



# VISVESVARAYA NATIONAL INSTITUTE OF TECHNOLOGY (VNIT), NAGPUR

---

## Machine Learning with Python (ECL 443)

### Mini Project

---

*Submitted by :*

Ajinkya Deshpande (BT19ECE024)

Ajay Jondhale (BT19ECE046)

Pranav Kathar (BT19ECE058)

Shubham Patil (BT19ECE084)

Semester 7

*Submitted to :*

Dr. Saugata Sinha

(Course Coordinator)

Department of Electronics and Communication Engineering,  
VNIT Nagpur

# Contents

- 1    Edge Detector using Artificial Neural Networks . . . . . 2
  - 1.1    Edge Detector . . . . . 2
  - 1.2    Conclusion . . . . . 4
  - 1.3    Appendices . . . . . 4

## Edge Detector using Artificial Neural Networks

### 1.1 Edge Detector:

**Edge detection** is a technique of image processing used to identify points in a digital image with discontinuities, simply to say, sharp changes in the image brightness. These points where the image brightness varies sharply are called the edges (or boundaries) of the image.

In this Mini-Project we aim to achieve to the same results using machine learning techniques specially using **Artificial Neural Networks (ANN)**. An artificial neural network is build based on the idea to mimic how the human brain works.

The dataset we have used is **Furniture Images Dataset**. It consists of 9346 images of various household furnitures. Images are scraped from Sri Lanka's most famous online marketplace: [Ikman.lk](https://www.ikman.lk). For making the edge detector Illumination, Scale , Light , Rotation Invariant we used various transforms available in PyTorch.

- Input Image size 64 x 64. 9277 images of furnitures downloaded from kaggle.
- Output image size 64x64. Edge mapped image with laplacian filter - created manually.
- Number of FC layers 8 with ReLU activation and BatchNormalization
- trainable hyperparametrs 8994656
- Gradient descent with Adam optimizer and Cosine annealing warm restart as learning rate scheduler
- trained for 50 epochs.
- Applied randomRotation, RandomFlipping, RandomCrop, CenterCrop, ColorJitter, padding as image transforms that would make model invariant of scale, size, brightness etc.

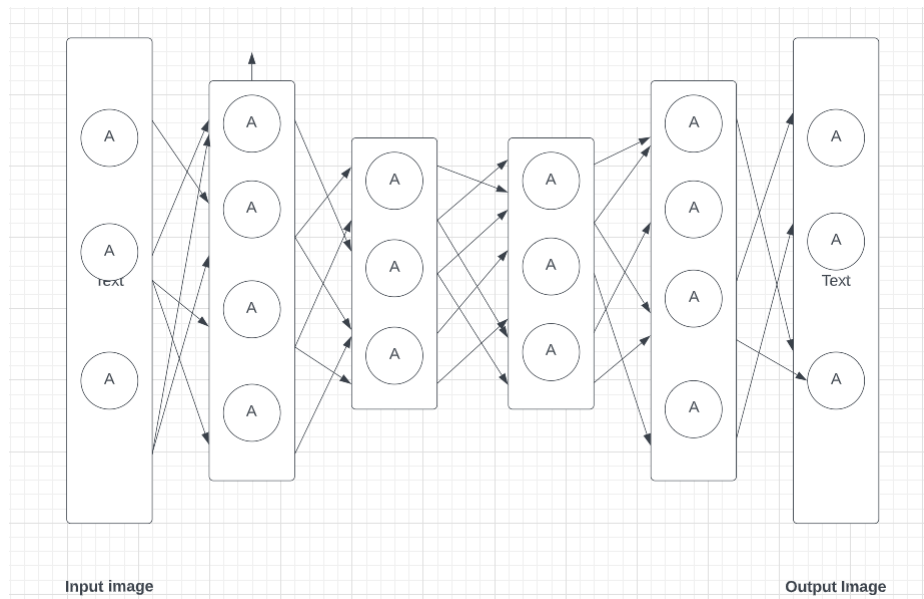


Figure 1: Model Artificial Neural Network

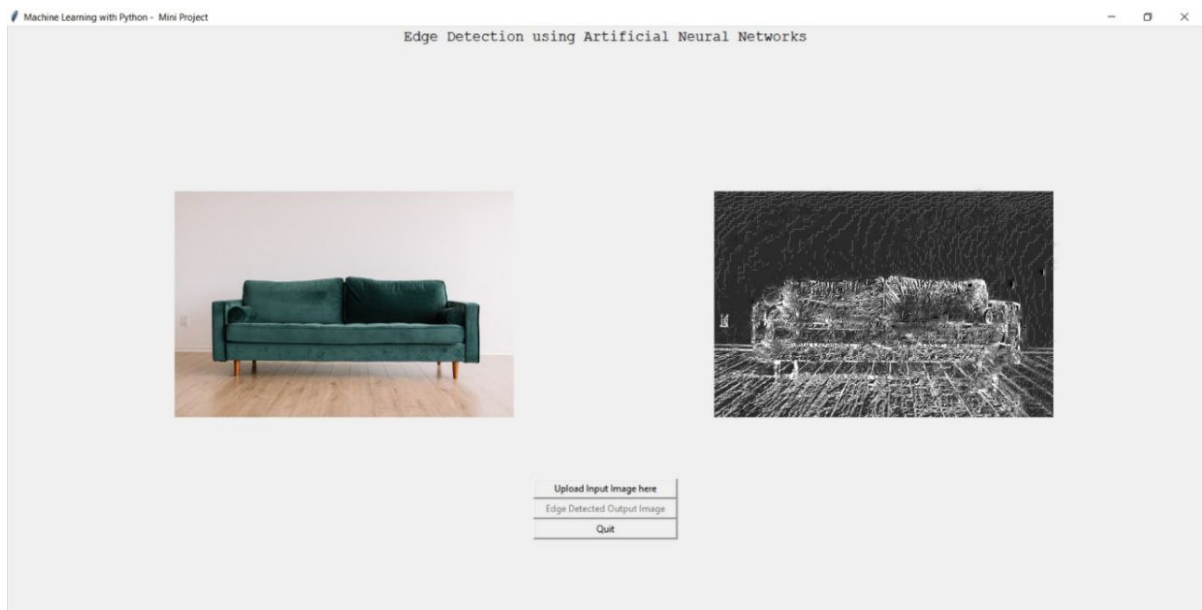


Figure 2: GUI

## 1.2 Conclusion:

- We successfully built an edge detector using Artificial Neural Networks which is Illumination, Scale, Light, Rotation Invariant and we compared our output with pre-defined edge detectors such as **Canny, Robert, Sobel, Prewitt**, etc

## 1.3 Appendices:

1. **Model Training Code** <https://colab.research.google.com/drive/1Xx104lr09PodPZOUfy5SnYHQc-BsyTzx?usp=sharing>
2. **Link for GUI** <https://colab.research.google.com/drive/1uaLnIm3C-DneuhKKEoz85fgJSJtTVVZH?usp=sharing>