Control Chart Patterns Recognition using Convolutional Neural Network

Steps for the recognition of patterns of control charts using Improved 1-D-CNN:

1. Firstly take the input data saved in a csv file or excel sheet which carries the production process data varies over time.

Date	Measurement	Mean	UCL	LCL
7/1/2020	12	13.9	19.31051608	8.489483924
7/2/2020	13	13.9	19.31051608	8.489483924
7/3/2020	15	13.9	19.31051608	8.489483924
7/4/2020	16	13.9	19.31051608	8.489483924
7/5/2020	13	13.9	19.31051608	8.489483924
7/6/2020	12	13.9	19.31051608	8.489483924
7/7/2020	15	13.9	19.31051608	8.489483924
7/8/2020	18	13.9	19.31051608	8.489483924
7/9/2020	14	13.9	19.31051608	8.489483924
7/10/2020	12	13.9	19.31051608	8.489483924
7/11/2020	13	13.9	19.31051608	8.489483924
7/12/2020	11	13.9	19.31051608	8.489483924
7/13/2020	13	13.9	19.31051608	8.489483924
7/14/2020	14	13.9	19.31051608	8.489483924
7/15/2020	16	13.9	19.31051608	8.489483924
7/16/2020	15	13.9	19.31051608	8.489483924
7/17/2020	16	13.9	19.31051608	8.489483924
7/18/2020	13	13.9	19.31051608	8.489483924
7/19/2020	12	13.9	19.31051608	8.489483924
7/20/2020	15	13.9	19.31051608	8.489483924

2. Now we find the different parameters like mean, standard deviation , distribution from the above datasets present in the csv file.

3. Now we apply a Data simulation method like monte carlo simulation method to generate different patterns of control charts by setting different parameters according to required patterns.

Table 1. Parameters and formulas of data simulation

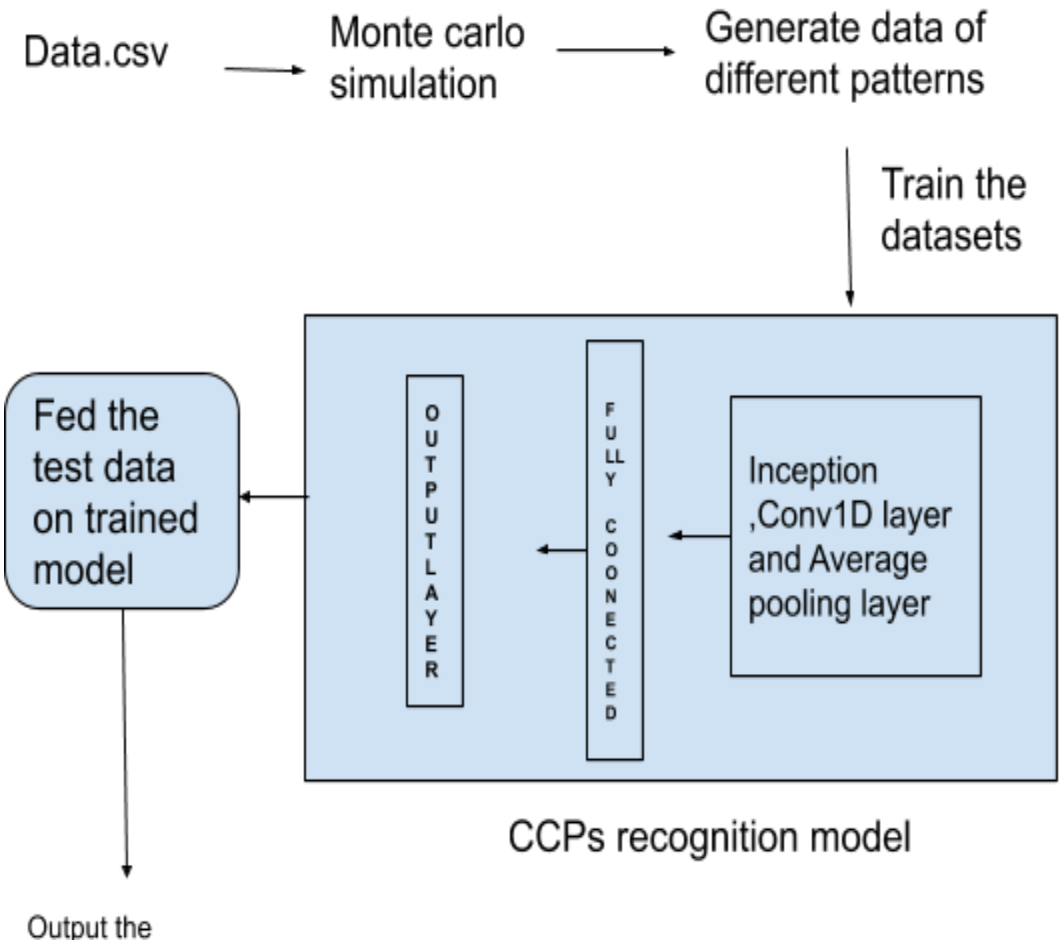
Class	Description	Equations	Remarks
0	Normal, NOR	$y_t = \mu + r(t) \times \delta$	$\mu = 0, \sigma = 1$ $r(t) \sim N(0,1)$ $\delta = 1\sigma$ $\delta' \in (0.2\sigma, 0.4\sigma)$ $d \in (1\sigma, 3\sigma)$ $a \in (1.5\sigma, 2.5\sigma)$ $T = 16$ $g \in (0.05\sigma, 0.25\sigma)$ $P \in (10, 20)$ $s \in (1\sigma, 3\sigma)$ $t = 1,2, \ldots, L$
1	Cyclic, CYC	$y_t = \mu + r(t) \times \delta + a \sin(2\pi t/T)$	
2	Systematic, SYS	$y_t = \mu + r(t) \times \delta + d \times (-1)^t$	
3	Stratification, STR	$y_t = \mu + r(t) \times \delta'$	
4	Upward Trend, UT	$y_t = \mu + r(t) \times \delta + t \times g$	
5	Downward Trend, DT	$y_t = \mu + r(t) \times \delta - t \times g$	
6	Upward Shift, US	$y_t = \mu + r(t) \times \delta + k \times s$ $k = 1 \; if \; t \geq P, else \; k = 0$	
7	Downward Shift, DS	$y_t = \mu + r(t) \times \delta - k \times s$ $k = 1 \; if \; t \geq P, else \; k = 0$	

Now by running the simulation we can generate data of various patterns which can be used further for training the CNN model

4. Now we fed the generated CCPs datasets to train the model (Improved 1-D-CNN) which extracts the feature from different patterns and helps to classify the 8 different Control chart patterns.

5. In the last step we fed the test data to the trained model to recognise the pattern of the control chart.

Flow chart


Data.csv
Monte carlo simulation
Generate data of different patterns
Train the datasets
Inception ,Conv1D layer and Average pooling layer
FULLY COONECTED
OUTPUTLAYER
CCPs recognition model
Fed the test data on trained model
Output the