Journal of Korea Multimedia Society Vol. 20, No. 5, May 2017(pp. 769-781)
https://doi.org/10.9717/kmms.2017.20.5.769

Accurate Human Localization for Automatic Labelling of Human from Fisheye Images

Van Pha Than[†], Thanh Binh Nguyen[††], Sun-Tae Chung[†††]

ABSTRACT

Deep learning networks like Convolutional Neural Networks (CNNs) show successful performances in many computer vision applications such as image classification, object detection, and so on. For implementation of deep learning networks in embedded system with limited processing power and memory, deep learning network may need to be simplified. However, simplified deep learning network cannot learn every possible scene. One realistic strategy for embedded deep learning network is to construct a simplified deep learning network model optimized for the scene images of the installation place. Then, automatic training will be necessitated for commercialization. In this paper, as an intermediate step toward automatic training under fisheye camera environments, we study more precise human localization in fisheye images, and propose an accurate human localization method, Automatic Ground-Truth Labelling Method (AGTLM). AGTLM first localizes candidate human object bounding boxes by utilizing GoogLeNet-LSTM approach, and after reassurance process by GoogLeNet-based CNN network, finally refines them more correctly and precisely(tightly) by applying saliency object detection technique. The performance improvement of the proposed human localization method, AGTLM with respect to accuracy and tightness is shown through several experiments.

Key words: Human Localization; Fisheye Camera; CNN (Convolutional Neural Networks); GoogLeNet; Long Short Term Memory; Saliency Detection

1. INTRODUCTION

Recent successes of deep convolution neural network (CNN) in computer vision applications motivate researchers to enable CNN-based algorithms to be running on embedded systems [1].

As opposed to workstation computing environments with powerful processing capability and large memory, embedded systems usually have limitation on those. By now, it is well-known that in general, deeper layers (increased number of layers), wider layer (increased layer size) of CNN can learn more about different scenes so that it can perform better with respect to vision tasks (image classification, object detection, and etc.) for every possible scene. However, deeper and wider CNN requires more computational power and memory, which may not be allowed in many embedded systems. One realistic strategy for embedded CNN is to construct a simplified CNN network model optimized for the scene of the installation place. Since a simplified CNN network with reduced number of layers and narrower layers can learn enough about a specific scene (a scene about installation place) and processes fast, it may run in real-time with reasonably good performance on an

※ Corresponding Author : ,Sun-Tae Chung, Address: (06978) Dept. of Smart Systems Software, Soongsil Univ., 369, Sangdo-Ro, Dongjak-Gu, Seoul, Korea, TEL : +82-2-820-0638, FAX : +82-2-821-7653, E-mail : cst@ssu.ac.kr
Receipt date : Feb 15, 2017, Approval date : April. 28, 2017

[†] Dept. of Information and Telecommunication Engineering, Soongsil University
(E-mail : phatv@ssu.ac.kr)
[††] Embedded Vision, Inc., Seoul, South Korea
(E-mail : binh.nguyen@ieev.org)
[†††] Dept. of Smart Systems Software, Soongsil Uniersity
(E-mail : cst@ssu.ac.kr)

embedded system. However, as expected, the performance of such a simplified CNN drops significantly when it is tested against a new environment different from the training environment. For example, Fig. 1 illustrates this fact, which was obtained from experiments about our simplified YOLO model [2] which consists of 5 convolution layers, 4 max pooling layers followed by 2 fully connected layers, dropout layer and output detection layer.

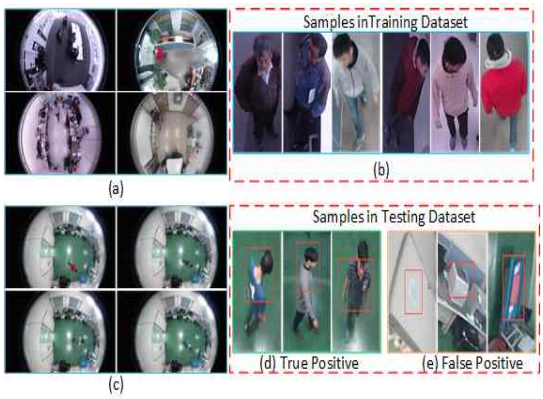

Fig. 1. (a) Images from training dataset, (b) Positive samples in training dataset, (c) Images from testing dataset, (d) Imprecise True Positives, (e) False Positives.

The YOLO simplified model [2] has been trained from omni-images of training datasets as shown in Fig. 1 (a) and (b). Here, omni-images means the images obtained from omni-directional cameras. When the YOLO simplified model was tested against omni-images obtained from a scene (Fig. 1 (c)) different from that of training, it performed poorly as shown in Fig. 1 (d) and (e). The lack of robustness of the YOLO simplified model with respect to human detection against different environments from training ones was shown more thoroughly in [2].

Then, an issue is how to train the simplified CNN model again appropriately for each installation place from image datasets which are collected from each installation place. Manual training for every installation is very costly and practically prohibitive.

In this paper, we propose a precise human localization method, AGTLM (Automatic Ground-Truth Labelling Method) under omni-directional camera environments for the purpose of automatic labelling humans in omni-images which are captured from omni-directional cameras. These labelled data can be utilized for reference ground-truth data in training CNN to learn human localization in an omni-image scene frame. It is critical that the ground-truth data set should not have incorrect data and had better have tighter and more precise data. Wrong ground-truth data (non-human objects) or not tight enough ground-truth data affects negatively on training so that they lead to yielding deteriorated performance of the simplified CNN network. One can see this fact in an experimental data of Table 2 in Section 4. Thus, the correct and tight object detection is much more important than missing objects.

The proposed human localization method, AGTLM consists of three stages; first localization of candidate human regions, second reassurance of the candidate regions, and third final refinement of human localization.

The GoogLeNet-LSTM approach adopted in the first stage generates candidate bounding boxes of humans in a scene. The second stage filters out false positives and can obtain more precise and tighter candidate human bounding boxes by applying the reassurance CNN processing. Finally, saliency object detection technique in the third stage refines bounding boxes of human localization resulting from the second stage processing.

The experimental results show that our AGTLM can generate the good ground-truth data for omni-images. The rest of the paper is organized as follows. Section II introduces technical backgrounds and related works necessary for understanding the contributions of the paper. Section III describes our proposed method and its implementation details. Experimental results are dis-

cussed in Section IV, and finally the conclusion is presented in Section V.

2. TECHNICAL BACKGROUNDS AND RELATED WORK

2.1 GoogLeNet

By now, it is well-known that deeper layers (increased number of layers) and wider layer (increased layer size) of CNN in general performs better. However increase the CNN size comes with two major drawbacks.

Bigger size typically means a larger number of parameters, which makes the expanded network more prone to overfitting, especially if the number of labeled examples in the training set is limited. Another drawback of uniformly increased network size is the dramatically increased use of computational resources. In order to solve two major drawbacks above, 'network-in-network' approach has been proposed by Lin et al. [5] which can increase the representational power of neural networks.

By the Inception architecture, a carefully crafted design based on network-in-network approach, GoogLeNet [4] designed by Google team increased the depth and width of the network while keeping the computational budget constant. GoogLeNet consisting of a 22 layers deep network, shows that on the ImageNet large-scale classification and detection challenges (ILSVRC 2014) [6], it significantly outperforms the current state of the art.

2.2 GoogLeNet–LSTM Approach to Object Detection

[3] proposed a combination of GoogLeNet and LSTM for object detection. Unlike traditional recursive neural networks, LSTM network is known to be well-suited to learn from experience to classify, process and predict systems with memory like time series when there are time lags of unknown size and bound between important events. For further detailed understanding, the reader needs to refer to [7].

In this paper, we will call the approach suggested in [3] as GoogLeNet-LSTM approach. Goog LeNet part of the GoogLeNet-LSTM approach encodes an image into high level descriptors via a convolutional architecture and LSTM part of GoogLeNet-LSTM approach decodes that representation into a set of bounding boxes of object.

Compared to the state-of-the-art methods, GoogLeNet-LSTM can handle to detect multiple persons, even overlapped persons under complex scenes and can detect objects with extremely small false alarm for not much complicated image data sets.

2.3 Saliency Detection via Graph–Based Manifold Ranking

The task of saliency detection is to identify the most important and informative part of a scene, and can be used to perform some difficult tasks such as generating bounding boxes [8], binary foreground and background segmentation, or saliency maps which indicate the saliency likelihood of each pixel. In this paper, salient object detection is adopted to refine bounding box of objects. Among many salient object detection methods, the one based on graph-based manifold ranking is known to be very competitive with respect to precision and recall and also with respect to processing speed [9].

Saliency detection via graph-base manifold ranking consists of two stages; first candidate saliency map obtained from 4 sides of an image problem and second final salient object detection coming from correction of the candidate saliency map by relevance computed via graph-based manifold ranking. It is known that even though the candidate saliency maps after the first stage are not precise in the Saliency detection via graph-base manifold ranking, salient object can be well detected by the saliency maps after the foreground queries in the second stage.

2.4 Related Works

For annotating(labelling) and evaluating video analysis such as object detection, and tracking, there have been developed some tools such as VATIC [10] , ViPER [11], LabelMe [12], and labelImg [13]. All these annotation tools are semi-automatic and need human intervention for manipulation.

As well-known by now, the performance of an object detector depends much on its training dataset and drops significantly when the detector is applied to a new scene. Even though most object detectors are learned with generic annotated datasets that are sampled from a large number of situations to cover the maximum variability of the object, a new scene can still have variations different from generic training dataset.

One of approaches to tackle this problem has been tried by proposing methods to transfer generic learning to scene-specific learning [14, 15, 16, 17]. The generically trained detectors can be used for extracting features from scene specific datasets and the extracted scene specific features are utilized for training detectors to be adapted for the scene [15]. Also, the generically trained classifiers can be fine-tuned for a specific scene from scene specific labelled dataset [16, 17]. All these efforts still need labelled dataset about the specific scene.

Some research works to automate the transfer learning have been reported in the literature [14, 18, 19]. All of [14, 18, 19] proposed an automatic adapting a generic classifier to specific scene iteratively. In [14], it starts with a generic pedestrian detector, which is applied to unlabeled samples in videos collected from the target scene. Based on detection results and context cues, some positive and negative samples from the target scene are automatically selected. Since the labels of the selected samples are predicted by detection scores and context cues, and could be wrong, their confidence scores are estimated. The selected samples and their confidence scores are used to retrain the scene-specific detector by transfer learning. The updated scene-specific detector is applied to the samples from the target scene again to select more samples for the next round of training. It repeats until convergence.

In similar spirits, [18] and [19] have dealt with transfer learning in the area of deep learning. [18] proposed a deep model to automatically learn scene-specific features and visual patterns in static video surveillance without any manual labels from the target scene. [19] proposed a novel approach to automatically specialize a generic pedestrian detector to specific scene by utilizing the sequential Monte Carlo filter and the Faster R-CNN deep model. In [20], at the first iteration, a generic classifier is used to predict a set of samples from the target dataset. Then, the update step determines the relevance of each sample by using an observation function. After that, the sampling step proposes the first specialized dataset from target and source samples. The process is the same at a different iteration, but the prediction step uses a specialized classifier, trained on a dataset built at the previous iteration, to propose new samples belonging to the target dataset.

In automatic adaptation of generic learning to scene-specific learning, [14, 18, 19] utilizes confidence concept for more reliable object prediction from unlabeled data sets.

In our problem setting, our targeted CNN for embedded systems is simplified and is not large enough to learn generic data sets. Thus, transfer learning may not be appropriate for our problem. In this paper, the research focus is concentrated on extracting a more reliable ground-truth human localization from the scene-specific data sets automatically so that the extracted ground-truth human localization labelled data can be utilized to retrain the embedded CNN model in the later stage.

3. THE PROPOSED AGTLM FOR PRECISE HUMAN LOCALIZATION

3.1 Outline of the Proposed AGLTM

Fig. 2 shows the workflow of our proposed AGTLM (Automatic Ground Truth Labelling Method). AGTLM consists of three stages; first stage finds out candidate human regions, second stage reassures the candidate regions, and final third stage refines human localization regions.

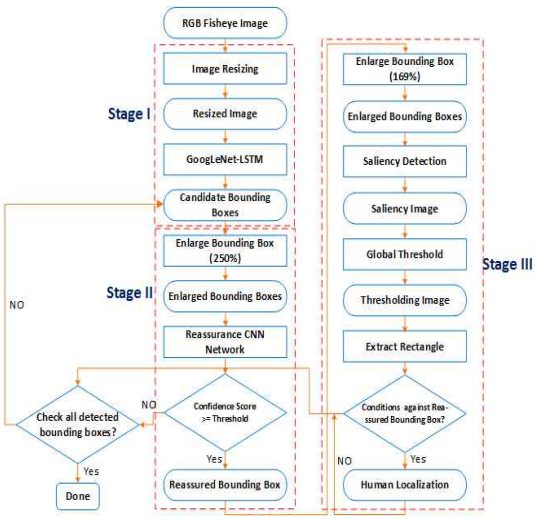

Fig. 2. The working flow of AGTLM for human localization.

In the first stage of the proposed AGTLM, GoogLeNet-LSTM approach [3] is adopted to predict candidate bounding boxes of humans. Since GoogLeNet-LSTM approach is not perfect, obviously GoogLeNet-LSTM approach predicts false positive bounding boxes or predicts bounding boxes which are not precise or tight enough for labelling purpose. Thus, in the second stage, additional reassurance CNN processing based on GoogLeNet [4] is applied to filter out false positives and to obtain more precise bounding boxes.

The second stage starts with enlarging the candidate bounding boxes resulting from the first stage to increase the possibility that the enlarged bounding boxes may include the missing human body parts not covered by the first stage bounding boxes. Next, the reassurance CNN processing is applied to the enlarged candidate region. Since the GoogLeNet-based reassurance CNN network is trained with human bounding box images without any cluttered background objects, its performance to detect human in the restricted region is observed to be very reliable. The reassurance CNN network can reliably either filter out the incorrect candidate regions which do not contain a human or produce more precise candidate bounding boxes. The reassurance CNN network produces a confidence score about the human detection in the enlarged candidate region and bounding boxes of humans. If the confidence score is less than the threshold, the candidate region will be discarded. Otherwise, the proposed method moves to the third stage.

The third stage begins with another enlarged region of the bounding box generated by the reassurance CNN network in the second stage. In the third stage, further localization refinement based on saliency detection is applied. The saliency region in the enlarged candidate region is extracted by applying saliency object detection technique and one calculates the minimum bounding box of the extracted saliency region. Finally, after the newly calculated bounding box is checked against the second stage bounding box and satisfies conditions about similar size and similar aspect ratio, then the bounding box obtained by the saliency process is determined to be a final tight bounding box. The applied saliency detection technique in the third stage utilizes graph-based manifold ranking [9] and some traditional image processing method such as global thresholding, and median filtering.

Fig. 3 demonstrates major processes in the three stages of the proposed AGTLM.

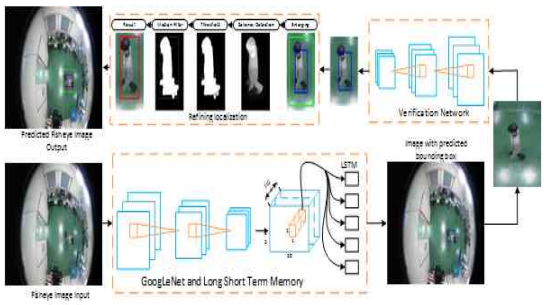

Fig. 3. Workflow and processes of the proposed AGTLM.

3.2 Human Localization based on GoogLeNet-LSTM approach

We construct GoogLeNet part of adopted Goog LeNet-LSTM approach to encode the fisheye images with 480×720 resolution into a 15×25 grid of 1024-dimensional top level GoogLeNet features. By changing into the 480x800 resolution. Each cell in the grid has a receptive field of size 139×139, and is trained to produce a set of distinct bounding boxes in the center 64×64 region. 300 distinct LSTM controllers are constructed to run in parallel, one for each 1×1×1024 cell of the grid. The LSTM units are designed with 250 memory states, no bias terms, and no output nonlinearities. At each step, the GoogLeNet features are concatenated with the output of the previous LSTM unit, and the result is fed into the next LSTM unit. The image is only fed into the first LSTM unit, indicating that multiple presentations of the image may not be necessary. Producing each region of the full 480x720 image in parallel gives an efficient batching of the decoding process.

Fig. 4 demonstrates GoogLeNet-LSTM processing in the first stage of the proposed method.

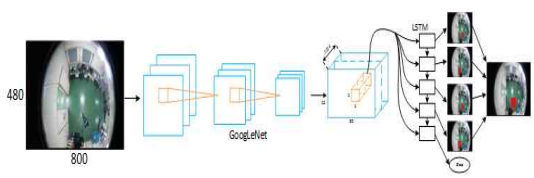

Fig. 4. GoogLeNet-LSTM processing in the first stage of the proposed method.

3.3 Reassurance CNN Network

It is reported in [3] that human detection by GoogLeNet-LSTM approach performs better compared to other human detection methods, but shows some not negligible false alarm rate and missing alarm rate. For ground-truth data generation purpose, missing alarm (missing human detection) is not so dangerous, but false alarm (non-human detection or imprecise human detection) is critical and avoidance of false alarm is strongly desirable. In order to filter out the false alarm, the proposed AGTLM applies reassurance process via another deep learning network based on GoogLeNet. The applied reassurance CNN network is constructed to produce a confidence score about human detectability for the candidate bounding box obtained from the first stage processing, and bounding box for the human object in the enlarged region of the candidate bounding box if the reassurance CNN network predicts a human region. The reassurance CNN network is trained with restricted images of 160×160 like the input image in Fig. 5. Thus, in the second stage processing, firstly the candidate bounding box image from the first stage is enlarged 2.5 times and it is rescaled into 160×160 size. Then, the rescaled image is input into the trained reassurance CNN network.

Fig. 5 illustrates the processing of the reassurance CNN network architecture.

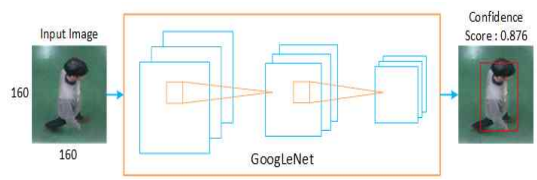

Fig. 5. Reassurance Process of the GoogLeNet-based deep CNN network.

3.4 Refining Human Localization

The predicted bounding boxes from GoogLeNet-LSTM in the first stage occasionally miss an important part of human body, and are not precise.

Yellow boxes in Fig. 6(a) show such cases. Or, the predicted bounding boxes are sometimes not tight enough compared to ones obtained manually by a ground-truth tool, which can be seen in yellow boxes in Fig. 6(b). Also, the predicted bounding boxes from the reassurance CNN network in the second stage shows similar cases which can be seen from blue boxes in Fig. 6.

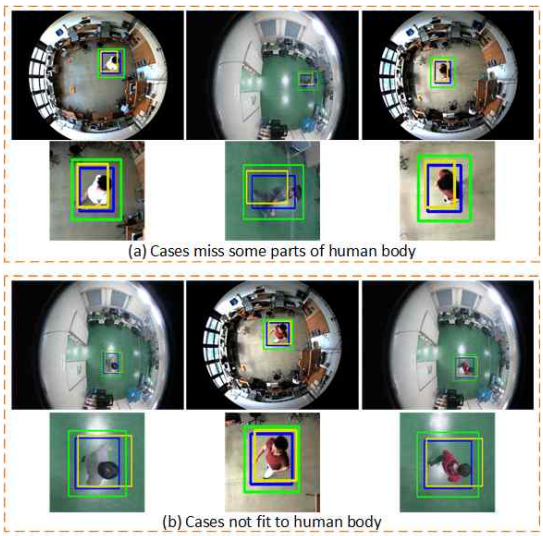

Fig. 6. Predicted Bounding boxes (yellow) from GoogLeNet-LSTM in the first stage, predicted bounding boxes (blue) from the reassurance CNN network in the second stage and enlarged bounding boxes (green) from the results in the second stage.

As we mention earlier, the predicted bounding boxes form the reassurance CNN network occasionally also miss a part of human body or are not tight enough. So we apply the refinement process (the third stage) to make the bounding boxes more correctly and tighter.

We start the third stage by enlarging the predicted bounding boxes from the reassurance CNN network up to 169% more, with same the purpose as in stage 2. We use the enlarged rectangle as input into the refining human localization process. Fig. 7 illustrates the refinement human localization process.

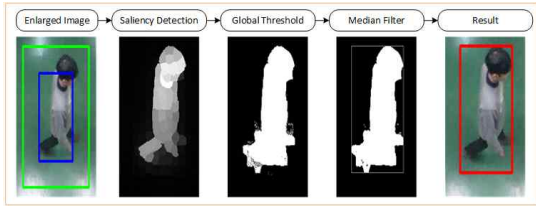

Fig. 7. Refinement Human Localization Process

In order to find the tight bounding boxes, we apply several processes, such as saliency detection via graph-based manifold ranking [9], global thresholding, and median filter. Sometimes, the refinement process leads to incorrect results as shown in Fig. 8 (c), (f) and (i).

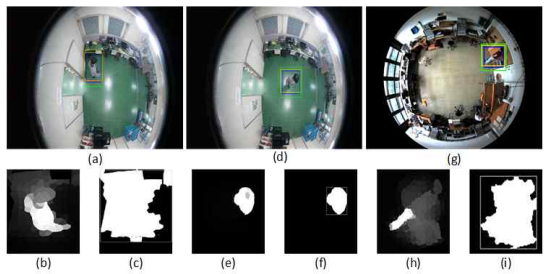

Fig. 8. Some incorrect results of refinement process: (c), (f), and (i)

To filter out the wrong results of refinement process, we compare the area of the final refined bounding box to a corresponding bounding box from the reassurance CNN network.

If the condition below is satisfied, we decide the refined bounding box is correct otherwise we discard it.

$$Th_L < \frac{Area\,(refined bounding box)}{Area\,(reassurance\ CNN's\ bounding box)} < Th_H \quad (1)$$

From our experiments we set the value of low thresholding and high thresholding to be 0.4, 1.4 respectively.

4. EXPERIMENTAL RESULTS

4.1 Experimental Environments

In order to evaluate our proposed solution, we

utilized two home-grown datasets, DB-A and DB-B. DB-A and DB-B which consists of images with 480×800 resolution, collected at offices of some companies in Seoul, South Korea. The DB-A contains 1500 Omni-images taken under 4 different indoor environments. And, DB-B contains 700 images consisting of images taken under 2 environments which are different from DB-A's environment and some images from Bomni DB [24]. In Fig. 9 shows some sample images of DB-A and DB-B.

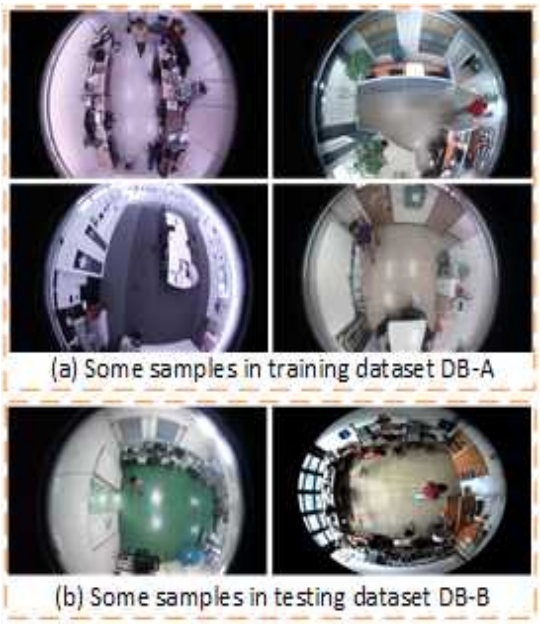

Fig. 9. Sample images in DB-A and DB-B.

To evaluate the performance of our proposed AGTLM, we utilized the simplified CNN network [2]. The simplified CNN network is obtained by simplifying and optimizing YOLO model so as to be able to run on our fisheye camera's DSP.

4.2 Evaluation Methodology

For detection measure, we adopt IOU (Intersection Over Union) as PASCAL measure in [23]. IOU is defined as follows.

$$IOU := \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \quad (2)$$

Where Bp and Bgt means the detected bounding box, and the ground-truth bounding box, respectively. If IOU > 50%, we decide that the detected object is correctly detected.

Next, if we denote the number of correctly detected human objects (true positives), the number of falsely detected human objects (false positives), and the number of missed human objects (false negatives) as TP, FP, and FN, respectively. Then, precision is defined to be the percentage of detected true positives compared to the total number of items detected by the method.

$$\mathrm{Precision} := \frac{TP}{TP+FP} \quad (3)$$

Recall means the percentage of detected true positives compared to the total number of true positives in the ground truth.

$$Recall := \frac{TP}{TP+FN} \quad (4)$$

Actually, Precision = 1- false alarm rate, and Recall = 1- miss rate.

4.3 Experimental Results

4.3.1 Robustness of the simplified CNN network

In order to see the performance of the simplified YOLO CNN network [2] drops considerably when it is applied to new scenes, we trained the simplified YOLO and Tiny YOLO with DB-A and tested the trained ones with DB-B.

Experimental data in Table 1 shows that the simplified YOLO network is not robust with respect to new environments (different backgrounds, different illumination, and so on) and needs to be retrained for each scene. But, manual labelling of new training datasets for retraining costs expensive. Thus, need for development of automatic labelling tool may follows. Table 1. Testing results of applying the simplified YOLO and Tiny YOLO against the dataset, DB-B, which has different environment from the training data set, DB-A.

Table 1. Testing results of applying the simplified YOLO and Tiny YOLO against the dataset, DB-B, which has different environment from the training data set, DB-A

Simplified YOLO	11.65	14.04
Tiny YOLO	68.25	72.16

4.3.2 The effect of the tightness of ground-truth data on the performance of CNN network

To understand why one may need tighter ground-truth bounding-box data for better performance of the CNN network, we train the simplified YOLO network model [2] by two kinds of ground-truth data, separately; tight ones (red rectangles in Fig. 10) and not tight ones (green rectangles in Fig. 10). Those ground-truth data are obtained from DB-A by using our manual ground-truth tool.

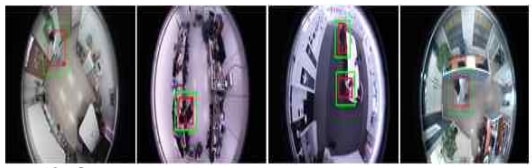

Fig. 10. Some Images from DB-A. Not tight ground truth-green rectangle, precise ground truth-red rectangle.

Then, we tested two trained simplified YOLO network against DB-B, and obtained experimental data of Table 2. The experimental data in Table 2 shows that training by tight bounding box ground-truth data is important for better accuracy performance.

Table 2. Accuracy comparison with respect to loose and tight ground-truth bounding box data

Training Ground Truth Data	Simplified YOLO	
	Precision	Recall
Not Tight Bounding Box Data	67.35%	78.25%
Tight Bounding Box Data	92.68%	94.35%

4.3.3 Accuracy of The Proposed AGTLM

Through experiments about DB-A and DB-B, it is verified that the proposed AGLTM generates more tighter bounding-boxes compared to GoogLeNet-LSTM approach (first stage) and reassurance CNN network (second stage). First, Fig. 11 shows that AGTLM generates the bounding boxes more precisely and tightly than GoogLeNet-LSTM and the reassurance CNN network by illustrating and comparing bounding boxes (1^{st}stage; yellow-ones, 2^{nd}stage; blueones, and 3^{rd}stage; red ones) predicted from the first stage, the second stage, and the final stage of the proposed AGLTM, respectively.

Fig. 12 illustrates again the effectiveness of the proposed AGTLM in multiple person images.

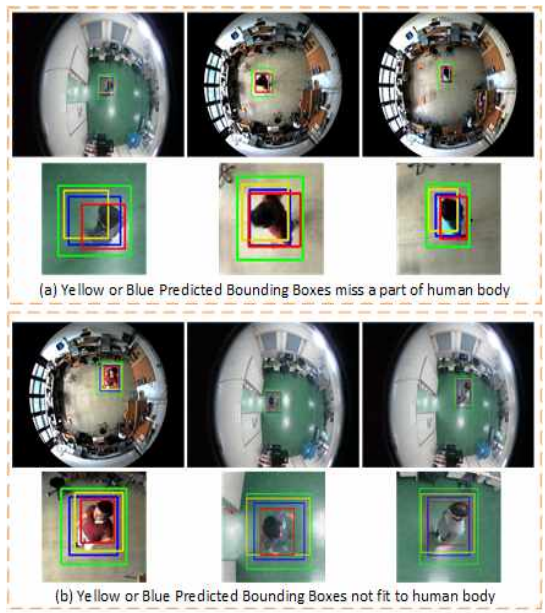

Fig. 11. Predicted Bounding boxes (yellow) from GoogLe Net-LSTM in the first stage, predicted bounding boxes (blue) from the reassurance CNN network in the second stage and enlarged bounding boxes (green) of the second stage bounding boxes, and the final bounding boxes (red) refined in the final third stage.

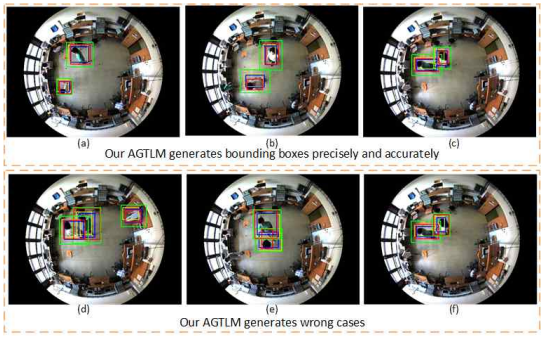

Fig. 12. Predicted Bounding boxes (yellow) from GoogLe Net-LSTM in the first stage, predicted bounding boxes (blue) from the reassurance CNN network in the second stage and enlarged bounding boxes (green) from the results in the second stage, refined bounding boxes (red)

Since the enlarged bounding boxes are expanded based on the predicted bounding boxes from the second stage (reassures the candidate regions), if the enlarged bounding boxes overlap with the others a lot or have complex background, the proposed AGTLM may generate the unreliable results, as shown in Fig. 12 (d), (e), and (f). All these wrong cases will be filtered out by applying the condition (1) in Section 3.4. When bounding boxes are separate or overlaps a little, then it is observed that the proposed AGTLM works well as shown in Fig. 12 (a), (b), and (c).

Table 3 obtained by quantitative analysis on DB-B shows that the proposed AGTLM can filter out false alarm cases (non-human objects) and correct imprecisely detected human objects. Goog LeNet-LSTM in the first stage may pass false alarms, and reassurance network in the second stage filters out some false alarm by checking confidence scores, and finally the condition (1) in the third stage filters out some false alarms survived from the second stage. Saliency processes in the third stage, especially global thresholding process may generate worsen bounding boxes which are supposed to be filtered out by condition (1). On the other hand, saliency process applied to the enlarged region of the bounding box of the 2^{nd} stage may correct the second stage's imprecise bounding boxes. Anyway, filtering out may increase missing of human objects, which leads to decreased recall performance as shown in Table 3. However, as explained before in Introduction, missing is not critical for ground-truth data generation purpose.

Table 3. Comparison with respect to accuracy

Method Name	*Precision (%)*	*Recall (%)*
GoogLeNet-LSTM	87.27	93.69
Reassurance CNN network	90.82	94.03
Our Proposed AGTLM	98.44	92.23

The performance with respect to tightness of the proposed AGTLM is seen by experimental data Table 4, which was obtained on 673 images from DB-B which have survived as true positives after the second stage. IOUs was calculated between the ground-truth bounding boxes and each ones among three bounding box classes; ones after GoogLe Net-LSTM in the first stage, or ones after reassurance CNN network in the second stage, or ones after the final third stage. Data in Table 4 is average IOU computed from IOUs for each class.

Table 4. Comparison of Tightness Performance by average IOU

Method Name	IOU Rate (%)
GoogLeNet-LSTM	76.06
Reassurance CNN network	86.23
The Proposed AGTLM	89.31

4.3.4 The Effectiveness of training ground-truth data by the Proposed AGTLM

To see that the more precise and tighter bounding box ground-truth data generated by the proposed AGTLM is more effective for training simplified YOLO network, we trained the simplified YOLO network by the four different kinds of ground-truth data; one by GoogLeNet-LSTM, one by reassurance CNN network, one by the proposed AGTLM method, and one by manual tool, respectively. All these ground-truth data are ex-

tracted from the same 400 images which are randomly chosen among 673 images in DB-B which have been detected as true positive by AGTLM.

Then, we tested four differently trained simplified YOLO networks with the remaining 300 images in DB-B. Table 5 shows the testing results.

Table 5. Comparison the accuracy of simplified YOLO with different ground-truth data

Training Ground-Truth Dataset	Simplified YOLO	
	Precision (%)	*Recall (%)*
From Manual Labelling Tool	95.71	98.14
From GoogLeNet-LSTM	88.86	95.29
From Reassurance CNN network	92.15	96.31
From the Proposed AGTLM	94.14	97.43

Actually, manual labelling tool generates ground-truth data with 100% IOU as opposed to the automatic labelling tool, GoogLeNet-LSTM and the proposed AGTLM.

Experimental data in Table 5 shows that more precise and tighter ground-truth bounding box data would train CNN networks better so as to generate better testing result.

The experiment results in Table 3, Table 4 and Table 5 show that our proposed AGTLM generates the ground truth more correctly and tightly compared to GoogLeNet-LSTM method and can produce better testing result.

5. CONCLUSIONS

In this paper, we presented an accurate human localization method for automatic labelling of human from fisheye Images. The proposed human localization method, AGTLM, improves the pre-existing good performing human localization method, GoogLeNet-LSTM, by applying further reassurance process based on GoogLeNet, and refining process based on saliency detection. From experiments, it was shown that the proposed human localization method generates more precise and tighter human object bounding boxes. And, it is also observed the more precise and tighter ground-truth data is more effective to train CNN with respect to accuracy performance, at least for the simple CNN networks.

REFERENCES

[1] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, et al., "Recent Advances in Convolutional Neural Networks," *arXiv:1512.07108*, 2017.

[2] N.T. Binh, N.V. Tuan, and S.T. Chung, "Real-time Human Detection under Omni-directional Camera based on CNN with Unified Detection and AGMM for Visual Surveillance," *Journal of Korea Multimedia Society*, Vol. 19, No. 8, pp. 1345-1360, 2016.

[3] R. Stewart and M. Andriluka, "End-to-end People Detection in Crowded Scenes," *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2325-2333, 2016.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al, "Going Deeper with Convolutions," *Proceeding of Computer Vision and Pattern Recognition (CVPR)*, pp. 1-9, 2015.

[5] M. Lin, Q. Chen, and S. Yan, "Network in Network," *arXiv:1312.4400*, 2013.

[6] O. Russakovsky, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, Vol. 115, No. 3, pp. 211-252, 2015.

[7] C. Olah, Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed Feb., 14, 2017).

[8] K.Y. Chang, T.L. Liu, H.T. Chen, and S.H. Lai, "Fusing Generic Objectless and Visual Saliency for Salient Object Detection," *Proceeding of International Conference on*

Computer Vision, pp. 914-921, 2011.

[9] C. Yang, L. Zhang, H. Lu, X. Ruan, and M. Yang, "Saliency Detection via Graph-Based Manifold Ranking," *Proceedings of IEEE Conferenceon Computer Vision and Pattern Recognition,* pp. 3166-3173, 2013.

[10] VATIC: Video Annotation Tool from Irvine, California, http://web.mit.edu/vondrick/vatic/ (accessed Feb., 14, 2017).

[11] ViPER: The Video Performance Evaluation Resource, http://viper-toolkit.sourceforge.net (accessed Feb., 14, 2017).

[12] LabelMe, http://labelme.csail.mit.edu /Release 3.0/ (accessed Feb., 14, 2017).

[13] LabelImg, https://github.com/tzutalin/labelImg (accessed Feb., 14, 2017).

[14] X. Wang, M. Wang, and W. Li, "Scene-Specific Pedestrian Detection for Static Video Surveillance," *IEEE Transactionson Pattern Analysis and Machine Intelligence*, Vol. 36, No. 2, pp. 361-374, 2014.

[15] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How Transferable are Features in Deep Neural Networks?," *Advances in Neural Information Processing Systems 27*, pp. 3320-3328, 2014.

[16] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the Devil in the Details: Delving Deep into Convolutional Nets," *Proceedings of British Machine Vision Conference*, pp. 18-19, 2014.

[17] D. Xing, W. Dai, G.R. Xue, and Y. Yu, "Bridged Refinement for Transfer Learning," *Proceeding of European Conference on Principles and Practice of Knowledge Discovery in Databases, Lecture Notes in Computer Science*, pp. 324-335, 2007.

[18] X. Zeng, W. Ouyang, and M. Wang, "Deep Learning of Scene-Specific Classifier for Pedestrian Detection," *Proceeding of Europe an Conference on Computer Vision*, pp 472-487, 2014.

[19] A. Mhalla, T. Chateau, and S. Gazzah, "Scene-Specific Pedestrian Detector Using Monte Carlo Framework and Faster R-CNN Deep Model," *Proceeding of International Conference on Distributed Smart Camera*, pp. 228-229, 2016.

[20] H. Maâmatou, T. Chateau, S. Gazzah, Y. Goyat, and N. Essoukri Ben Amara, "Transductive Transfer Learning to Specialize a Generic Classifier Towards a Specific Scene," *Proceeding of International Conference on Computer Vision Theory and Applications*, pp. 411-422, 2016.

[21] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, et al., "Learning to Detect a Salient Object," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 353-367, 2011.

[22] L. Wang, J. Xue, N. Zheng, and G. Hua, "Automatic Salient Object Extraction with Contextual Cue," *Proceeding of International Conference on Computer Vision*, pp.105-112, 2011.

[23] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of th State of the Art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, Issue 4, pp. 743-761, 2011.

[24] Bomni-DB Hompage, https://www.cmpe.boun.edu.tr/pilab/pilabfiles/databases/bomni/ (accessed Feb., 14, 2017).

Van Pha Than

He received the Engineer degree in electronics and computer engineering from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 2013. He is currently a research assistant at Embedded Real-time Computing Laboratory, Soongsil University, Seoul, South Korea. His main areas of research interest are embedded systems, image processing, visual surveillance, recognition systems, deep learning.

Thanh Binh Nguyen

He received the B. Eng. degree in computer science from the University of Science, Ho Chi Minh, Vietnam, in 2005, the M. Sc. degree in information and telecommunication engineering from the University of SoongSil, Seoul, South Korea, in 2010, and the Ph.D. degree in engineering at the University of SoongSil, Seoul, South Korea, in 2017. He is currently a principal software R&D researcher at Embedded Vision Inc., Seoul, South Korea, assistant at Embedded Real- time Computing Lab, University of Soongsil, Seoul, South Korea. His research interests cover the design and analysis of various smart embedded software system, I.O.T and also intelligent image, video analytic algorithms which is applied to visual surveillance, recognition systems, and etc.

Sun-Tae Chung

He received B.E. degree from Seoul National University, and M.S. degree and Ph.D. degree in Electrical Eng. and Computer Science from the University of Michigan, Ann Arbor, USA, in 1986 and 1990, respectively. Since 1991, he had been with the School of Electronic Eng. at the Soongsil university, Seoul, Korea where he is now a full professor. Now, he has been with the Dept. of Smart Systems Software, at the Soongsil Univ. since 2015. His research interests include: computer vision, visual surveillance, biometrics, and embedded systems.