high level of semantics in the image, only the regional difference, the deep network is not necessary.

To train the model, Binary Cross Entropy based on each pixel is used by the model.

$$L(y, \widetilde{y}) = -\frac{1}{m}\sum_{i=1}^{m}[y_i log(\widetilde{y}) + (1 - y_i) log(1 - \widetilde{y})]$$

Training details: The model is implemented by using keras based on tensorflow. In order to improve the robustness of the model, during training, random cropping and flipping are applied. The model uses the Adam optimizer with an initial learning rate of 1e-3 to train, and then keeps reducing learning rate when the metric has stopped improving. After learning rate is reduced to zero, the model reaches its optimal state.

(a) (b) (c)

Figure 5: The input and label of the algorithm for training are shown here. (a) Intensity map acts as background for the input. (b) Intensity map acts as contrast for the input. (c) Foreground (Actual Difference) acts as label for output.

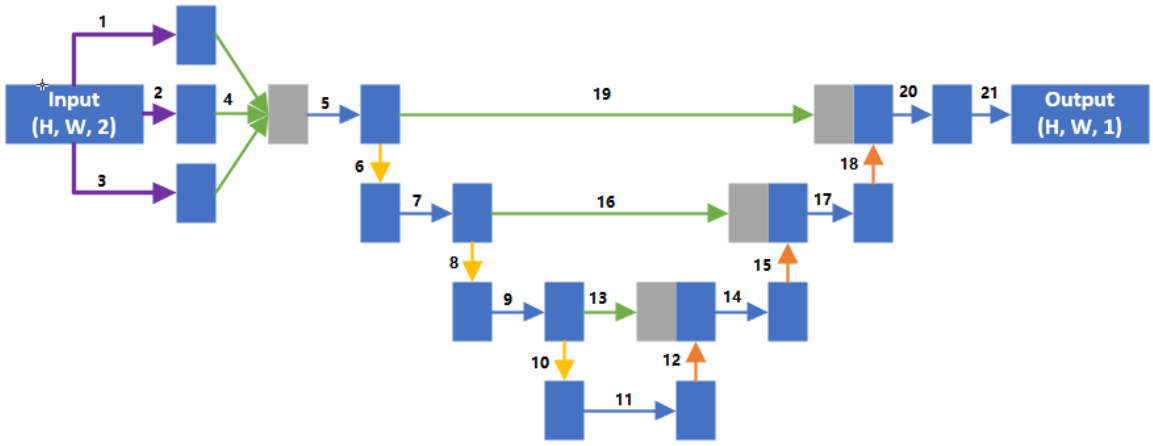


Figure 6: The architecture of the model is shown here. It contains a head, a contraction path and an expansion network. The information of each layers are displayed in Table 1.

#	**Layers**
1	Convolution: 16 filters, kernel size = 3, stride = 1
2	Convolution: 16 filters, kernal size = 3, stride = 1, dilation = 3
3	Convolution: 16 filters, kernal size = 3, stride = 1, dilation = 5
4	Concatenate
5	Convolution: 48 filters, kernal size = 3, stride = 1 Batch Normalization Relu
6, 8, 10	MaxPooling: kernal size = 2, stride = 2
7	Convolution: 32 filters, kernal size = 3, stride = 1 Batch Normalization Relu
9	Convolution: 16 filters, kernal size = 3, stride = 1 Batch Normalization Relu
11	Convolution: 16 filters, kernal size = 3, stride = 1 Batch Normalization Relu
12, 15, 18	UpSampling: kernal size = 2, stride = 1
13, 16, 19	Skip / Concatenate Connection
14	Convolution: 32 filters, kernal size = 3, stride = 1 Batch Normalization Relu
17	Convolution: 48 filters, kernal size = 3, stride = 1 Batch Normalization Relu
20	Convolution: 16 filters, kernal size = 1, stride = 1 Batch Normalization Relu
21	Convolution: 1 filter, kernal size = 3, stride = 1 Sigmoid

Table 1: The information of all layers in the model are shown here.

4. Data collection and augmentation

The datasets used in this paper was collected by two ToF cameras, one is based on Espros EPC660 imager, and the other is based on Sony IMX556 imager. The datasets are a collection of members that each one includes one intensity map acts as background, one intensity map acts as contrast and their actual difference. The resolutions of EPC660 and IMX556 are 320*240 pixels and 640*480 pixels separately. In order to unify the shape of input and improve the performance of model, the images they captured are all reduced to 160*120 pixels.

The datasets contain a total of 560 sets that contains different integration times and multipath distortions, such as a pair of the intensity map without multipath distortions

captured under 1000us integration time and the intensity map with multipath distortions captured under 600us integration time. During the experiment, 448 sets of datasets are used for training and 112 sets of datasets are used for testing.

The datasets contain diverse indoor scenes, such as office rooms, meeting rooms and laboratories. When capturing a set of datasets, the ToF camera was mounted on a fixed bracket for prevent vibrating. After camera was mounted, use an appropriate integration time that will not cause regional overexposure, but can capture clear intensity map to capture an intensity map as background. Then, put an object, named "A", into the scene and repeat the previous step (select another integration time) to capture an intensity map as contrast (multipath distortions can be reproduced if the object is close enough to the camera). Finally, label the actual difference (the object "A") between the previous two intensity maps on a binary image.

Data collection and labeling are laborious. The following method can be used to simulate the multipath distortions and dynamic integration time adjustment on the original datasets for augmenting the training datasets (Figure 7). Firstly, cutout the actual difference between two intensity maps. Secondly, apply a same series of transformations, such as translation, rotation and shear, on both actual difference image and actual different cutout. Thirdly, paste the transformed actual difference cutout on the intensity map acts as background to create a new intensity map to simulate objects at different location. Fourthly, blend the new intensity map and noise image (put several random two-dimensional gaussians with random height, width, μ and σ at different locations) to create another intensity map to simulate different integration times and multipath distortions.

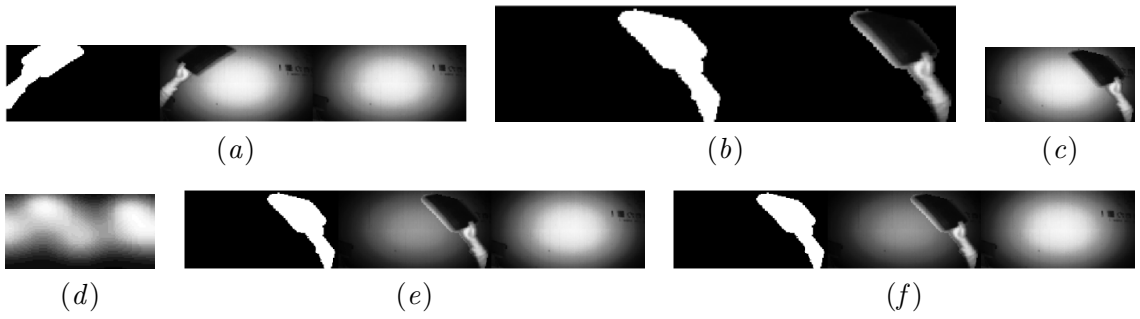


Figure 7: How to simulate the multipath distortions and dynamic integration time adjustment is shown here. (a) Select a set from datasets which contains actual difference, intensity map acts as contrast and intensity map acts as background. (b) Apply a same series of transformations on actual difference and actual different cutout. (c) Paste the transformed actual difference cutout on the intensity map acts as background. (d) Create noise image(put several random two-dimensional gaussians with random height, width, μ and σ at different locations). (e) Blend the intensity map from part (c) and noise image from part (d). (f) Create a new set of datasets by using the previous images.

Methods	**mIOU**
Inter-frame differencing (Threshold >=20)	23.4%
Inter-frame differencing (Threshold >=30)	31.8%
Mog (Threshold >=20)	21.3%
Mog (Threshold >=30)	23.8%
K-Nearest (Threshold >=20)	23.3%
K-Nearest (Threshold >=30)	24.8%
Ours	**82.3%**

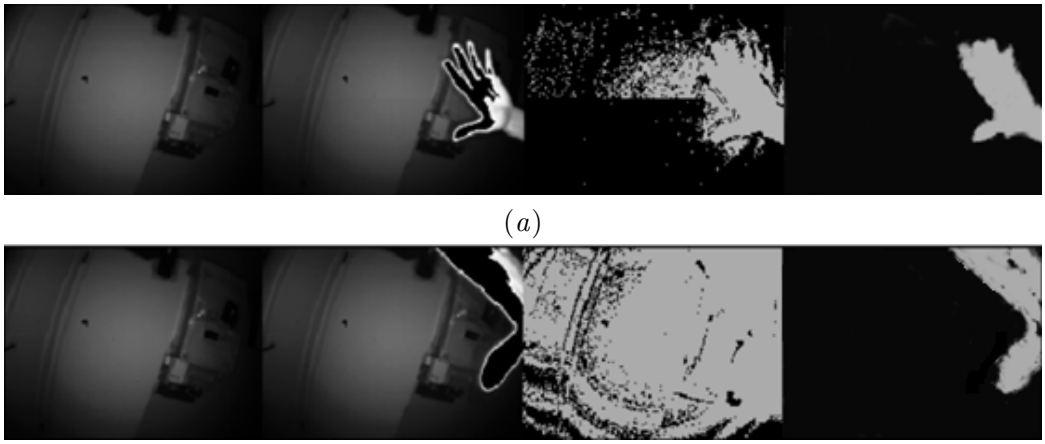
Table 2: The performance of different algorithms on the test datasets are compared by using the metric called mean Intersection Over Union. The inter-frame differencing algorithm requires only one background image to work, therefore the datasets in this paper can be nicely adopted to it. The Mog and K-Nearest algorithm requires multiply images to learn the background, therefore random scaling all the values of entire background image is used to simulate multiple background images.

5. Experiment and result

In the experiment, the error of foreground detection algorithms is quantified by the metric called mean Intersection Over Union (Table 2). The reason why pixel accuracy is not used here is when the area of actual difference between two intensity maps is relatively small, it cannot nicely represent the performance of foreground detection. In the case of using fixed integration time and no multipath distortions existed, the performances of the new algorithm and the traditional foreground detection algorithms are close. In the case of using dynamic integration time and no multipath distortions existed, the performance of the new algorithm dramatically surpasses the traditional one. Besides, another advantage of the new algorithm is no need to set any parameters, such as the threshold of pixel values difference. The traditional algorithms are sensitive to the threshold of pixel values difference which is difficult to pick, therefore, different thresholds are used in the test to detect the foreground.

Performance under multipath distortions: In the case of multipath distortions existed, the intensity map fluctuates greatly. The traditional algorithm treats a large amount of background as the foreground, but the new algorithm can detect the foreground well. The performances of both algorithms can be seen in Figure 8, and the multipath distortions were reproduced by controlling the distance between the hand and the camera.

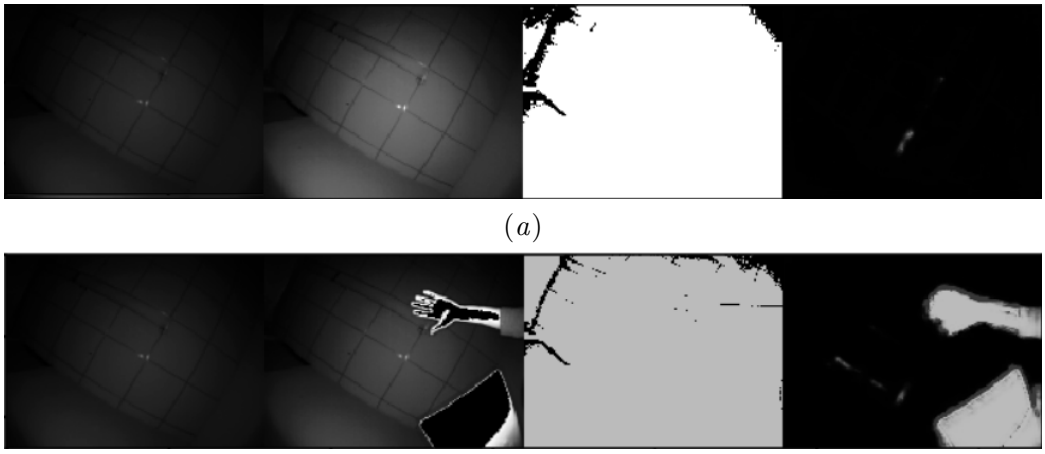
Performance under dynamic integration time adjustment: In the case of dynamically adjusting the integration time, the change of value in intensity map is approximately proportional to the integration time. When the integration time changes greatly, the traditional algorithm will consider most areas of the entire image as foreground, in contrast, the new algorithm can detect the foreground pretty well. Figure 9 demonstrates the performances of both algorithms, 500us integration time and 800us integration time are used separately.



(a)

(b)

Figure 8: The performances of traditional algorithms and new algorithm are compared based on two sets of test datasets. In each part, intensity map acts as background, intensity map acts as contrast, performance of traditional algorithm, performance of new algorithm are displayed sequentially. (a) Performances of traditional algorithms and new algorithm in the case of multipath distortions with light magnitude. (b) Performances of traditional algorithms and new algorithm in the case of multipath distortions with heavy magnitude.



(a)

(b)

Figure 9: The performances of traditional algorithms and new algorithm are compared based on two sets of test datasets. In each part, intensity map acts as background, intensity map acts as contrast, performance of traditional algorithm, performance of new algorithm are displayed sequentially. (a) Performances of traditional algorithms and new algorithm in the case of dynamic integration time adjustment. (b) Performances of traditional algorithms and new algorithm in the case of dynamic integration time adjustment.

6. Discussion and Conclusion

Using neural networks to detect foreground on ToF images is a novel algorithm. Despite the additional computation introduced by neural networks, it is acceptable on mainstream computing platforms, and the input image size can be scaled down to reduce the computation complexity. The new algorithm can be used in many ToF camera-based applications, such as passenger flow statistics, object detection on convey belt. It can also be a sub-module in the machine vision platforms.

In this paper, it shows the feasibility of using neural networks to learn and detect the foreground on the intensity map of ToF, especially, dynamically adjusting the integration time or encountering multipath distortions. In order to realize this learning process, two real ToF cameras are used to collect comprehensive datasets. In the experience, it shows the performance of the new algorithm significantly surpasses the traditional algorithms in the scenes mentioned above. This also gives people who develop ToF based applications a new way of thinking.

References

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

Carlo Dal Mutto, Pietro Zanuttigh, and Guido M Cortelazzo. Fusion of geometry and color information for scene segmentation. volume 6, pages 505–521. IEEE, 2012.

Valeria Garro, Carlo Dal Mutto, Pietro Zanuttigh, and Guido M Cortelazzo. Edge-preserving interpolation of depth data exploiting color information. *annals of telecommunications-annales des télécommunications*, 68(11-12):597–613, 2013.

Tae-Hoon Hwang and Jin-Heon Kim. A real time low-cost hand gesture control system for interaction with mechanical device. *Journal of IKEEE*, 23(4):1423–1429, 2019.

Hand gesture recognition with leap motion and kinect devices, 2014. IEEE.

Robert Lange and Peter Seitz. Solid-state time-of-flight range camera. *IEEE Journal of quantum electronics*, 37(3):390–397, 2001.

Benjamin Langmann, Seyed E Ghobadi, Klaus Hartmann, and Otmar Loffeld. Multi-modal background subtraction using gaussian mixture models. In *ISPRS Symposium on Photogrammetry Computer Vision and Image Analysis*, pages 61–66, 2010.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Ajmal Shahbaz, Joko Hariyono, and Kang-Hyun Jo. Evaluation of background subtraction algorithms for video surveillance. In *2015 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, pages 1–4. IEEE, 2015.

Ali Shahnewaz and Ajay K Pandey. Color and depth sensing sensor technologies for robotics and machine vision. In *Machine Vision and Navigation*, pages 59–86. Springer, 2020.

Kilho Son, Ming-Yu Liu, and Yuichi Taguchi. Learning to remove multipath distortions in time-of-flight range images for a robotic arm setup. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3390–3397. IEEE, 2016.

Samuel Verdú, Eugenio Ivorra, Antonio J Sánchez, Joel Girón, Jose M Barat, and Raúl Grau. Comparison of tof and sl techniques for in-line measurement of food item volume using animal and vegetable tissues. *Food control*, 33(1):221–226, 2013.

Weihang Wang, Peilin Liu, Rendong Ying, Jun Wang, Jiuchao Qian, Jialu Jia, and Jiefeng Gao. A high-computational efficiency human detection and flow estimation method based on tof measurements. *Sensors*, 19(3):729, 2019.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.