Detection of Unstable Objects by Using Deep Learning for Domestic Environment

Takaaki Fukui*, Nobutaka Shimada*, Tadashi Matsuo*
*Ritsumei University, Shiga, Japan

***Abstract*— In this study, we estimate whether or not an object is unstable under the situation that an object exists on a desk by using deep learning. We also show that this method is effective for images taken in the real world of similar situations. In the experiment, the model is learned using images generated from physical simulation using Unity, and the structure of the model is designed using UNet.**

I. Introduction

Even if a human has never seen an object, you can guess how the object will move according to the dynamic state and external force of the object. For example, suppose there is a situation where a pet bottle is placed on the edge of the table, the person who sees it can imagine that it will fall down if you press it a little and you can understand that it is unstable. In this way, humans can intuitively grasp the state of motion without solving the equation of motion. In recent years, not only industrial use for robots, but also applications that support people's lives by being close to people have been studied. In order for robots to make appropriate decisions based on experience, even in unknown environments, it is indispensable to express human-like abilities as computers, and this is becoming increasingly important.

As a related study, Wenbin Li et al. [3]Using the deep learning model, we directly calculated the stability of the entire image from the still image to the input image. In this study, the stability of each pixel is estimated. As a related study, Adam Lerer et al. [2]Created a model to determine how an object moves, observed how the results changed with occlusion, and which pixels contributed to stability and instability. However, in this study, it is possible to detect unstable places more quickly by using the results of physical phenomena directly as a teacher. In this study, we estimate whether an object is stable or unstable by using a deep convolutional neural network for images of a situation where objects exists on a desk, which is generated using physical simulation. In this study, instability is defined by the frequency of large changes in potential energy when an external force is applied multiple times to an object. We also show that this method is effective for images taken in the real world of similar situations. The method of this research is expected to be applicable to tasks that must infer physical behavior from observations such as product alignment and tidying up.

II. Problem Setting

This chapter describes the issues addressed in this study. The purpose of this study is to detect unstable parts of the input image. Here, instability is defined by the frequency with which large energy changes occur when an external force is applied to an object multiple times. Create a model that outputs a map assigned the instability of objects belonging to each pixel of the input image. When learning this model, the input image uses an image taken from a simulation of the home environment created by the simulation.

III. Proposed Method

In order to solve this problem, we created a UNet(Fig.1) using SE-ResNeXt50 [1] as an encoder. The reason for choosing this network is that UNet can be used to estimate the instability of each pixel so that it can be predicted even when the position, shape, and amount of the object are unknown. Next, the loss function L will be described. The loss function is the expression1 that is the function similar to MSE.

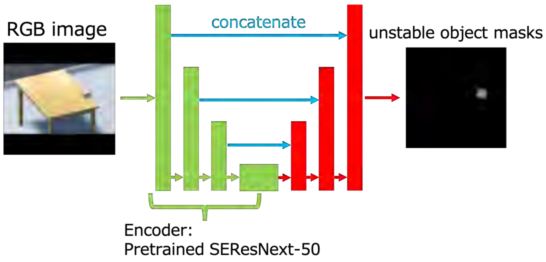

Fig. 1. Network Architecture

$$L(\hat{U}(x), u) = \sum_k \frac{1}{|M_k|} \sum_{(i,j)\in M_k} \left\{u_{(i,j)} - \hat{U}(x)_{(i,j)}\right\}^2 + \frac{\alpha}{|M_0|} \sum_{(i,j)\in M_0} \left\{u_{(i,j)} - \hat{U}(x)_{(i,j)}\right\}^2 \quad (1)$$

In the expression 1, u is the teacher signal, $u_{(i,j)}$ is the pixel value of the teacher signal, the output when the RGB image x is input to the function $\hat{U}$ representing the deep learning model is represented by $\hat{U}(x)$, and the pixel value is represented by $\hat{U}(x)_{(i,j)}$. M_k is the area where the object k appears in the input image x, and $|M_k|$ represents the number of pixels in this area. M_0 is a background area where object k is not displayed in input image x. Similarly, $|M_0|$is the number of pixels in the area where the object is not displayed. The loss function parameter α is used to suppress incorrect output in the background area. The teacher signal u was substituted with the frequency of large potential energy changes in the object area of the input image, and 0 in other area. This is defined as the expression 2.

$$u_{(i,j)} = \begin{cases} \frac{D_k}{N} & ((i,j) \in M_k) \\ 0 & (otherwise) \end{cases} \quad (2)$$

Where D_k is the number of large potential energy changes when an external force of constant magnitude N is applied to the object k.

IV. Experiment

This section describes the experimental method and procedure for evaluating the proposed method.

A. Dataset Used in The Experiment

First, the creation of the dataset used for learning is described. Since it is unrealistic to prepare a lot of real-world images necessary for learning to train the deep learning model in the problem dealt with this time, a simulation environment simulating a room like Fig.2 Was created in Unity and the dataset. Two types of tables such as Fig.3 and four types of objects such as Fig.4 were prepared. A desk and an object were selected at random, and the objects were placed on the desk and photographed at various angles.

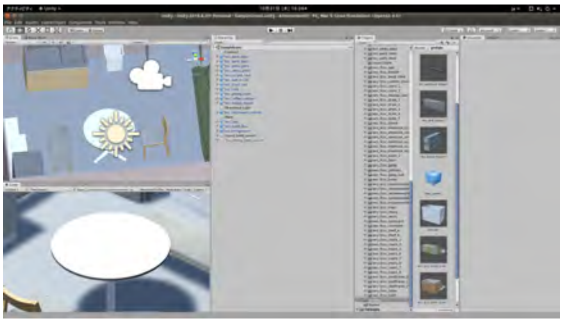

Fig. 2. Simulation Environment

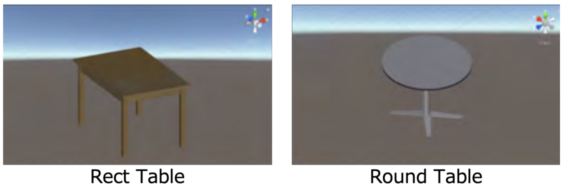

Fig. 3. Tables

In the simulation, the mass of the object is set to a constant $1kg$, and the number of times the object has been dropped is recorded by adding 20 momentum changes of $1.0kg \cdot m/s$ from a random angle. 9945 combinations of 4 types of objects, 2 types of tables and object arrangements were created, and RGB images and teacher signals used for input were acquired. During the simulation, the table size was changed randomly between 0.6 and 1.4 times in the x-axis and z-axis directions. In addition, one object and a table were selected at random, and the objects were arranged randomly. Fig.5 gives an example of the acquired RGB image and teacher signal.

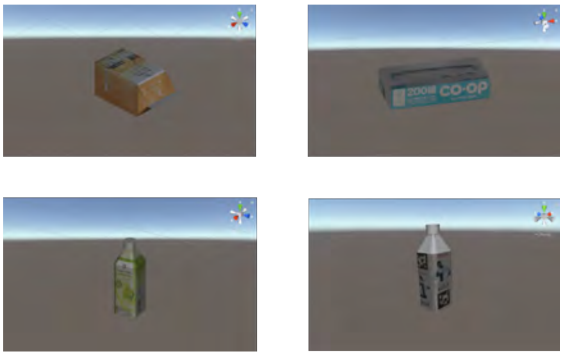

Fig. 4. Objects

	Object Area MSE	Background Area MSE
alpha=1.0	0.017761745138193598	0.0023926512621310378
alpha=1.5	0.019256748776808455	0.0004358408037372762
alpha=3.0	0.029173045109157563	0.00019384744103669927

TABLE I
Comparison of MSE by area

B. Train Deep Learning Model

Next, the training of the deep learning model is described. First, 10% test set was cut out from the data set. The data used for direct learning was cut out from the remaining data at 80%, and the rest was used as data for over-fitting and parameter verification. Each data was divided so that the ratio of the number of four types of objects was the same. Using the loss function defined in the III chapter, the parameter α of the expression 1 was tested for three patterns of $1.0, 1, 5, and 3.0$. We learned 20 epoch (2238 iteration) and tested using the weight of epoch with the smallest loss value of validation data. Adam was used for optimization, and parameters were learned using $alpha = 0.0001, beta1 = 0.9, beta2 = 0.999, eps = 1e-08$.

C. Experimental Result

This chapter describes the results of the experiment described in IV. Figure 6 shows a summary of the results estimated from some test data generated by simulation using the proposed method with the loss function parameter set to $\alpha = 1.0$. The results estimated using the proposed method with the loss function parameter set to $\alpha = 1.5$ are shown in Fig.7, and the result that the loss function parameter is set to $\alpha = 3.0$ are shown in Fig.8.

Next, Fig.9, Fig.10, and Fig.11 summarize the estimation results using the loss function parameters set in $\alpha = 1.0$, $\alpha = 1.5$, and $\alpha = 3.0$ of images taken in the real world. The truth and predict in these figures are corrected with the color map on the right in the figure. Next, TableI shows the mean square error of test data not used for learning generated by simulation of the object region and background region for each parameterα of the loss function.

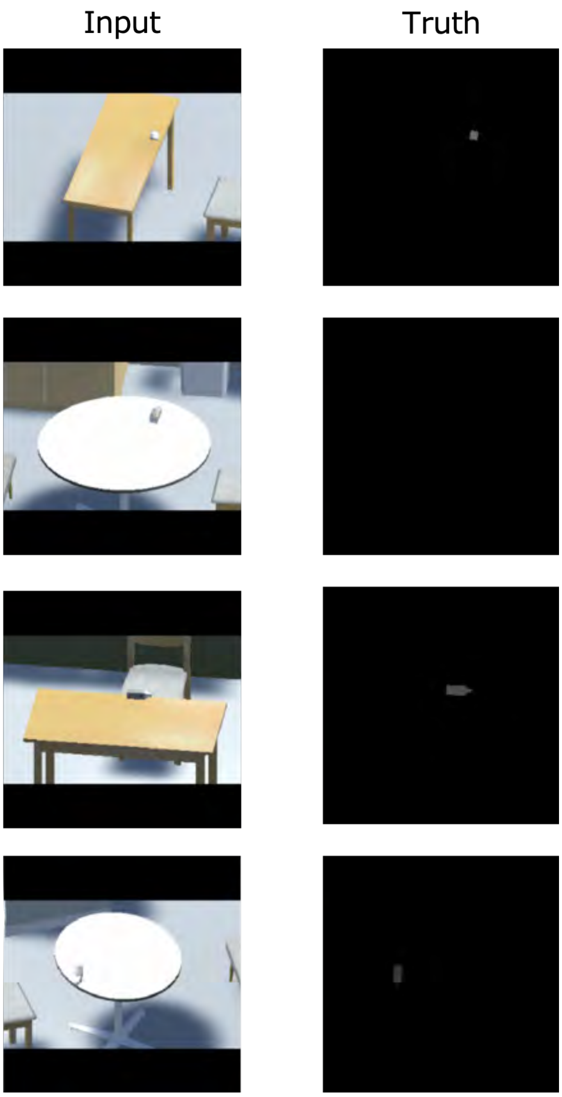

Fig. 5. Data Example

D. Discussion

Looking at Fig.9, Fig.10, and Fig.11, you can see that increasing the value of parameter α of the loss function can reduce noise outside the object area of interest. TableI compares and compares the MSE of the teacher signal with the model output of each object and background area. Compared to $\alpha = 1.0$, you can see that $\alpha = 1.5$ lightly increases the MSE of the object area, but significantly decreases the MSE of the background area. In Figure.11, the output value of the object area is almost zero. Looking at $\alpha = 3.0$ in TableI, the MSE of the object area has increased. From this point, this study considers $\alpha = 1.5$ to be the best parameter for learning.

Figures 9, 10, and 11 show the results of entering the model using images taken under the same conditions as the simulation. Figure 11 is set to $\alpha = 1.5$, but since the noise is hardly removed, the optimal parameters are different in real-world images and other methods such as weak teacher learning can be used to improve accuracy.

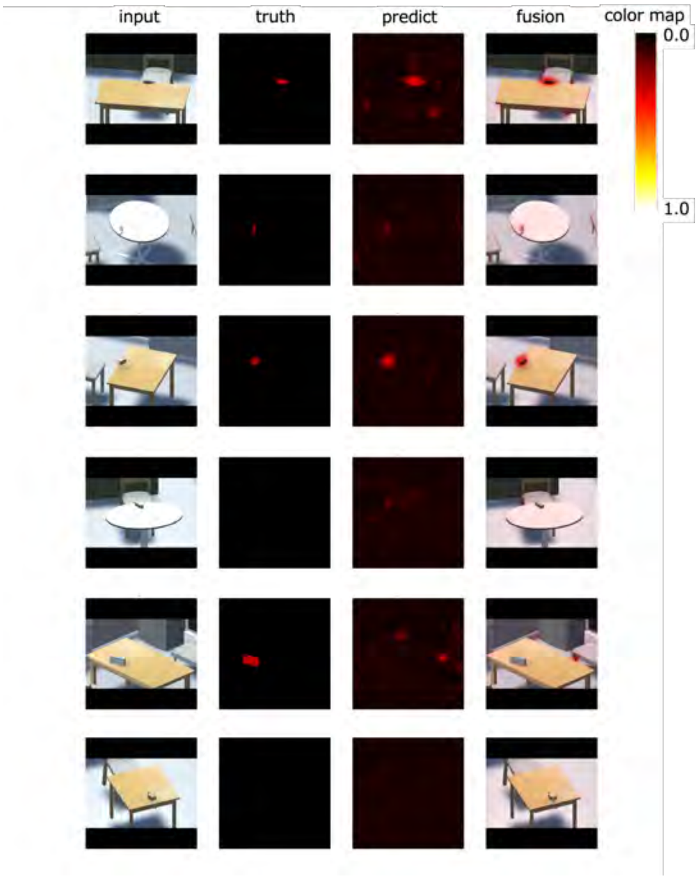

Fig. 6. Simlation Image Results($\alpha = 1.0$)

V. Conclusion

In this study, instability is defined as the frequency at which large energy changes occur when an external force is applied to an object multiple times. A model has been created that uses convolutional deep learning to output a map that assigns object instabilities to each location in the input image. Although the model uses deep learning, it was difficult to prepare many images in the real world to be used for learning, so a data set was created using 3D simulation using Unity. For the purpose of learning the model, we have devised a loss function that takes into account the imbalance between the area of the object region and the area of the background region. The trained model was validated with simulation images that were not used for training, and real-world images in a similar situation to the simulation. As a result, we were able to show that simulation images can be estimated almost accurately, and that real-world images in similar situations can also be estimated.

A. Issues in The Future

In this study, the simple problem of guessing was solved by placing objects on a table. However, it is expected that the physical constraints of objects in real situations will not be guessed correctly because they are not only in the table but

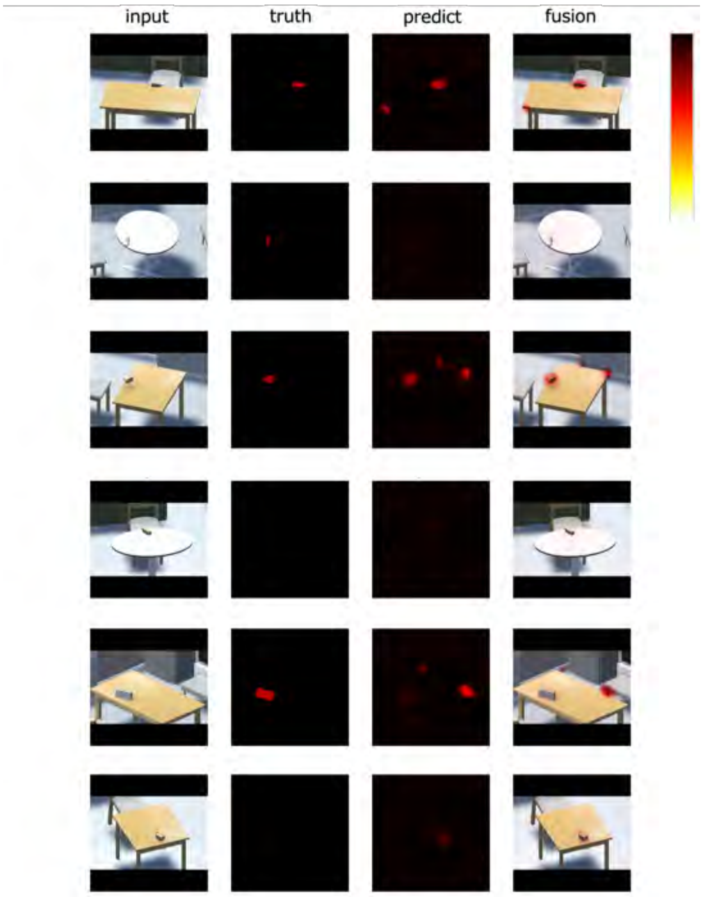

Fig. 7. Simlation Image Results($\alpha = 1.5$)

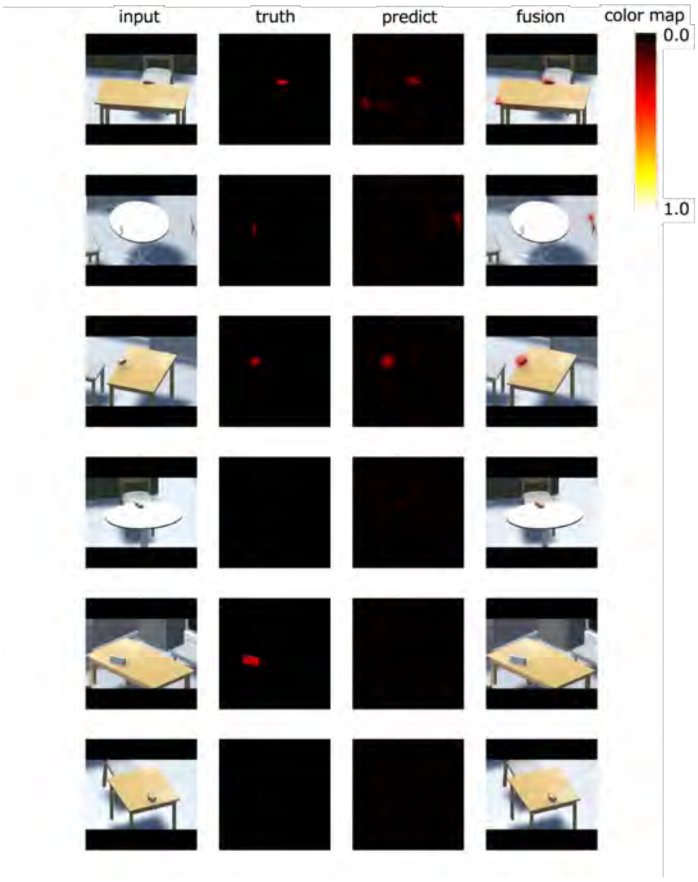

Fig. 8. Simlation Image Results($\alpha = 3.0$)

also in contact with other objects on the table or in complex situations.

B. *Future Perspective*

As a future perspective, this study estimated unstable parts from a single RGB image, but using multiple images from different viewpoints can establish physical relationships between objects in more complex situations. Although the data set used in this study omits friction and elasticity, we aim to build a model that takes these important parameters into account in the real world.

References

[1] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *CoRR*, abs/1709.01507, 2017.

[2] Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. *CoRR*, abs/1603.01312, 2016.

[3] Wenbin Li, Seyedmajid Azimi, Ales Leonardis, and Mario Fritz. To fall or not to fall: A visual approach to physical stability prediction. *CoRR*, abs/1604.00066, 2016.

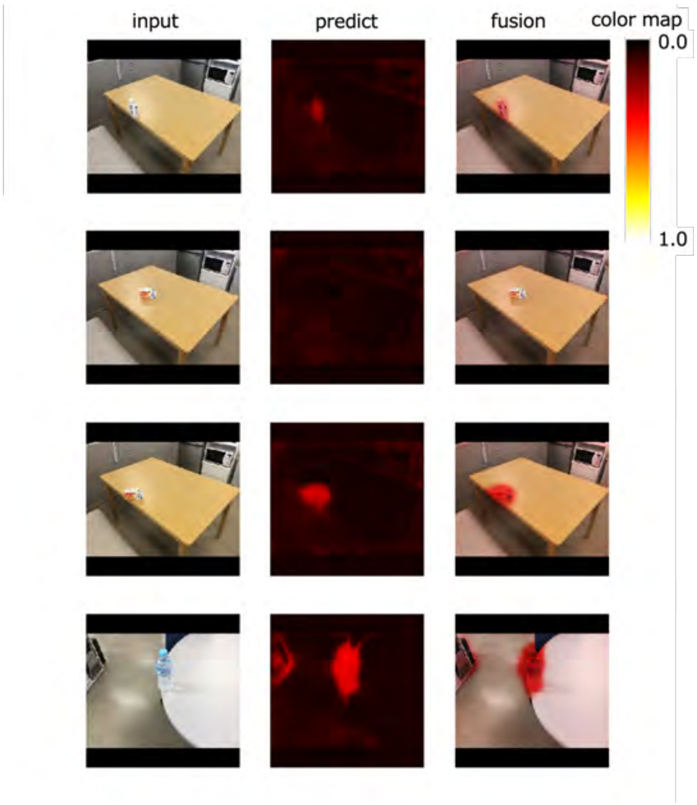

Fig. 9. Real Image Results($\alpha = 1.0$)

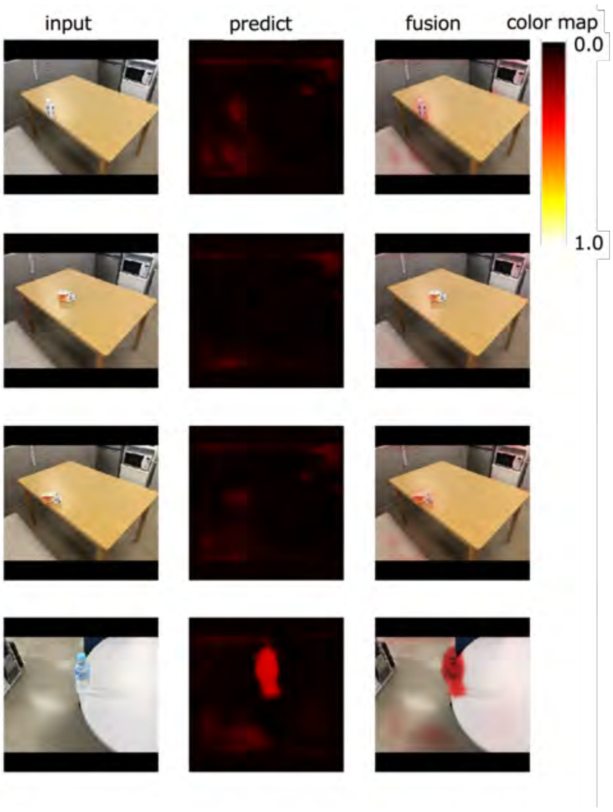

Fig. 10. Real Image Results($\alpha = 1.5$)

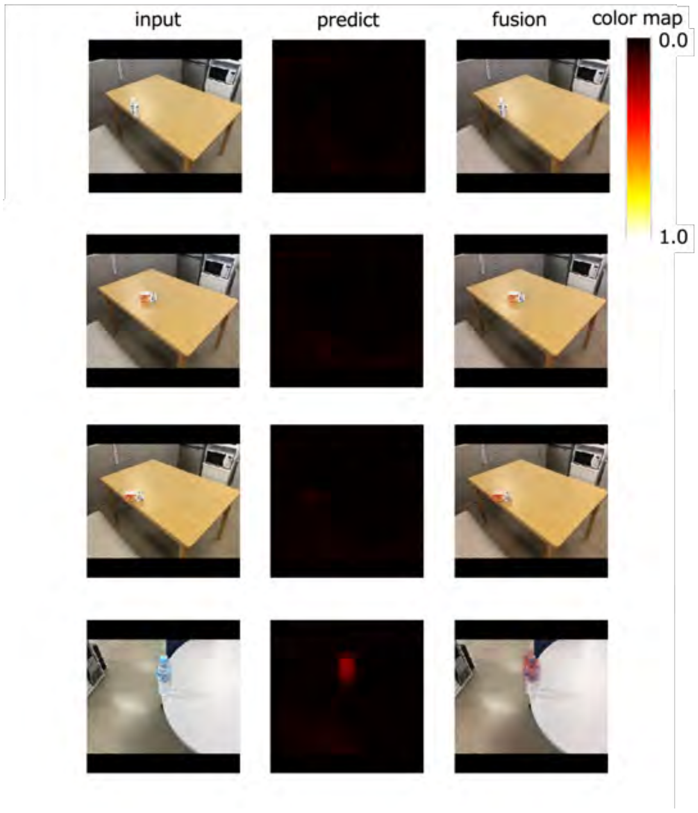

Fig. 11. Real Image Results($\alpha = 3.0$)