# Autonomous Intelligence Systems
## Labortory

Exercise Nr. 2

# Implementing a CNN in Tensorflow

Author : Basharat Basharat
Matriculation Nr. 4053110
Email : Basharat.basharat@uranus.uni-freiburg.de

## Introduction:

Implementing of a Convolution Neural Network using the Tensorflow package. Convolution neural network consists of two convolution layers with (16 filters with size [3x3] and a stride of 1, and each convolution layer followed by a ReLU as an activation function and a max pooling layer with (pool size 2). Fully connected (Dense layer) consists of 128 units (neurons) and a final softmax layer for classification. The model is trained by optimizing the cross-entropy loss with stochastic gradient descent.

## Learning Rates :

By using different (learning rates) in the stochastic gradient descent, the following training accuracy and training loss with training set and evaluation set.

**Results :**
**Training Set & Eval Set:**

| Learning Rate | 0.1 | 0.01 | 0.001 | 0.0001 |
|---|---|---|---|---|
| Training loss | 0.02879 | 0.0165 | 0.0238 | 0.0113 |
| Evaluation loss | 0.0376 | 0.0289 | 0.0287 | 0.0287 |
| Training accuracy | 0.990 | 0.990 | 0.9901 | 0.9901 |

The above learning graphs are in the json files, which are uploaded in my github. Unfortunately I was not able to plot the graphs due to less amount of time for this particular exercise.

I was also not able to implement the random_search.py as to lack of time. If I am provided more time I can try to solve the second part of the exercise.