

# Training of convolutional neural networks with simulated image data for logistics object detection

supervisor: Hagen Borstell (ILM)



# overview

- Team
- Motivation
- Working methodology
- Implementation
  - Workflow
  - Tools overview
  - Rendering
  - Training
  - Initial Results
- Futurework
- Estimated time

# Team

Jagriti Labh

Karri Aravind Reddy



OTTO VON GUERICKE  
UNIVERSITÄT  
MAGDEBURG

# Motivation



Interesting Subject



Opportunities towards



Learning new Implementations



OTTO VON GUERICKE  
UNIVERSITÄT  
MAGDEBURG



FAKULTÄT FÜR  
INFORMATIK

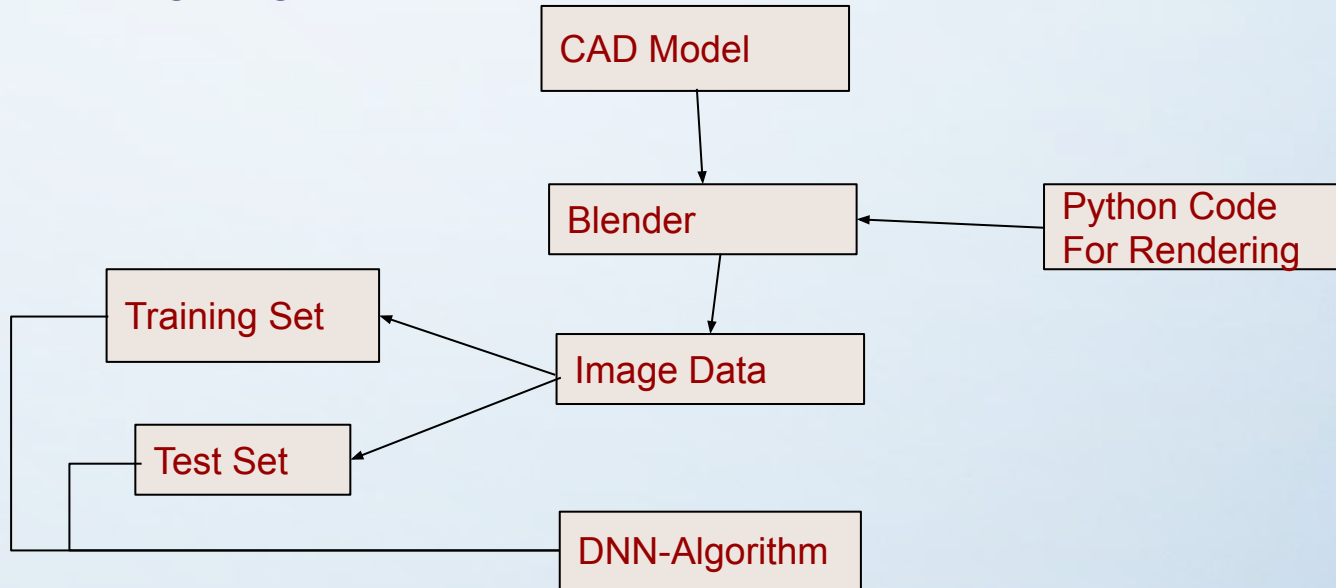
# Working Methodology

Scrum based Agile methodology



# Implementation

## workflow

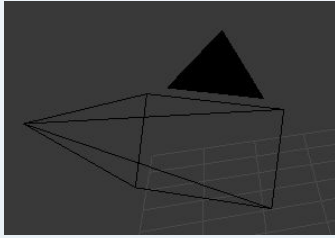


# Tools overview

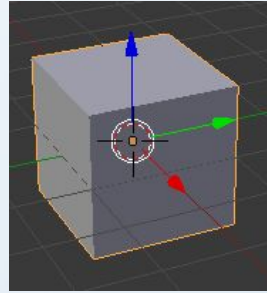


- Opensource
- Used for creating animated films, visual effects ect
- Written on python

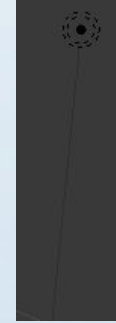
## Elements in blender



Camara



object



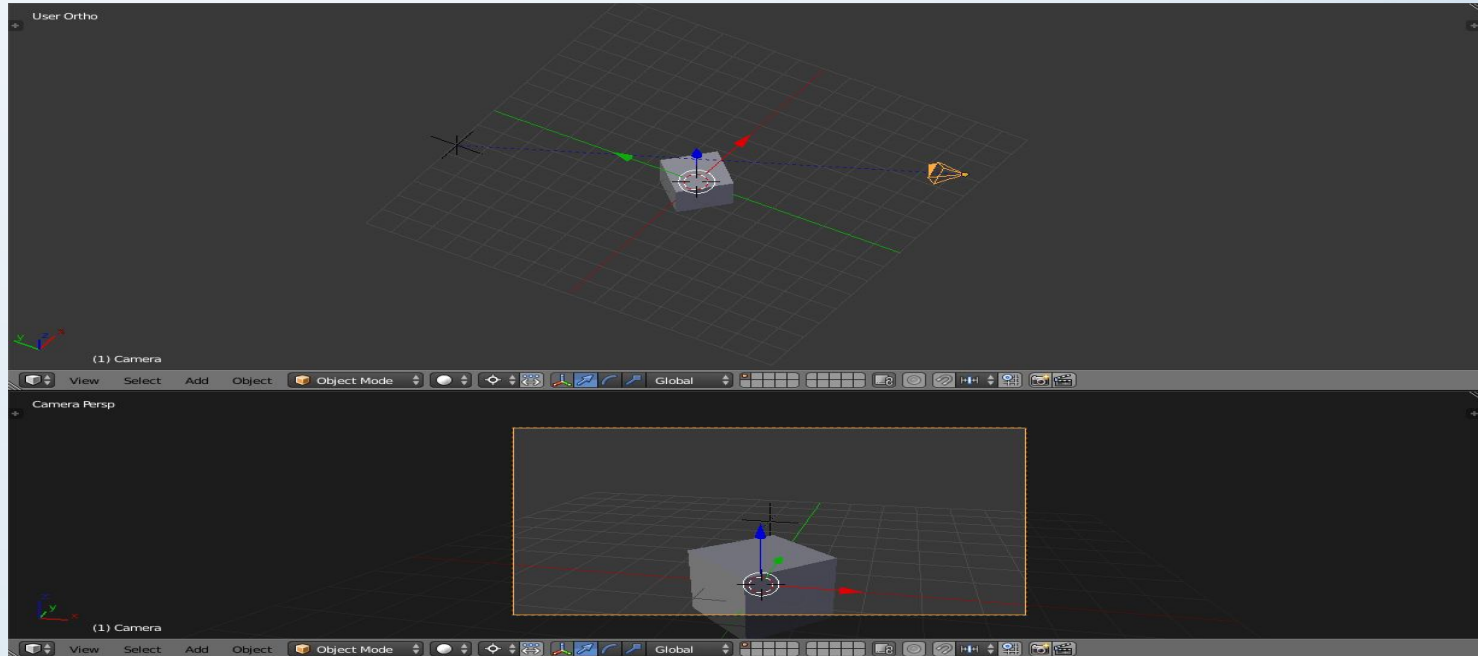
Light

- Access each element as object  
ex: `bpy.camara`, `bpy.object`



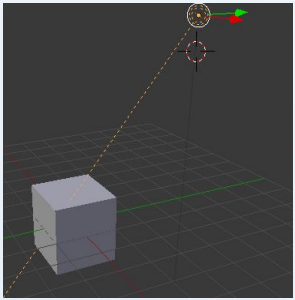
# Rendering

Rotate camera and render images with a particular interval

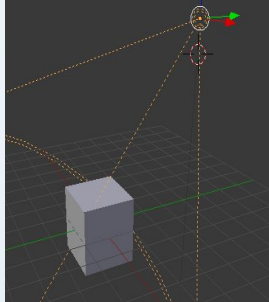


# Rendering

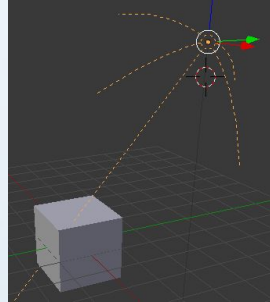
## Variation in light



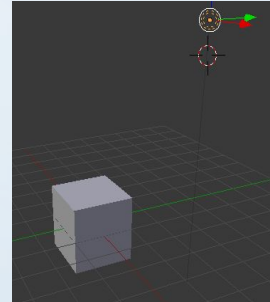
Area



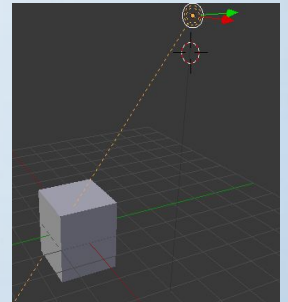
spot



hemi

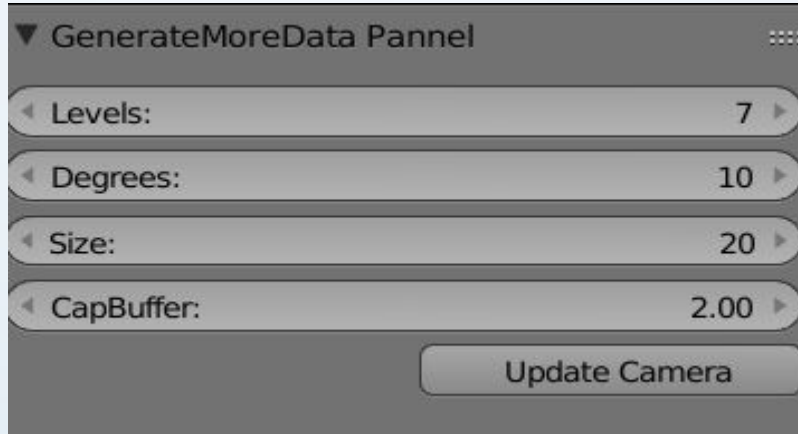


point



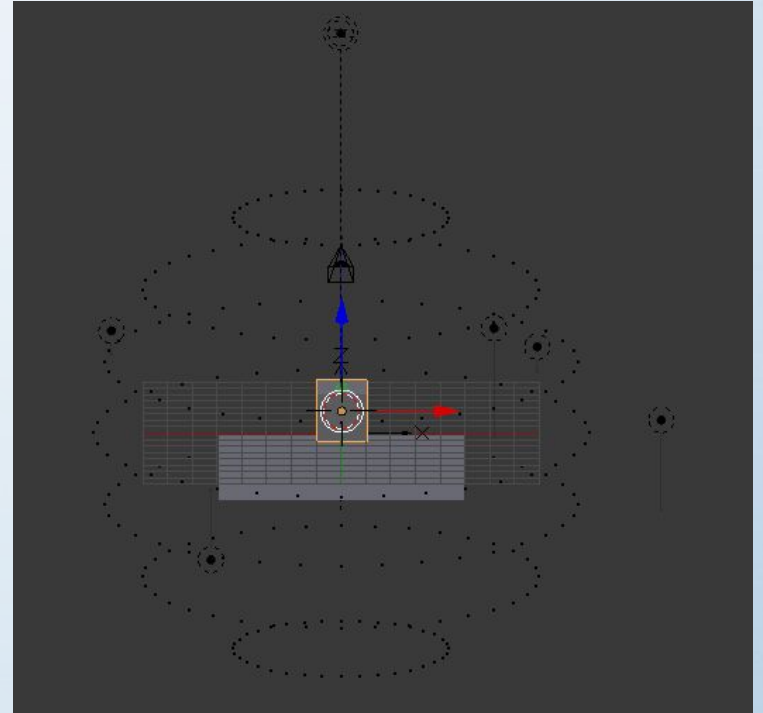
sun

Lamp Type	Buffer Shadow	Ray Shadow
Point	No	Yes
Sun	No	Yes
Spot	Yes	Yes
Hemi	No	No
Area	No	Yes



- Levels- Number of height levels
- Degrees-Degrees to rotate camera  
Ex:  $(360/10)$
- Size- Size of object
- CapBuffer- set the buffer at caps

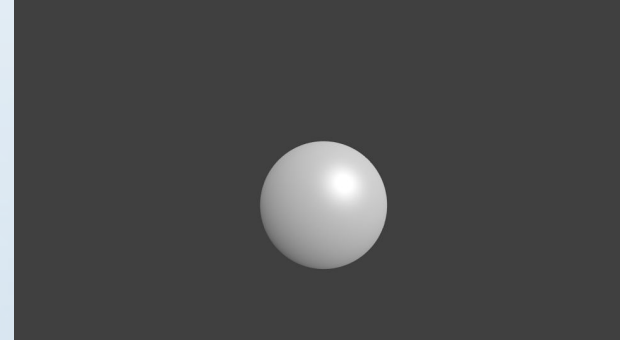
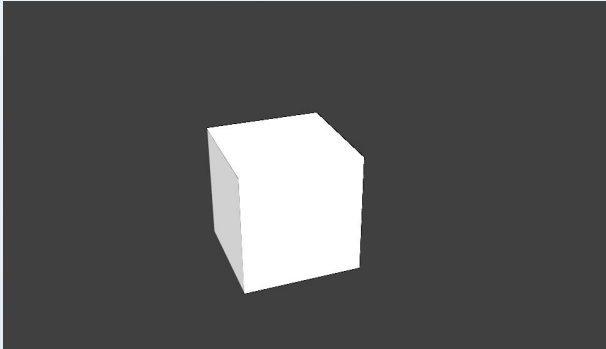
# Rendering



# Data

Initial Image DataBase created with simple models (Cube , Sphere) with

- Variation in Light sources and Number of light sources
- Variation in camera angles (pose)
- Variation in material colour

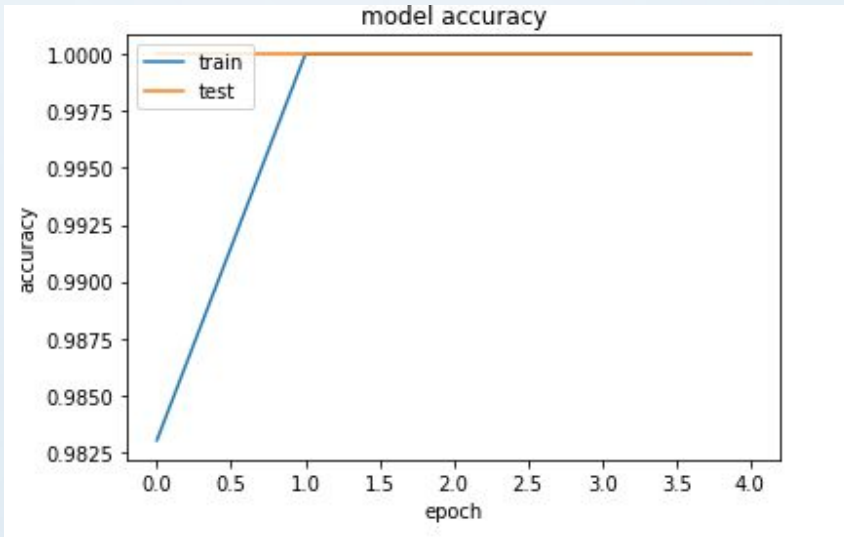


Sample Database Images of cube and sphere

# Simple CNN- Algorithm






Created a simple CNN Algorithm in keras(with tensorflow backend) to test our data



- With two convolutional layers



# Where to find our code

playhard21 simpleCNN Latest commit a575973 an hour from now

 CubeRendering.py	Rendering files	just now
 FirstCNN.py	simpleCNN	just now
 README.md	update	just now
 RotateAndRender.py	Rendering files	just now
 generateMoreData.py	Rendering files	just now

 README.md 

## Simulated-image-data

This is code for student project to Detection of logistic objects: "Training of deep neural networks with simulated image data "

Do not forget to add Empty while using RotateAndRender.py

Add Empty and select source as light while using CubeRendering.py

Add CamaraBoss while using generatemoreData.py

Code : <https://github.com/playhard21/Simulated-image-data>

# Problems with blender

## Problems in fully Automating the work

- Blender objects can't be accessed until we invoke them
  - Ex : Adding empty with code
  - Accessing the objects in light variation until we invoke light
- Invoking some objects can't be done through code in python had to be done manually

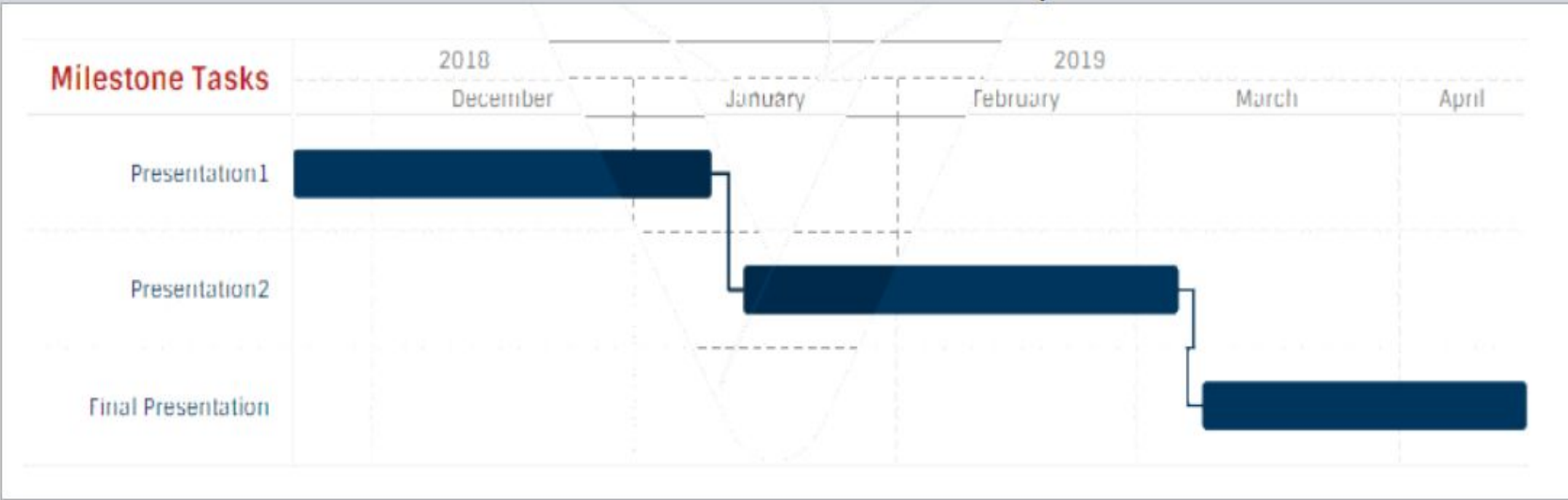


## Future work

- Create Database of logistic objects
- Automate some of the workflow
- Research activities on deep neural networks for the purpose of identification and Localization of logistics objects
- Evaluate Keras Implementation of Yolo-Network
- Compare several parametrization of the neural networks
- Building complex scene Database (Multiple combination of logistic objects )
- Fine tuning of existing neural network (eg Yolo) with virtual data
- Apply and evaluate neural network with real data



# Timeline



*Thank you*