**HO CHI MINH UNIVERSITY OF SCIENCE**

**HO CHI MINH NATIONAL UNIVERSITY**

**A picture containing logo

Description automatically generated**

**FINAL SEMESTER PROJECT**

**Subject: Introduce to Machine learning**

**YOLO – ANIMAL DETECTION**

**COURSE CODE**

**CSC14005**

**Ho Chi Minh city – 2022**

**TABLE OF CONTENT**

[**1.** **Overview** 3](#_Toc91597719)

[**2.** **Detail** 3](#_Toc91597720)

[2.1 Dataset 3](#_Toc91597721)

[2.2 Web interface 3](#_Toc91597722)

[2.3 Code 3](#_Toc91597723)

[2.4 Evaluate the model 3](#_Toc91597724)

[**3.** **Advantage and Disadvantage** 3](#_Toc91597725)

[3.1 Advantages 3](#_Toc91597726)

[3.2 Disadvantages 3](#_Toc91597727)

[**4.** **Result** 4](#_Toc91597728)

[**5.** **Necessary files** 4](#_Toc91597729)

[**6.** **Instruction for running code:** 4](#_Toc91597730)

[**7.** **Reference** 4](#_Toc91597731)

# **Overview**

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Full name** | **Role** |
| 19127392 | Tô Gia Hảo | Design UI + Collect dataset |
| 19127525 | Nguyễn Thanh Quân | Code UI + Code detection |
| 19127625 | Lâm Chí Văn | Design UI + Code Train + Train model |

# **Detail requirement**

## Dataset

We has 8 class for training and 750 sample for each, for each sample, will label all objects of that class.

Each template includes a picture of the object and a txt file that stores the labels of those objects:

The format file txt:

Diagram

Description automatically generated

## Require 1:

* We have done:
  + Build web interface which allows insert an image and return object detection result
  + Install YOLO to detect object
* Reference from source code:

## Require 2:

* We have done:
* Build Desktop app run base on HTML, CSS, JavaScript and Python
* Build a complete app use for detect animal in picture
* Reference from source code:

Average Precision

|  |  |  |  |
| --- | --- | --- | --- |
| class\_id | name | TP | FP |
| 0 | Cow | 333 | 34 |
| 1 | Buffalo | 12 | 6 |
| 2 | Cat | 62 | 13 |
| 3 | Dog |  |  |
| 4 | Deer |  |  |
| 5 | Lion |  |  |
| 6 | Eagle |  |  |
| 7 | Tiger |  |  |

F1-score & Average IoU

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| conf\_thresh | precision | recall | F1-score | TP | FP | FN | average IoU |
| 0.25 | 0.88 | 0.95 | 0.92 | 407 | 53 | 21 | 69.55 % |

* Data
  + Train: 60 %
  + Test: 20 %
  + Validation: 20 %
* score on these sets and the score by epoch

# **Advantage and Disadvantage**

## 3.1 Advantages



## 3.2 Disadvantages



# **Instruction for running code:**

**Train model:**

**Detect animal:**

**Step 1:** Run python animalDetection.py in Animal-Detection\Src

Graphical user interface, application

Description automatically generated

Figure 1: app interface

. It will auto download file weight, you need not do anything, just run

**Step 2:** Choose Image by click **upload new photo** button

Graphical user interface, text

Description automatically generated

This image will display in left frame and auto detect

# **Result**

# **Necessary files**

File weights:

# **Reference**

Video demo:

For more access: <https://github.com/QuanBlue/Animal-Detection.git>