# Models and diagrams

As described in the Methodology section, we will employ a selections of UML and SysML to model the systems structure and behaviour, in that order. These models will help when deciding how the system will be build, as well as where the different components should be implemented.

## Clarifying the problem

First step in modelling the system is describing the system, which in part was done in earlier sections, with the description of the goal, and the use-case.

Based on that a domain diagram is constructed which shows what is neassesary to realize the use-case. Below in figure XX is the domain diagram

This shows the different domains in the system and how they connect to each other. In this system, a SD-card with images to classify is loaded into the Zybo board. Where a button on the system activates sending an image for pre-processing, before being sent to the neural network for classification which in turn is displayed on the led’s on the Zybo board.

## Modelling the Systems

When modelling the system, a decision was made to start with a behavioural model, in the form of activity diagrams to define how the system was to achieve the solution to the problem after that the activity diagram was used as the base to construct block diagrams.

### System Behaviour

To describe the system behaviour an activity diagram has been made to show the different actions the system should perform as well as the inputs and outputs of the system. In figure XX the activity diagram describing the top-level of the system can be seen.



The top-level activity diagram shows how the program flow is. With inputs and outputs, as well as other activities performed when the system runs, in the form of load from SD, Pre-process image and Neural network classification prediction. Now that the overall system behaviour has been described in the activity diagram, a dive into the other activities contained in figure XX. Load from SD is much like a standard IO file read class, and as such will not be described. So, the Next one is Pre-processing shown in figure XX.



In the pre-processing a blur, and down sample, combo, is used several times to get the desired output size, for use in the neural network. A flatten function is also used to get a 16 by 1 output. For ease of use with the network. which the activity diagram of is shown below in figure XX



The neural network uses a couple of different functions, that might need some explaing first is the first layer of the network, it uses something called neurons which essentially multiplies the input with a set of weights to get a single value, this is done with 32 different weight vectors to get 32 outputs. Then an activation function is run, in this case one that is called RELU, which just sets all negative values to zero. A second network layer, this time with 10 different weights, to get the 10 possible results. And a max activation function which returns the index of the highest value of the input to get our classification prediction.

### Structure.

From these activity diagrams a block diagram of the system has been constructed shown in figure XX

Here the system is broken into the main components. SD-card for containing the images, GPIO to interact with the user with buttons and LED’s. Pre-processing on the image, before finally putting it through the neural network for a classification of the image. Each part contains more sub blocks, but this is just to get an idea of what the main functionality of the system is comprised of. There is no internal block diagram for this block diagram, since the interfaces between them is somewhat described in the activity diagram. Instead a more in-depth block diagram that also describes whether the blocks should be in software, or hardware, this is shown in figure XX

In the SW/HW BDD we can se the Zynq processor where software will run, and the hardware blocks will be called from, and in the Programmable logic, the GPIO will be made, mostly to provide a interface for the processor to the hardware of the LED’s and Buttons, these are two separate hardware blocks, but has been combined under the GPIO block for clarity. The neural network was chosen to be implemented in hardware since it as great candidate for parallelizing the processing.

In figure XX some interfaces have been added to the blocks as well in the IBD showing what data is expected to flow between the different blocks in slightly more detail than the activity diagram.



In the figure the Zynq processor is shown as containing all the software, the SD-Card reader, and the pre-processing as well as a main program where they are called from, and the LED, Buttons and Neural network is shown outside of the Zynq as to indicate the fact that they are to be implemented in hardware. Instead of using inOut ports, both in and out port is shown with the data they transfer, this is because all the functions take at most 1 input and 1 output. As such the flow specifications has been foregone in favour of just showing it with in and out ports.

Software architecture.

Since the system is very sequential, and just reacts to the user pressing a button, the event driven architecture has been used.