Curiosity Guided Fine-Tuning for Encoder-Decoder-Based Visual Forecasting

Yuta KAMIKAWA††, Nonmember, Atsushi HASHIMOTO††, Member, Motoharu SONOGASHIRA†, Nonmember, and Masaaki IYAMA†, Member

SUMMARY An encoder-decoder (Enc-Dec) model is one of the fundamental architectures in many computer vision applications. One desired property of a trained Enc-Dec model is to feasibly encode (and decode) diverse input patterns. Aiming to obtain such a model, in this paper, we propose a simple method called curiosity-guided fine-tuning (CurioFT), which puts more weight on uncommon input patterns without explicitly knowing their frequency. In an experiment, we evaluated CurioFT in a task of future frame generation with the CUHK Avenue dataset and found that it reduced the mean square error by 7.4% for anomalous scenes, 4.8% for common scenes, and 6.6% in total. Some other experiments with the UCSD dataset further supported the reasonability of the proposed method.

**key words:** imbalance data, pixel-wise prediction, visual forecasting

1. Introduction

Pixel-wise prediction by neural networks has recently revealed its usefulness in a wide variety of applications, such as future frame/motion synthesis [1]–[3], semantic segmentation [4], pose estimation [5], and so on. In addition, the auto-encoder (AE), which is another type of pixel-wise prediction architecture, is used as an auxiliary module for multiple purposes. For example, it is used for CycleGANs [6] to preserve some consistencies. The variational auto-encoder (VAE) is used to boost the performance of unsupervised domain adaptation [7], domain generalization [8], disentangled representation learning [9], [10].

An encoder-decoder (Enc-Dec) architecture is a common structure of pixel-wise prediction. Despite its generality, however, the imbalance with a Enc-Dec model is not well investigated. When training an Enc-Dec model wildly, it tends to minimize losses for frequent local patterns of images more than losses for uncommon patterns. This can narrow down the distribution of predictable input patterns. This is undesirable property for many practical applications.

As the first step to solve this fundamental problem, this paper proposes a very simple weighting approach. In the method, a pixel-wise weight map, which we call curiosity, is calculated at a specific epoch in the training. The epoch is manually selected as the timing where the model can roughly deal with the common inputs, but not with uncommon inputs. The weight map is preserved after the epoch and used to weight the prediction loss.

To prove the effectiveness of our concept, we used a simple auto-encoding task with MNIST and another task of future frame generation with two datasets, which were originally proposed for anomaly detection. Uncommon test samples are given explicitly in the datasets. To the authors’ best knowledge, there are no methods that enhance diversity of predictable samples for pixel-wise regression tasks.

The contribution of this paper is threefold. First, we import the idea of curiosity from reinforcement learning [11] to the problem of pixel-wise prediction with a reasonable implementation of visual forecasting application. Second, we present a method of curiosity guided fine-tuning, which is a simple strategy for enhancing the accuracy on uncommon inputs. Third, we experimentally confirm that such a strategy is safe even when training data does not include such uncommon samples. Note that this is an important property for a practical use because we generally do not know whether the training dataset includes uncommon samples or not.

2. Related Work

2.1 Weights for Prediction Loss

To predict diverse input patterns, many methods are proposed to control loss weight at various component levels (e.g., pixel-wise and sample-wise). We overview these weighting strategies used in CV applications for reviewing current knowledge for pixel-wise loss weight.

U-net [4] is one of the most important methods for semantic segmentation. In this method, to achieve better segmentation, a problem-specific weight function was handcrafted to focus more around boundaries of ground truth segments. Another pixel-wise weighting method was proposed by Luo et al. [2] for the problem of future motion forecast. In this task, one problem that should be fixed is the imbalanced observation frequency of pixel motion. That is, there are many more flows with small velocity than with large velocity, mainly due to the imbalance of background/foreground areas. To deal with this problem, they used a weight function that is a linear combination of uniform weight and a normalized reciprocal of frequency. This function is applicable only when the pixel-wise outputs are quantized and normalized reciprocal of frequency. This function is applicable only when the pixel-wise outputs are quantized and normalized reciprocal of frequency. This function is applicable only when the pixel-wise outputs are quantized and normalized reciprocal of frequency. This function is applicable only when the pixel-wise outputs are quantized and normalized reciprocal of frequency. This function is applicable only when the pixel-wise outputs are quantized and...
countable. A similar approach is also used by Eigen and Fergus [12].

The focal loss is a sample-wise weighting strategy proposed by Lin et al. [13]. This is a weighted loss function proposed for an object detection task. The task essentially has a problem of imbalance in the number of foreground/background samples. To overcome this problem, the method uses confidence of classification by decreasing weight when a sample is easy to classify. This strategy is similar to our curiosity-based approach: when an input is less curious, there is less weighting on the output. On the other hand, this strategy is applicable only with classification tasks since the definition of the focal loss assumes a prediction that outputs class probabilities. This paper aims to propose a method that is applicable even with regression, such as auto-encoding and future frame generation.

2.2 Anomaly Detection

Anomaly detection is a task to detect uncommon inputs. It is a counter-task of that treated in this paper in the sense that unpredictability for an input is a useful property for anomaly detection. There is even a method that artificially restrict model to fail with uncommon inputs [14], but such a restriction is not always required. Many papers have reported that anomalous samples can be detected without such an artificial restriction but with wildly trained auto-encoding models [15]–[19] or future forecasting models [9].

The success of these past anomaly detection methods supports the fact that a general implementation of auto-encoder and feature forecasting model suffer from predicting uncommon patterns. Our purpose is opposite to anomaly detection: widening the predictable input distribution as much as possible.

2.3 Visual Forecasting

Visual forecasting is a class of recently-emerging tasks in CV. Zeng et al. [3] summarized current existing visual forecasting tasks as trajectory forecasting [20], activity prediction [21], and future frame/motion generation [1], [2]. Those tasks are challenging because they involve estimation of human intention, which is important for smart human-computer interaction (e.g., an autonomous car must detect humans who may invade the car’s route). Note that forecasting uncommon scenes is often more informative than forecasting common scenes.

One of the key structures for visual forecasting is convolutional long short-term memory (ConvLSTM), which was proposed by Shi et al. [22] for weather nowcasting. ConvLSTM is a convolution filter that has the same hidden nodes and gate functions as LSTM. Owing to its memory-efficient filter-wise LSTM architecture, ConvLSTM can generate future frames with a practical memory size and is often used in Enc-Dec models for future frame generation [1], [2].

In the experiment, we test our method on future frame generation. Future frame generation is a fundamental task for visual forecasting in the sense that it requires the recovery of all visual elements, such as trajectory, motion, and activity, at once. This indicates that the task requires a model to be versatile and, in this sense, it is challenging.

3. Curiosity Guided Fine-Tuning

3.1 Curiosity as a Weight Map

The intention of introducing curiosity is to imitate the human way of observing things. That is, when we see something usual, we pay no attention to it, but when we see something unusual, we pay a lot of attention to it. This idea was first implemented for reinforcement learning in [11], where a curiosity value was calculated as the error between an observation and its time-development result with a selected action. To customize the idea of curiosity for the training of an Enc-Dec model, we define curiosity simply as the inverse of a pixel-wise regularity score, which has been used in anomaly detection methods [15].

Let $W_e(x) \in \mathcal{R}^P$ be a pixel-wise curiosity weight matrix for an input sample $x$ (e.g., image or video clip), at the $e$-th epoch of the training, where $P$ is the number of pixels in the prediction target (e.g., the input image for auto-encoders or a future frame of the input video clip for visual forecasting), respectively. It is calculated as

$$W_e(x) = \frac{L_e(x) - L_{e,\min}}{L_{e,\max} - L_{e,\min}},$$

where $L_e(x) \in \mathcal{R}^P$ is a matrix of pixel-wise losses calculated by a loss function $L$ with an input $x$ at the $e$-th epoch. $L_{e,\min} \in \mathcal{R}$ and $L_{e,\max} \in \mathcal{R}$ are the minimum and maximum values of $L$ all over the pixels of training samples at the $e$-th epoch, respectively. The motivation of curiosity-based weighting is to assign a larger weight to uncommon input patterns, whose frequency is not given explicitly. In a classification task, we can directly count the pixel-wise frequency of each category but cannot with regression. Instead of counting the frequency, Eq. (1) estimates the pixel-wise irregularity of the input sample $x$ on the basis of its loss $L_e(x)$.

Here, $L_e(x)$ potentially consists of two components: frequency-related and unpredictability-related components. The former gets a value when prediction error with the input pattern is caused by its low appearance frequency in training. The latter gets a value simply when the model is well trained but not capable of predicting the patterns. To obtain curiosity, we need to set the weight more based on the former component.

To reduce the undesirable effect of the unpredictability in curiosity calculation, we select the epoch $e$ based on the following assumption: at the very beginning of training, the model can deal with only very easy patterns regardless of its appearance frequency. As the training progresses, at a certain epoch, $L$ depends maximally on the frequency difference. After that, the model will reach a tight convergence.
and then $L$ depends largely on the unpredictability-related component. In this final phase, the frequency-related component is hidden by the unpredictability-related component; thus, we cannot use the loss $L_e$ at this phase for curiosity calculation.

Under this assumption, we select $e$ at the epoch where the frequency difference is highlighted in $L$. After deciding $e$, our proposed method is executed as the following three-step algorithm.

1. Train the model as usual until $e$-th epoch. (pre-training)
2. Calculate $W_e(x)$ for each sample in the training dataset
3. Train the pre-trained model with $L_{curio}$, which is calculated with $W_e(x)$ (fine-tuning)

Here, $L_{curio}$ is defined to emphasize the loss with a large weight of $W_e(x)$ as following:

$$L_{curio}(x) = \lambda L(x) + (1 - \lambda)W_e(x) \circ L(x),$$

where $\lambda$ is a parameter that balances uniform weighting and curiosity-based weighting, and $\circ$ denotes element-wise multiplication. We simply set $\lambda = 0.5$ in all experiments in this paper.

Note that, until $e$-th epoch (the first step of the algorithm), the above training strategy of the proposed method is exactly same with that of Orig. Because the dataset is switched to the same but differently weighted one at the $e$-th epoch, we regard the first step as a kind of pre-training and the latter as a fine-tuning. From this perspective, we call this training strategy curiosity guided fine-tuning (CurioFT).

3.2 Proof of the Concept with MNIST Dataset

To prove the concept shortly, we implemented the method with a simple auto-encoder for MNIST dataset [23]. To emphasize the difference of input pattern, we used digits of ‘3’ and ‘4’ because their handwritten images hardly share their shape: ‘3’ is organized by curves and ‘4’ by straight lines. Note that the number of training samples for ‘3’ and ‘4’ are 6,131 and 5,842, respectively. Hereafter, when we use the digits as normal samples, all those samples are involved in training. When we use them as anomalies, only five samples are involved in training. As the loss function $L$, we always used the mean square error (MSE).

Figure 1 visualizes the difference of error distribution under different epochs and different conditions of irregularity. The left two columns are error distributions obtained with five training samples of ‘3’. When ‘3’ was used as normal samples (the first column of Fig. 1), no samples of ‘4’ were involved in training. In contrast, when it was used as anomalies (the second column), all the samples of ‘4’ were involved in training with the five samples of ‘3’. The right two columns are obtained in the same manner for ‘4’. Note that the scale of errors is different in each error map and thus only the distribution shape can be compared. At the very early epochs, the model can predict only the patterns in the red rectangle in Fig. 1, the difference of loss distribution between normal and anomalous samples becomes obvious, where the loss with the normal samples spreads while that of the anomalous samples relatively concentrates on uncommon local patterns (more significant with ‘4’). Finally, the loss distribution spreads over the digit (more significant with ‘3’). This results experimentally support the assumption that we described in 3.1.

Figure 2 shows the history of (non-weighted) $L(x)$ under the condition where ‘4’ was used as normal samples and ‘3’ as anomalies. From the observation in Fig. 1, the difference in the error distribution is less significant than the opposite case of normal ‘3’ and anomalous ‘4’. In this sense, the choice of normal ‘4’ and anomalous ‘3’ is the more difficult setting. We compared two methods of Orig. and CurioFT, where Orig. is trained without $W_e(x)$ and CurioFT is with $W_e(x)$ where $e = 200$. The training was iterated over 10,000 epochs. For the normal samples, two methods behaved similarly. On the other hand, for the anomalous samples, CurioFT resulted in less MSE than Orig.

It is often expected that a sudden change in loss calculation can drastically affect its optimization quality; this was not in the case with CurioFT. As long as we see in Fig. 3,
which focuses on the history of MSE loss around the 200th epoch, no sudden oscillation was observed. Different from a case of adding a totally new criterion, $W_c(x)$ is calculated by $L$ and the tendency of $L^{curio}(x)$ at 201th epoch was similar to $L(x)$ at 200th epoch. Hence, no sudden impact was observed at the change of loss function with CurioFT.

3.3 Implementation for Future Frame Generation

We test CurioFT on future frame generation. In this section, we describe the definition of the problem and our implementation.

3.3.1 Problem Definition

Let $\{x_{t-M+m}\}_{1}^{M} = \{x_{t-M+m} | 0 < m \leq M\}$ be a sequence of video frames, which is the input. Future frame generation is a problem to predict the visual appearance of frame $x_{t+N}$, where $N$ is the number of frames after $x_t$, the last frame of the input $\{x_{t-M+m}\}_{1}^{M}$.

3.3.2 The Network Architecture

Although there are several different architectures for this task [1]–[3], their common element is a pair of an encoder and a decoder that are connected via ConvLSTM layers [22]. To validate the effectiveness of CurioFT for those different architectures, we prepared a minimal Enc-Dec model with ConvLSTM (Fig. 4). For this purpose, we customized the model used in [24], which is designed to generate future frames for anomaly detection.

In this network, the output is a sequence of generated future frames, which we describe as $\{\hat{x}_t\}_{0}^{N} = \{\hat{x}_{t+n} | 0 < n \leq N\}$. $\{\hat{x}_t\}_{0}^{N}$ consists of an estimation of $x_{t+N}$ and all the interpolating frames between $x_t$ and $x_{t+N}$.

Each input frame $x_{t-M+m}$ is fed to a three-layered convolution module $f_{conv}$ (Fig. 5), and we obtain $x'_{t-M+m} = f_{conv}(x_{t-M+m})$. The sequence $\{x'_t\}_{1}^{M}$ is fed to ConvLSTMs, and further compressed into a single latent feature vector $z_t$. $z_t$ is then fed to the decoder part, which has another set of ConvLSTMs and the deconvolution module $f_{deconv}$ (Fig. 6). Finally, $\{\hat{x}_t\}_{0}^{N}$ is obtained.

Note that each input frame $x_{t-M+m}$ is formatted to a 224×224 gray-scale image, which is a common setting in anomaly detection models [19], [24].

1 Convolution Module

The convolution module $f_{conv}$ (Fig. 5) consists of three convolution blocks, each of which has layers of 5×5 convolution, activation by ReLU, and 2×2 max-pooling, as in a previously proposed anomaly detection model [24]. While the numbers of output channels at the convolution layers were...
originally 256, 256, and 512, we use a more compact setting of 128, 128, and 256, as we observed no accuracy decrease with this in preliminary experiments. The padding parameter is set to zero at all convolution layers, as in the original ConvLSTM paper [22]. We also add a batch normalization layer after every convolution layer for better convergence.

(2) ConvLSTM

We set the filter size of ConvLSTM to be 5×5 and \( N = M = 10 \), as in the anomaly detection model [24]. The number of channels is the same as the output of the convolution module, i.e., 256.

(3) Deconvolution Module

The deconvolution module \( f_{\text{decon}} \) (Fig. 6) consists of three deconvolution blocks, each of which has a 5×5 convolution layer, activation by ReLU, and 2×2 upsampling layer with stride 2, the same as the anomaly detection model [24]. The numbers of output channels at these convolution layers are set to be symmetrical with those in the convolution module, i.e., 256, 128, 128. We also add a batch normalization layer after every convolution layer.

3.3.3 Weight Calculation

In the future frame generation, we have multiple output frames \( \{\hat{x}_{t+1}\}_{t=1}^{N} \). Hence, it is not very obvious with which output frame we calculate \( W_e((x_t - M + m)_{t=1}^{M}) \). There are several options for it: e.g., averaging \( L_e(\hat{x}_{t+1}) \) among \( \{\hat{x}_{t+1}\}_{t=1}^{N} \), using the loss with its first frame \( L_e(\hat{x}_{t+1}) \) or the last frame \( L_e(\hat{x}_{t+N}) \), and so on. From these options, we decided to use \( L_e(\hat{x}_{t+N}) \) for \( W_e((x_t - M + m)_{t=1}^{M}) \) due to its simplicity of analysis and importance for the application.

4. Experiments

We tested the proposed method with the task of future frame generation. This section explains the details of our experimental setting and results with multiple datasets.

4.1 Dataset

In the experiment, we used the following two datasets that are originally prepared for anomaly detection because the test data consist of common (normal) and uncommon (anomalous) scenes.
CUHK Avenue [25] consists of 16 training video clips and 21 test video clips. Each video clip is about one minute long with a frame size of 640×360 pixels. Test video contains some anomalous activities by pedestrians, such as running, loitering, and throwing. Training data is uncensored and may involve above anomalous activities by chance.

UCSD pedestrian [26] comprises two different datasets: UCSDped1 and UCSDped2. UCSDped1 contains 34 training video clips and 36 test video clips. Each clip has about 200 frames, and the frame size is 238×158 pixels. UCSDped2 contains 16 training video clips and 12 test video clips. Each clip has about 170 frames, and the frame size is 360×120. Test video clips contain anomalous scenes with bicycles, skateboarders, wheelchairs, and vehicles on a sidewalk. Training data is censored and the above anomalous activities are removed from training data, which is an important difference between CUHK and UCSD datasets.

Due to its censored property, we used UCSD as a control group: the proposed method is not expected to improve its accuracy on the anomalous scenes with this dataset. In addition, we check whether or not the proposed method degrades the prediction quality in use with such a clean dataset. Moreover, we check whether or not the proposed method degrades the accuracy on the anomalous scenes with this dataset. In this sense, we can say that we can safely use the proposed method without checking the existence of uncommon scenes in the training data.

4.2 Variations of the Proposed Method

To analyze the property of the proposed method, we prepared CurioFT with several different schedules of \( e \) as following:

CurioFT(10) Updating \( W_e(x) \) only at the end of the tenth epoch.

CurioFT(20) Updating \( W_e(x) \) only at the end of the twentieth epoch.

CurioFT(10-20) Updating \( W_e(x) \) at the end of tenth epoch, then update it again at the end of the twentieth epoch.

The difference between CurioFT(10) and CurioFT(20) is their timing of calculating \( W_e(x) \). CurioFT(10-20) has two scheduled timings. The target of this configuration is to evaluate the possibility of two-stage updates. After increasing weights for uncommon input patterns, the model will fit better for those patterns. Then, the recalculation may find other left-over uncommon patterns that are not weighted at the first stage.

To decide the above schedules, we checked the history of \( L \) obtained by Orig. with the CUHK Avenue dataset, which is shown in Fig. 7. The training stage did not reach a tight convergence at the tenth epoch but becomes more stable at the twentieth epoch (Fig. 7). Hence, CurioFT needs to calculate the curiosity weights before convergence but also after learning common input patterns. The tenth and twentieth epochs were chosen from this observation. We also confirmed that the history of training loss for UCSDped1 and USCDped2 exhibited the same trend as those in Fig. 7.

In addition, we prepared two more variations that periodically update \( W_e(x) \). We further investigated the MSE scores by dividing the test data into hard and easy: hard data is the scenes annotated as anomalies, and easy data is the rest.
These annotations are given by the third party of the dataset providers. For our purpose of enhancing the diversity of predictable input, it is more preferable when the method is more effective on the hard data. Here, CurioFT(10) and CurioBat, which performed best in the categories of scheduled and periodical variants, respectively, were compared with Orig.

Table 2 shows the results on the three methods with the ratio against the MSE score for Orig. CurioFT(10) reduced MSE by 7.4% for the hard data compared with Orig. CurioFT(10) also reduced MSE by 4.8% even for the easy data, and thus, 6.6% in total. This may be caused by the increased generalization ability of the curiosity-based weighting strategy. In contrast, CurioBat performed even worth than Orig. on the easy data. One possible reason is a kind of overfitting caused by the noisy distribution of $W(x)$ after a certain level of convergence. As shown in Fig. 1, the more the training proceeds, the noisier the weights distribute. The locations with large weights might be the pixels that are difficult to predict, and putting heavier weights on such unpredictable locations can cause overfitting and can decrease the generalization ability.

Figure 8 shows another effect of CurioFT(10). Namely, you can see periodical spikes with Orig, but not with CurioFT(10). The spikes are typically observed when the model parameters leave from local minima by stimuli from sudden large loss values. This would imply that these stimuli were yielded by anomalous samples appearing when least expected for Orig. This was not seen with CurioFT(10) owing to its weighting strategy.

In addition, Tables 3 and 4 show the comparison by the same protocol but with the UCSDped1 and UCSDped2 datasets. CurioBat performed the best with these clean datasets owing to its emphasized MSE loss. In contrast, CurioFT(10) did not perform significantly well. Because the method is designed to improve the performance collaboratively with uncommon inputs in the training dataset, this is an expected result.

### 4.3.2 Qualitative Evaluation

The main aim of the proposed method is to encode and decode uncommon inputs. To focus on concrete examples of uncommon input patterns, we visualized the error for anomalous scenes temporally and spatially. For temporal examination, we plotted the frame-wise MSE scores for video clips #01 in the CUHK Avenue dataset (9). This video clip contains an anomalous scene annotated as an “abnormal object.” Both CurioFT(10) and CurioBat were less affected by this anomalous event than Orig, as seen from their smaller MSE values around the anomaly.

![History of training loss on CUHK with Orig. and CurioFT(10).](image)

![Image](image)
but a small error with CurioFT(10) and CurioBat. This indicates that the proposed method was able to learn uncommon scenes as well as common scenes.

Figures 11 and 12 show the temporal and spatial examinations, respectively, for another video clip of #20 in CUHK, which is annotated as “throw object.” MSEs on this scene by CurioFT(10) and Orig. were better than that by CurioBat. However, their spatial visualizations (Fig. 12) show little qualitative difference. This is simply because no actions similar to the anomaly in this clip (elaborately throwing paper documents) were contained in the training data, whereas actions similar to the anomaly in the scene of clip #01 might be observed by chance in natural scenes.

4.3.3 Additional Evaluation on modified UCSD

As an additional experiment, we prepared a new training/test division of UCSD datasets, UCSDped1* and UCSDped2* (or UCSD* as an abbreviation). From each of UCSD datasets, we randomly selected half of the test video clips and moved them to its training dataset. In other words, UCSD and UCSD* are controlled so as to exclude and include anomaly in the training set, respectively.

Tables 5 and 6 show the results in UCSDped1* and UCSDped2*, respectively. In the tables, the difference from those in the (non-modified) UCSD dataset (shown in Tables 3 and 4) is given in the rows of “Δ.” Because CurioBat has large gains (from Orig.) with UCSDped1 as shown in Table 3, it has also performed better than CurioFT(10), especially with the easy samples in UCSDped1*. On the other hand, when we focus on the improvement in both UCSDped1* and UCSDped2* from the clean setting of UCSD, CurioFT(10) has decreased MSE on hard samples but CurioBat has increased the errors.

The results with CUHK, UCSD, and UCSD* imply that, with wilder training data, CurioFT(10) can make the model predict a wider variety of input while not degrading the prediction quality for common samples but CurioBat does not always work well.
5. Conclusion

In this paper, we proposed CurioFT, a method that makes an Enc-Dec model applicable to uncommon input patterns. To let the model focus on uncommon samples without knowing their frequency, we imported the idea of curiosity from reinforcement learning. In our case, it is simply calculated as an estimation error normalized by the minimum and maximum values. Using the curiosity as a pixel-wise weight on decoder’s regression loss, we confirmed that our implementation in a curiosity-guided fine-tuning scheme (pre-training without curiosity and post-training with curiosity) achieves the best performance when applying an Enc-Dec model to a dataset with anomalous scenes.

In the experiment, we evaluated our curiosity guided fine-tuning method (CurioFT) in the task of future frame generation. On the CUHK Avenue dataset, we found that CurioFT reduced the average error by 7.4% for uncommon scenes, 4.8% for common scenes, and 6.6% in total. This result was further supported by several controlled experiments with the UCSD dataset.

References

[1] N. Srivastava, E. Mansimov, and R. Salakhudinov, “Unsupervised learning of video representations using LSTMs,” International conference on machine learning, pp.843–852, 2015.
[2] Z. Luo, B. Peng, D.-A. Huang, A. Alahi, and L. Fei-Fei, “Unsupervised learning of long-term motion dynamics for videos,” The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.
[3] K.-H. Zeng, W.B. Shen, D.-A. Huang, M. Sun, and J.C. Niebles, “Visual forecasting by imitating dynamics in natural sequences,” 2017 IEEE International Conference on Computer Vision (ICCV), pp.3018–3027, IEEE, 2017.
[4] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” International Conference on Medical image computing and computer-assisted intervention, pp.234–241, Springer, 2015.
[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields,” CVPR, 2017.
[6] J.-Y. Zhu, T. Park, P. Isola, and A.A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” Proceedings of the IEEE International Conference on Computer Vision (ICCV), Oct. 2017.
[7] Y.-J. Li, F.-E. Yang, Y.-C. Liu, Y.-Y. Yeh, X. Du, and Y.-C. Frank Wang, “Adaptation and re-identification network: An unsupervised deep transfer learning approach to person re-identification,” Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp.172–178, 2018.
[8] Y. Liu, Z. Wang, H. Jin, and I. Wassell, “Multi-task adversarial network for disentangled feature learning,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
[9] Y. Liu, F. Wei, J. Shao, L. Sheng, J. Yan, and X. Wang, “Exploring disentangled feature representation beyond face identification,” The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
[10] M. Zwicker, Q. Hu, A. Szabo, T. Portenier, and P. Favaro, “Disentangling factors of variation by mixing them,” The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
[11] D. Pathak, P. Agrawal, A.A. Efros, and T. Darrell, “Curiosity-driven exploration by self-supervised prediction,” ICML, 2017.
[12] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” Proceedings of the IEEE International Conference on Computer Vision, pp.2650–2658, 2015.
[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” 2017 IEEE International Conference on Computer Vision (ICCV), pp.2999–3007, IEEE, 2017.
[14] P. Perera, R. Nallapati, and B. Xiang, “Ocean: One-class novelty detection using gans with constrained latent representations,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.
[15] M. Hasan, J. Choi, J. Neumann, A.K. Roy-Chowdhury, and L.S. Davis, “Learning temporal regularity in video sequences,” The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.
[16] Y.S. Chong and Y.H. Tay, “Abnormal event detection in videos using spatiotemporal autoencoder,” International Symposium on Neural Networks, 2017.
[17] R. Chalapathy, A.K. Menon, and S. Chawla, “Robust, deep and inductive anomaly detection,” European Conference on Machine Learning and Principles and Practice of Knowledge Discovery, 2017.
[18] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, “Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes,” IEEE Transactions on Image Processing, 2017.
[19] W. Luo, W. Liu, and S. Gao, “Remembering history with convolutional lstm for anomaly detection,” IEEE International Conference on Multimedia and Expo (ICME), July 2017.
[20] N. Lee, W. Choi, P. Veznaza, C.B. Choy, P.H.S. Torr, and M. Chandraker, “DESIRE: Distant future prediction in dynamic scenes with interacting agents,” 2017.
[21] C. Vondrick, H. Pirsiavash, and A. Torralba, “Anticipating visual representations from unlabeled video,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.98–106, 2016.
[22] S. Xingjian, Z. Chen, H. Wang, D.Y. Yeung, W.K. Wong, and W.C. Woo, “Convolutional lstm network: A machine learning approach for precipitation nowcasting:” Advances in neural information processing systems, pp.802–810, 2015.
[23] Y. LECUN, “The mnist database of handwritten digits,” http://yann.lecun.com/exdb/mnist/
[24] J.R. Medel and A. Savakis, “Anomaly detection in video using predictive convolutional long short-term memory networks,” arXiv preprint arXiv:1612.00390, 2016.
[25] C. Lu, J. Shi, and J. Jia, “Abnormal event detection at 150 fps in matlab,” IEEE International Conference on Computer Vision (ICCV), 2013.
[26] V. Mahadevan, W. Li, V. Bhadodia, and N. Vasconcelos, “Anomaly detection in crowded scenes,” The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.
[27] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.
[28] T. Tieleman and G. Hinton, “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” 2012.
Yuta Kamikawa received B.S. degree in Engineering from Doshisha University, Kyoto, Japan in 2017 and received M.S. degree in informatics from Kyoto University, Kyoto, Japan in 2019. His current research interests include visual forecasting, action recognition, action anticipating.

Atsushi Hashimoto is currently a senior researcher at Omron SINIC X Corp, from 2018. He received B.S. degree in Engineering, M.S. and Ph.D. degrees in Informatics from Kyoto University, Japan, in 2005, 2008, and 2013, respectively. He has been working on activity-forecast-based human-computer interaction. He is a member of IEEE, IEICE, and IPSJ.

Motoharu Sonogashira received his B.S. degree in engineering, and M.S. and Ph.D. degrees in informatics from Kyoto University, Kyoto, Japan in 2013, 2015, and 2018, respectively. He is currently a researcher at the Academic Center for Computing and Media Studies, Kyoto University. His research interests include image restoration, blind deconvolution, super-resolution, optical flow, and variational Bayes.

Masaaki Iiyama received the B.S. degree in engineering informatics, and the M.S. and Ph.D. degrees in informatics from Kyoto University, Kyoto, Japan, in 1998, 2000, and 2006, respectively. Currently, he is an Associate Professor with the Academic Center for Computing and Media Studies, Kyoto University. His research interests include 3-D shape reconstruction, reflectance and scattering modeling, and computational photography.