Improving work detection by segmentation heuristics pre-training on factory operations video

Shotaro Kataoka¹*, Tetsuro Ito², Genki Iwaka², Masashi Oba², Hirofumi Nonaka²

1 Department of Science of Technology Innovation, Nagaoka University of Technology, Nagaoka, Niigata, Japan, 2 Department of Information Management Systems Engineering, Nagaoka University of Technology, Nagaoka, Niigata, Japan

* s183347@stn.nagaokaut.ac.jp

Abstract

The measurement of work time for individual tasks by using video has made a significant contribution to a framework for productivity improvement such as value stream mapping (VSM). In the past, the work time has been often measured manually, but this process is quite costly and labor-intensive. For these reasons, automation of work analysis at the worksite is needed. There are two main methods for computing spatio-temporal information: by 3D-CNN, and by temporal computation using LSTM after feature extraction in the spatial domain by 2D-CNN. These methods have high computational cost but high model representational power, and the latter has low computational cost but relatively low model representational power. In the manufacturing industry, the use of local computers to make inferences is often required for practicality and confidentiality reasons, necessitating a low computational cost, and so the latter, a lightweight model, needs to have improved performance. Therefore, in this paper, we propose a method that pre-trains the image encoder module of a work detection model using an image segmentation model. This is based on the CNN-LSTM structure, which separates spatial and temporal computation and enables us to include heuristics such as workers’ body parts and work tools in the CNN module. Experimental results demonstrate that our pre-training method reduces over-fitting and provides a greater improvement in detection performance than pre-training on ImageNet.

1 Introduction

Recognition of employee activities in the production process is gaining attention [1–12]. Bell developed a conceptual framework about artificial intelligence data-driven Internet of Things systems [1]. Elvira et al carried out to evaluate and analyze artificial intelligence-supported workplace decision method such as big data algorithmic analytics, sensory and tracking technologies [2]. Ren et al carried out a comprehensive review of big data analytics throughout product lifecycle to support sustainable smart manufacturing [3]. Smith analyzes the outcomes of cyber-physical manufacturing systems such as real-time sensor networks, and Internet of
Things smart devices [4]. Clarke drew on a substantial body of theoretical and empirical research on big data-driven manufacturing, and to explore this, I inspected, used, and replicated survey data from Accenture, Capgemini, PwC, Software AG, and we.CONNECT, performing analyses and making estimates regarding sensing, smart, and sustainable technologies [5]. Nica et al performed analyses and made estimates regarding the link between smart connected sensors, industrial big data, and real-time process monitoring to examine cyber-physical system-based manufacturing [6]. Leng et al presents a digital twin-driven manufacturing cyber-physical system (MCPS) for parallel controlling of smart workshop under mass individualization paradigm [7]. Hyers et al analyzed big data-driven decision-making processes, Industry 4.0 wireless networks, and digitized mass production to evaluate utility of cyber-physical system of smart factories [8]. Keane et al performed structural equation modeling to analyze cognitive automation, big data-driven manufacturing, and sustainable industrial value creation using data collected from BCG, CompTIA, Deloitte, IW Custom Research, Kronos, McKinsey, PAC, PwC, and Software AG [9]. Miricica et al. analyzed and estimated regarding the main areas of interest for manufacturers within Industry 4.0 and the effects of Industry 4.0 on the workforce [10]. Graessley et al performed analyses and making estimates regarding how Industry 4.0 is delivering revenue, cost and efficiency gains and top technologies being considered in-line with organizations’ strategic plan [11]. Meyers collected data from Bright & Company, Corporate Research Forum, Deloitte, Management Events, McKinsey, and Top Employers Institute and estimated data-driven approaches in industry [12]. Among them, employee behavior recognition in the manufacturing industry is attracting increasing attention from both industry and academia [13]. For example, to improve management and effectiveness of employees’ learning processes, Patalas-Maliszewska et al. [14] used CNN-SVM to extract features from video material concerning each work activity, and comparison with the features of the instruction picture. The importance of real-time process monitoring and analysis of employee behavior based on various sensors and other devices has been increasing from both industry and academia [15–18]. In such a situation, the measurement of work time for individual tasks by using video and other methods has made a significant contribution to a framework for productivity improvement such as value stream mapping (VSM) [19] which is a method for reviewing and improving productivity by dividing tasks into two categories—value-adding tasks and non-value-adding tasks—and by analyzing the time spent on the tasks [20–24]. Monteiro et al performed case study analysis of lean tool. In the case of metalworking company project, flowcharts and VSM (Value Stream Mapping) approaches were conducted to identify and map key processes and improved setup times. As a result, setup times were reduced in 40% on the vertical milling machine of the company, and in 57% on the horizontal milling machine [20]. UlfKTeichgraber et al. have empirically demonstrated that VSM can be applied to identify NVAs and improve productivity in the manufacturing process of endovascular stents. Specifically, they found that 5 out of 15 processes were unnecessary and could be eliminated by using VSM [21]. Mareike Heinzen et al. clarified the effect of co-location in the new drug development process through VSM. The results suggest that communication increases especially in child-located teams [22]. Heravi et al. used value stream mapping (VSM) as a lean method in the manufacturing process of prefabricated steel frames (PSF) and showed that 34% of the lead time could be reduced [23]. Wang et al. developed a training method combining VR and VSM, and showed that it can contribute to the efficiency of the construction process [24]. It has made a significant contribution to improve work efficiency and has been successfully introduced to a variety of industries [25–28]. SMED provides the most complete and detailed procedures available anywhere to transform the manufacturing environment in a way that speeds production and makes small lot inventory feasible [25]. Zhang et al. found that lean manufacturing contributes to knowledge creation through a
questionnaire survey [26]. Sousa et al. applied VSM to improve production efficiency in the cork industry, and found that production time could be reduced by 43% [27]. Adanna et al. showed that VSM can reduce set-up time by 45% when applied to automotive equipment manufacturing [28]. An important part of implementing such a VSM is to measure work times in order to visualize them. For this purpose, video analysis is often used. For example, [29] used video to extract the cycle time and other factors that are important in analyzing the value added by a task. However, this type of work time analysis requires experts who are well versed in the work process and also requires considerable labor and involves economic costs because of the need to check the video footage during the work. Thus, the implementation hurdles are high, especially in small and medium-sized companies. For these reasons, automation of work analysis at the worksite is needed. In the mainstream research on this topic, sensors attached to the operator have traditionally been used for collecting measurement data. However, this method has certain problems, such as the burden of attaching a sensor to the worker, the resulting limitation on the worker’s flexibility of movement, and the high cost of introduction. Another approach is to detect action from videos; among such methods, convolutional neural network (CNN)–based methods are the most popular. There are two main methods for computing spatio-temporal information: by 3D-CNN, and by temporal computation using LSTM after feature extraction in the spatial domain by 2D-CNN. The former has high computational cost but high model representational power, and the latter has low computational cost but relatively low model representational power. In the manufacturing industry, the use of local computers to make inferences is often required for practicality and confidentiality reasons, necessitating a low computational cost, and so the latter, a lightweight model, needs to have improved performance. In addition, both methods tend to overlearn when the training dataset is small, and it is difficult to prepare a large dataset with a large number of work annotations for actual applications in the manufacturing industry. Therefore, the automation of work time analysis is a highly important topic of research. However, conventional video analysis and sensor-based methods are difficult to apply to the analysis of work hours in the manufacturing industry in environments where the granularity of work is not uniform and workers can move around freely. This has posed a major problem, especially in applying the system to manufacturing sites with high-mix low-volume production. In this paper, therefore, we propose a method using machine learning to train a model to extract the features of a worker’s position and posture from a set of video clips taken in a factory and to then use the model to detect the content of the worker’s work.

Our contributions can be summarized as follows:

1. We propose the printing factory video dataset, which is a dataset of videos taken from a third person’s point of view using a fixed camera and containing second-by-second work content annotations. The dataset is available at https://www.kaggle.com/datasets/shotarokataoka/factory-operation-video-dataset.

2. We propose a method for considering heuristics in a model that uses an image segmentation task as a pre-training method. To our best knowledge, there is no research on the application of such a method to work videos in factories. In particular, the video proposed in this study has completely new characteristics, as described in Table 2, and it would be of great academic value to confirm the effectiveness of heuristic pre-training on this video.

3. We present our findings that with our dataset, the proposed pre-training method significantly reduced over-fitting and improved the F1 score on the test set by 0.2906 points over the ImageNet pre-training method. The code and raw result are available at https://github.com/ShotaroKataoka/WorkDetectionFactoryPretrain.
The structure of this paper is as follows. In section 3, we review related works. Then, in Section 4 we define the task. In section 4, we explain about proposal method. In section 5, we explain about experiment. In section 6, we showed the experiment result. Finally, in Section 5, we discussed about the result. The paper finishes with some conclusions and points out some future work.

2 Related work

In order to automate the work process analysis, it is necessary to estimate the work content and the time it took to perform it from some data. In this paper, we refer to this sequence of steps as "work detection." There are two approaches to work detection: using a sensor (e.g., an accelerometer) to collect data on the physical movement of a worker and using a camera to record videos of the work. In this section, we describe previous studies on sensor-based and video-based task estimation, video datasets, and methods of action recognition.

2.1 Action recognition from sensors vs. videos

With the development of sensing devices, research on motion recognition using sensor information has been attracting attention [30–50]. Peterek compared different algorithms for human physical activity recognition from accelerometric and gyroscopic data which are recorded by a smartphone [30]. Chang et al. proposed a hierarchical hand motion recognition method based on one inertial measurement unit (IMU) sensor and two surface electromyogram (sEMG) sensors. An SVM classifier was used for the EMG signal and a decision tree classifier was used for the IMU signal [31]. Ronao developed a two-stage continuous hidden Markov model (CHMM) approach for the task of activity recognition using accelerometer and gyroscope sensory data gathered from a smartphone [32]. Uddin et al. propose a model that extracts feature vectors from multi-sensor data such as ECG based on Gaussian kernel-based principal component analysis and Z-score normalization, and then trains them with CNN [33]. Wang et al. also use CNN to analyze human activity [34]. Lee et al. have developed a method to analyze time-series acceleration signals using 3D acceleration sensors of smart phones using a hierarchical hidden Markov model [35]. Ravi et al. found that using Plurality Voting showed good motion recognition performance even with a single accelerometer [36]. Kwapisz also used a single accelerometer to recognize human activities [37]. Casanova et al. have shown that recognition of motion features from acceleration data can be applied to biometric authentication [38]. Albinali proposed a motion recognition algorithm for individuals with the Autism Spectrum Disorder (ASD) by combining the Fourier transform of acceleration time series with decision trees [39]. Khan developed a model that combines an artificial neural network (ANN) and an autoregressive model (AR) to recognize motion from acceleration data [40]. Kaghyan et al. proposed an algorithm that uses accelerometer data and the K-NN algorithm to identify common activities performed by the user [41]. Brezmes also used K-NN to recognize human activities [42]. Mitchell et al. have developed a method for sports motion recognition from accelerometers using Discrete Wavelet Transform (DWT) and SVM [43]. Subasi et al. showed that an ensemble classifier based on the Adaboost algorithm using acceleration data can significantly improve the performance of automatic human activity recognition (HAR) [44]. Wang et al. proposed a real-time, hierarchical model to recognize both simple gestures and complex activities using a wireless body sensor network [45]. Garcia-Ceja used hidden Markov models and conditional probability fields to recognize actions based on time series of sensor information [46]. Hossain et al. proposed a method that combines accelerometers with LoRaWAN sensors to recognize basic actions such as walking, staying, and running by KNN and LDA [47]. Ryu et al. showed that accelerometer information can be used...
to classify a simple routine task consisting of four subtasks using a multi-class SVM [48]. Kim et al. analyzed IMU data by dynamic time warping (DTW) to classify the movements of construction machines [49]. These studies are of general action recognition. Sensors have been used in a number of studies, such as in an analysis of work motions in bicycle maintenance using ultrasonic and IMU sensors [51], an analysis of line motions using only IMU sensors [52], the estimation of hammering and other motions on an assembly line using wrist-mounted IMU sensors [53], and the estimation of work types using wristwatch IMU sensors [54, 55]. Modern sensor-based work recognition methods that use deep learning include those proposed by [56, 57], which use a convolutional neural network (CNN) to analyze data from IMU sensors.

Such methods use IMU sensors and other wearable sensors primarily to sense the movement of a specific body part such as the hand and are intended for the analysis of assembly work in which the worker is stationed at a workbench and is not required to walk around. However, it is difficult to apply a sensor-based method to an environment in which workers can move freely because 1) parts of the body that are not equipped with sensors can move, 2) the direction of movement ranges widely, and 3) the body position must be accurately estimated. To capture workers’ movements in such high-flexibility environments, expensive devices such as multi-purpose, infrared, and line-of-sight sensors may be required. Under such conditions, it is difficult to sense the work as performed in a natural state because of the burden of mounting the sensor. Therefore, the application of sensor-based work recognition methods has been limited to the analysis of production lines in which only specific tasks with fixed actions are performed.

Unlike sensors, video is a rich source of information and can extract the features of the work even if the degree of flexibility in the worker’s movement is high; in addition, locations can be accurately identified. Furthermore, the use of a small camera has the advantage of facilitating the construction of an inexpensive system.

Table 1 summarizes the characteristics and capabilities of sensors and videos for action recognition.

As described in Section 3 below, in a printing factory such as the one used for the work analysis in this study, the worker has a high degree of movement flexibility; therefore, gathering data by using a sensor was not considered suitable from the standpoint of the burden on the worker. Therefore, in this study, we adopted a video-based work detection method.

### 2.2 Video datasets used for action recognition

Recently, there have been many studies on video-based general action recognition, focusing primarily on large public video datasets created through crowdsourcing. The characteristics of datasets commonly used in action recognition/detection research are listed in Table 2. UCF101 [63], Kinetics [59, 60], Something-Something [62] HMDB51 [79], and Sports-1M [65] are particularly well-known. However, they are used for action recognition tasks rather than for action detection because unlike our proposed dataset, each of their videos contains only one action. “AVA” is a movie action dataset, which consists of multi-label annotations of basic actions such as “talk to” and “watch” [66]. “NTU RGB+D” contains 60 classes of human

### Table 1. Comparison of sensors and video.

| Device   | Movement flexibility | Cost     | Representation |
|----------|----------------------|----------|----------------|
| Sensors  | x                    | x        | x              |
| Video    | o                    | o        | o              |

https://doi.org/10.1371/journal.pone.0267457.t001
activity and 56,880 video samples [67]. The labels in these two datasets are in three major categories: daily actions, mutual actions, and medical conditions. These labels consist of basic actions or states such as “drink water” or “staggering” [67]. “EgoGesture” is a multi-modal large scale dataset for egocentric hand gesture recognition. This dataset has 83 classes of static or dynamic gestures such as “OK” [68]. “THUMOS Challenge 2014 Temporal Action Detection” dataset defines specific motions in sports, such as dunking and pitching [69]. “ActivityNet” Temporal Action Localization is a dataset for action recognition consisted of 200 different daily activities such as: “walking the dog”, “long jump” [70]. Ibrahim et al. collected a new dataset using publicly available YouTube volleyball videos. They annotated 4830 frames that were handpicked from 55 videos with 9 player’s basic action labels such as “spiking” [71].

“Win-Fail Action Recognition” is a task for judging the success or failure of an action in the following domain: “General Stunts,” “Internet Wins-Fails,” “Trick Shots,” & “Party Games.” [72]. “HAA500” contains fine-grained atomic actions where only consistent actions are classified under the same label (e.g., “baseball pitching” and “basketball free throws” to minimize ambiguity in action classification) [73]. “YouTube-8M” is the largest multi-label video classification dataset, consisting of up to 8 million videos (500,000 hours of video), annotated with a vocabulary of 4800 visual entities [74]. The Moments in Time Dataset is a large human-annotated collection of one million short videos corresponding to dynamic events that unfold in less than three seconds [75]. HACS is a dataset for human action recognition using a taxonomy of 200 action classes, which is identical to that of the ActivityNet-v1.3 dataset [76]. It has 504K videos retrieved from YouTube and each one is strictly shorter than 4 minutes, and the average length is 2.6 minutes. A large-scale “Holistic Video Understanding Dataset” (HVU) is organized hierarchically in a semantic taxonomy that focuses on multi-label and multi-task

| Dataset name            | Fixed camera? | Multi-label&time detection? | Action/Work   | Content     | Time range |
|-------------------------|---------------|-----------------------------|---------------|-------------|------------|
| Charades [58]           | no            | yes                         | general work  | general     | seconds    |
| Kinetics [59, 60]       | no            | no                          | action        | general     | seconds    |
| EPIC-Kitchen [61]       | no            | yes                         | specific work | cooking     | minutes    |
| Something-Something [62]| no            | no                          | action        | general     | seconds    |
| UCF101 [63]             | no            | no                          | action        | general     | seconds    |
| HMDB51 [64]             | no            | no                          | action        | general     | seconds    |
| Sports-1M [65]          | no            | no                          | action        | general     | seconds    |
| AYA [66]                | no            | no                          | action        | general     | seconds    |
| Ntu rgb+ d [67]         | no            | no                          | action        | general     | seconds    |
| Egogesture [68]         | no            | no                          | action        | general     | seconds    |
| THUMOS14 [69]           | no            | no                          | action        | general     | seconds    |
| Activitynet [70]        | no            | no                          | action        | general     | seconds    |
| Volleyball [71]         | no            | no                          | action        | general     | seconds    |
| Win-Fail [72]           | no            | no                          | action        | general     | seconds    |
| HAA500 [73]             | no            | no                          | action        | general     | seconds    |
| Youtube-8M [74]         | no            | no                          | action        | general     | seconds    |
| Hollywood in Homes [58] | no            | no                          | action        | general     | seconds    |
| Moments in Times [75]   | no            | no                          | action        | general     | seconds    |
| Hacks Clip [76]         | no            | no                          | action        | general     | seconds    |
| HVU [77]                | no            | no                          | action        | general     | seconds    |
| AVID [78]               | no            | no                          | action        | general     | seconds    |
| Ours                    | yes           | yes                         | specific work | factory     | seconds to minutes |

https://doi.org/10.1371/journal.pone.0267457.t002

Table 2. Comparison of video datasets.
video understanding [77]. AViD is a large-scale video dataset which has 467k videos and 887 basic action classes. The AViD dataset consists of similar actions to those in Kinetics, plus some additional actions such as talking, explosion, boating [78]. HDMB is the largest action video database with 51 basic action categories, which in total contain around 7,000 manually annotated clips extracted from a variety of sources ranging from digitized movies to YouTube [79]. “something something” is a large collection of labeled video clips that show humans performing pre-defined basic 174 actions with everyday objects [80]. “Charades” is composed of 9,848 annotated videos with an average length of 30 seconds involving 157 action classes such as “pouring into cup” [58]. “Kinetics-600” is a large-scale action recognition dataset which consists of around 480K videos from 600 action categories [58]. Each video in the dataset is a 10-second clip of action moment annotated from raw YouTube video [81]. “Kinetic-700” is an extensions of the Kinetics-600 dataset [82].

Charades [58] and EPIC-Kitchen [61] contain multiple actions per video, and, like our dataset, they are intended for action detection tasks. Charades is a dataset of videos consisting of common actions such as “Holding a laptop” or “Watching TV.” Each of these actions ranges from a few seconds to tens of seconds in length; the length of time required for each action is small. EPIC-Kitchen is a video dataset with clips of first-person work activities in the kitchen; it differs from general action datasets such as Charades in that it consists of complex work activities in the specialized domain of cooking. In addition, the duration of each work activity is on the order of minutes, and it includes a wide range of work durations, although not as wide as our dataset.

Given these characteristics, EPIC-Kitchen is the dataset most similar to ours. There are three important differences, however, between our dataset and EPIC-Kitchen. First, our dataset covers work in the factory. To the best of our knowledge, there is no other video dataset for work detection in the factory. Therefore, it has been difficult to automate the operation process analysis using video; we believe that the proposed dataset will contribute to the progress of future research. In addition, unlike conventional general action datasets, this dataset is intended for the analysis of specialized work activities consisting of series of complex body movements. Second, our dataset includes a wider range of time periods than EPIC-Kitchen, from a few seconds to several tens of minutes per work activity. In general, having a wider range of work times increases the difficulty of work detection because the model needs to capture both micro and macro changes in the video. As mentioned above, the range of work times has been limited in the past, and therefore models that address this problem have not been well studied. We believe that this dataset will contribute to the development of research on work detection over long periods. Third, the videos in our dataset were captured by a fixed-point camera. Considering actual application to the work site, the method whereby a camera is affixed to the operator and acquires video, such as was done for the videos in EPIC-Kitchen, is not appropriate from the standpoint of the burden on the operator; it is also costly because it is necessary to prepare a camera for each worker. In addition, it has been pointed out that in cases where the background information varies, the work content may be estimated using only the features of the background [83]. An example that is mentioned is that if the background is a tennis court, it is possible to infer that the action in the video is “Tennis Swing” even if the person is removed from the video. If a first-person camera is used for task detection as in EPIC-Kitchen, we can expect that each work activity will have its own characteristic background (e.g., a sink when the activity is washing dishes). Because our dataset uses a fixed-point camera, the background is the same throughout the video. Therefore, it is possible to avoid the problems pointed out by [83] and set up a task that will infer the work content from the worker’s movements rather than the background.
2.3 Action recognition methods

Until the use of CNNs became widespread, manually defined features were proposed as the analysis model for use with the above action recognition datasets. Examples include Oreifej and Liu et al.'s HOND4 feature [84], the HOG feature-based model [85], STIP [86], Gradient and Histogram of Optical Flow [87], 3D Histogram of Gradient [88], SIFT-3D [89], and SURF [90]. The method using Fisher vector representation [91] was also mainstream.

Since the publication of the study by [92], CNN methods have become mainstream in the field of computer vision, and in the video processing field, the SOTA method is a model based on CNN. Video processing requires not only spatial computation, as in the case of image processing, but also temporal computation. There are two main methods for computing the temporal dimension in CNN-based methods (see Table 3 for a summarized comparison). The first is a 3D convolutional neural network (3D-CNN) such as C3D [93]. This is a method for computing both spatial and temporal convolutions simultaneously; it has high representational capability. Following the proposal of C3D, 3D-CNN has been actively studied, such as in I3D [60], which employs Inception, and in work by [94], which is based on ResNet. Qui et al. proposed the Local and Global Diffusion (LGD) model. This model is equipped with LGD blocks that learn local and global representations in parallel [95]. D3D is a method for spatio-temporal recognition by distillation. Specifically, it is based on a teacher network that grasps the features of the optical flow and learns from the features of RGB only with a student network [96]. Tran et al. devised a model consisting of 3D ResNet by searching for the optimal architecture of convolutional neural networks [97]. PERF-Net, which uses the rendering information of poses in RGB frames to capture the motion features, has also been proposed [98]. In addition to the two streams, Hong et al. propose to use additional pose and pairwise streams which can be extracted contextual action cues to improve the performance of action recognition [99]. Shuiwang et al. have proposed a method for action recognition using 3D convolutional neural networks [100]. FstCN [101] also decompose a 3-dimensional convolution into 2-dimensional and 1-dimensional convolutions the model to reduce the amount of computation. ResNet [102] and DenseNet [103] has also been proposed as a model to reduce computational complexity. X3D is a model that searches for the optimal network structure [104]. A method to reduce the number of channels, called R(2+1)D [105], has been proposed, and the computation efficiency is being improved. Many other methods based on 3D-CNN such as P3D [106], S3D [107], and CSN [108] have been proposed and are SOTA for Kinetics and other datasets [109].

The second method of CNN-based video processing is one that first calculates spatial directions with an ordinary 2D-CNN and extracts features of the frame and then calculates temporal direction with LSTM or CNN [110]. This method is inferior to the 3D-CNN method in terms of representational capability because the spatial and temporal calculations are separate from each other, but it is advantageous in terms of computational cost. In the case of a large model such as 3D-CNN, the computer specifications are too high to introduce it to a factory, and it might not be possible to infer the actual environment. Therefore, there is a strong need to develop a relatively lightweight model such as CNN-LSTM. In addition, in order to create a work detection model for operation process analysis, it is necessary to create a video dataset

| Model type | Segmentation? | Inference memory | Time range |
|------------|---------------|------------------|------------|
| 3D-CNN     | no            | large            | short      |
| CNN-LSTM   | yes           | small            | long       |

Table 3. Comparison of 3D-CNN and CNN-LSTM.

https://doi.org/10.1371/journal.pone.0267457.t003
with work annotations for each work site. As the cost of creating such a dataset is very high, it is difficult to create a large dataset. With consideration of this background, we investigated ways of creating a better model by training a conventional basic CNN-LSTM model on a small dataset.

### 3 Task definition

In this study, we wished to analyze the work time of the main worker by categorizing his or her work content by the second from video recordings of factory work. The particular focus was to develop a method of factory work analysis that could be used in real-world applications. However, we did not find any video datasets available in the market that met the requirements for video content and labeling. Therefore, a dataset was created for this study by recording three days’ work content in an actual factory using a common video camera. Fig 1 shows an example of one frame of video recorded for the dataset. The task was then to classify the content of the main worker’s work by the second to annotate this dataset.

The proposed work video dataset has several features that differ from public video datasets that are conventionally used:

- The videos are captured by a fixed camera from a bird’s eye view in an actual printing factory.
- The videos are untrimmed and have a length of about 10 h per day. There is approximately 30 h of labeled video for the entire dataset.
- Videos show the full body of the worker and include complex movements. The background environment may also vary independently of the work.
- The worker uses multiple machine tools and small tools to perform the work.
- Twelve work labels were defined to categorize the work content. (These are not the same as action labels. Therefore, even if two physical actions are similar, they will be assigned different work labels if they differ in terms of work content.) With consideration of this background, we investigated ways of creating a better model by training a conventional basic CNN-LSTM model on a small dataset.
- The length of time to perform the same work consecutively varies considerably, ranging from a few seconds to tens of minutes, and tasks depend on one other over a very long period.

### 4 Method

The data were collected after prior explanation to the company (Echigo Fudagami Inc) that the data would be used for research purposes and verbal consent was obtained from the company manager and the subject workers. Permission to conduct the research was obtained from the company manager and subject workers. Our university’s ethics committee confirmed that ethical approval was waived. And no third-party ethical oversight was provided.

In this section, we first describe the work detection model used in our method, that is, the series of processes for classifying the operator’s work by the second based on a series of input images, as shown in Fig 2.

Then, we explain our approach for improving the generalization performance by pre-training this model on a task to segment workers. Even with pre-training on the segmentation task, the model structure in the inference process is almost identical to that of the work detection
model described above. Therefore, we expected the pre-training to improve performance considerably without increasing the computational cost of inference.

For the pre-training, an encoder–decoder segmentation model [111–113] is used; this makes it possible for the model to effectively learn important information such as the body parts of the worker from a small quantity of data. During the training of the work detection model, only the encoder module is used, to enable the feature extraction to focus on the important information.

The dataset videos were filmed using a fixed camera for three days at a specific work site in a printing factory (Echigo Fudagami Inc: Uenoyama 1-2-8, Ojiya, Niigata, Japan.), with a data size of 10 h per day, i.e., 30 h for the entire dataset.

4.1 CNN-LSTM model

The CNN-LSTM model for work detection is shown in Fig 2. The algorithm of this step is explained in Algorithm 1. In the hyper-parameters for each layer in Fig 2, the default values described in Fig 3 are used unless a value is specified. The hyper-parameters of the LSTM were determined through grid search by comparing validation scores. Each frame image of the video data is entered into the same CNN model [92]; the CNN model computes the images in a convolutional calculation and converts them into feature vectors. Then, a series of these feature vectors are input to LSTM [114]. LSTM performs feature extraction by considering the temporal patterns of the feature vectors of the frames. After that, the fully connected layer and Softmax infer the work content for each frame.

For video-based action detection, let $X_n = \{x_{n}, \ldots, x_{n+T}\}$ denote a series of frames starting from time $n$, where $T$ is the length of the input sequence.
The CNN-Module performs the following calculation:

\[ f_t = \text{CNN - Module}(x_t) \]  

where \( x_t \) is the \( t \)-th frame of \( X_n \), and \( \text{CNN-Module}(\cdot) \) is CNN feature extractor that transforms a 2D image into a vector.

The CNN-Module used in our experiment is shown in Fig 4. As the CNN module, we adopted the Encoder of DeepLab v3+[113] that we utilize for pre-training, as described in the following section. Since the output of DeepLab v3+ Encoder is a feature map, we added
Vectorizer right after Encoder to convert it into a vector. Hyper-parameters for each layer were determined as follows: the hyper-parameters for Encoder of DeepLab v3+ were the values recommended in the original paper. To effectively reduce the feature resolution before performing GAP, we added three additional layers of CNNs with stride = 2 as Vectorizer. The hyper-parameters of these CNNs were determined through grid search by comparing validation scores. The mean and standard deviation parameters used in the standardization function were calculated from the entire image of training set. Therefore, the result of the calculation, \( f_t \), is a \( d \)-dimensional feature vector that represents the spatial information of \( x_t \).

The LSTM module performs the following calculation:

\[
Y_n = LSTM(F_n)
\]  

(2)

where \( F_n = \{f_n, \ldots, f_{n+T}\} \) denotes a series of output feature vectors from \( CNN(X_n) \), which are computed independently. The result, \( Y_n = \{y_n, \ldots, y_{n+T}\} \), denotes a corresponding series of output action labels.

Fig 3. Class diagram of the layers used in the model diagrams in Figs 2 and 4. This figure describes the hyper-parameters that can be set for each class and their default values.

https://doi.org/10.1371/journal.pone.0267457.g003
The loss function is computed as

$$\text{Loss}(Y_n) = \sum_{t=n}^{n+T} CE(g_t, y_t)$$  \hspace{1cm} (3)$$

where $g_t$ is the ground truth at time $t$, and $CE(\cdot)$ is a cross-entropy loss function.

**Algorithm 1** Training factory segmentation pre-trained CNN-LSTM
Require: \( X_n = \{x_n, x_{n+1}, ..., x_{n+T}\}, \) \( \text{gt}_t \in \text{GT} \) (GT: work ground truth set)
Ensure: \( Y_n = \{y_n, y_{n+1}, ..., y_{n+T}\} \) (\( y \) = predicted work label)
1: while Until Convergence do
2: for NUMBER_of_IMAGES / BATCH_SIZE do
3: \( L \leftarrow 0 \)
4: for BATCH_SIZE do
5: \( \text{feature\_maps}_n \leftarrow \text{Encoder}(X_n) \) # Parallel, Freeze weight
6: \( \text{feature\_vectors}_n \leftarrow \text{Vectorizer} \text{feature\_maps}_n \) # Parallel
7: \( Y_n \leftarrow \text{LSTM} \text{feature\_vectors}_n \)
8: \( L \leftarrow L + \sum_{t=T}^{T} \text{Loss}(\text{gt}_t, y_{n+t}) \)
9: end for
10: \( \text{Adam}(L) \) then update params
11: end for
12: end while

4.2 Factory segmentation pre-training

The factory segmentation pre-training of the CNN-Module is shown in Fig 5. The algorithm of this step is explained in Algorithm 2.

Pre-training [116] is a learning method in which models are trained previously using a different dataset than the target task uses; the use of pre-trained models can be expected to reduce over-fitting in the target task. Traditionally, large-scale general image datasets such as ImageNet [117] and MSCOCO [118] are often used as pre-training datasets in neural net models for image processing. The CNN-Module used in the baseline model in this study was also pre-trained on ImageNet. However, pre-training with general image datasets such as ImageNet is only useful for capturing general features in images and is not suitable for extracting task-specific features. [119] successfully improved model classification performance in a study of pattern classification of interstitial lung diseases from lung CT images by pre-training the model on general texture datasets in addition to lung CT images to address the lack of training data. Thus, a more domain-specific form of pre-training may be more effective than pre-training on a general dataset alone for applied tasks for which training data are limited.

There are heuristics for identifying the important features in a frame image (e.g., position of a worker’s body part, position and state of a machine tool) for work detection in practical
applications. Because the CNN-Module used in this study is only an image feature extractor (the so-called encoder mechanism), the CNN-Module is generally pre-trained for an image recognition task that uses only the encoder. However, the image recognition task only determines whether or not there is a target in the image, which makes it difficult to utilize the heuristics. Therefore, in this study, we constructed an encoder–decoder model by adding a decoder mechanism after the encoder, which is not used in the work detection model but only during pre-training, so that the object segmentation task can be learned. Because segmentation performs pixel-level object classification, it is possible to extract more highly representative features. In this study, we trained the model for the segmentation of the worker’s body part, which is the most common heuristic in factory work detection. More advanced heuristics, such as the segmentation of tools and specific machine parts, can be included in applications in the real-world. It is expected that this pre-training will allow the encoder to extract more domain-specific features with an emphasis on heuristics.

We used DeepLab v3+ [113] as a segmentation model for pre-training because it is a simple encoder–decoder model consisting of a single stage and is thus naturally applicable.

We randomly sampled 337 frames from the training set of our factory video dataset so that work labels were distributed as evenly as possible. We annotated the worker’s head, upper body, lower body, and arm segments in these images to create a dataset for the segmentation task. We used this dataset to train the ImageNet pre-trained DeepLab v3+ model. The dataset was holdout validated with train = 269 images and test = 68 images, and the evaluation result on the test set was mIoU = 0.86.

After the above pre-training, learning is transferred to the task of work detection. For this, we remove the decoder in the segmentation-pre-trained model because the decoder is not needed for work detection. In addition, because the output of the encoder is a feature map, it is transformed into a vector by a CNN consisting of three layers and GAP (Global Average Pooling) [120].

Algorithm 2 Pretraining Encoder with factory segmentation

Require: \( x_i \in X, g_t \in GT \) (GT: segmentation ground truth set)
Ensure: \( y_i \in Y \) (Y: predicted segmentation set)
1: while Until Convergence do
2: for \( \frac{\text{NUMBER_of IMAGES}}{\text{BATCH_SIZE}} \) do
3: \( L \leftarrow 0 \)
4: for \( \text{BATCH_SIZE} \) do
5: \( \text{feature_map}_i \leftarrow \text{Encoder}(x_i) \)
6: \( y_i \leftarrow \text{Decoder}(\text{feature_map}_i) \)
7: \( L \leftarrow L + \text{Loss}(g_t, y_i) \)
8: end for
9: \( \text{Adam}(L) \)
10: end for
11: end while

5 Experiment

To ascertain the effect of the segmentation task pre-training on work detection performance, we compared the work detection performance of two methods: the CNN-LSTM model (baseline), in which the CNN-Module was pre-trained with ImageNet only, and the CNN-LSTM model (ours), which was pre-trained with the segmentation task.

In the first section, we describe the details of our dataset. Then, we describe the experiment implementation details and the results for the two methods.
5.1 Dataset

Here, we provide details about our factory operations video dataset. The videos were filmed using a fixed camera for three days at a specific work site in a printing factory, with a data size of 10 h per day, i.e., 30 h for the entire dataset. The dataset is annotated by the second with 12 labels according to the work content. Because this dataset has a time-series feature, it is appropriate to divide it into the training, validation, and test sets along the time series. Thus, we divided the dataset such that the training set consists of all of day 1 and the morning of day 2, the validation set consists of the afternoon of day 2, and the test set consists of all of day 3.

Table 4 shows the distribution of the labels in the dataset. It can be seen that the frequency of class occurrences is biased. This is a common situation in real-world applications of the work detection model, and it is important to train the model so that the tasks that appear less frequently can be correctly inferred.

4.2 Implementation details

In our experiments, we sampled a fixed length $T$ at a sampling rate $FPS$ from each frame sequence as the input. We set $T$ to 100 frames and $FPS$ to 1 fps. Hence, the frame sequence we utilized had a time length of 100 s. The output channel of the CNN-Module was set to 256. During training, we used the Adam optimizer [121] to optimize the network. The initial learning rate was set to $10^{-3}$. The batch size was 128.

Table 5 shows the equipment hardware and software infrastructure details of our experiment. Computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) was used.

5.3 Results and comparisons

In this section, we compare the results from our proposed segmentation heuristics pre-training method with those of the ImageNet-pre-trained model.

Fig 6 shows a loss curve of our factory-segmentation-pre-trained CNN-LSTM. For prediction, we used the weights of the epochs with the highest class accuracy in the validation set. In this experiment, the best validation class accuracy was achieved at 9 epochs (training took 21 minutes), so the weights at 9 epochs were used for the experiment. In the loss curve, the validation loss and training loss both show a decreasing trend, and there is no significant overfitting.
Although there is a gap between these two, it is considered to be within the acceptable range for learning progress.

Table 6 summarizes our experimental results. In the dataset used in this study, the number of data in each class differs considerably, and the micro average is not suitable because the influence of some classes is large. For this reason, we used the macro average as the average score in the results shown in the table.

From the table, we can see that the proposed method achieved an F1 performance of 0.6173, better than that of the baseline (ImageNet pre-training) method of 0.3267. This result shows that the baseline model is in a quite over-fitted state. In contrast, our method has a smaller performance gap between training and testing, which demonstrates that our segmentation heuristics pre-training method contributes to the reduction in over-fitting.

In addition, we performed a Welch's t-test to confirm the significance of the difference between the output of each method. As a specific procedure, we randomly divided the dataset into 100 parts, calculated the F1 score for each class in each part, and performed Welch's t-test
on each class. Consequently, the null hypothesis on all classes except class 7 F1 score can be rejected at the 1% significance level for the difference between baseline and ours.

Tables 7–9 show the recall, precision, and F1 score, respectively, for each class. In these tables, the “Diff” column was calculated as \( \text{Diff} = \text{Ours} - \text{Baseline} \), which indicates the amount by which the performance of our method differed from that of the baseline method.

### Table 6. Comparison of the two CNN-LSTM models

The baseline is ImageNet-pre-trained CNN-LSTM; ours is the factory-segmentation-pre-trained CNN-LSTM. Scores were calculated using the macro average. The highest scores are expressed in a bold font.

| Model | Accuracy | F1 | Recall | Precision |
|-------|----------|----|--------|-----------|
| Baseline | training | 0.9954 | 0.9946 | 0.9947 | 0.9945 |
| | validation | 0.6364 | 0.3940 | 0.3846 | 0.4578 |
| | testing | 0.3685 | 0.3267 | 0.3528 | 0.4955 |
| Ours | training | 0.8991 | 0.8691 | 0.8614 | 0.8823 |
| | validation | 0.7152 | 0.5528 | 0.5678 | 0.6096 |
| | testing | 0.7125 | 0.6173 | 0.6221 | 0.6407 |

https://doi.org/10.1371/journal.pone.0267457.t006

### Table 7. Comparison of label-by-label recall between the two LSTM-CNN models (test set)

The highest scores are expressed in a bold font.

| Class | Label | Baseline | Ours | Diff |
|-------|-------|----------|------|------|
| Class 1 | Chores | 0.2138 | 0.5481 | +0.3343 |
| Class 2 | Paperwork | 0.8271 | 0.6530 | −0.1741 |
| Class 3 | Cylinder preparation | 0.0775 | 0.7797 | +0.7022 |
| Class 4 | Material preparation | 0.3680 | 0.8647 | +0.4967 |
| Class 5 | Ink roller adjustment | 0.2642 | 0.4786 | +0.2144 |
| Class 6 | Winding preparation (top) | 0.2651 | 0.5336 | +0.2685 |
| Class 7 | Winding preparation (bottom) | 0.3839 | 0.7198 | +0.3359 |
| Class 8 | Creation of sample | 0.1891 | 0.4427 | +0.2536 |
| Class 9 | Ink adjustment | 0.1641 | 0.4660 | +0.3019 |
| Class 10 | Workbench | 0.2335 | 0.3684 | +0.1349 |
| Class 11 | Product check | 0.8273 | 0.6775 | −0.1498 |
| Class 12 | Nonhuman work | 0.4204 | 0.9324 | +0.5120 |

https://doi.org/10.1371/journal.pone.0267457.t007

### Table 8. Comparison of label-by-label precision between the two LSTM-CNN models (test set)

The highest scores are expressed in a bold font.

| Class | Label | Baseline | Ours | Diff |
|-------|-------|----------|------|------|
| Class 1 | Chores | 0.1856 | 0.3539 | +0.1683 |
| Class 2 | Paperwork | 0.1923 | 0.7523 | +0.5600 |
| Class 3 | Cylinder preparation | 0.8037 | 0.7922 | −0.0115 |
| Class 4 | Material preparation | 0.9007 | 0.8849 | −0.0158 |
| Class 5 | Ink roller adjustment | 0.5671 | 0.8591 | +0.2920 |
| Class 6 | Winding preparation (top) | 0.3986 | 0.5883 | +0.1897 |
| Class 7 | Winding preparation (bottom) | 0.4836 | 0.5883 | +0.1897 |
| Class 8 | Creation of sample | 0.3496 | 0.5564 | +0.2069 |
| Class 9 | Ink adjustment | 0.7222 | 0.4343 | −0.2879 |
| Class 10 | Workbench | 0.0219 | 0.4468 | +0.4249 |
| Class 11 | Product check | 0.3584 | 0.7305 | +0.3721 |
| Class 12 | Nonhuman work | 0.9623 | 0.9694 | +0.0071 |

https://doi.org/10.1371/journal.pone.0267457.t008
Our method improved F1 in all classes. The most notable improvement was in class 3, where F1 was 0.7859, an increase of 0.6445 points from 0.1414. Class 7, which showed the least improvement, still improved by 0.0149 points, and the other classes generally improved by 0.18 to 0.4 points. Class 3, which showed the greatest improvement in F1, is the cylinder preparation process; this work content is similar to that of classes 2 and 10 and requires a detailed analysis of the worker’s behavior. Because our method significantly improved the detection accuracy of these work activities, we believe that our proposed factory segmentation pre-training method is particularly effective for the detailed analysis of these work activities.

Besides, the occurrence rate of classes 9 and 10 in the dataset is extremely low (1.75% for class 9 and 0.85% for class 10), and it is difficult to extract features from the work content dataset alone, as the baseline model showed. However, our method improves F1 scores of classes 9 and 10 by 0.2036 and 0.3693 points, respectively. This suggests that pre-training of heuristics can generalize the model even for tasks with a low frequency of occurrence.

6 Discussion

6.1 Analysis of model architecture

In the dataset we used, the training set and the test set have different work days, and the worker’s environment can vary throughout the dataset. These aspects reflect the reality of the situation, and there would be considerable academic value in being able to generalize the model for application to a setting other than that of training.

As the baseline model uses ImageNet for pre-training, there is a high degree of flexibility in the worker’s movement in the training of the task detection model. In our factory segmentation pre-training, by contrast, the heuristics are included in the pre-training so that the direction of learning can be specified at the time the work detection model is trained. For example, we pre-trained for the segmentation of the worker’s body part, and this provided the direction during the training of the work detection model that the worker’s pose be considered. If necessary, by pre-training for other heuristics such as segmentation of work tools or materials, we can adjust the direction of learning to focus on these movements.

As described in the previous section, our method significantly reduced over-fitting. In cases such as the present one, in which the quantity of data is limited, we believe that giving the model a heuristic direction for learning may contribute effectively to reducing over-fitting.
The annotation of the worker segmentation performed for pre-training was much less costly than the annotation of the work content for work detection. The work detection performance improvement of 0.2906 points for macro F1 was obtained simply by pre-training on a segmentation task with low annotation costs; this suggests that our method can improve performance while keeping annotation costs low.

6.2 Failure cases

Here we discuss the reasons for the cases on which detection failed using this method. Table 10 shows the confusion matrix of result of our CNN-LSTM model of the test set. The rows show the classes of the groundtruths (GT). The columns show the classes predicted by the model (PR). The value of each element is a count of the pairs of the GT and the PR. The diagonal line is the true positive and is expressed in a blue cell. The class numbers in the table correspond to the class IDs in Table 4.

Table 10. Confusion matrix of our model’s result (test set). The rows show the classes of the groundtruths (GT). The columns show the classes predicted by the model (PR). The value of each element is a count of the pairs of the GT and the PR. The diagonal line is the true positive and is expressed in a blue cell. The class numbers in the table correspond to the class IDs in Table 4.

| GT | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 946 | 67 | 161 | 38 | 4 | 23 | 77 | 162 | 41 | 49 | 118 | 40 |
| 2 | 232 | 1227 | 253 | 31 | 6 | 27 | 9 | 37 | 4 | 7 | 38 | 8 |
| 3 | 160 | 12 | 3080 | 30 | 1 | 2 | 443 | 31 | 131 | 47 | 7 | 6 |
| 4 | 117 | 11 | 13 | 2199 | 2 | 2 | 9 | 0 | 56 | 0 | 66 | 68 |
| 5 | 144 | 24 | 77 | 5 | 683 | 61 | 240 | 44 | 0 | 0 | 149 | 0 |
| 6 | 31 | 15 | 48 | 8 | 61 | 793 | 124 | 352 | 4 | 0 | 39 | 11 |
| 7 | 18 | 0 | 10 | 12 | 12 | 3 | 668 | 197 | 0 | 0 | 8 | 0 |
| 8 | 429 | 246 | 28 | 7 | 19 | 427 | 220 | 1193 | 14 | 0 | 94 | 18 |
| 9 | 29 | 0 | 0 | 1 | 0 | 0 | 268 | 16 | 370 | 0 | 109 | 1 |
| 10 | 76 | 9 | 42 | 5 | 0 | 0 | 0 | 0 | 84 | 0 | 12 |
| 11 | 369 | 5 | 2 | 110 | 7 | 9 | 27 | 48 | 231 | 0 | 1754 | 27 |
| 12 | 122 | 15 | 174 | 39 | 0 | 1 | 3 | 64 | 1 | 1 | 19 | 6052 |

The annotation of the worker segmentation performed for pre-training was much less costly than the annotation of the work content for work detection. The work detection performance improvement of 0.2906 points for macro F1 was obtained simply by pre-training on a segmentation task with low annotation costs; this suggests that our method can improve performance while keeping annotation costs low.

6.2 Failure cases

Here we discuss the reasons for the cases on which detection failed using this method. Table 10 shows the confusion matrix of result of our CNN-LSTM model of the test set. The rows show the classes of the groundtruths (GT). The columns show the classes predicted by the model (PR). The value of each element is a count of the pairs of the GT and the PR. The model predicted once per second.

Fig 7 shows the output feature vectors of factory segmentation pre-trained CNN-LSTM dimensionally reduced using t-SNE and mapped onto a two-dimensional graph. The 64-dimensional feature vector, which is the output of the LSTM prior to the FC+Softmax layer in Fig 2, was compressed into a 2-dimensional vector by t-SNE. The parameters of t-SNE were set to perplexity = 30 and n_iter = 1000. t-SNE dimensionality reduction was performed on the entire test set, from which 12,000 samples were randomly sampled so that all classes were evenly distributed, and a scatter plot was created.

We consider the causes for the classes having an F1 score of less than 0.5 (Table 9, namely, classes 1, 7, 8, 9, and 10. First, let us consider classes 1 and 10. These are the labels having low F1 scores with the baseline method as well and are presumably difficult to classify in the first place. As shown in Table 4, class 10 is a rare class, constituting only 0.85% of the dataset. In addition, our review of the dataset suggests that the physical movements of the workers in classes 1 and 10 are particularly diverse, making it difficult to capture their work patterns. This can be seen from the distribution of the output feature vectors in Fig 7. Class 1 (red) and class 10 (greenyellow) are widely and finely distributed around the upper right area in the figure and are mixed with other classes such as class 2 and 3.

All of the remaining classes—7, 8, and 9—are work activities performed around the rotary press. As can be seen from Table 10, these work activities are frequently misclassified for each
other. In Fig 7, these classes are widely distributed from the center to the lower left of the figure. However, they differ from classes 1 and 10 in that they form a rough clump. In many cases, even humans cannot classify these work activities correctly unless they carefully observe the operation of the rotary press and the time series of the work content. The operation of the rotary press in particular is closely related to the content of the work, and so it is quite important to take this into account.

Overall, there are sparse and dense regions in the feature space of Fig 7. Classes 3, 4, and 12 occupy a relatively large space, while the other classes are clustered in a close space. In order to deal with this, it is necessary to be able to perform feature extraction that can separate the classes that are densely clustered. For this purpose, two approaches can be considered: 1. improvement of spatial feature extraction, and 2. improvement of temporal feature extraction. For the improvement of spatial feature extraction, as mentioned above, it is possible to use
segmentation models to pre-train the machines and tools that are important for work identification. As for the improvement of temporal feature extraction, we can expect that it can be improved by using a model with higher expressive power than LSTM (Bi-LSTM, Transformer, etc.) or by using a new model that can consider the flow of work over a longer span.

6.3 Limitations
In this dataset, the same person performs works throughout the entire dataset. Since our method includes abstraction of the body parts of the worker by segmentation, it is considered to be robust to changes in the workers. However, the method may not be able to infer correctly when the procedures and motions of the workers change significantly due to differences in the workers. In this case, we need to add new training data.

There were cases where the working time for the same task varied from several minutes to several tens of minutes. This is thought to be because the work procedures may differ even for the same work. For example, “Product check” work is performed continuously for several tens of minutes, but it may be interrupted irregularly when defective products are found, so the work may last from a few seconds to several minutes. Since the training data contains multiple work order patterns even for the same task, the model can predict when the work order changes to some extent. However, if the order of the tasks is highly irregular, or if the tasks are completely new and unknown, the inference may not be correct. In such cases, it is necessary to add training data or review the contents of the work labels.

Our method is robust to changes in background information that is irrelevant to the task by using the segmentation of critical elements. However, since the method assumes a fixed camera, the trained model cannot be used as-is when the layout of the machines and the work environment related to the work changes. In these cases, it is considered necessary to re-collect the training data.

7 Conclusion
In this paper, we have proposed a factory segmentation pre-training method for video-based work detection, which is the first attempt using our proposed printing factory video dataset. This method does not interfere with the worker’s work operations because it detects his or her work solely from the video. Compared with conventional methods that involve attaching sensors to workers, this system enables the detection of work activities in environments where workers have a higher degree of flexibility. The proposed pre-training method can not only learn to capture heuristics that are found important for distinguishing among similar work classes but also substantially reduce over-fitting with only a few additional segmentation annotations. Furthermore, our factory segmentation pre-training method does not result in a detection model so large that it would seriously impair the inference speed.

In addition, we proposed a printing factory video dataset, which is 30 h in length and contains annotations (12 classes) for every second. On this dataset, the proposed factory segmentation pre-training method outperforms the baseline (ImageNet pre-training) method, with an improvement of 0.2906 points in the macro F1 score.

As future research, we would like to improve our spatial and temporal feature extraction methods respectively. As for spatial features, we are interested in being able to take into account the state of machines and tools (e.g., rotation speed of the rotary press). As for temporal features, we are interested in time-series analysis methods that can extract and take into account the flow of factory operations over a longer time range.
Author Contributions

Data curation: Shotaro Kataoka, Genki Iwaka, Masashi Oba.

Formal analysis: Shotaro Kataoka, Tetsuro Ito, Genki Iwaka.

Investigation: Shotaro Kataoka, Masashi Oba, Hirofumi Nonaka.

Methodology: Shotaro Kataoka.

Project administration: Shotaro Kataoka.

Resources: Hirofumi Nonaka.

Software: Shotaro Kataoka, Tetsuro Ito.

Supervision: Hirofumi Nonaka.

Validation: Shotaro Kataoka, Masashi Oba.

Writing – original draft: Shotaro Kataoka, Hirofumi Nonaka.

Writing – review & editing: Hirofumi Nonaka.

References

1. Bell E. Cognitive automation, business process optimization, and sustainable industrial value creation in artificial intelligence data-driven internet of things systems. Journal of Self-Governance and Management Economics. 2020; 8(3):9–15. https://doi.org/10.22381/JSME8320201

2. Nica E, Miklencicova R, and Kicova E. Artificial intelligence-supported workplace decisions: Big data algorithmic analytics, sensory and tracking technologies, and metabolism monitors. Psychosociological Issues in Human Resource Management. 2019; 7(2):31–36. https://doi.org/10.22381/PIHRM7120195

3. Ren S, Zhang Y, Liu Y, Sakao T, Huisingh D, and Almeida CM. A comprehensive review of big data analytics throughout product lifecycle to support sustainable smart manufacturing. A framework, challenges and future research directions. Journal of cleaner production. 2019; 210:1343–1365. https://doi.org/10.1016/j.jclepro.2018.11.025

4. Smith A. Cognitive decision-making algorithms, real-time sensor networks, and Internet of Things smart devices in cyber-physical manufacturing systems. Economics, Management, and Financial Markets. 2020; 15(3):30–38. https://doi.org/10.22381/EMFM15320204

5. Clarke G. Sensing, smart, and sustainable technologies in big data-driven manufacturing. Journal of Self-Governance and Management Economics. 2020; 8(3):23–29. https://doi.org/10.22381/JSME8320203

6. Nica E, Janoškova K, and Kovacova M. Smart connected sensors, industrial big data, and real-time process monitoring in cyber-physical system-based manufacturing. Journal of Self-Governance and Management Economics. 2020; 8(4):29–36. https://doi.org/10.22381/JSME8420203

7. Leng J, Zhang H, Yan D, Liu Q, Chen X, and Zhang D. Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop. Journal of ambient intelligence and humanized computing. 2019; 10(3):1155–1166. https://doi.org/10.1007/s12652-018-0881-5

8. Hyers D. Big data-driven decision-making processes, Industry 4.0 wireless networks, and digitized mass production in cyber-physical system-based smart factories. Economics, Management, and Financial Markets. 2020; 15(4):19–28. https://doi.org/10.22381/EMFM15420202

9. Keane E, Zvarkova K, and Rowland Z. Cognitive automation, big data-driven manufacturing, and sustainable industrial value creation in Internet of Things-based real-time production logistics. Economics, Management, and Financial Markets. 2020; 15(4):39–48. https://doi.org/10.22381/EMFM15420204

10. Mircică N. Cyber-physical systems for cognitive Industrial Internet of Things: Sensory big data, smart mobile devices, and automated manufacturing processes. Analysis and Metaphysics. 2019; (18):37–43. https://doi.org/10.22381/AM1820195

11. Graessley S, Suler P, Kliestik T, and Kicova E. Industrial big data analytics for cognitive internet of things: wireless sensor networks, smart computing algorithms, and machine learning techniques. Analysis and Metaphysics. 2019; 18:23–23. https://doi.org/10.22381/AM1820193

12. Meyers T D, Vagner L, Janoskova K, Grecu I, and Grecu G. Big data-driven algorithmic decision-making in selecting and managing employees: Advanced predictive analytics, workforce metrics, and digital
innovations for enhancing organizational human capital. Psychosociological Issues in Human Resource Management. 2019; 7(2):49–54. https://doi.org/10.22381/PIHRM7220198

13. Andronie Mihai and Lăzăroiu George and Ștăfănescu Roxana and Uță Cristian and Dijmărescu Irina. Sustainable, Smart, and Sensing Technologies for Cyber-Physical Manufacturing Systems: A Systematic Literature Review. Sustainability. 2021; 13(10). https://doi.org/10.3390/su13105495

14. Patala s-Maliszew ska Justyna and Halikowski Daniel. A Model for Generating Workplace Procedures Using a CNN-SVM Architecture. Symmetry. 2019; 11(9). https://doi.org/10.3390/sym11091151

15. White T, Grecu I, and Grecu G. Digitized mass production, real-time process monitoring, and big data analytics systems in sustainable smart manufacturing. Journal of Self-Governance and Management Economics. 2020; 8(3):37–43. https://doi.org/10.22381/JSMSE8320205

16. Harrower K. Algorithmic decision-making in organizations: Network data mining, measuring and monitoring work performance, and managerial control. Psychosociological Issues in Human Resource Management. 2019; 7(2):7–12. https://doi.org/10.22381/PIHRM7220191

17. Meilă AD. Regulating the sharing economy at the local level: How the technology of online labor platforms can shape the dynamics of urban environments. Geopolitics, History, and International Relations. 2018; 10(1):181–187. https://doi.org/10.22381/GHR101201810

18. Davis R, Vochozk a M, Vrbka J, and Negurită O. Industrial artificial intelligence, smart connected sensors, and big data-driven decision-making processes in Internet of Things-based real-time production logistics. Economics, Management and Financial Markets. 2020; 15(3):9–15.

19. Hines P, Rich N. The Seven Value Stream Mapping Tools. International journal of operations & production management. 1997.;

20. Monteiro C, Ferreira LP, Fernandes NO, Sá JC, Ribeiro MT, Silva FJG. Improving the Machining Process of the Metalworking Industry Using the Lean Tool SMED. Procedia Manufacturing. 2019; 41:555–562. https://doi.org/10.1016/j.promfg.2019.09.043

21. Teichgräber UK, de Bucourt M. Applying Value Stream Mapping Techniques to Eliminate Non-Value-Added Waste for the Procurement of Endovascular Stents. European Journal of Radiology. 2012; 81(1):e47–e52. https://doi.org/10.1016/j.ejrad.2010.12.045 PMID: 21316173

22. Hines P, Mettler S, Coradi A, Boutilier R. A New Application of Value-Stream Mapping in New Drug Development: A Case Study within Novartis. Drug discovery today. 2015; 20(3):301–305. https://doi.org/10.1016/j.drudis.2014.10.009 PMID: 25448754

23. Harrer K. Algorithmic decision-making in organizations: Network data mining, measuring and monitoring work performance, and managerial control. Psychosociological Issues in Human Resource Management. 2019; 7(2):7–12. https://doi.org/10.22381/PIHRM7220191

24. Dillon AP, Shingo S. A Revolution in Manufacturing: The SMED System. CRC Press; 1985.

25. Zhang L, Chen X. Role of Lean Tools in Supporting Knowledge Creation and Performance in Lean Construction. Procedia Engineering. 2016; 145:1267–1274. https://doi.org/10.1016/j.proeng.2016.04.163

26. Sousa E, Silva FJG, Ferreira LP, Pereira MT, Gouveia R, Silva RP. Applying SMED Methodology in Cork Stoppers Production. Procedia manufacturing. 2018; 17:611–622. https://doi.org/10.1016/j.promfg.2018.10.103

27. Adanna IW, Shantharam A. Improvement of Setup Time and Production Output with the Use of Single Minute Exchange of Die Principles (SMED). International Journal of Engineering Research. 2013; 2(4):274–277.

28. Rajenthirakumar D, Caxton RJ, Sivaguruathanath S, Balasuadakhar A. Value Stream Mapping and Work Standardization as Tools for Lean Manufacturing Implementation: A Case Study of an Indian Manufacturing Industry. International Journal of Engineering Science and Innovative Technology. 2015; 4(3):156–163.

29. Peterek T, Penhaker M, Gajdoš P, Dohnálek P. Comparison of classification algorithms for physical activity recognition Innovations in Bio-Inspired Computing and Applications; 2014. p. 123–131.

30. Chang W, Dai L, Sheng S, Tan JTC, Zhu C, Duan F. A hierarchical hand motions recognition method based on IMU and sEMG sensors Robotics and Biomimetics (ROBIO). 2015 IEEE International Conference on, IEEE (2015). 2015; p. 1024–1029.

31. Ronao CA, Cho S-B. Human activity recognition using smartphone sensors with two-stage continuous hidden Markov models Natural Computation (ICNC). 2014 10th International Conference on, IEEE (2014). 2014; p. 681–686.
33. Uddin MZ and Hassan M M. Activity recognition for cognitive assistance using body sensors data and deep convolutional neural network. IEEE Sensors Journal. 2018; p. 1–1.

34. Wang P, Liu H, Wang L, Gao RX. Deep learning-based human motion recognition for predictive context-aware human-robot collaboration. CIRP Ann. 2018; 67(1):17–20. https://doi.org/10.1016/j.cirp.2018.04.066

35. Y-S Lee and S-B Cho. Activity recognition using hierarchical hidden markov models on a smartphone with 3d accelerometer, in Hybrid Artificial Intelligent Systems; 2011. p. 460–467.

36. Ravi N, Dandekar N, Mysores P, and Littman ML. Activity recognition from accelerometer data. AAAI. 2005; 5:1541–1546.

37. J R Kwapisz, G M Weiss, and S A Moore. Cell phonebased biometric identification, Proc 4th Int Biometrics: Theory Applications and Systems Conf, Washington DC, USA. 2010; p. 1–7.

38. J G Casanova, C S A vila, A de Santos Sierra, G B del Pozo, and V J Vera. A real-time in-air signature biometric technique using a mobile device embedding an accelerometer, in Networked Digital Technologies; 2010. p. 497–503.

39. Albinali F, Goodwin M S, and Intille S. Detecting stereotypical motor movements in the classroom using accelerometry and pattern recognition algorithms. Pervasive and Mobile Computing, 2012; 8 (1):103–114. https://doi.org/10.1016/j.pmcj.2011.04.006

40. Khan A M, Lee Y.-K, Lee S Y, and Kim T.-S. A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer. Information Technology in Biomedicine, 2010; 14(5):1166–1172. https://doi.org/10.1109/TITB.2010.2051955

41. Kaghyan S and Sarukhanyan H. Activity recognition using k-nearest neighbor algorithm on smartphone with triaxial accelerometer. International Journal of Informatics Models and Analysis (IJIMA), ITHEA International Scientific Society, Bulgaria. 2012; 1:146–156.

42. T Brezmes, J-L Gorricho, and J Cotrina. Activity recognition from accelerometer data on a mobile phone, in Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living; 2009. p. 796–799.

43. Mitchell E and Monaghan D. Classification of sporting activities using smartphone accelerometers. Sensors. 2013; 13(4):5317–5337. https://doi.org/10.3390/s130405317

44. Subasi, D H Dammas, R D Alghamdi, R A Makawi, E A Albiety, T Brahim, et al. Sensor based human activity recognition using adaboost ensemble classifier. Procedia Computer Science. 2018; p. 104–111.

45. Wang L, Gu T, Tao X, and Lu J. A hierarchical approach to real-time activity recognition in body sensor networks. Pervasive and Mobile Computing, 2012; 8(1):115–130. https://doi.org/10.1016/j.pmcj.2010.12.001

46. Garcia-Ceja E and Brena R. Long-term activity recognition from accelerometer data. Procedia Technology. 2013; 7:248–256. https://doi.org/10.1016/j.protcy.2013.04.031

47. Hassain T, Ahad M A R, Tazin T, and Inoue S. Activity recognition by using lorawan sensor. UbiComp Adjunct. 2018; p. 58–61.

48. Ryu J, Seo J, Jebellii H, and Lee S. Automated action recognition using an accelerometer-embedded wristband-type activity tracker. Journal of construction engineering and management. 2019; 145(1):04018114. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001579

49. Kim H, Ahn C R, Engelhaupt D, and Lee S. Application of dynamic time warping to the recognition of mixed equipment activities in cycle time measurement. Autom Constr. 2018; 87:225–234. https://doi.org/10.1016/j.autcon.2017.12.014

50. C Seeger, A Buchmann, and K Van Laerhoven, Myhealthassistant. A phone-based body sensor network that captures the wearer’s exercises throughout the day. Proc 6th Int Body Area Networks Conf, Beijing, China. 2011; p. 1–7.

51. Stiefmeier T, Ogris G, Junker H, Lukowicz P, Troster G. Combining Motion Sensors and Ultrasonic Hands Tracking for Continuous Activity Recognition in a Maintenance Scenario. In: 2006 10th IEEE International Symposium on Wearable Computers. IEEE; 2006. p. 97–104.

52. Stiefmeier T, Roggen D, Troster G. Fusion of String-Matched Templates for Continuous Activity Recognition. In: 2007 11th IEEE International Symposium on Wearable Computers. IEEE; 2007. p. 41–44.

53. Koskimaki H, Huijker V, Siritola P, Laurinen P, Roning J. Activity Recognition Using a Wrist-Worn Inertial Measurement Unit: A Case Study for Industrial Assembly Lines. In: 2009 17th Mediterranean Conference on Control and Automation. IEEE; 2009. p. 401–406.

54. Maekawa T, Nakai D, Ohara K, Namioka Y. Toward Practical Factory Activity Recognition: Unsupervised Understanding of Repetitive Assembly Work in a Factory. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing; 2016. p. 1088–1099.
55. Qingxin X, Wada A, Korpela J, Maekawa T, Namioka Y. Unsupervised Factory Activity Recognition with Wearable Sensors Using Process Instruction Information. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. 2019; 3(2):1–23. https://doi.org/10.1145/3328931

56. Tao W, Leu MC, Yin Z. Multi-Modal Recognition of Worker Activity for Human-Centered Intelligent Manufacturing. Engineering Applications of Artificial Intelligence. 2020; 95:103866. https://doi.org/10.1016/j.engappai.2020.103868

57. Al-Amin M, Tao W, Doell D, Lingard R, Yin Z, Leu MC, et al. Action Recognition in Manufacturing Assembly Using Multimodal Sensor Fusion. Procedia Manufacturing. 2019; 39:158–167. https://doi.org/10.1016/j.promfg.2020.01.288

58. Sigurdsson GA, Varol G, Wang X, Farhadi A, Laptev I, Gupta A. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In: European Conference on Computer Vision. Springer; 2016. p. 510–526.

59. Kay W, Carreira J, Simonyan K, Zhang B, Hillier C, Vijayanarasimhan S, et al. The Kinetics Human Action Video Dataset. arXiv preprint arXiv:170506950. 2017;.

60. Carreira J, Zisserman A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. p. 6299–6308.

61. Damen D, Doughty H, Farinella GM, Fidler S, Furnari A, Kazakos E, et al. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 720–736.

62. Goyal R, Ebrahimi Kahou S, Michalski V, Materzynska J, Westphal S, Kim H, et al. The "Something Something" Video Database for Learning and Evaluating Visual Common Sense. In: Proceedings of the IEEE International Conference on Computer Vision; 2017. p. 5842–5850.

63. Soomro K, Zamir AR, Shah M. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. arXiv:12120402 [cs]. 2012;.

64. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T. HMDB: A Large Video Database for Human Motion Recognition. In: 2011 International Conference on Computer Vision; 2011. p. 2556–2563.

65. A Karpathy , G Toderici, S Shetty, T Leung, R Sukthankar, F Li. Large-scale video classification with convolutional neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2014; p. 1725–1732.

66. Gu, C, Sun, C, Ross, D A, Vondrick, C, Pando, C, Li, Y, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018; p. 6047–6056.

67. Shahroudy, A, Liu, J, Ng, T T, and Wang, G. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016; p. 1010–1019.

68. Zhang Y, Cao C, Cheng J, and Lu H. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. Egogesture: a new dataset and benchmark for egocentric hand gesture recognition. IEEE Transactions on Multimedia. 2018; 20(5):1038–1050.

69. Wang L, Qiao Y, and Tang X. Action recognition and detection by combining motion and appearance features. THUMOS14 Action Recognition Challenge. 2014; 1(2):2.

70. Caba Heilbron, F, Escorcia, V, Ghanem, B, and Carlos Niebles, J. Activitynet: A large-scale video benchmark for human activity understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015; p. 961–970.

71. Ibrahim, M S, Muralidharan, S, Deng, Z, Vahdat, A, and Mori G. A hierarchical deep temporal model for group activity recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016; p. 1971–1980.

72. Parmar, P, and Morris, B. Win-Fail Action Recognition. arXiv:210207355. 2021; p. preprint.

73. Chung, J, Wu, C H, Yang, H R, Tai, Y W, and Tang, C K. HAA500: Human-centric atomic action dataset with curated videos. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021; p. 13465–13474.

74. Abu-El-Haija S, Kothari N, Lee J, Natsev P, Toderici G, Varadarajan B, et al. Youtube-8m: A large-scale video classification benchmark. arXiv:160906875. 2016; p. preprint.

75. Monfort M, Andonian A, Zhou B, Ramakrishnan K, Bargal SA, Yan T, et al. Moments in time dataset: one million videos for event understanding. IEEE transactions on pattern analysis and machine intelligence. 2019; 42(2):502–508. https://doi.org/10.1109/TPAMI.2019.2901464 PMID: 30902849

76. H Zhao, A Torralba, L Torresani, Z Yan. Hacos: Human action clips and segments dataset for recognition and temporal localization. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019; p. 8668–8678.
77. Diba A, Fayyaz M, Sharma V, Paluri M, Gail J, Stiefelhagen R, et al. Large scale holistic video understanding. European Conference on Computer Vision. 2020; p. 593–610.

78. Piergiovanni A, Ryoo M S. Avid dataset: Anonymized videos from diverse countries. arXiv:200705515. 2020; p. preprint.

79. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T. HMDB: a large video database for human motion recognition. Proceedings of the IEEE International Conference on Computer Vision (ICCV). 2011;.

80. Goyal, R, Ebrahimi Kahou, S, Michalski, V, Materzynska, J, Westphal, S, Kim, H, et al. The “something something” video database for learning and evaluating visual common sense. In Proceedings of the IEEE international conference on computer vision. 2017; p. 5842–5850.

81. Carreira J, Noland E, Banki-Horvath A, Hillier C, Zisserman A. A short note about kinetics-600. arXiv:180801340. 2018; p. preprint.

82. Carreira J, Noland E, Hillier C, Zisserman A. A short note on the kinetics-700 human action dataset. arXiv:190706987. 2019; p. preprint.

83. He Y, Shirakabe S, Satoh Y, Kataoka H. Human Action Recognition without Human. arXiv:160807876 [cs]. 2016;.

84. Oreifej O, Liu Z. Hon4d: Histogram of Oriented 4d Normals for Activity Recognition from Depth Sequences. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2013. p. 716–723.

85. Baumann F. Action Recognition with Hog-of Features. In: German Conference on Pattern Recognition. Springer; 2013. p. 243–248.

86. Laptev I. On space-time interest points. International journal of computer vision. 2005; 64(23):107–123.

87. Laptev, M Marszalek, C Schmid, and B Rozenfeld. Learning realistic human actions from movies. IEEE Conference on Computer Vision and Pattern Recognition. 2008; p. 1–8.

88. A Klaser, M Marszalek, and C Schmid. A spatio-temporal descriptor based on 3d-gradients. 19th British Machine Vision Conference. 2008; p. 275–1.

89. Scovanner P, Ali S, and Shah M. A 3-dimensional sift descriptor and its application to action recognition. In Proceedings of the 15th ACM international conference on Multimedia. 2007; p. 357–360.

90. Bay H, Ess A, Tuytelaars T, and Van Gool L. Speeded-up robust features (SURF). Computer vision and image understanding. 2008; 110(3):346–359. https://doi.org/10.1016/j.cviu.2007.09.014

91. Wang H, Kläser A, Schmid C, Liu CL. Dense Trajectories and Motion Boundary Descriptors for Action Recognition. International journal of computer vision. 2013; 103(1):60–79. https://doi.org/10.1007/s11263-012-0594-8

92. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. Advances in Neural Information Processing Systems 25. Curran Associates, Inc.; 2012. p. 1097–1105.

93. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M. Learning Spatiotemporal Features with 3d Convolutional Networks. In: Proceedings of the IEEE International Conference on Computer Vision; 2015. p. 4489–4497.

94. Haritaoglu I, Aramaki E, Arikan B,_rgb-fusion: Multi-camera surveillance system for tracking people across multiple FOVs. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2001; 23(7):691–705.

95. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M. Learning Spatio-Temporal Features with 3d Convolutional Networks. In: Proceedings of the IEEE International Conference on Computer Vision; 2015. p. 4489–4497.

96. Qiu, Z, Yao, T, Ngo, C W, Tian, X, and Mei, T. Learning spatio-temporal representation with local and global diffusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019; p. 12056–12065.

97. Stroud, J, Ross, D, Sun, C, Deng, J, and Sukthankar, R. D3d: Distilled 3d networks for video action recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2020; p. 625–634.

98. Tran, D, Ray, J, Shou, Z, Chang, S F, and Paluri, M. Convnet architecture search for spatiotemporal feature learning. arXiv:170805038. 2017; p. preprint.

99. Li, Y, Lu, Z, Xiong, X, and Huang, J. Perf-net: Pose empowered rgb-flow net. arXiv:200913087. 2020; p. preprint.

100. Ji S, Xu W, Yang M, and Yu K. 3D Convolutional Neural Networks for Human Action Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2013; 35(1):221–231. https://doi.org/10.1109/TPAMI.2012.59 PMID: 22392705
101. Sun, L, Jia, K, Yeung, D Y, and Shi, B E. Human Action Recognition Using Factorized SpatioTemporal Convolutional Networks. Proceedings of the IEEE International Conference on Computer Vision. 2015; p. 4597–4605.

102. He, K, Zhang, X, Ren, S, and Sun, J. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016;.

103. Landola F, Moskewicz M, Karayev S, et al. DenseNet: Implementing Efficient ConvNet Descriptor Pyramids. Eprint Arxiv. 2014; Eprint Arxiv.

104. Feichtenhofer C. X3D: Expanding Architectures for Efficient Video Recognition. In Proceedings of the IEEE/ CVF Conference on Computer Vision and Pattern Recognition. 2020; p. 203–213.

105. Tran D, Wang H, Torresani L, Ray J, LeCun Y, Paluri M. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 6450–6459.

106. Qiu Z, Yao T, Mei T. Learning Spatio-Temporal Representation with Pseudo-3d Residual Networks. In: Proceedings of the IEEE International Conference on Computer Vision; 2017. p. 5533–5541.

107. Xie S, Sun C, Huang J, Tu Z, Murphy K. Rethinking Spatiotemporal Feature Learning for Video Understanding. arXiv preprint arXiv:171204851. 2017;1(2):5.

108. Tran D, Wang H, Torresani L, Feiszli M. Video Classification with Channel-Separated Convolutional Networks. In: Proceedings of the IEEE International Conference on Computer Vision; 2019. p. 5552–5561.

109. Duan H, Zhao Y, Xiong Y, Liu W, Lin D. Omni-Sourced Weblly-Supervised Learning for Video Recognition. arXiv preprint arXiv:200313042. 2020;.

110. Joe Yue-Hei Ng, Hausknecht M, Vijayanarasimhan S, Vinyals O, Monga R, Toderici G. Beyond Short Snippets: Deep Networks for Video Classification. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA: IEEE; 2015. p. 4694–4702.

111. Badrinarayanan V, Kendall A, Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017; 39 (12):2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615 PMID: 28060704

112. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF, editors. Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015. Lecture Notes in Computer Science. Cham: Springer International Publishing; 2015. p. 234–241.

113. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 801–818.

114. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997; 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 PMID: 9377276

115. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. Mobilenetv2: Inverted Residuals and Linear Bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 4510–4520.

116. Erhan D, Courville A, Bengio Y, Vincent P. Why Does Unsupervised Pre-Training Help Deep Learning? In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings; 2010. p. 201–208.

117. Deng J, Dong W, Socher R, Li L, Kai Li, Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition; 2009. p. 248–255.

118. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: Common Objects in Context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T, editors. Computer Vision—ECCV 2014. Lecture Notes in Computer Science. Cham: Springer International Publishing; 2014. p. 740–755.

119. Huang S, Lee F, Miao R, Si Q, Lu C, Chen Q. A Deep Convolutional Neural Network Architecture for Interstitial Lung Disease Pattern Classification. Medical & Biological Engineering & Computing. 2020; p. 1–13. PMID: 31965407

120. Lin M, Chen Q, Yan S. Network In Network. arXiv:13124400 [cs]. 2014;.

121. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. arXiv:14126980 [cs]. 2017;.