ABC - Automatic background change using deep learning

Quoc Viet Tran, Trung Nguyen Nguyen, Gia Huy Le

Advisor: TrungNQ46

Abstraction: Given the worldwide health crisis caused by COVID-19, most governments throughout the world have implemented a lockdown strategy to prevent community spread over the last two years. The fact that everyone works from home has pushed online meetings to become extremely popular. Most people turn on the camera to their colleagues after a long time. However, unwanted background objects are often captured by the camera. So it needs to be removed and replaced with a suitable background. Based on that idea, we decided to create an application that allows any user to change the background with a personal image to suit their requirements, or according to the background suggested by our AI model.

Keywords: *background removal, change background, image processing, object detection, image segmentation, deeplabv3+, yolov5,...*

1 Introduction

1.1 Background and context

Artificial intelligence in general and deep learning, in particular, have been developing rapidly and are applied in many aspects and specific tasks. Among them, computer vision is an outstanding field that is applied a lot in modern life today, such as face recognition, self-driving car technology and optical character recognition - OCR . . . In general, computer vision consists of two main tasks: object detection and image segmentation. In this problem, we will use both of these tasks to solve the problem posed above.

1.2 Project solution overview

Regarding the solution, we will initially use the input of a photo through the YOLOv5 model to identify the outfit of the person in the photo and then suggest a background suitable for the context of that type of outfit, next, the Deeplabv3 model will segment the image to separate the human subject and the background, and finally make changes automatically or according to the background that the user wants.

To make the problem clear, we will show it step by step through the following flowchart:

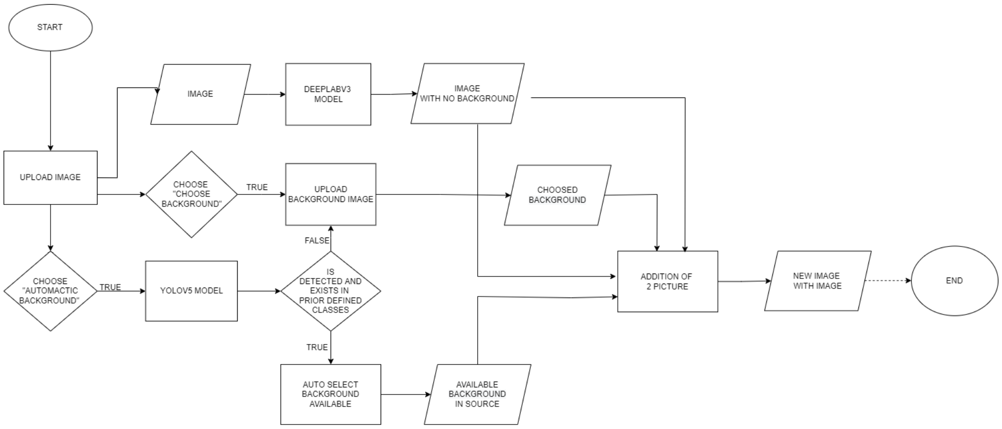

Figure 1: Flowchart for automatic background change application

2 Related worked

Computer Vision: Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information [1]. If AI enables computers to think, computer vision enables them to see, observe and understand.

Figure 2: Computer vision is the eye in digital [4]

Computer vision works much the same as human vision, except humans have a head start. The human sight has the advantage of lifetimes of context to train how to tell objects apart, how far away they are, whether they are moving and whether there is something wrong in an image.

Computer vision trains machines to perform these functions, but it has to do it in much less time with cameras, data and algorithms rather than retinas, optic nerves and visual cortex. Because a system trained to inspect products or watch a production asset can analyze thousands of products or processes a minute, noticing imperceptible defects or issues, it can quickly surpass human capabilities.

2.1 Object detection

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos [2]. Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance.

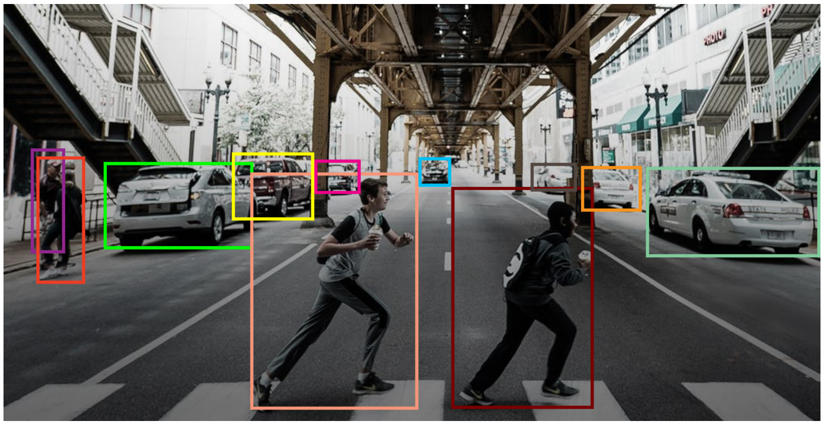

Figure 3: Description of particular task in object detection [5]

Methods for object detection generally fall into either neural network-based or non-neural approaches. For non-neural approaches, it becomes necessary to first define features using one of the methods below, then use a technique such as a support vector machine (SVM) to do the classification. On the other hand, neural techniques are able to do end-to-end object detection without specifically defining features, and are typically based on convolutional neural networks (CNN).

- Non-neural approaches:
  - ∇ Viola–Jones object detection framework based on Haar features
  - ∇ Scale-invariant feature transform (SIFT)
  - ∇ Histogram of oriented gradients (HOG) features

- Neural network approaches:
  - ∇ Region Proposals (R-CNN, Fast R-CNN, Faster R-CNN, cascade R-CNN)
  - ∇ Single Shot MultiBox Detector (SSD)
  - ∇ You Only Look Once (YOLO)
  - ∇ Single-Shot Refinement Neural Network for Object Detection (RefineDet)
  - ∇ Retina-Net
  - ∇ Deformable convolutional networks

2.2 Image segmentation

In digital image processing and computer vision, image segmentation is the process of partitioning a digital image into multiple image segments, also known as image regions or image objects (sets of pixels) [3]. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image or a set of contours extracted from the image (see edge detection). Each of the pixels in a region is similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different colors with respect to the same characteristic(s). When applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like marching cubes.

Method for image segmentation:

- Thresholding method
- Clustering methods
- Motion and interactive segmentation
- Compression-based methods
- Histogram-based methods
- Edge detection
- Dual clustering method
- Region-growing methods
- Partial differential equation-based methods

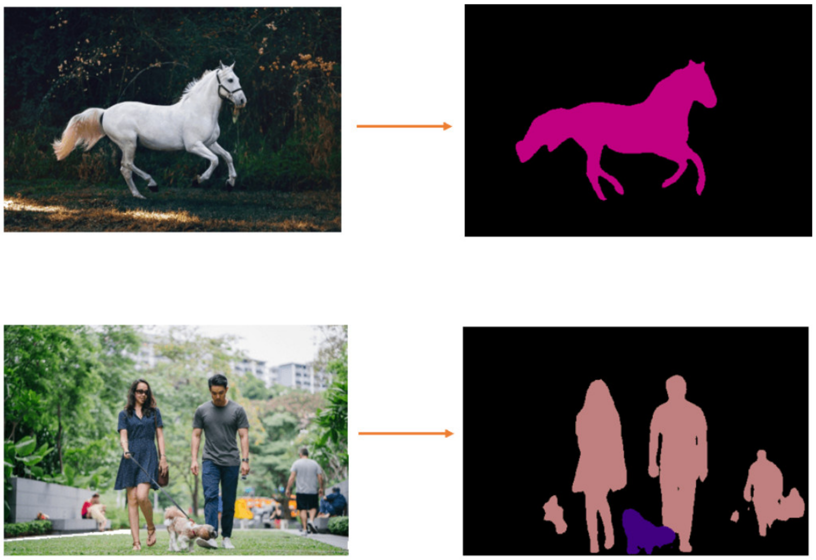

Figure 4: Example about image segmentation [6]

- Variational methods
- Graph partitioning methods
  - ∇ Markov random fields
- Watershed transformation
- Model-based segmentation
- Multi-scale segmentation
  - ∇ One-dimensional hierarchical signal segmentation
  - ∇ Image segmentation and primal sketch
- Semi-automatic segmentation
- Trainable segmentation
- Segmentation of related images and videos

3 Data preparation

3.1 Data for image segmentation

The data used is the ”Person Segment” dataset which is public on Kaggle, after downloading this data is enhanced by 5 times to create data diversity. This is a data set that is used a lot in the "person segmentation" problem. It includes a set of images containing human images and their binary mask.

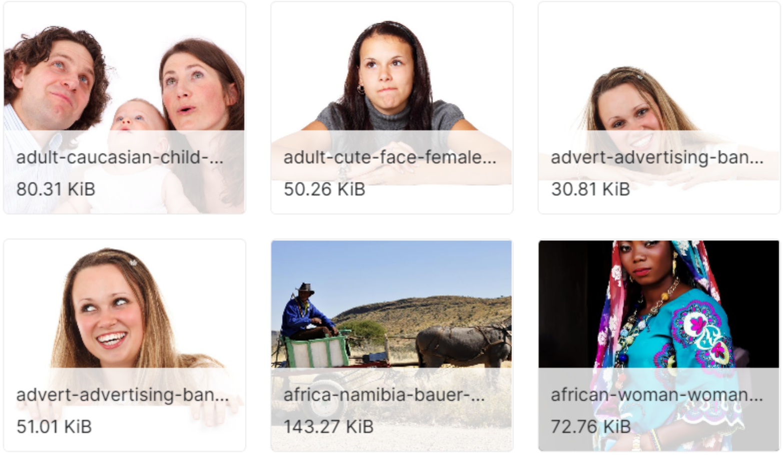

Figure 5: Description of person segment dataset

3.2 Data for object detection

In this section, we temporarily define three types of clothing that the AI model needs to detect, including beachwear, workwear, and sportswear. Then we crawl from google for all 3 previously defined classes respectively. In each class, we crawl about 300 related photos to do manual labeling. In the data enhancement step, we perform 3x data enhancement before feeding the model to perform the training process.

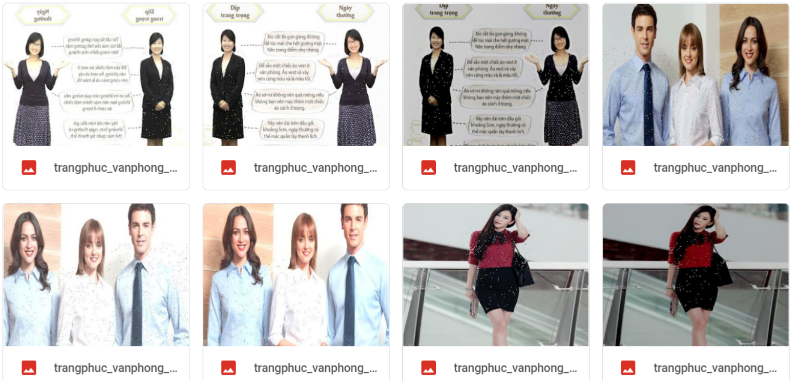

Figure 6: Description of clothes detection dataset

4 Methodology

4.1 Overview

The background model is a preliminary component for the foreground extraction. In this study, we will use models and deep learning algorithms to extract the background on photos and videos, and then, allow users to make changes to them. Specific methods will be shown in the following subsections.

4.2 Background modeling

4.2.1 DeepLabv3+

DeepLabv1 and v2 are authored by Liang-Chieh Chen and George Papandreou, who are longtime Semantic Segmentation researchers and were first published in 2016 and for the second time in 2017. The second draft is an addition to the first, so DeepLabv1 and v2 are not much different [8]. Immediately after its publication, the architecture achieved impressive results on the validation dataset of Pascal VOC 2012. This is an architecture that flexibly applies Atrous convolution instead of the previous methods. Apply Transposed Convolution. Besides, the author also applies the Conditional Random Field method to refine the forecast results more accurately. After Deeplabv1 and Deeplabv2 were invented, authors tried to RETHINK or restructure the DeepLab architecture and finally come up with a more enhanced DeepLabv3.

In DeepLabv3 the author put in 2 tweaking improvements:

1. Conduct ASPP parallel convolution at many different scales and add batch normalization, inheriting ideas from the Inception network.

2. In particular, removing Fully Connected CRF at the last processing step helps to increase calculation speed.

The DeepLabv3+ network is one of the most excellent semantic segmentation models at present, but there are some shortcomings in the models. In order to increase the ability to segment multi-scale targets

DeepLabv3+ connects to the ASPP structure after extracting the network with hollow convolutional abstract features. The structure consists of 3×3 hole convolutions with expansion ratios of 1,6,12,18, respectively, and global average pooling operations. Excessive expansion rate cannot accurately extract the target features of the image edge, and at the same time, it cannot completely simulate the relationship between the local features of the large-scale target, so there is a hole phenomenon in the large-scale target segmentation. This results in the DeepLabv3+ network segmentation accuracy of remote sensing image edge targets and large-scale targets being reduced.

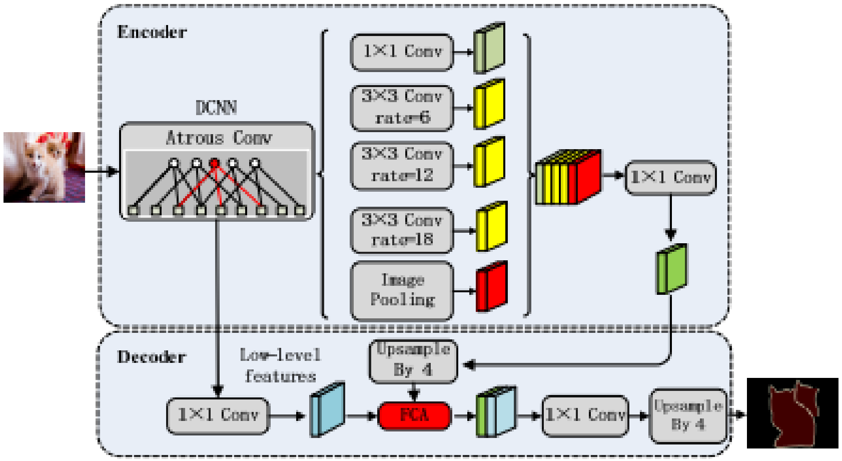

Figure 7: The Deeplabv3+ model with FCA module added

4.2.2 YOLOv5

YOLO (You only look once) is an effective real-time object recognition algorithm, first described in the seminal 2015 paper by Joseph Redmon et al, different from the family of RCNN (Region-based Convolutional Neural Network) - not suitable for real-time objects because of its slow processing speed [7].

Yolo is created from a combination of convolutional layers and connected layers. In which convolutional layers will extract the features of the image, while fully-connected layers will predict the probability and the coordinates of the object.

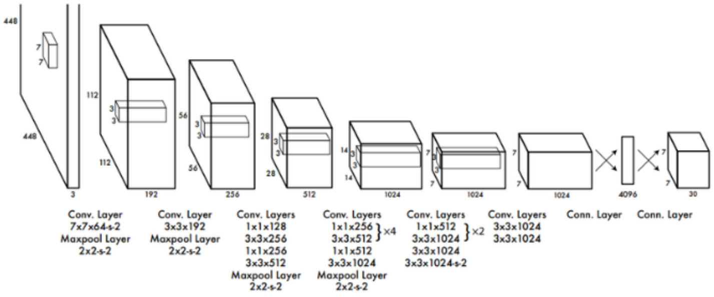

Figure 8: YOLO network architecture

YOLO currently has many versions, but here we will use the most modern version, which is YOLOv5. YOLOv5 is a family of object detection architectures and models pretrained on the COCO dataset, and represents Ultralytics open-source research into future vision AI methods, incorporating lessons learned and best practices evolved over thousands of hours of research and development.

5 Result and discussion

5.1 Clothes detection result

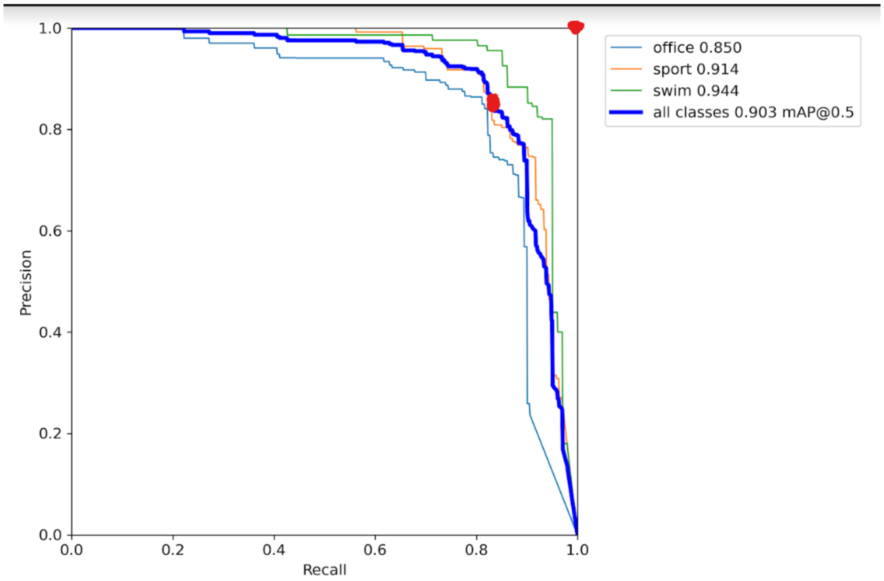

Figure 9: PRcurve of YOLOv5 model for this task

There are actually a lot of learning curves created and we can use them in the model, but here we consider PRcurve. Based on the above figure, we can see that the current model is quite good with both precision and recall greater than 0.85. Or to be more clear, we look at the 2 red points on the image, the higher point is the point where the model is perfect and the other point is very close to that point, which means the current model is a good model.

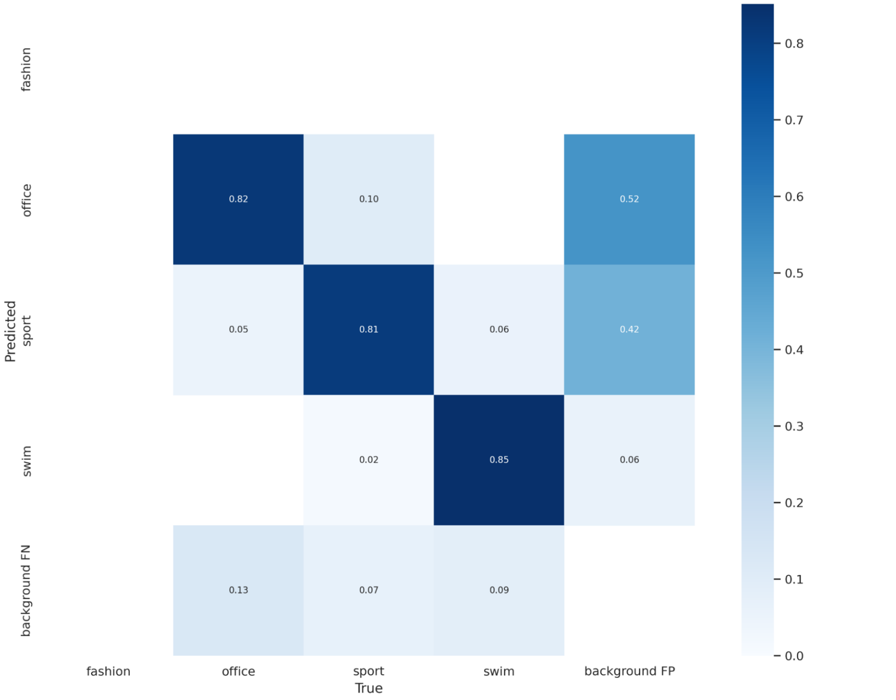

Figure 10: Confusion matrix on this model

Similarly, for the confusion matrix, we can see that the accuracy in each class is quite high, demonstrating the practical ability of the model.

5.2 Image segmentation result

Accuracy	**F1**	**Jaccard**	**Recall**	**Precision**
0.97	0.95	0.92	0.95	0.95

Table 1: Quality metrics for image segmentation model with Deeplabv3+

In addition to the usual measurement parameters like accuracy, precision, recall, F1score. . . , we will refer to Jaccard, this is a metric often used in image segmentation, it only corresponds to the extent at the actual mask and the prediction mask. The basic idea is to regard the image masks as sets. These sets can overlap within the picture. If both masks are completely identical, both sets have exactly the same size and do overlap to 100%, so that intersection equals union.

In this case, the IoU score is 1 and optimal. On the other hand, if the predicted mask is shifted or changed in size compared to the original mask, then the union gets bigger than the intersection. The IoU score decreases.

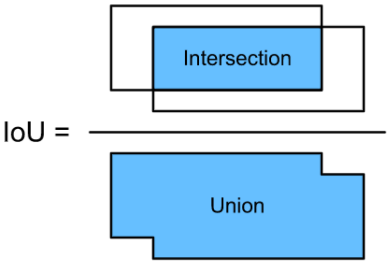

Figure 11: Formula for IoU score

6 Conclusion

With the above methods, the application to change the image background has quite good performance, but there are still many limitations, such as the models will be better if there are more quality data and there are even better conditions to perform model training on more batches and epochs. In fact, the app is only working fine on photos with 1 subject on it, which can also be an issue that the app needs to improve to further enhance the user experience.

References

[1] IBM Solutions, *What is computer vision?*

[2] Wikipedia, *Object Detection*

[3] Wikipedia, *Image Segmentation*

[4] Vaisala, *Computer Vision*

[5] AmazonAWS, *Real time object detection*

[6] Learnopencv, *Semantic Segmentation*

[7] Dinh Khanh Pham Blog, *YOLO You Only Look Once*

[8] Dinh Khanh Pham Blog, *DeepLab Sentiment Segmentation*

[9] Liang-Chieh Chen, George Papandreou, Florian Schroff, Hartwig Adam. *Rethinking Atrous Convolution for Semantic Image Segmentation arXiv:1706.05587*, 2017

Appendix A. Project Plan management

Description	Priority	Status	Deadline	Task recipient
Research and choose the right algorithm	High	Done	16/05/2022 - 26/05/2022	All
Configure, adjust hyperparameters, and define support methods	High	Done	27/05/2022 - 06/06/2022	All
Collect data, data cleaning, data labeling	High	Done	06/06/2022 - 13/06/2022	All
Evaluate models (using test datasets)	High	Done	13/06/2022 - 20/06/2022	All
More hyperparameter tuning for optimal performance	High	Done	20/06/2022 - 27/06/2022	All
Deploy the model	High	Done	27/06/2022 - 04/07/2022	All
Deploy to server or app	High	Done	04/07/2022 - 11/07/2022	All
Listen for events to remove and change the background	High	Done	11/07/2022 - 18/07/2022	All

Table 2: Project Plan

Appendix B. Source code & Data

Data & source code

Appendix C. Application

Demo: ABC - Automatic Background Change App