

## Pedestrian Recognition Technology Using YOLO

Min-Hui Heo<sup>a</sup>, Jin Lee<sup>b</sup>, Dong-Jin Shin<sup>c</sup>, Yong-Soo Lee<sup>d</sup>, Jeong-Joon Kim<sup>e</sup>

<sup>a</sup>Department of Computer Mobile Convergence, Gyeonggi University of Science and Technology, South Korea, [hmh4902@naver.com](mailto:hmh4902@naver.com)

<sup>b</sup>Department of Computer Mobile Convergence, Gyeonggi University of Science and Technology, South Korea [ing013@naver.com](mailto:ing013@naver.com)

<sup>c</sup>Department of Computer Engineering, Anyang University, South Korea, [djshin@ayum.anyang.ac.kr](mailto:djshin@ayum.anyang.ac.kr)

<sup>d</sup>Professor, Department of Software and Communications Engineering, Yeosu Institute of Technology, South Korea, [diclee@yit.ac.kr](mailto:diclee@yit.ac.kr)

<sup>e</sup>Assistant Professor, Department of ICT Convergence Engineering, Anyang University, South Korea

**\*Corresponding author:** [jjkim@anyang.ac.kr](mailto:jjkim@anyang.ac.kr)

### Abstract

Recently, the performance of pedestrian recognition techniques has been improved rapidly with the introduction of deep learning, and the scope of their use has also been widely used in various fields. Currently, most pedestrian recognition technology focused on is detecting a standing person. These general pedestrian recognition techniques are inappropriate for situations where detailed pedestrian information is needed, such as complex traffic conditions, child protection zones, and disasters that make it difficult to identify children. Well-known algorithms for pedestrian recognition techniques include Faster R-CNN, YOLO, and SDD. In this paper, we use the Yolo algorithm to differentiate between adults and children among pedestrians. Furthermore, we confirm the change in detection results with the number of learning images by gradually increasing the number of learning images, and propose an improvement method to improve the detection accuracy of pedestrians.

**Keywords:** Yolo, pedestrian recognition, Image Detect

### 1. Introduction

As deep learning has been combined with AI, one of the main technologies of the Fourth Industrial Revolution, pedestrian recognition technologies, including computer vision, are being used in various fields. Pedestrian recognition technology is being used in safety protection systems to identify pedestrians in real time on devices capable of filming images such as CCTV and to inform them when a risk is detected. However, although it is still determined whether the object is a pedestrian, detailed research on detecting objects such as adults, children, the elderly, and the disabled is still insufficient. Therefore, in this paper, we attempt to distinguish pedestrians using Yolo, which has recently been in the spotlight among various image recognition algorithms. The used version is v3, and the photos required for learning were collected by implementing a crawling program, and the tools used for labeling were Yolo-produced Yolo\_Mark. Paper is composed as follows: Chapter 2 introduces a study related to the algorithms used in this study. Chapter 3 describes the labeling method and main code for object recognition in this work. Finally, Chapter 4 discusses the conclusions of this study and the direction of improvement in future research.

## 2. Related Technologies and Research

### 2.1. Labeling Operations

There is a HOG detector as a basic method of pedestrian recognition technology. The HOG detector divides the area of the object into a constant cell format, obtaining a histogram of the slope or higher for each cell, and then extracting the distribution features to identify pedestrians.[1] Traditional HOG detectors have a slow drawback, and there is a Cascade HOG method that improves this. Cascade HOG is a method of applying cascade technology to a HOG detector, which computes the HOG from a variety of blocks and selects the appropriate blocks through boosting learning.[2]

### 2.2. YOLO

yolo (You Only Look Once) is an algorithm for detecting multiple objects and predicting images using a representative single network.[3] Algorithms can predict the type and location of objects, with single convolutional probabilities of bounding boxes and class probabilities within images trained around the center of the grid using grids of the same size. Based on the prediction box, we calculate the reliability from the learned image to identify the categories by selecting objects with high object reliability.

### 2.3. Yolo labeling

Data labeling refers to the process of categorizing human-made photos, documents, etc. into categories using data processing tools and deriving location coordinate data information values from objects for images. YOLO labeling uses YOLO mark, a subprogram of YOLO, as a process of collecting learning data and labeling them to generate individual models based on YOLO. yolo mark is a labeling tool for learning directly yolo's image data information[4] The labeling tool labels objects in the image to be learned by drawing them with bounding boxes by constructing the center coordinates  $x$ ,  $y$ , and bounding box width and height  $w$ ,  $h$ .

### 2.4. Related Studies

Based on video analysis, there is a study that uses biometric functions to classify children/adults.[5] This study uses the Haar cascade algorithm, which orientation of pixels with a constant magnitude of aims to classify and detect children and adults by measuring biometrics considering the relative

proportional length of the head and body. The results of the study show a slightly higher accuracy for adults, with 100% and 64.5% for children and adults, respectively. Furthermore, there is a study that constructs and learns a novel artificial neural network model based on Deep Learning YOLO's artificial neural network model, which belongs to a public data portal.[6] This study aims to implement and study real-time pedestrian detection and alarm systems that can be linked to intelligent transportation systems. The results of this study implement My\_Tiny\_Model3 artificial neural network model with higher detection accuracy and optimized real-time processing than conventional Yolo models.

In addition, there is a study based on YOLO v3 that implements real-time license plate character recognition and model recognition systems using virtual data.[7] As a result of this study, license plate characters and vehicle recognition systems are divided into YOLO v3 tiny and YOLO v3 to reveal their respective accuracy.

## 3. YOLO Labelling Operations

Labeling based on the YOLO algorithm used in this paper is a necessary process to create a weight file for object detection. The tool that provides labeling based on the YOLO algorithm is the YOLO mark. To build a labelling environment using GPUs in Yolomark, we installed CUDA 8.0 and opencv 3.2.0 versions available on Windows 10 and GTX1060, and Visual Studio 2015 compatible with Yolov3. If you open yolo\_mark.vcxproj, you can see where CUDA is used. It modifies the corresponding code and connects it with CUDA to establish an environment where GPU is used. To build an environment where opencv is available in yolo\_mark.sln, we edit the user variable Path for Rey to create a path to access opencv. When a path is set and a directory is added, the environment for building the program is configured.

### 3.1. Labeling Operations

Before proceeding with the labeling task, we modify the yolo-cfg file to determine the number of objects to be learned and change the value of the filter according to the number of objects. The filter value is  $5 * (\text{number of object classifications} + 5)$ . Choose pictures of objects under 40 years of age and infants to learn put them

inside the /Release/data/img folder. Open the obj.data file that exists inside the x64/Release/data folder and save the classes as few as the number of object classes to learn. It also enters and stores the classification names Kid and Adult of the objects to be learned in the obj.names file. The object id of the object is given in order from zero in the object name stored in the obj.names file. This paper is designated as Kid No. 0 Adult No. 1

After successful construction of yolo\_mark.sln, yolo\_mark.exe is created in the x64/Release folder, and yolo\_mark.cmd can run to proceed with labeling as shown in Figure 1.



Fig 1. adult & kid is Labeled appearance

The labeling opera selects the desired learning object among the objects in the picture and draws a bounding box, so only the objects that exist in the bounding box are used for learning. Select the object id you want to learning is draw a bounding box, obj.names the location coordinates of the bounding box (four vertex) with the object name you specified in the names file are created in the img folder as the object photo name.txt file, At the same time, a path to the photo file located in the data/imge file is created in the train.txt file. Afterwards, will proceed with the learn by dividing the object image into grids around the position coordinates written in the .txt inside the img folder.

In this paper, the criteria for images set to clearly indicate the distinction between adults and children are kid/adult Whole body, adults are adults in their 40s under, child was chosen as an infant.

The reason chose a kid as an infant is because in the process of defining the characteristics of adults and children, with infants or older were not clearly recognized after learning and two objects were confused, becoming they were recognized kid and adult at the same time so chose to be an infant.

### 3.2. Ccheck labeling

Related files after labeling are the img directory, obj.data, obj.data obj.names, train.txt, yolo-obj.cfg file exists.

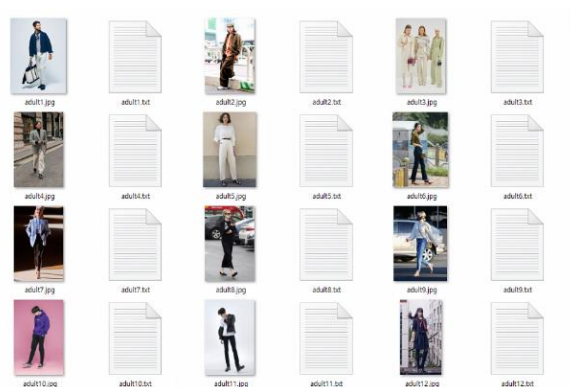


Fig 2. The img directory used for yolo training

## Pedestrian Recognition Technology Using YOLO

As shown in Figure 2, the img directory contains photo needed for labeling, and when the object id of the labeled object and the object name with the coordinates of the corresponding bounding box are generated when, indicating that the bounding box was drawn normally.

```
classes= 2
train = data/train.txt
valid = data/train.txt
names = data/obj.names
backup = backup/
```

Fig 3. obj.data file for configuration of Yolo\_mark

As shown in Figure 3, the obj.data file contains the number of classes to classify, train.txt that contains the location of the data to be learned, Path to the names file where the object's name is stored, location of the backup copy file that will be saved when the weights file is created.

obj.names file is a file that stores the name of the object to be learned. object id is generated from zero in the order in which it is written in the file, and when it is labeling, it draws a bounding box by distinguishing objects through that object id. object id is generated from zero in the order in which it is written in the file, and when it is labeling, it draws a bounding box by distinguishing objects through that object id.

```
data/img/adult1.jpg
data/img/adult10.jpg
data/img/adult100.jpg
data/img/adult1000.jpg
data/img/adult1001.jpg
data/img/adult1002.jpg
data/img/adult1003.jpg
data/img/adult1004.jpg
```

Fig 4. train.txt to save the labeled image path file.

while assigning a bounding box to the object through a labeling making as shown in Figure 4, train.txt file to the determined saves the path of the object picture Automatically. Stored paths are used to load images as they progress through learning.

```
[convolutional]
size=1
stride=1
pad=1
filters=35
activation=linear

[region]
anchors = 1.08,1.19, 3.42,4.41, 6.63,11.38, 9.42,5.11, 16.62,10.52
bias_match=1
classes=2
coords=4
num=5
softmax=1
jitter=2
rescore=1
```

Fig 5. yolo-obj.cfg for setting of yolo

The yolo-obj.cfg file is a file that defines blocks unit in Yolo's layout, as shown in Figure 5, this is a setup file for labeling. Before starting labeling, filters and classes must be modified to label the desired number of objects. The classes number, which specifies the number of objects, were modified by setting to kids and adult, two, and the filter value described in this section 1 was calculated according to the number of objects and entered below [convolutional], filters as 35.

### 3.3. Course of study (maincode)

The Learning is proceeded using the Darknet Yolov3 algorithm.

Yolomark's labelling information files img directory, train.txt, obj.names, obj.data, yolo-obj-cfg files is move to Darknet Yolo file and proceed with the learning. Depending on the environment, the batch of yolo-obj-cfg with the amount of RAM needs to be modified to a lower level.

Figure 6 shows darknet.c, the main code of the algorithm used in the learning process

```

1  #include "darknet.h"
   #include <time.h>
   #include <stdlib.h>
   #include <stdio.h>
   #if defined(_MSC_VER) && defined(_DEBUG)
   #include <crtdbg.h>
   #endif

   #include "parser.h"
   #include "utils.h"
   #include "dark_cuda.h"
   #include "blas.h"
   #include "connected_layer.h"

2  extern void nn_detector(int argc, char **argv);

3  } else if (0 == strcmp(argv[1], "detector")){
   nn_detector(argc, argv);
   } else if (0 == strcmp(argv[1], "detect")){
   float thresh = find_float_arg(argc, argv, "thresh", 24);
   int ext_output = find_arg(argc, argv, "ext_output");
   char *filename = (argc > 4) ? argv[4] : 0;
   test_detector("cfg/coco.data", argv[2], argv[3], filename, thresh, 0.5, 0, ext_output, 0,
   NULL, 0, 0);
   }

```