Video-based Human Tracking

Group ID:55

Xuwei Xu u6688636 Weifan Jiang u6683698 Haozhan Sun u6688485
Hongming Zhang u6693651 Guanyang Zhang u6688675

Abstract. Tracking human in videos becomes increasingly significant in CV application. In this project, we innovatively presented a combination of human detection and human re-identification, having created an industrial-level end-to-end product to achieve human tracking. For human detection, we presented an algorithm based on YOLO modifying its output tensor. For re-identification, we presented a modified ReID algorithm with combined loss function and construction of feature gallery, achieving a better mAP result with nearly real-time tracking speed. Finally, we combine the two parts as a pipeline to make up the whole product, from getting a simple video to returning a video with bounding boxes and ReID function.

Keywords: YOLO, ReID

1 Introduction

With the development of computer vision techniques, object tracking has become a new topic in recent years. Among all the tracking tasks, tracking human in videos is most essential and has the biggest potential in daily applications. Human-tracking can be applied in diverse situations, such as safeguard, surveillance, and human computer interaction. Moreover, this technique will be a milestone to the industrial computer vision as well as a link between other related tasks.

In order to prevent ambiguity, we define the meaning of human tracking in our project as labeling the same person with the same tag among a sequence of videos or across cameras. The aim of this project can be divided into two relevant parts. The first goal is to implement human detection in videos and to take the human image by a bounding box. The second aim is retrieving the human image to determine which person it belongs to as well as its corresponding label.

A modified network under YOLO concept is designed to achieve nearly real-time human detection. A feature extraction network is built based on DenseNet to calculate the feature vector. The two parts are integrated by a feature gallery system to eventually construct the video-based human tracking system.

2 Related Work

Representation learning. Geng, Mengyue, et al. (2017) used a two-branch Siamese network. That is, take a pair of input person detection images as input. After feature extraction of CNN, the image is sub-predicted by the classification subnet; the Verification Subnet is used to fuse the features of the two pictures to determine whether the two pictures belong to the same pedestrian, but the representation learning The method is more dependent on the data set, making it easier to overfit the trained data set [1].

Metric learning. In 2016, Cheng et al. made better results for person re-identification by using a special multi-channel parts-based convolutional neural network (CNN) model under the triplet framework [2].

According to Varior, Haloi and Wang (2016), the accuracy of ReID could be improved if contrastive loss and gating function are utilized on Siamese Convolutional Neural Network (S-CNN) [3].

ReID method based on image splitting. In 2016, Varior et al. proposed a method based on local features which split the graph horizontally [4]. By using Long Short-Term Memory (LSTM), the final feature is the combination of the local features of all image blocks. However, if two images are not aligned up and down, it is likely that the head and upper body are compared, which will lead to an incorrect judgement.

Local feature extraction with key body points. In paper [5], a new CNN model called Spindle Net is introduced, which is based on multi-stage feature de-composition and tree-structured competitive feature fusion guided by human body region.

3 Method

3.1 YOLO

3.1.1 YOLO Introduction

In object detection area, You Only Look Once (YOLO) is a network model firstly raised by Redmon, J. et al in 2015 [6]. It differs from the R-CNN series since YOLO uses only a single convolutional neural network to predict multiple bounding boxes, which outperforms most of the current object detection methods, especially on speed.

Despite of its training skills and network-building techniques, the most innovative idea of YOLO is that it regards the whole image as being split into $K \times K$ grids and each grid is supposed to encode the information of bounding boxes as well as objectness. It is assumed that the center of

bounding box must locate at some certain grid, hence running a single convolutional neural network to compute out a $K \times K \times N$ 3-D matrix is satisfiable for encoding bounding boxes.

3.1.2 YOLO Principle

In the basic YOLO, an input image will be divided into $S * S$ grids and each grid cell predicts B bounding boxes. If the total number of object classes is denoted as C, then the prediction is encoded as an $(S * S * (B * 5 + C))$ tensor, shown as Fig1. The objectness, i.e. the possibility of objects, is indicated by the C vectors for each grid. And there will be five figures encoding the position, shape and confidence of each bounding box. The bounding box with maximum confidence in one grid will be selected to represent the predicted bounding box if its ojbectness is higher than the threshold.

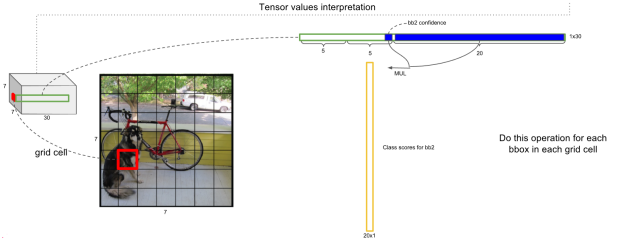

Fig. 1: Basic YOLO concept

To improve the basic YOLO, YOLOv2 comes up with some new methods, including batch normalization, high resolution classifier, etc. The most thrilling improvement is introducing the anchor box from faster R-CNN. Anchor box enables the prediction if multiple objects are in the same grid. Thus, the output will become an $(S * S * (B * (5 + C)))$ tensor, which shows that there will be at most B possible objects in one grid and the $(5 + C)$ vectors will show what and where the object is.

In YOLOv3, the latest version, it uses Darknet-53 network which is built based on ResNet to replace the old Darknet-19 network. The new network is adequately deep for extracting more small features. It also adopts 9 anchor boxes derived from different layers for predicting across scales. Although there are several new methods are applied to improve the performance of YOLO, the overall concept is the same as the basic YOLO.

3.1.3 Our Modified YOLO Method

We build our own model by modifying the YOLOv3 network.

First, since we only want to know whether there is a human in the bounding box, the (5+C) vectors for each bounding box is changed into (5+1) vectors where the first five numbers encode the bounding box and the last number indicates the possibility of human. And we suppose there are at most two people appearing in the same grid, otherwise they would be severely overlapped and unsuitable for re-identification. The final output tensor is shown as Fig2

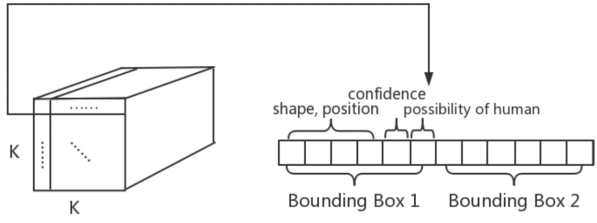

Fig. 2: Modified output tensor

Second, we maintain the most part of Darknet-53, including its bounding box prior. In the original Darknet-53 network, three distinctly scaled detections are calculated from the 82nd layer (19*19), 94th layer (38*38) and 106th layer (76*76) separately, shown as Fig 3. The three detections are used to predict multiple possible scales of object. The larger the size is, i.e. the image being divided into more grids, the smaller the detected objects will be. This relation between the three predicting layers in distinct scales is called bounding box prior which helps to predict objects across scales.

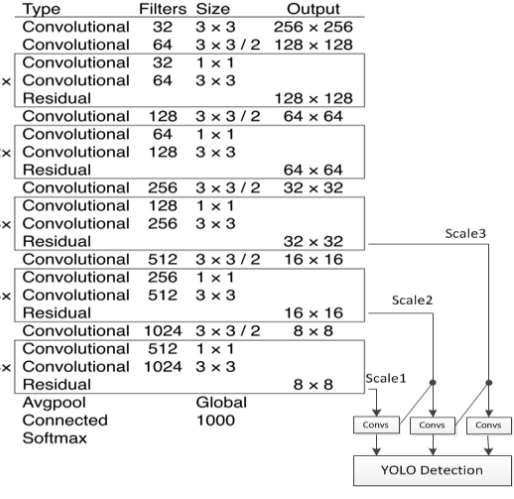

Fig. 3: Three layers of anchor boxes in different scales to form the final detection result

Third, considering the shape of pedestrians, which is thin and tall, we eliminate the wide and short anchor box. In the original Darknet-53, there will be three different anchor boxes for each scale, hence totally nine anchor boxes. Although anchor box does not affect the prediction, it performs to improve the training processes. Among the nine anchor boxes, we drop one anchor box for each scale so that the model is specialized for detecting human-shaped objects. Since we do not tend to retrain a brand-new model, we simply use the two human-shaped anchor boxes and ignore the result of the third box. Therefore, the last twenty layers of the network do not need to be changed.

If the four coordinates predicting bounding box are denoted as t_x, t_y, t_w, t_h, respectively representing the relative x and y coordinates in the grid and relative width and height. And the left top coordinates of the image are denoted as c_x, c_y. Then we can derive the parameters of bounding box as Eq1.

$$\begin{cases} b_x = \sigma(t_x) + c_x \\ b_y = \sigma(t_y) + c_y \\ b_w = p_w e^(t_w) \\ b_h = p_h e^(t_h) \end{cases} \tag{1}$$

Where b_i represents the offset of the real bounding box and p_w, p_h are the width and height of bounding box prior.

3.1.5 Non-Maximum Suppression

In the output tensor above, there will be 76 * 76 * 2 = 11,552 possible predicted bounding boxes. The number of bounding boxes is much greater than the number of possible existing humans, which might lead to bounding box overlapping. Hence, non-maximum suppression is adopted to eliminate those bounding boxes.

For each bounding box, if the confidence is higher than the threshold (0.5) and the possibility of detecting human is higher than another threshold (0.85), then we assume that the bounding box successfully detects a human. Assume the two bounding boxes in one grid are denoted as $bbox_1$ and $bbox_2$.

Prior to all, we sort all the $bbox_1$ that successfully detect a human by its confidence. Then we iteratively traverse the $bbox_1$ sequence. During each iteration, eliminate all those less confident $bbox_1$ which have high IoU (Fig 4) with the current $bbox_1$, shown as Eq 2. Repeat the steps above until the

$bbox_1$ sequence is traversed. Adopt the same non-maximum suppression operations to $bbox_2$ as well.

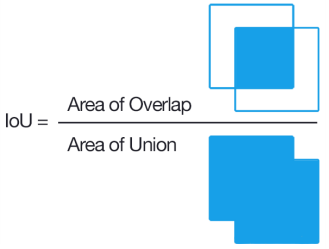

Fig. 4: Definition of intersection over union

$$IoU(bbox_{current}, bbox_{maxconfidence}) > 0.5 \tag{2}$$

Hence, when multiple grids detect humans, non-maximum suppression ensures we identify the optimal cell among all candidates where the human belongs.

3.2 ReID

3.2.1 ReID Introduction

What is ReID. Person Re-identification, also referred to as ReID, is a technique in computer vision to determine whether a specific pedestrian exists in an image or video sequence. It is widely considered to be a sub-question of image retrieval. The aim of ReID is: given a pedestrian image in the monitor, retrieve the pedestrian image across different devices (i.e. ReID requires all of the images of a certain pedestrian under other cameras to be retrieved).

Why ReID. In surveillance video, due to camera resolution and shooting angle, high quality face images are often not available. In case of face recognition failure, ReID has become a significant alternative technique.

Example application: Intelligent security, unmanned market, photo album clustering, etc.

3.2.2 Our Modified ReID Method

In ReID section, CNN is used to extract features of human images. After training, CNN could extract human features rather than background or other useless information.

Loss function. The Triplet loss with batch hard mining is adopted [8]. During the training process, every training batch contains a group of images randomly sampling P classes (person identities), and then randomly

sample K images of each class. We denote A as a group of images which have the same labels in each batch while B has different labels, and α as a pre-set constant. Triplet loss with batch hard mining loss is defined as follows:

$$L_{th} = \frac{1}{P \times K} \sum_{a \in batch} \left(\max_{p \in A} d_{a,p} - \min_{n \in B} d_{a,n} + \alpha \right)_+ \tag{3}$$

Trihard loss function will compute Euclidean distance between the anchor image and every image in the batch. Then, to form the Triplets, a positive image with the largest distance (the hardest positive sample) and a negative image with the smallest distance (the hardest negative sample) are selected to compute the loss.

In addition, softmax cross-entropy loss is adopt as well, which is widely used in classification tasks. We denote $pred_i$ as one of the outputs of CNN (a 1-by-751 vector in market-1501). The Softmax cross-entropy loss is defined as:

$$p_i = \text{softmax}\left(\text{pred}_i\right) = \frac{e^{\text{pred}_i}}{\sum_{j=1}^{n} e^{\text{pred}_i}} \tag{4}$$

$$L_{cross\_entropy} = -\sum_{i} \log p_{i,label_i} \tag{5}$$

Where the softmax function (Eq 4.) converts every element in a vector to a set of probabilities corresponding to each class. Cross-entropy (Eq 5.) returns the error between prediction and ground truth.

Training Process:

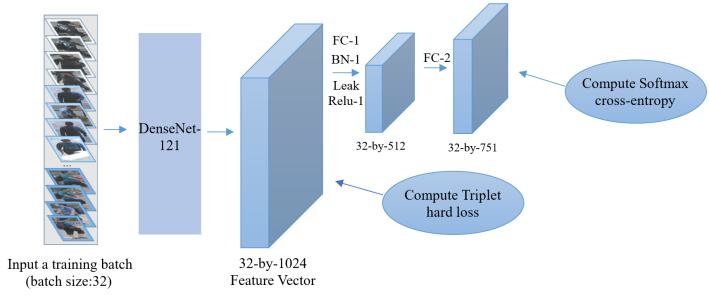

Fig. 5: The DenseNet neuron network architecture is used for learning how to extract human features properly. First, images are resized into 224-by-224. After feature extraction, we get a 32-by-1024 feature vector group which contains feature vector of images in the training batch. Then, the Triplet hard loss and Softmax cross-entropy loss are adopted jointly to optimize parameters. We also adopted fine-tuning (pre-trained on ImageNet), warm-up learning rate training techniques.

Label Assignment:

Now we get a CNN that is ready to extract features. Designing a robust label return function does matter.

Algorithm 1 Framework of ensemble learning for our system.

Require: We denote v as the feature vector of the query image. Each row in the feature gallery, which represents k features of the same person, is denoted as G_i. We introduce g_k, where $g_k \in G_i$, as the representation of the kth dimension of G_i, which also means one of the k features of a person. Note that $G_i \in G$, where $G = G_1 G_2 \dots G_i$ is the feature gallery. The $threshold$ is a variable that is pre-set. i represents the count of entities in the feature gallery.

Ensure:

v = CNN(img) // Extract features via CNN
if G is empty: //Initialize the feature gallery
  G = v * k //Replicate v for k times, making G
  //a 5-by-1024 matrix
  label = i
  i += 1
else:
  $min_{dist} = min(sum(Euclideandistance(v, G_i \in G)/k))$
  if $min_{dist} > threshold$ //Regard G_i as a new person
    G_i = v * k
    G = append(G, G_i) //Append G_i into feature gallery
    label = i
    i += 1
  else:
    label = $\arg\min_i \{sum(Euclideandistance(v, G_i \in G))/k\}$
    Randomly place one feature g_k in G_i with v
return label

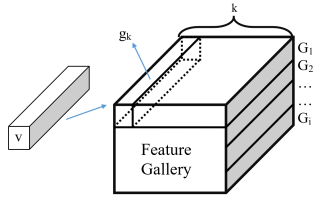

Fig. 6: Building the feature gallery

3.3 Generating Tracking Video

To generate a human tracking video, in which humans will be boxed with labels and the same person will have the same label, we integrate the former methods as below.

The input video will be processed frame by frame. For each frame, we use our modified darknet to detect its bounding boxes and deliver the human image into re-identification system.

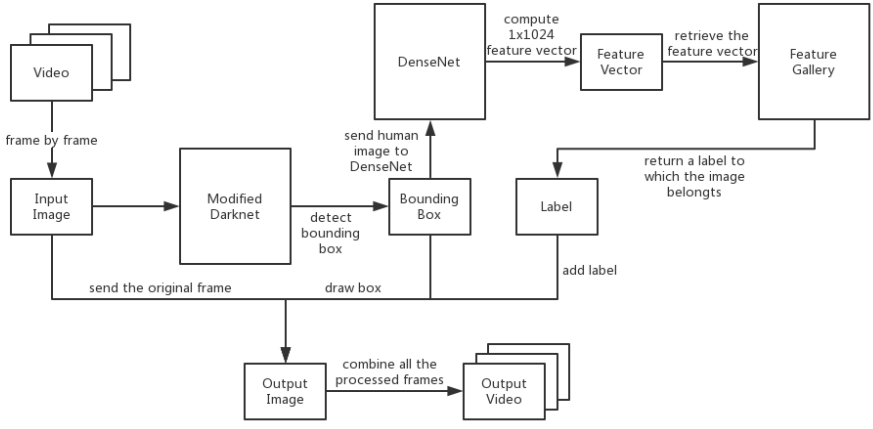

Fig. 7: Workflow of this project

Re-identification system uses DenseNet to compute the feature vector of this human image and retrieve the feature vector in our feature gallery. If current feature vector is close to one group in the gallery, the corresponding label of this group will be sent to detection system. Otherwise this feature vector will be added to the feature gallery. After receiving the label from re-identification system, detection system will draw the bounding box with corresponding label. As soon as all the frames in the video is processed, they will be connected to form a new video result.

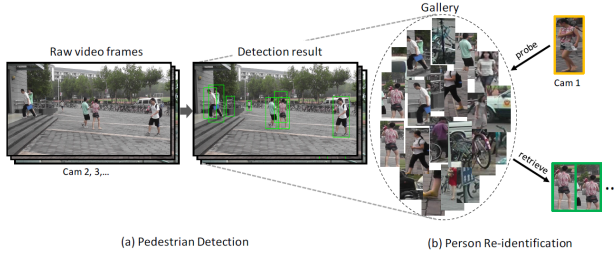

Fig. 8: Combination of human detection and re-identification

4 Results

4.1 Human Detection Fraction

Example Output:
The example output of modified YOLO model is shown as below:

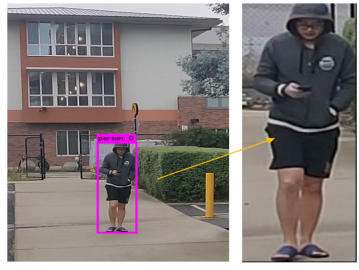

Fig. 9: Example human detection. A person is detected and given a bounding box. The bounded image is also cut and sent to ReID part.

In human detection part, our modified network is much faster but not as accurate as the other models. Our model is tested on COCO dataset and the running speed is tested on GTX2070 under darknet framework. Since our model cannot detect objects other than human and most other models only provide mean average precision, we choose to compare our average precision on humans with the other models' mean average precision.

Method	mAP-50	Speed(fps)
SSD321	45.4	16.4
R-FCN	51.9	11.8
SSD513	50.4	8
FPN FRCN	59.1	5.8
RetinaNet-101-800	57.5	5.1
YOLOv3-609	57.9	19.6
	AP-Person	
Ours	49.3	18.2

Table 1: Performances of different networks on COCO dataset

4.2 Human Re-identification Fraction

The ReID part result fraction is shown below:

Fig. 10: Example ReID output

Three models and a baseline model are listed in order to compare the performances. We trained and tested our model on RTX 2070 and market-1501 [9]. Note that all models adopted Adam optimization algorithm.

No.	Model name	Rank-1	Rank-5	mAP
1	DensNet-121+SCE (baseline)	86.4%	93.4%	65.4%
2	DensNet-121+Tri	88.2%	94.8%	70.3%
3	**DensNet-121+SCE+Tri**	**90.0%**	**96.4%**	**74.5% (0.049 secs)**
4	DensNet-161+SCE+Tri	91.4%	97.1%	77.8% **(0.065 secs)**

Table 2: SCE: Softmax cross-entropy loss. Tri: Triplet hard loss. For each human image, the cost of feature extraction for once using DenseNet-121 is 0.049s while DenseNet-161 is 0.065s. Although model 4 performs the best, however, to achieve real-time performance, we choose faster DensNet-121 (i.e. model 3) as our base neuron network, where the learning rate is 0.0003, batch size is 32, and trained for 230 epochs.

4.3 Final result

As shown in the DEMO presentation, the same person is 'remembered' by the model across different cameras, as shown below.

Fig. 11: Human tracking and re-identification based on video. Note that this is a capture of a single frame and is not exactly real-time, thus there would for sure exist 'laggy' phenomenon on the bounding box.

5 Conclusion and Discussion

5.1 Summary

In this project, we improved our indistrial-level deep learning skills in human detection and re-identification together with some tricks on param tuning. We also practiced our paper reviewing abilities, which is signifcant in our future academic careers. We felt the responsibility of teamwork and improved our communication skills. This is a meaningful project experience.

5.2 Limitation

Firstly, our model cannot distinguish multiple people in the same region. In the aspect of network structure, our modified model assumes there are at most two bounding boxes locating at one grid.

Secondly, our model finds it difficult to track the human who is either too close or too distant to the camera. The main causation is that our model cannot extract adequate feature from those humans.

5.3 Future Work

For improving our human tracking system, there are some potential future work concerning different aspects.

1. Our model can be trained on the other objects so that it can be used to trace other objects. For instance, it can be trained to detect and trace the motions of dogs or kangaroos in some area for scientific researches.
2. Temporal information can be analyzed in order to make more accurate prediction and re-identification based on the video. Recurrent neural networks, e.g. LSTM, are efficient in extracting temporal-spatial features of objects.
3. Combined with the 3-D reconstruction techniques or triangulation methods, our model can be adopted to draw one human's trajectory of motions in real world.

References

1. Geng, M., Wang, Y., Xiang, T., & Tian, Y. (2016). *Deep transfer learning for person re-identification*. arXiv preprint arXiv:1611.05244.
2. Cheng, D., Gong, Y., Zhou, S., Wang, J., & Zheng, N. (2016). *Person re-identification by multi-channel parts-based CNN with improved triplet loss function*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1335-1344).

3. Varior, R. R., Haloi, M., & Wang, G. (2016, October). *Gated siamese convolutional neural network architecture for human re-identification*. In European conference on computer vision (pp. 791-808). Springer, Cham.
4. Varior, R. R., Shuai, B., Lu, J., Xu, D., & Wang, G. (2016, October). *A siamese long short-term memory architecture for human re-identification*. In European conference on computer vision (pp. 135-153).
5. Zhao, H., Tian, M., Sun, S., Shao, J., Yan, J., Yi, S., ... & Tang, X. (2017). *Spindle net: Person re-identification with human body region guided feature decomposition and fusion.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1077-1085).
6. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You only look once: Unified, real-time object detection.* In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
7. Redmon, J., & Farhadi, A. (2018). *Yolov3: An incremental improvement.* arXiv preprint arXiv:1804.02767.
8. Hermans, A., Beyer, L., & Leibe, B. (2017). *In defense of the triplet loss for person re-identification.* arXiv preprint arXiv:1703.07737.
9. Zheng, L. , Shen, L. , Tian, L. , Wang, S. , Wang, J. , & Tian, Q. . (2015). *Scalable Person Re-identification: A Benchmark.* 2015 IEEE International Conference on Computer Vision (ICCV). IEEE Computer Society.

6 Peer Review

Group Member	Workload Distribution	Contribution Ratio
Xuwei Xu u6688636	YOLO coding and related document	26.6%
Weifan Jiang u6683698	ReID coding and related document	26.6%
Haozhan Sun u6688485	Paper writing.	26.6%
Hongming Zhang u6693651	Optimization and related works	15%
Guanyang Zhang u6688675	Photo taking	5%

Table 3: Peer review table