

Specifications

11. November 2019

1 Preface

2 Goal

The goal is a software which performs image classification and is able to switch between deploy platforms and working modes. It also should have a GUI to control the software and to show the results.

3 Product use

Image classification

4 Acceptance criteria

4.1 Must

MAC010 : Image classification

MAC020 : Running NN on heterogenous platforms, CPU and FPGA

MAC030 : Different operating modes

MAC040 : GUI for interacting with software

MAC050 : Performance/power prediction

4.2 Can

KAC060 : Training a nn for classification
KAC070 : Illustration of a topology of the nn
KAC080 : Object detection
KAC090 : Choosing between different models
KAC100 : Creating new models
KAC110 : Voting of multiple nn
KAC120 : Using video for classification
KAC130 : Using camera for classification input
KAC140 : Running NN on GPU

5 Functional Requirements Must

- MFR010]: Use neural network for image classification

MFR011 : Deploy pre-trained nn with the corresponding layers
MFR012 : Reading and parsing nn config/weight file
MFR020 : Have high performance mode
MFR021 : Have low power consumption mode
MFR022 : Have high energy efficiency mode
MFR023 : Calculator for power consumption
MFR024 : Calculator for performance
MFR025 : Dispatching the calculation process defined from the mode
MFR030 : Support CPU for calculation
MFR031 : Support FPGA for calculation
MFR040 : Communication between Host-PC and platform
MFR041 : Send image for classification
MFR042 : Receive result
MFR050 : GUI
MFR060 : Showing results
MFR070 : Choosing image for classification in format .jpg, .png, .bmp
MFR080 : Choosing platform/hardware
MFR081 : Look which devices are available
MFR082 : Testing communication
MFR090 : Choosing mode
MFR091 : Show how much energy and power on which platform one needs/has

6 Functional Requirements Can

KFR100 : Choosing between different models for example AlexNet, GoogleNet
KFR110 : Train nn for classification of imageset (with transfer learning)

KFR111 : Saving new trained nn (config an weights)
KFR112 : Choosing/Reading data set
KFR032 : Support GPU for calculation
KFR113 : Backpropagation
KFR114 : Choosing parameters like learning rate
KFR120 : Illustrating nn topology
KFR130 : Object detection algorithm
KFR131 : Showing detected object
KFR132 : Choosing between detection and classification mode
KFR140 : Creating new topology
KFR150 : Choosing between training and inference mode
KFR160 : Choosing video in format .avi
KFR161 : Apply classification for a certain amount of frames
KFR170 : Connect with camera
KFR171 : Receive video stream from camera
KFR180 : Detecting object
KFR181 : Drawing bounding box

7 Productdata

PD010 : Images for classification
PD020 : Labeled image set for training
PD030 : Config/weight file of pretrained model

8 Demarcation

D010 : No real time / no performance optimization
D020 : No mobile support
D030 : No neural network size optimization
D040 : No low-level (Assembler) optimization

9 Non-functional requirements

NF10

10 System models

10.1 Scenarios

10.2 Usecases

10.2.1 Seminarorganisation