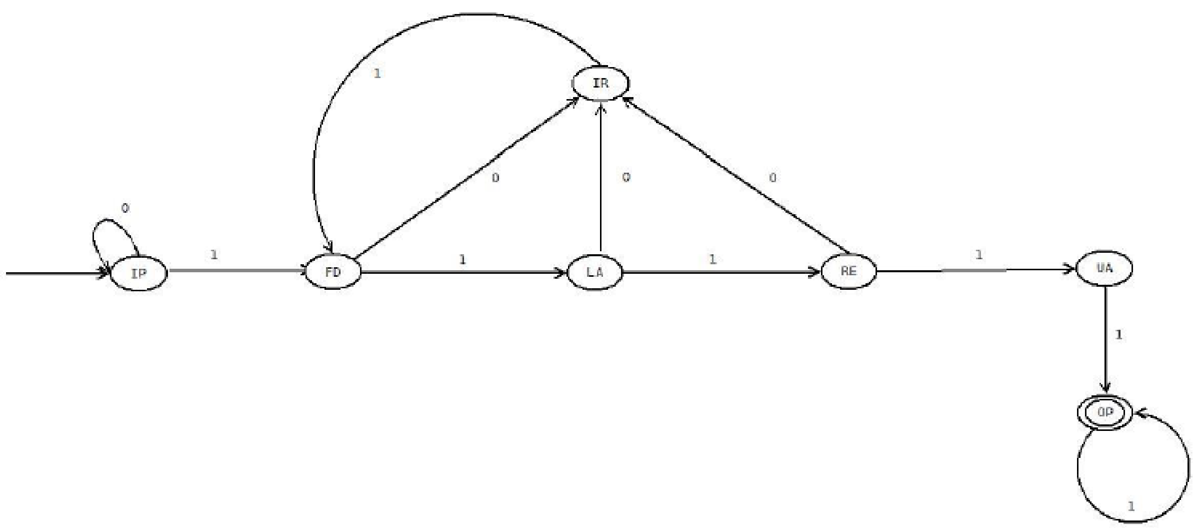
F(R,1) --> (UA)

F(R,0) --> (IR)

F(UA,1) --> (OP)

F(OP,1) --> (OP)

Finite Automata



Pushdown Automata

A PDA is deterministic if its transition function satisfies **both** of the following properties

For all $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, and $X \in \Gamma$,
the set $\delta(q, a, X)$ has **at most** one element
there is only one move for any input and stack combination

For all $q \in Q$ and $X \in \Gamma$,
if $\delta(q, \varepsilon, X) \neq \{\}$, then $\delta(q, a, X) = \{\} \ \forall a \in \Sigma$
an ε-transition has no input-consuming alternatives, i.e., there cannot exist another move with stack = *X* from the *same state q* .

As this project's automata obeys the rules of the DPDA

$L(M) = \{ 0^m (1^x (01+1)^y 1^z \mid 1<=x<4, y>=0, z>=4, m>=0\}$

M = $(Q, \Sigma, \Gamma, \delta, IP, Z, F)$

Q = {IP,FD,LA,RE,IR,UA,OP}

Σ = {0,1}

F = {OP}

Γ = {0,1,Z}

Z is the initial stack symbol

Main conditions required to construct the proper DPDA

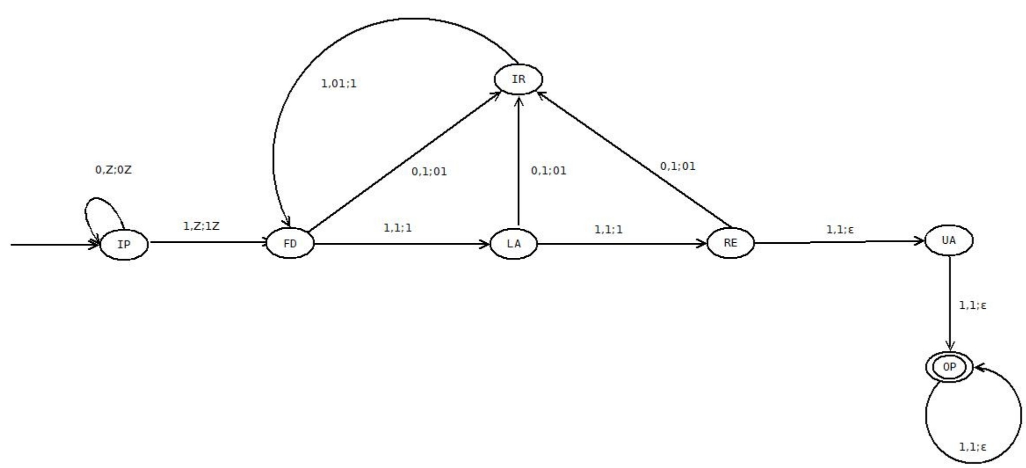
Condition to be in state IP is m>=0
Condition to be in state FD is x=1
Condition to be in state LA is x=2
Condition to be in state RE is x=3

δ→*{*

δ(IP,0,Z)→(IP,0Z)
δ(IP,1,Z)→(FD,1Z)
δ(FD,0,1)→(IR,01)
δ(FD,1,1)→(LA,1)
δ(LA,0,1)→(IR,01)
δ(LA,1,1)→(RE,1)
δ(RE,0,1)→(IR,01)
δ(RE,1,1)→(UA,ε)
δ(UA,1,Z)→(OP,ε)
δ(OP,1,Z)→(OP,ε)

}

DPDA construction



Conclusion

The overall process of image recognition is understood.

- Construction of language description for the automata is successfully done.
- Construction of grammarfor the automata is successfully done.
- Construction of the finite automata for the entire process.
- Construction of PDA for the automata is successfully implemented.

References

Geeksforgeeks :
https://www.geeksforgeeks.org/

learnopencv:
https://www.learnopencv.com/image-recognition-and-object-detection-part1/