Workers Supervision for Construction Safety

PHAM Trung Kien, DO Van Quyet
Department of Mathematics
The Hong Kong University of Science and Technology
{tkpham, vqdo}@connect.ust.hk

Abstract

Along with the growth of society, many problems emerge and need our attention. Specifically, in construction industry, more and more severe accidents occur, raising the need for construction safety enhancement. Many solutions that focus on monitoring worker, mainly about whether they wear enough safety equipment, has been proposed. Unfortunately, they share the same drawbacks as heavily relying on human supervisors and using standard camera input with limited surveillance range.

In this report, we introduce an approach for autonomous monitoring that deploys an AI system on images retrieved from 360° fisheye overhead camera on construction site. It consists of two components, a Multi-Person Detector and a pair of Binary Classifiers. Here we hire the State-of-the-art model for each subtask, RAPiD and EfficientNetV2 respecively. With the concern about heterogeneity between fisheye and standard images in training dataset for Classification, we implement and apply fisheye effect on standard instances, then compare to the case of not adding fisheye distortion.

Overall, our AI model achieves lower but close to the reported performance of two components in their original papers. That suggests the capability of our model in this real-world problem.

1. Introduction

The construction industry has never been less worth attention with number of buildings, infrastructures, skyscrapers… significantly growing every year [6]. However, going along with this development is the fact that the number of workers injured or even died in tragic accidents happened in construction sites is also increasing excessively [2] [1], making the need to enhance construction safety hotter and hotter.

There are two main aspects in construction safety enhancement. Firstly, in term of non-human factor, construction tools and machine should be regularly examined, repaired, and upgraded to mitigate malfunctions, which can cause severe accidents. Secondly, in term of human factor, workers need to equip themselves with high-quality safety suits. In this project, we focus on human factor, that we want to monitor workers in construction site and check if all workers wear all necessary safety items.

At early period, the solution mostly relied on human effort with fatal limitations such as slow reaction ability or inaccurate decision making [1]. Therefore, it is essential to minimize or even eliminate that human factor and introduce an automatic approach to cope with the problems. Many recent popular methods involve distant monitoring systems with standard cameras properly setup on construction areas to keep track of construction progress and workers. Despite of laying the groundwork for the ultimate goal, they still require human supervisors to manipulate the monitors and their selected type of cameras shows biased performance due to limited range of angle the cameras can be placed.

Attempting to overcome those drawbacks, this project will focus on building an accurate and fast AI supervisor to manage fisheye frames retrieved from onsite 360° overhead camera adopting State-of-the-art (SOTA) Object Detection and Image Classification models. Our AI monitor consists of two components (Figure 1) which are a Multi-Person Detector to locate and keep track of every worker inside input frames and two Binary Image Classifiers to determine whether those workers are wearing required safety helmet and safety jacket or not.

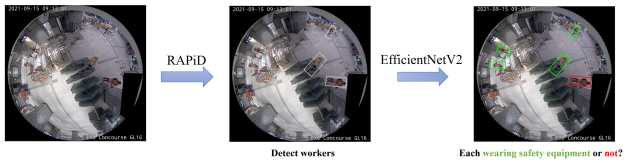


Figure 1: End-to-end pipeline of our AI monitor.

As we put concentration on fisheye frames retrieved from 360° cameras, we will attempt to replicate and retrain Rotation-Aware People Detection (RAPiD) model, which

has shown outstanding performance on detecting people inside fisheye frames, to be our detector. From the fact that our target worker object can indeed be considered as person object, we do expect qualitatively that our detector will have competitive performance on locating workers in construction sites through 360° cameras after being sufficiently trained on the CEPDOF and HABBOF datasets. Quantitatively, we aim to test our detector's implementation on the two datasets adopting Average Precision (AP), i.e., the area under the Precision-Recall curve, as well as F-measure metrics. Again, we expect to have similar statistics comparing to its original paper.

After workers' positions are predicted, bounding boxes will subsequently be cropped and fed into our image classifier instances to make inferences. For each targeted safety equipment, every cropped boxes will be classified as Wearing / Not wearing. The model we selected to be our classifiers is EfficientNetV2 which can help speed up both training and inference process. We expect our classifiers to produce high-quality inferences within a blink of an eye after its training process. For quantitative evaluation, our classifiers will be tested in terms of Precision, Recall, and F-measure metrics and are expected to reach 90 percent or more in every criterion for any category of safety equipment.

2. Dataset

To train the detection model, we use two prevalent datasets [3], HABBOF (Human-Aligned Bounding Boxes from Overhead Fisheye cameras) and CEPDOF (Challenging Events for Person Detection from Overhead Fisheye image). Both are publicly available, captured by overhead fisheye cameras, and importantly annotated with rotated bounding boxes. The setting in images from the two datasets more or less similar to images from construction site (Figure 2), as they contain many people and other objects. Table 1 shows various statistics of these two datasets. In fact, HABBOF has the number of frames and people within a frame limited, also lack of challenging cases. On the other hand, CEPDOF contains many more frames and people, especially challenging scenerios such as crowed room, low-light condition, or various human body poses.

Simultaneously, we collected images for training our classifiers in two ways: one is by scraping, crawling, and cropping from open sources on the internet like Google Images and Shutterstock; the other is by cropping from CEPDOF, HABBOF frames and videos from construction sites. The two most common safety equipment that we are focusing on are the safety helmet and safety jacket. For the first method, images obtained are in standard rectangle form and they include both people and background, thus were processed by published pretrained person detection model such as YOLOv5 to extract our targeted single-person frames. For the second one, we can retrieve human objects from images in CEPDOF and HABBOF datasets by the groundtruth annotation. Also, we utilized our detector after well-trained to do the extraction on construction-site videos since those are captured in fisheye setting. After properly labelling those cropped images, we obtain two datasets, so-called helmet and jacket datasets, each consists of around 30000 images with 50:50 ratio of wearing to not wearing according safety equipment.

Figure 2: Sample from two datasets. The left one is from HABBOF, the right one is from CEPDOF.

3. Methodology

As mentioned, the AI monitor consists of two task-specified models: RAPiD for object detection on images with fisheye frame and EfficientNetV2 for classification on cropped boxes as the output of RAPiD. Apart from some common data augmentation methods, fisheye distortion is applied in training process to enhance model performance.

3.1. RAPiD (Rotation-Aware People Detection) [3]

RAPiD is an end-to-end method for human detection on overhead fisheye images. Noticing the appearance of standing people in the images that is not always in upright position, this method use arbitrarily-rotated bounding boxes to cope with this problem. It extends the architecture of YOLO – one of the most successful algorithms for object detection.

Unlike some other YOLO-based methods which use radial-aligned bounding boxes or rotated anchor boxes with step of 15 degree and require require significant pre/post-processing, RAPiD only introduce a new entry $\hat{t}_\theta$ in output tuples to predict angle of the corresponding bounding box which do not produce additional computational complexity (Figure 3). With the removal of $\hat{t}_{class}$ (since the only interest class is human) and addition of $\hat{t}_\theta$, the loss function is modified accordingly.

As end-to-end algorithm for object detection, RAPiD can be effectively and comfortably trained or fine-tuned on annotated fisheye images dataset. Comparing performance of RAPiD versus previous SOTA methods when they were

Dataset	# of videos	Avg. # of people	Max # of people	# of frames	Resolution	FPS
HABBOF	4	3.5	5	5,837	2,048	30
CEPDOF	8	6.8	13	25,504	1,080 to 2,048	1-10

Table 1: Statistics of datasets for object detection on overhead fisheye images.

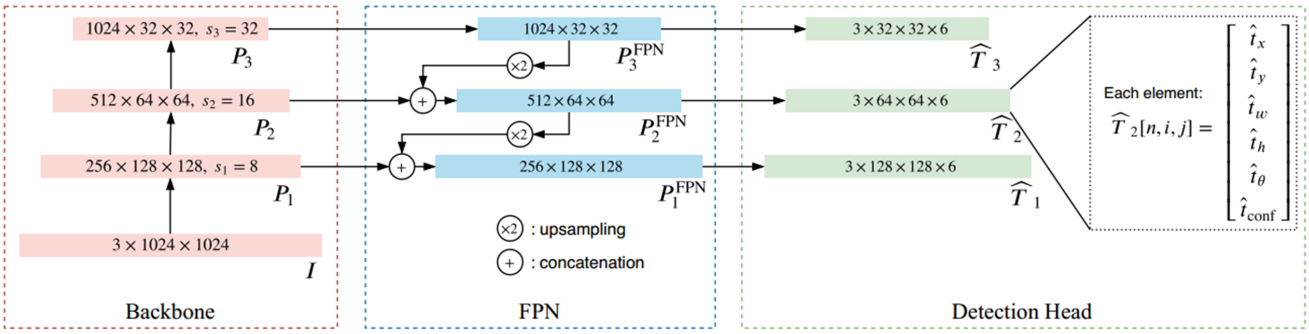
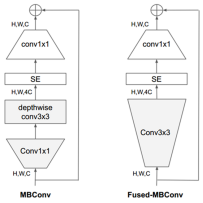


Figure 3: RAPiD's architecture.

first pretrained on MS COCO 2017, then cross-validated on three datasets MW-R, HABBOF, and CEPDOF, it clearly outperforms other previous SOTA models in both accuracy and inference time. Thus, it suits our need in this project.

3.2. EfficientNetV2 [5]

EfficientNetV2 is new family of convolutional neural network mainly for classification. Its primary aim is to have better paramater efficiency and faster training speed. Applying Neural Architecture Search (NAS) method, models sampling from space of MBConv [4] and Fused-MBConv blocks are experimented and reported, then the best architecture that optimizes training and parameter efficiency is chosen (Figure 4).



Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

Figure 4: Structure of MBConv, Fused-MBConv blocks, and EfficientNetV2-S.

To further speed up training process, progressive learning is adopted. It starts with small images, then gradually increase the image size. To overcome overfitting problem when training models with large images, EfficientNetV2 has been equipped adaptive regularization – weak in the first few epochs (corresponding to small images) and stronger in later epoch. Apart from that, due to slow speed of training model with very large-size images, the maximum size of input images is set to 380x380 instead of usual 512x512.

In comparison to previous SOTA models on ImageNet ILSVRC2012 dataset, EfficientNetV2 networks is up to 6.8x smaller in parameter size, much faster in both training and inference while maintaining approximately the same or higher accuracy (Figure 5). In downstream tasks like CIFAR-10, CIFAR-100, Flowers and Cars, after fine-tuned on those datasets, EfficientNetV2 also surpass performance of previous models. That suggested the capability of the model in specific tasks, such as ours.

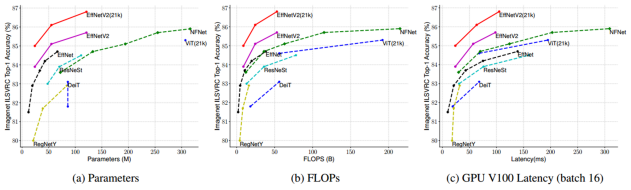


Figure 5: Model size, FLOPs, and Inference latency of models. Latency is measured with batch size 16 on V100 GPU. 21k denotes pretrained on ImageNet21k images.

3.3. Fisheye distortion

We notice that the characteristics of standard images may be different from those of fisheye images, so are crawled images in training dataset for classificaion. That raises the out-domain issue that additional training data from not the same source as test data may not improve or even lower

performance of model. Thus, to overcome this problem, we consider applying fisheye effect on crawled instances, as give the same level of distortion to those images.

Specifically, we simulate the way fisheye cameras capture the scence with the focus on a particular object. For each crawled image, we place it to a random-size empty square texture at a random location, then apply fisheye effect on the whole texture while keeping track of the image location as coordinates of four corners, and finally use those information to extract the image back. In this process, the whole texture plays as original scene.

To better understand where to paste crawled images in empty textures, we take a close look at fisheye frame retrieved from onsite camera. In most the cases, worker objects are aligned in radius of the images, i.e straight line segments that connect center point to points in boundary, with foot-to-head direction pointing out, and they are relatively far from the images' center. Thus, when we apply fisheye effect, we need to put crawled images in the same setting. Furthermore, because of the symmetry, putting crawled images in any radius will give us the same distorted version of it. Therefore, to simplify our implementation of fisheye effect, we only render crawled images in the upper vertical radius. The process is shown in Figure 6.

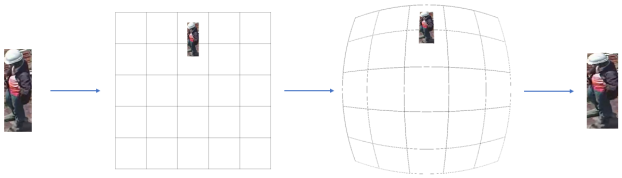


Figure 6: Illustration of applying fisheye distortion.

4. Experimental Results

4.1. Workers Detection

We have finished implementing the architecture for our multi-person detector, i.e. replicating RAPiD model based on its original paper, as well as training and simultaneously fine-tuning the detector on approximately 32000 frames from CEPDOF and HABBOF datasets. Preliminarily in our midterm milestones, since there is no predefined train-test split in these two datasets, all the images were exerted for the training process for us to get a first glance at our detector's behavior during this procedure. Simple resizing augmentation was adopted for preprocessing the images into 1024x1024 resolution before they were fed into our detector with a batch size of 16. Besides, the detector weights were updated by Stochastic Gradient Descent (SGD) optimizer with the following parameters: learning rate 0.0001, momentum 0.9, and weight decay 0.0005. Until then, the training process of our Multi-Person Detector could be considered as partially finished after over 100 epochs, which is equivalent to around 200000 feedforward iterations. To ensure that the training process progressing in the right direction, we kept track of the loss (see Figure 7) after every epoch throughout the whole process. The fact that value of the loss gradually decreases epoch by epoch have affirmed that there is neither implementation error nor logical error occurred in our detector's architecture and training code. Other than the loss, we also saved checkpoint

Figure 7: Plot of training loss over epoch.

for the weights of the detector after each epoch throughout the training process. We selected the best checkpoint corresponding to the epoch that has smallest loss for making inferences and evaluation. As expected, with the weights loaded from that checkpoint, our detector performs reasonably well, and its statistics are lower but close to the benchmark published in the original paper. In details, for quantitative results, we calculated Average Precision (AP) score at IoU = 0.5 (AP_{50}) and F-measure at a fixed 0.3 confidence threshold to examine our detector's performance on HABBOF and CEPDOF datasets. All metrics are averaged over all the videos in each dataset and recorded in Table 2.

	CEPDOF		HABBOF	
	AP_{50}	F	AP_{50}	F
RAPiD	85.8	0.836	98.1	0.969
Ours (Preliminary)	80.2	0.789	95.3	0.912
Ours (Best)	82.8	0.803	96.2	0.935

Table 2: Quantitative results of our detector comparing to original paper.

In terms of qualitative results, we ran inference of our

detector on all the videos from the two datasets. Additionally, to justify our assumption that worker objects can be detected similarly to person objects, we also ran inference on two more videos retrieved from two 360° cameras placed at two different positions monitoring a construction site. Some of the best illustrative frames with bounding boxes drawn for each detected person or worker can be observed in Figure 8.

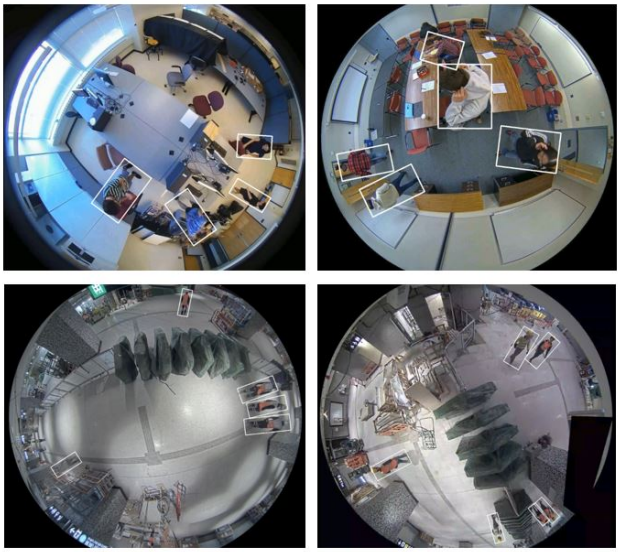


Figure 8: The top left frame is from HABBOF, the top right frame is from CEPDOF, and the two bottom frames are from a construction site. Bounding boxes are drawn for each detected person or worker. The two bottom ones justify the correctness of our assumption.

After midterm, we have experimented and adopted some techniques with purpose to further improve the detector's performance. Specifically, we first practiced holdout cross-validation process by randomly splitting the original dataset combined from CEPDOF and HABBOF into training set and validation set with respective ratio 70:30, preventing the detector from overfitting issue and enhancing generalizability. Simultaneously, we performed hyperparameters tuning by trying out different combinations of values for batch size and optimizer configuration in order to select the most appropriate setup for training. Finally, more complex augmentation methods including vertical and horizontal flip, random rotation, brightness and contrast adjustment, and color jitter are utilized to diversify the dataset. With all these being applied, our detector is now acquiring moderately better performance and corresponding statistics can be observed in Table 2.

4.2. Classification

For our binary classifier, during midterm we partially finished implementing its architecture which is EfficientNetV2, and integrating its training code. However, we had not started the training processes for its two instances, one for determining workers wearing safety helmet or not and one for the same task but targeting safety jacket, yet because we have limited resources to do it concurrently with the detector. All of our available GPUs were reserved and utilized for training the detector as it is more difficult yet essential part to ensure the success of our AI monitor. Fortunately, we were able to collect datasets, which is not a GPU-dependent work, for training our classifier's instances.

Up to this point, the architecture implementation of our binary classifier is completed as well as its training and inference code, wrapping up the pipeline of our AI monitor. Additionally, the data collecting procedure for our two aforementioned datasets is also finished not only by retrieving images from open sources but also utilizing our own well-trained detector to extract person objects in CEPDOF, HABBOF datasets and the two videos from a construction site.

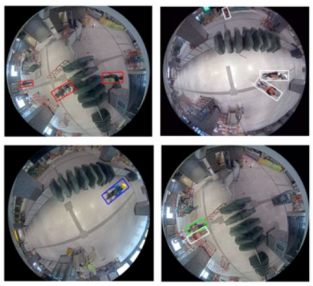
In terms of data preprocessing, recognizing the effectiveness of holdout cross-validation process in preventing overfitting problem and improving generalizability, we also randomly split both the helmet and jacket datasets into their corresponding training set and validation set with ratio 70:30. Various augmentation methods such as vertical and horizontal flip, resizing to 224x224 resolution, and color jitter are applied onto the images before being fed into the training processes with a batch size of 32. We also try fisheye distortion on crawled standard images before training to examine its effectiveness on counteracting the out-domain issue. Similar to the detector's, the classifiers' weights are updated by Stochastic Gradient Descent (SGD) optimizer. Together with the batch size value, optimizer's parameters are also selected to be: learning rate 0.001, momentum 0.9, and weight decay 0.0005, based on hyperparameters tuning process.

For each of the two binary classifier instances, the training process went for over 120 epochs, which is equivalent to nearly 100000 feedfoward iterations. Checking similarly as for the detector, it is confirmed that there is neither implementation error nor logical error occurred in our classifier's architecture and training code integrated in our pipeline. We also kept track and selected the best checkpoints for each instance for making inferences and evaluation. With the weights loaded from those checkpoints, our two classifier instances perform significantly well and its statistics are positive as well. In details, for quantitative results, we calculated Precision, Recall, and F-measure to examine the performances of our classifier instances on their corresponding datasets and recorded them in Table 3.

	Precision	Recall	F
Helmet Classifier	0.922	0.915	0.918
Helmet Classifier + FD	0.903	0.895	0.899
Jacket Classifier	0.937	0.923	0.930
Jacket Classifier + FD	0.915	0.897	0.906

Table 3: Quantitative results of our classifier instances' performance. FD denotes Fisheye Distortion.

In terms of qualitative results, we simultaneously ran inference of our detector and two classifier instances, i.e. ran the whole pipeline process, on the two videos from a construction site. Some of the best illustrative frames with color-defined bounding boxes drawn for each detected person or worker can be observed in Figure 9.



Jacket \ Helmet	Yes	No
Yes	White	Green
No	Blue	Red

Figure 9: Typical inference results. Table on the right describes the color definition according to classification results for displaying bounding boxes.

4.3. Result Analysis and Further Improvement

Overall, both detection and classification models yield sufficiently high performance, meet our original goal.

The deficit in performance of our RAPiD in comparison to the original one likely comes from the pretraining process, that the original were trained on a big dataset MS COCO 2017 before being fine-tuned on CEPDOF and HABBOF. This gives us an idea to further improve the system. For qualitative result, in the test images, we observed that some workers are not detected by our system, mainly because they are too small within the frame, under relatively low-light condition, or surrounded and blocked by some construction tools. To solve this case, we need to build a larger dataset with more instances from construction sites.

About our Classifier, its performance unexpectedly drops when applying fisheye effect on standard instance. We suspect that the effect is not properly applied, say standard instances are overly distorted. In fact, at construction site, workers are normally several meters distant from the vertical projection of cameras in the ground, making their images in fisheye frame more or less the same as in standard one. In future, we would focus on this aspect to improve our classifier.

5. Conclusion

In this project, we proposed an autonomous AI supervisor to help monitor workers safety in construction sites through fisheye imaging. The system consists of two main components, a multi-person detector and a pair of binary classifier instances. The detector, which is a replication of RAPiD model, holds responsible for locating and correspondingly extracting spatial information of every worker inside input frames. Subsequently, single-person images will be cropped out based on those information and be justified by the two classifier instances, which are powered by EfficientNetV2, if their according person is wearing safety helmet and safety jacket or not. The classification results will then be reflected respectively by the predefined color of the bounding boxes in the output frames. We believe that our AI monitor can be extended and applied into many more use cases not limited in construction safety area.

References

[1] T. Anojan and C. Siriwardana. A study to investigate the factors affecting construction worker performance. In *2021 Moratuwa Engineering Research Conference (MERCon)*, pages 13–18, 2021.

[2] CPWR – The Center for Construction Research and Training. Data bulletin: Fatal injury trends in the construction industry, 2021.

[3] Z. Duan, M. O. Tezcan, H. Nakamura, P. Ishwar, and J. Konrad. Rapid: Rotation-aware people detection in overhead fisheye images, 2020.

[4] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[5] M. Tan and Q. V. Le. Efficientnetv2: Smaller models and faster training, 2021.

[6] United Nations Environment Programme. 2020 global status report for buildings and construction: Towards a zero-emission, efficient and resilient buildings and construction sector, 2020.