Fast Human Head and Shoulder Detection Using Convolutional Networks and RGBD Data

Wassim A. El Ahmar, Farzan Erlik Nowruzi, Robert Laganiere
Department of Electrical Engineering and Computer Science, University of Ottawa
Ottawa, Ontario, Canada
welahmar@uottawa.ca, fnowr010@uottawa.ca, laganier@eecs.uottawa.ca

Abstract

We introduce a new real-time approach for human head and shoulder detection from RGB-D data based on a combination of image processing and deep learning approaches. Candidate head-top locations (CHL) are generated from a fast and accurate image processing algorithm that operates on depth data. We propose enhancements to the CHL algorithm making it three times faster. Various deep learning models are then evaluated for the tasks of classification and detection on the candidate head-top locations to regress the head bounding boxes and detect shoulder keypoints. We propose three different models based on convolutional neural networks for this problem. Experimental results for different architectures of our model are discussed. We also compare the performance of our models to other state of the art methods in terms of accuracy of detections and computational cost and show that our proposed models are on par with the state of the art in terms of precision-recall of head detection and precision of shoulders detection, with the biggest advantage of our models being in terms of computation time. We also analyze the effect of adding the depth channel on the performance of the network.

1. Introduction

Deep learning has dominated the field of machine vision in the last few years. The advancements in GPU technology has allowed deep learning methods to become more feasible as the computing power of GPU can be harnessed to train and run deep learning models. The accurate detection of humans in videos is a prominent challenge in machine vision with several applications ranging from pedestrian detection in autonomous driving systems, to security and congestion analysis in surveillance videos. There are several approaches to human detection depending on the type of application.

Some systems only need to detect the head bounding boxes of humans in an image as head detection is sufficient to achieve the desired task. Applications such as people counting [5] and congestion analysis [6] are examples of such systems.

Other systems work on detecting bounding boxes of the entire human body in images. Pedestrian detection for autonomous cars [11], person tracking and reidentification [20] are examples of systems requiring the detection of the full body of humans.

Human pose estimation is the process of detecting keypoint locations that define important joints of the human body such as the shoulders, elbows, hips, and knees. Pose estimation is used in applications relating to activity recognition [10] and action prediction [13].

1.1. Challenges

Even with the great advancements achieved in the field of human detection, certain challenges of object and keypoint detection still exist. The main two challenges are:

- Occlusion: It is difficult for a model to detect objects that are fully or partially occluded. This is specially true for congested scenes like a crowded street.
- Feature broadness: With deep learning, the performance of a model is reliant on the size of the training data. The more examples a model trains on, the better it is expected to be in predicting test images. However, generating training data requires manual labour for data annotations.

1.2. Contribution

We developed a system that is able to efficiently and reliably detect humans in indoor environments in real time given limited computational resources. Our contributions can be summarized as follows:

- We propose a system that combines traditional image processing techniques to generate object proposals with deep learning. The proposed model efficiently

and reliably detects human heads and shoulders in indoor environments. Our approach addresses occlusion challenges and does not require a large amount of training data.

- We introduce enhancements to the CHL algorithm [16] that reduces its computational complexity by three folds.

- We introduce three variations of our model and compare our approach to other state of the art solutions in the tasks of object and shoulder keypoint detection and show that our model is more computationally efficient while still performing well on the tasks of head and shoulders detection.

The remainder of this paper is organized as follows. Section 2 provides a review of the previous methods related to our task. We introduce our approach in section 3. The experimentation results and analysis are given in section 4. Finally, we conclude this paper in section 5.

2. Related Work

Spinello et al[14] propose the histogram of oriented depth (HOD) descriptor which encodes the orientation of depth changes from depth frames that is then stored in a 1D histogram and the aggregation of blocks of those histograms result in the HOD features.

In [8], the authors use local maximum height pixels as plausible human head tops. A SVM (Support Vector Machine) [4] is then trained with two sources of features (height difference, and joint histogram of color and height)

SSD (Single Shot multibox Detector) [9] generates proposals and classifies them in one network pass (single shot) making it faster than Faster-RCNN [12]. Convolutional feature layers are added to the end of the base network that decrease in size to allow the detection of objects of varying sizes. Default boxes of varying sizes and aspect ratios are used to detect objects of different shapes and sizes.

Introduced in [17], selective search relies on heavy image segmentation to generate bounding boxes around segmented partitions representing object proposals. Segmented partitions are grouped with their neighbours based on similarity and bounding boxes are then created around the newly merged partitions to generate bigger object proposals for bigger objects. The process of combining neighbouring partitions to create bigger proposals continues until we end up with one proposal around the entire image.

Tian et al[16] scan a depth image from the top-left corner to the bottom right corner, projecting every pixel to the 3D-plane and connecting pixels that have a Eucledian distance less than a threshold allowing for the fast generation of connected regions in an image. The top pixel of every connected region is taken as a candidate human head-top location.

Zhang et al[19] propose the use of a cascade of three CNN (Convolutional Neural Networks) for the task of facial landmark (keypoint) detection. Each stage refines the predictions of the previous stage as a fully connected layer outputs the coordinates of facial keypoint locations.

OpenPose [3] is a real-time method for pose estimation proposed by Cao et al. The method consists of 2 stages. The first stage consists of a feature extractor while the second stage is composed of 2-branch multi-stage CNN that generate confidence maps and part affinity fields.

3. Proposed Method

Our approach is composed of two stages. The first stage involves the generation of object proposals using an improved implementation of the CHL algorithm [16]. In the second stage, generated object proposals are fed to a small CNN that classifies them as heads or background. For proposals classified as heads, the regressed head bounding box coordinates and shoulder keypoint locations are generated. Figure 1 shows an overview of our proposed system.

3.1. Generating Proposals

The CHL algorithm processes a depth image to generate candidate head-top locations. The image is scanned from the top left corner to the bottom right corner. The idea is that pixels that belong to the same object would have continuous depth values and vary in a specific range. Each pixel is projected to the 3D space and the euclidean distance between the pixel and its neighbouring pixels is calculated. If the distance is within a certain range, the connected region is expanded and the neighbours of the newly added pixel are processed next to check if they also belong to the same object. This continues until no more pixels are found that belong to this object. Then, the next pixel in the queue is processed to start searching for a new object until no more pixels are left unprocessed. This generates a number of connected regions. The top point (having the lowest y coordinate) of every accepted region is chosen as the candidate head top location. After generating the candidate head top locations, bounding boxes should be generated from them in order to be fed to the CNN. In the original CHL algorithm, the bounding box is a rectangle of aspect ratio $3:4$. For our application, we found that the bounding box needed to have a bigger width in order to assure that it contains the shoulder keypoints. Hence, the bounding box generation algorithm is altered to output square shaped objects. There is a linear relationship between the size of the proposal and the distance to the sensor. As the object is further away from the sensor, the proposal box length is smaller.

We have introduced a few changes to the CHL algorithm that allowed us to improve its speed by 3 times. The imple-

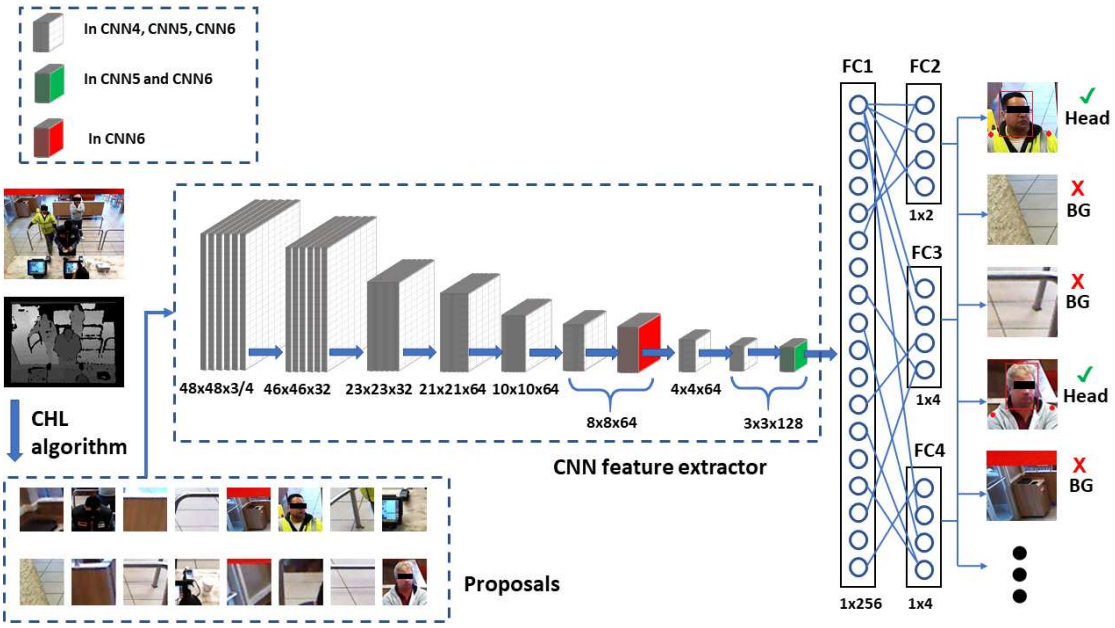

Figure 1: Complete overview of our system.

mentation of the CHL algorithm in [16] requires two passes over all image pixels. In the first pass, the (x, y, $depth$) values of a pixel are used to generate a pointcloud. In the second pass, the 3D euclidean distance between neighbouring pixels in the generated pointcloud is calculated and connected regions are generated accordingly. In our implementation, we combine the processes of 3D euclidean distance calculation and connected regions generation. Instead of going through all image pixels in the first pass and projecting them to the 3D space, then go over the generated pointcloud again to generate connected regions, we analyze the image pixels only once where pixels being processed are projected to the 3D space and their Eucledian distance is calculated. Hence, although we are still projecting the pixels to the 3D space, the access to pixel values is happening once instead of twice, significantly reducing computation time. In addition, we do not need to store the generated pointcloud in memory. In addition, we use pointer access to retrieve the value of depth pixels at specific locations. Finally, we use Cython [2] to create a bridge between Python and C to remove the overhead caused by the dynamic binding in Python. The enhanced CHL algorithm functions at 115 frames per second.

3.2. Feature Extractors

Since we already have proposals, there is no need to use a complex, deep CNN. We propose three slightly different CNNs for classifying head proposals and regressing bounding boxes and shoulder locations. We employ multi-task learning (multi-task: head classification, head bbox detection, shoulder location regression). The 3 architectures are different variations of the proposed system and would work independently of each other. The 3 architectures will be referred to in the remaining of this paper as CNN4, CNN5, and CNN6. The 3 models share a similar architecture that is composed of convolutional and pooling layers where ReLU (Rectified Linear Unit) is used as the activation function. The output of the last convolutional layer is flattened and input to a fully connected layer (FC1). 3 different fully connected layers (FC2, FC3, and FC4) are connected to FC1 and have the following functionalities:

- FC2: Has 2 outputs, used for binary classification of the input. It outputs 1 if the proposal is a head, and 0 otherwise.
- FC3: Has 4 outputs X_{min}, Y_{min}, X_{max}, Y_{max}, the bounding box coordinates of the head location in the proposal.
- FC4: Has 4 outputs X_{LS}, Y_{LS}, X_{RS}, Y_{RS}, the coordinates of the left and right shoulders of the person in the proposal.

3.3. Dataset

The dataset we use was taken in four different indoor locations at a restaurant. Every captured frame has an RGB image and a corresponding depth image captured using

an Orbbec Astra[1] 3D camera. In total, we captured 1610 images from which 1449 images were randomly selected for training, and 162 were selected for testing. However, since the input to our models is proposals (resized to a size of 48×48), the number of samples used for testing is the total number of heads in all test images, and the same applies for training images. This is because our system trains on samples containing a human head and shoulder, so the total number of train and test data is the total number of humans present in the train/test set. Thereby, the total number of training samples is 3212 positive samples and 28921 negative samples, and the total number of test samples is 357 positive samples and 3361 negative samples. Data augmentation (rotation) is applied to increase the size of the positive training samples to 16060. The dataset can be downloaded from our website `http://www.site.uottawa.ca/research/viva/projects/head_shoulder_detection/index.html`, and the complete thesis from this project is available at `https://ruor.uottawa.ca/handle/10393/39448`.

3.4. Training

The training was carried out on a machine with NVIDIA GeForce GTX 1080 Ti Graphics card[2]. Binary cross entropy loss is used for FC2 that classifies proposals, while FC2 and FC3 that respectively regress the head locations and shoulder keypoints within proposals use mean squarred error as loss function. The training was conducted using a batch size of 200, with an initial learning rate of 0.001 that decays with a factor of 0.1 at 500, 1000, and 1200 epochs. The model is trained for 2000 epochs. The source code for our work is available online at `https://github.com/wassimea/hs_detection_cnn/`.

4. Results and Analysis

4.1. Head Detection

When measuring the efficiency of a method in object detection, it is important to take both precision and recall into consideration. To generate the precision-recall curve, we calculate the precision and recall of different models on different confidence thresholds ranging from 0.05 to 0.9. As we increase the confidence of a model, we are decreasing the recall but increasing the precision, and vice versa.

IoU (Intersection over Union) is a metric used to evaluate the accuracy of a detection compared to its ground truth box. It is calculated by dividing the overlapping area of the detected box and the ground truth box by the union of the areas of the two boxes (Equation 1). The conventional minimum IoU threshold to consider a human detection to be correct is 0.5.

$$IOU = \frac{Areaof Intersection}{Areaof Union} \quad (1)$$

In our experimentation, two precision-recall curves are calculated:

- Precision recall curve of our models compared to the state of the art methods for object detection. In these experiments, we set the IoU threshold to 0.5 in order to consider a detection to be correct. These results are shown in Figure 2.

- Precision recall curve of our models compared to Mobilenet trained on the DMH (Depth map, Multi-order depth template, and Height difference map) representation proposed in [16]. In these experiments, the proposal box is evaluated instead of the regressed detection that is output by our models. Those experiments are necessary to compare with the work done in [16] that does not regress the head bounding box. It only classifies a proposed box as either a head or background based on a confidence threshold. Hence, we disregard the IoU of the proposal box with the ground truth similar to what the authors of [16] do. If a proposal box classified as a head contains a ground truth head, the proposal box is considered a true positive, regardless of the IoU value, and it is considered a false positive otherwise. A proposal box is considered to contain a ground truth if more than 50% of the ground truth box is contained in the proposal box. The results of those experiments are shown in Figure 3 and Figure 4.

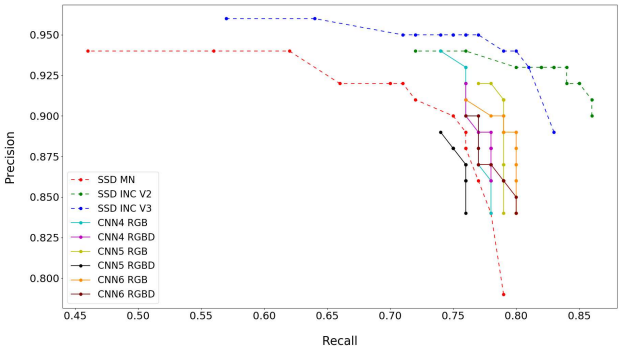

Figure 2: Precision-Recall curve applying 0.5 IoU criterion using confidence thresholds between 0.05 and 0.9

Figure 2 shows that the best method in terms of precision-recall is SSD-Inception-V3 [15] followed by SSD-Inception-V2 [15]. This was expected as the inception networks are among the best feature extractors. However,

[1] `https://orbbec3d.com/product-astra/`

[2] `https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1080-ti/`

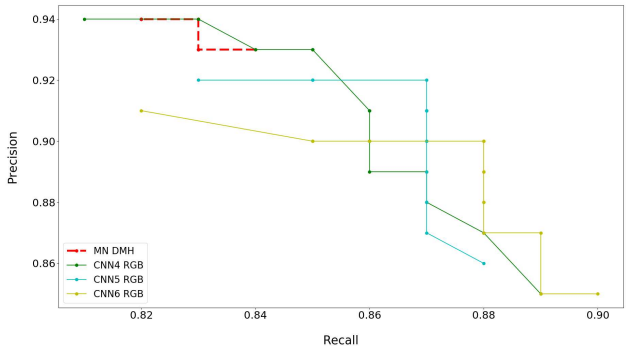

Figure 3: Precision-Recall curve using the ground-truth inclusion criterion using confidence thresholds between 0.05 and 0.9 for our RGB models

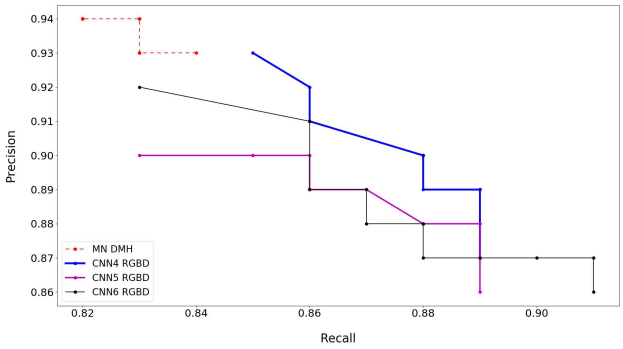

Figure 4: Precision-Recall curve using the ground-truth inclusion criterion using confidence thresholds between 0.05 and 0.9 for our RGBD models

this impressive performance comes at a hefty computational cost. This is elaborated in section 4.4. We see that our models perform relatively better than SSD Mobilenet [7], which is the fastest feature extractor usually used as a backbone for SSD.

The figure shows that our models that receive a RGB input perform better than the models that operate on RGBD input. This can be explained by the depth channel adding confusion to the head classification task. The depth channel gives information that allows the differentiation of objects from background due to the distance disparity between an object and the background. While this differentiation proves to be beneficial for the precision of shoulder detection (discussed in section 4.2), it adds confusion to the classification task as depth does not give information that helps in differentiating a human head from other objects.

We also observe that deeper models perform better. CNN6 achieves the best recall (0.8) at the same precision (0.88) of CNN5 and CNN4 (recalls of 0.79 and 0.77 respectively). This is also an expected result as deeper models extract better features that lead to better classification and bounding box detection accuracy.

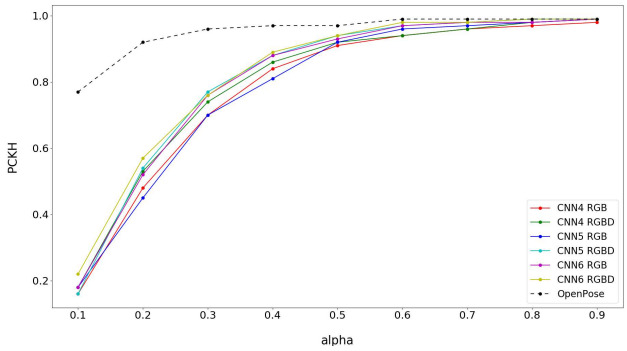

Figure 5: PCKh of shoulder detection of different tested models

Figure 3 and Figure 4 respectively compare the performance of our RGB and RGBD models with Mobilenet trained on the 3 channel representation proposed in [16] (DMH), which is generated solely from the depth image. Our experimental results show that [16] achieves high precision, but its recall is lower than the other compared models. This can be explained by the fact that DMH does not give a representation of the human head and shoulders region that is good enough to be distinguishable from other proposed objects.

4.2. Shoulder detection

For measuring the precision of our shoulder detections, we use the PCKh measure proposed in [1] which is inspired by the PCK (Probability of Correct Keypoints) measure proposed in [18]. The PCKh measure (h stands for head) determines a keypoint to be correctly predicted if the euclidean distance between the predicted and the ground truth points is less than a specific value, denoted as d calculated according to formula 2 below:

$$d = \alpha \times max(h, w) \tag{2}$$

Where α is a threshold that has a value between 0 and 1, h and w are respectively the height and width of the ground truth head bounding box. The results of calculating the PCKh precision using different values of α on our testing images are shown in Figure 5.

The results in Figure 5 show that OpenPose is by far the best model in terms of shoulder detection precision. However, this excellent performance comes at a heavy computational cost (discussed in section 4.4). The results clearly show that adding the depth channel always enhances the accuracy of shoulder detection. This can be seen through

Figure 6: 4 sample detection results using CNN4 RGB. Green represents ground truth data, red rectangles are regressed head detections, blue rectangles are proposal boxes, and yellow circles are predicted shoulder keypoint locations.

comparing models trained on RGB with the same models trained on RGBD. This can be explained by the distance disparity between an object and its background that is exploited by the depth channel. This disparity makes it easier for the model to correctly predict the keypoint locations of a proposal that has been classified as a head.

Figure 5 also shows that deeper models, in general, perform better in terms of shoulder precision. This is also expected as deeper models extract better features and are more efficient in keypoint detection.

4.3. Bounding box IoU

Our approach includes the generation of the bounding box of the exact location of the head within a proposal. This allows us to have a higher IoU with the ground truth bounding box. Table 1 shows the IoU of detected bounding boxes with the ground truth of different models. It is clear that our approach enhances the IoU of detections by more than twice. This is because the work done in [16] does not attempt to regress the bounding box location. The IoU is calculated based on the overlap between the proposal and the ground truth. For our methods, the IoU is calculated from the overlap of the detected bounding box and the corresponding ground truth. Figure 6 shows sample detection results from using CNN4 trained on RGB images.

4.4. Computational cost

To evaluate the computational cost of the different tested models, we benchmarked them on two platforms:

- Desktop workstation: An advanced workstation with powerful specifications and an NVIDIA GTX 1080 Ti graphics card
- Jetson TX2: An embedded system platform developed by NVIDIA [3]. Since we are building a model that should ideally run on embedded systems with low power consumption, we tested the models using the Jetson TX2.

Inference benchmarking (in milliseconds) of the different models on the GPU (Graphical Processing Unity) and CPU (Central Processing Unit) of the desktop workstation and the Jetson TX2 are shown in Table 2 and Table 3.

The processing time of the CHL algorithm is constant whether we are testing on the desktop GPU or CPU as it runs solely on the CPU. The proposals algorithm can not be parallelized as it functions on sequential analysis of pixels from the top left corner to the bottom right corner. However, this is not an entirely a negative effect as this keeps the system GPU free to only run inference.

The results show that the proposed models outperform the state of the art by a considerable margin on both GPU and CPU. On GPU, our slowest model CNN6 trained on RGBD data, runs at 80.65 FPS (Frames Per Second). It is more than two times faster than the closest competitor, SSD with base network Mobilenet which runs at 38.46 FPS. On CPU, the analytical results are the same as CNN6 trained on RGBD runs at 7.37 FPS where the closest competitor SSD with base network Mobilenet runs at 4.76 FPS. We can see that our models are more affected when running on the CPU compared to SSD Mobilenet. This is due to the fact that Mobilenet is optimized to train on low end devices.

We see that deeper networks are more computationally expensive and have a lower frame rate. In addition, the type of input affects the processing time. Models trained on RGBD input are slower than the same models trained on RGB input. These two effects are expected as adding more convolutions increases the computational cost and affects speed. Adding an extra input channel has the same effect of increasing the computational cost. However, the results show that the difference in processing times isn't significant between variations of our proposed models (around 4 FPS between our fastest and slowest models on GPU and around 1 FPS on CPU). This is primarily due to the small input size as all our proposals are resized to 48×48.

Mobilenet trained on the DMH representation [16] runs at 38.46 FPS on GPU and 4.76 FPS on CPU. The main overhead in this method comes from the generation of the DMH representation. Although the idea of relying solely on depth information for human head detection is interesting, the precision-recall performance of this approach (dis-

[3] https://developer.nvidia.com/embedded/buy/jetson-tx2

	Proposals	CNN4 RGB	CNN4 RGBD	CNN5 RGB	CNN5 RGBD	CNN6 RGB	CNN6 RGD
Avg IoU	0.28	**0.68**	**0.68**	**0.68**	**0.68**	**0.68**	**0.69**

Table 1: IoU of detected bounding box with the ground truth of different models

	Proposals	Extra	Inference	Total time (ms)	FPS
CNN4 RGB	8.7	0	3.1	11.8	**84.75**
CNN4 RGBD	8.7	0	3.1	11.8	**84.75**
CNN5 RGB	8.7	0	3.1	11.8	**84.75**
CNN5 RGBD	8.7	0	3.3	12	83.33
CNN6 RGB	8.7	0	3.5	12.2	81.97
CNN6 RGBD	8.7	0	3.7	12.4	80.65
MN DMH	8.7	12 (DMH Generation)	6.8	27.5	36.36
SSD MN	NA	0	26	26	38.46
SSD INC V2	NA	0	30	30	33.33
SSD INC V3	NA	0	39	39	25.64
OpenPose	NA	0	250	250	4

	Proposals	Extra	Inference	Total time (ms)	FPS
CNN4 RGB	8.7	0	110	118.7	**8.42**
CNN4 RGBD	8.7	0	110	118.7	**8.42**
CNN5 RGB	8.7	0	116	124.7	8.02
CNN5 RGBD	8.7	0	117	125.7	7.96
CNN6 RGB	8.7	0	127	135.7	7.37
CNN6 RGBD	8.7	0	127	135.7	7.37
MN DMH	8.7	12 (DMH Generation)	286	306.7	3.26
SSD MN	NA	0	210	210	4.76
SSD INC V2	NA	0	370	370	2.7
SSD INC V3	NA	0	580	580	1.72

Table 2: Processing times of different models on the desktop machine GPU (top) and CPU (bottom).

cussed in section 4.1), and the added overhead from the generation of the DMH representation make this approach not feasible for reliably detecting humans in different environments.

For object detection models running using SSD, Mobilenet proves the best feature extractor to be used as backbone for SSD in terms of processing speed followed by inception V2 and inception V3. OpenPose runs at only 4 FPS on the desktop GPU, which makes it unsuitable for embedded systems.

Table 3 shows the benchmarking results on the Jetson TX2. The analytical conclusions derived from benchmarking on the desktop machine are verified by the Jetson benchmarking. The interesting phenomenon is that SSD object detection models are not affected as much as our models, specially on the Jetson GPU. SSD with base network Mobilenet runs at almost the same frame rate as our slowest model (CNN6 trained on RGBD input). This can be explained through analysing how SSD works. SSD generates proposals and classifies them in a single pass. In addition, the number of default boxes evaluated by SSD does not rely on the number of people in the image. For our methods, the batch size is not constant. The proposals generated from an image are batched and passed to our models for classification, bounding box regression, and keypoint detection. In addition, it should be noted that the Jetson TX2 has modest CPU power, which significantly affects the generation of proposals. It should also be noted that while SSD models detect head locations only, our models detect head locations in addition to shoulder keypoints.

4.5. Drawbacks and failure cases

Analysing the results of our methods, we notice a significant drawback: bounding box regression and keypoint detection can't be separated from the proposal classification. We rely on the classification result of FC2 in order to generate the regressed bounding and predict the shoulder keypoints. Only when FC2 classifies the proposal as a

	Proposals	Extra	Inference	Total time (ms)	FPS
CNN4 RGB	44	0	30	74	**13.51**
CNN4 RGBD	44	0	31	75	13.33
CNN5 RGB	44	0	32	76	13.16
CNN5 RGBD	44	0	32	76	13.16
CNN6 RGB	44	0	33	77	12.99
CNN6 RGBD	44	0	33	77	12.99
MN DMH	44	50 (DMH Generation)	50	144	6.94
SSD MN	NA	0	77	77	12.99
SSD INC V2	NA	0	98	98	10.2
SSD INC V3	NA	0	130	130	7.69
OpenPose	NA	0	5500	5500	0.18

	Proposals	Extra	Inference	Total time (ms)	FPS
CNN4 RGB	44	0	300	344	**2.91**
CNN4 RGBD	44	0	300	344	**2.91**
CNN5 RGB	44	0	310	354	2.82
CNN5 RGBD	44	0	312	356	2.81
CNN6 RGB	44	0	330	374	2.67
CNN6 RGBD	44	0	333	377	2.65
MN DMH	44	50 (DMH Generation)	645	739	1.35
SSD MN	NA	0	500	500	2
SSD INC V2	NA	0	1190	1190	0.84
SSD INC V3	NA	0	1850	1840	0.54

Table 3: Processing times of different models on Jetson TX2 GPU (top) and CPU (bottom).

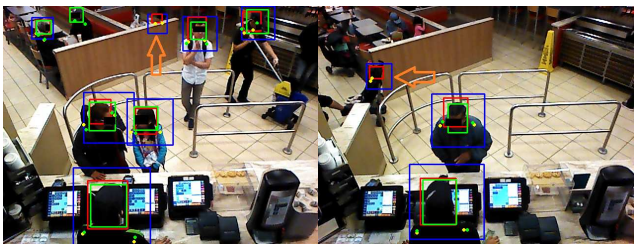

Figure 7: Sample failure cases

positive are the head bounding box and shoulder keypoints generated. In addition, two shoulder keypoints will always be generated, even if a part of the person is not visible or occluded and only one shoulder is visible. Finally, although our models can perform well when trained on RGB images only, the proposals algorithm operates solely on depth images. This makes it reliant on the quality of the depth images and adds range constraints. Not to forget that off the shelf depth sensors perform poorly in outdoor environments and our models trained on RGB input would not perform well at night. Figure 7 shows two samples where our system fails as it detects non human objects are humans, and generates a head bounding box and shoulder keypoint locations for those objects.

5. Conclusion

In this paper, we introduced a new method that is accurate and computationally efficient for human head and shoulders detection that uses a combination of image processing and deep learning techniques and is suitable for embedded systems. Object proposals are generated using an enhanced version of the CHL algorithm. Proposals are then fed to a small CNN for classification and regression of the head bounding box and shoulder keypoint locations. We show the effect of adding the depth channel on the accuracy of head and shoulder detections. We compare our work to state of the art methods in terms of speed and accuracy and show that our approach surpasses the current state of the art in both terms. Our approach that utilizes the CHL algorithm for generating object proposals solves the problem of occlusion to a decent extent. Our system would fail to propose the presence of an occluded person only if this person is within 15cm in 3D space from the person/object occluding him/her. In the future, our work can be expanded by detecting more body joints in a bottom up approach. In addition, introducing multi-threading to the CHL algorithm by dividing an image to a number of regions where each region runs the CHL algorithm on a separate thread can be investigated.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[2] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D.S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31 –39, 2011.

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017.

[4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[5] Farzan Erlik Nowruzi, Wassim A El Ahmar, Robert Laganiere, and Amir H Ghods. In-vehicle occupancy detection with convolutional networks on thermal images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[6] Chenqiang Gao, Pei Li, Yajun Zhang, Jiang Liu, and Lan Wang. People counting based on head detection combining adaboost and cnn in crowded surveillance environment. *Neurocomputing*, 208:108–116, 2016.

[7] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[8] Jun Liu, Ye Liu, Ying Cui, and Yan Qiu Chen. Real-time human detection and tracking in complex environments using single rgbd camera. In *2013 IEEE International Conference on Image Processing*, pages 3088–3092. ieee, 2013.

[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[10] Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5137–5146, 2018.

[11] André Mateus, David Ribeiro, Pedro Miraldo, and Jacinto C Nascimento. Efficient and robust pedestrian detection using deep learning for human-aware navigation. *Robotics and Autonomous Systems*, 113:23–37, 2019.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[13] Khurram Soomro, Haroon Idrees, and Mubarak Shah. Online localization and prediction of actions and interactions. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):459–472, 2018.

[14] Luciano Spinello and Kai O Arras. People detection in rgb-d data. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3838–3843. IEEE, 2011.

[15] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[16] Luchao Tian, Mingchen Li, Yu Hao, Jun Liu, Guyue Zhang, and Yan Qiu Chen. Robust 3d human detection in complex environments with depth camera. *IEEE Transactions on Multimedia*, 2018.

[17] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[18] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2878–2890, 2013.

[19] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[20] Liang Zheng, Yujia Huang, Huchuan Lu, and Yi Yang. Pose invariant embedding for deep person re-identification. *IEEE Transactions on Image Processing*, 2019.