

Sustainable Human-Robot Collaboration System in Mold Assembly

Subjects: Engineering, Industrial

Contributor: Yee Yeng Liao

Molds are still assembled manually because of frequent demand changes and the requirement for comprehensive knowledge related to their high flexibility and adaptability in operation. Human-robot collaboration (HRC) systems can be applied to improve manual mold assembly. A status recognition system based on parts, tools, and actions using a pre-trained YOLOv5 model is developed. This study improves the sustainability of the mold assembly from the point of view of human safety, with reductions in human workload and assembly time.

Keywords: human-robot collaboration ; transfer learning ; status recognition ; mold assembly

1. Introduction

The use of robots in manufacturing began in Industry 3.0 as industrial robots were introduced for automated mass production. However, there are challenges in expanding industrial robot systems' application in mass personalization. Industrial robot systems can work fast with a low error rate. Still, industrial robots are less flexible and require high-cost reconfiguration to cope with the frequent demand changes in mass personalization production. In contrast, manual systems can adapt to changes with lower investment costs, but human workers tend to become fatigued and have a higher error rate ^[1]. Therefore, the application of human-robot collaboration (HRC) systems in manufacturing has gained attention in Industry 4.0. HRC systems combine the cognitive ability of humans with the consistency and strength of robots to increase the flexibility and adaptability of an automated system ^[2]. Besides this, the key enabling technologies in Industry 4.0, such as artificial intelligence and augmented reality, are integrated into the HRC systems to support interaction and collaboration between humans and robots ^[3]. Although we are still at the stage of realizing Industry 4.0, the term Industry 5.0, which focuses on bringing humans back into the production line and collaboration between humans and machines, has been introduced ^[4]. Hence, HRC systems are foreseen to be an active research area in Industry 5.0 as well.

Herein focuses on the application of HRC systems in mold assembly. Most molds are still assembled manually, while full automation systems are implemented in various assembly systems, such as for automotive parts ^{[5][6][7][8]} and electronic parts ^{[9][10]}. A full automation assembly cell is insufficiently flexible to cope with the frequent changes in low-volume mold assembly production and the wide variety of mold components that vary in weight and geometry. However, musculoskeletal disorders (MSD) caused by heavy part handling and repetitive motion during mold assembly have increased the need for robots in mold assembly ^[11]. Therefore, we propose the application of the HRC mold assembly cell to overcome the problems in mold assembly. The implementation of collaborative robots can reduce repetitive strain injuries and relieve heavy part lifting for human workers. At the same time, the use of collaborative robots can increase the work consistency and productivity of the assembly cell by integrating computer vision into HRC assembly cells.

Herein aims to develop a vision-based status recognition system for a mold assembly operation to achieve a sustainable HRC mold assembly operation by reducing assembly time and improving human working conditions. An assembly operation comprises tasks performed to join or assemble various parts to create a functional model. Each task consists of a series of sub-tasks executed to assemble a specific part at a defined location using a defined tool.

2. Deep Learning-Based Recognition in HRC Assembly

The Convolutional Neural Network (CNN) is the most common deep learning method used in computer vision tasks. Image classification, object localization, and object detection are the three main computer vision tasks. Image classification seeks to classify the image by assigning it to a specific label ^[12]. AlexNet ^[13], ResNet ^[14], VGGNet ^[15], Inception Net and GoogleLeNet ^[16] are the most common CNN architectures that researchers on image classification have implemented. Object localization takes an image with one or more objects as input and identifies the objects'

location with bounding boxes. The combination of image classification and object localization results in object detection or recognition, which identifies the types of classes of the located objects [12]. The most common deep learning-based object recognition models are R-CNN (Region-based Convolutional Neural Network) [17], Fast R-CNN [18], Faster R-CNN [19], Mask R-CNN [20], and You Only Look Once (YOLO) [21][22][23][24] models. R-CNN families are two-stage object detectors that extract the regions of interest (ROIs), then perform feature extraction and classify objects only within the ROIs. Hence, two-stage object detectors require longer detection times than one-stage object detectors. YOLO models are one-stage detectors that directly classify and regress the candidate boundary boxes without extracting ROIs. Our study detects the parts and tools required during an assembly operation so as to recognize a task and then estimates the task's progress based on the position of parts or tools. Therefore, we focus on object recognition that involves classification and detection tasks.

The training data used in the existing recognition systems were raw data, collected using wearable sensors, and image data, such as an image captured during operation, and images of spatial and frequency domains derived from sensors. Uzunovic et al. [25] introduced a conceptual task-based robot control system that received human activity recognition and robot capability inputs. They recognized the ten human activities in the car production environment based on the data from nineteen wearable sensors on both arms using machine learning models. However, the attachment of wearable sensors on the human worker caused discomfort during the practical assembly operation. Furthermore, deep learning algorithms in computer vision allow us to perform motion recognition better using assembly videos or images. Researchers have developed deep learning-based approaches using images to recognize common tasks based on different recognition algorithms: gestures or motion recognition and combinations of part and motion recognition in manufacturing assemblies [26][27]. Therefore, this study focuses on applying deep learning algorithms to assembly videos or image data for task recognition.

The research on action and phase recognition has been developed and applied widely for common human activities and surgical applications. Still, the related research on manufacturing assembly applications is worth exploring. Wen et al. [28] used a 3D CNN to recognize seven human tasks in visual controller assembly for the learning process of the robot. They separated eleven assembly videos, collected into seven labeled segments representing seven tasks, and performed data augmentation to increase the dataset for training. The accuracy of the task recognition was only 82% because of the small training dataset and the environmental changes during the assembly operation. Wang et al. [27] used two AlexNets for human motion recognition and part tool identification, respectively. They recognized grasping, holding, and assembling motions to identify human intention. For the part tool identification of a screwdriver, small and large parts were the only parts and tools included in this study. However, they only tested the proposed method on a simple assembly that involved a single tool and limited types of parts.

Chen et al. [26] implemented the YOLOv3 algorithm to detect tools for assembly action recognition and the convolutional pose machine (CPM) to estimate the poses and operating times of the repetitive assembly actions. They tested the algorithm on three assembly actions, which were filling, hammering, and nut screwing. They only estimated the operating times using the cycle of action curve, and not the progress or the remaining operating times. Action recognition based on this tool is inefficient in monitoring the assembly progress because different tasks may require the same tool. Chen et al. [29] extended the previous study [26] and proposed a 3D CNN model with batch normalization for assembly action recognition to reduce the environmental effect and improve recognition speed. Besides this, they employed fully convolutional networks (FCN) to perform depth image segmentation, in order to recognize different parts from assembled products for assembly sequence inspection. They recognized parts using computer-aided design (CAD) models instead of original parts, and compared the accuracy and training time for RGB, binary, gray, and depth images. The results show that the RGB image data gave the highest accuracy, but the training time was longer than for the gray images dataset.

The performance of the developed recognition models in the existing research for assembly applications is worse than expected due to the limited dataset and the environmental changes during the assembly operation. Therefore, a transfer learning technique can be used to overcome these problems.

3. Status Recognition for HRC Mold Assembly Operation

Figure 1 illustrates the proposed conceptual framework of status recognition. An assembly operation consists of tasks for joining or assembling various parts to create a functional model. Herein defines a task as a series of sub-tasks executed to assemble a specific part at the designated item using a defined tool. Thus, herein decomposes the assembly operation into tasks and sub-tasks and define actions that represent the status of a sub-task. Status recognition identifies an assembly task based on the recognized unique part, and recognizes the status of the task based on the actions that are decomposed from sub-tasks.

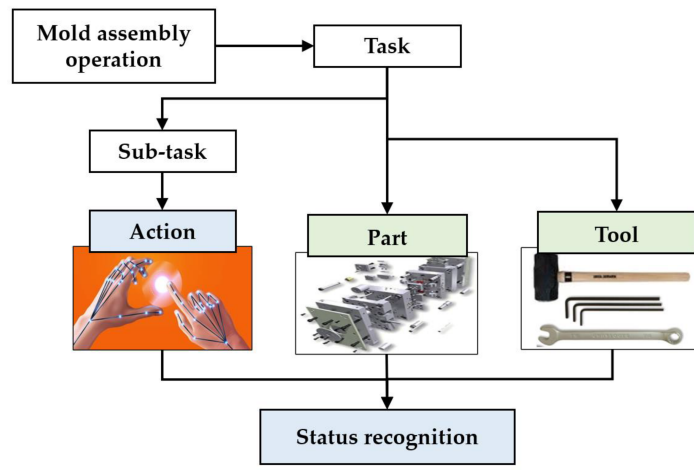


Figure 1. Proposed status recognition model for HRC system.

3.1. Decomposition of Mold Assembly Operation

Herein focuses on a two-plate mold assembly operation that consists of core and cavity sub-assemblies. **Table 1** lists the sixteen mold assembly tasks and each corresponding part. Each task assembles a unique mold part and joining part, such as screws and pins. Since the unique part assembled in each task is not repeated in other tasks, we can recognize a task by recognizing the corresponding unique part.

Table 1. List of tasks in a two-plate mold assembly.

No.	Task	Part
1	Prepare A side plate	A side plate
2	Assemble sprue bushing	Sprue bushing
3	Assemble top clamp plate	Top clamp plate, screws
4	Assemble location ring	Location ring, screws
5	Prepare outer B side plate	Outer B side plate
6	Assemble guide pin	Inner B side, guide pin
7	Assemble core	Core
8	Assemble ejection pin	Ejection pin
9	Assemble B side plates	Screws
10	Assemble ejection plate	Ejection plate, pin
11	Assemble return pin	Return pin
12	Assemble ejection support plate	Ejection support plate, screws
13	Assemble space plate	Space plate
14	Assemble bottom clamp plate	Bottom clamp plate, screws
15	Assemble core plate	Core plate, screws
16	Assemble core and cavity sub-assembly	Sub-assemblies

Herein defines a task as assembling a component consisting of a series of sub-tasks to assemble and join components. Herein also categorizes sub-tasks in mold assembly into nine categories ^[11]. **Table 2** lists the tools used in each sub-task. Herein must perform a series of sub-tasks on the component and corresponding joining components, such as screws and pins, to complete a task. Herein further decompose these sub-tasks into a series of actions for status recognition purposes. Mold assembly requires two types of tools, which are the hammer and hex-keys. Some sub-tasks need to be executed with a tool or without any tool. For sub-tasks that require a tool, we must include the actions to handle the tool.

Table 2. Categories of sub-tasks for mold assembly and tool used.

Code	Description of Sub-Tasks	Tool
A	Lift and position plate with rough tolerance	No
B	Lift and position plate with fair tolerance	No
C	Lift and position plate with tight tolerance	No
D	Pick and locate component with fair tolerance	No
E	Pick and locate component with tight tolerance	No
F	Pick, locate and insert screw	No
G	Tighten screw	Hex-key
H	Insert small component with force	Hammer
I	Insert plate with force	Hammer

The identification of actions plays an essential role in status recognition. Generally, a sub-task starts with a hand approaching the part or tool, and ends with an empty hand leaving the assembly area or returning the tool. Based on the common actions, we can summarize that a sub-task starts when the hand approaches the part and ends when the hand leaves the assembly area. The common actions in a sub-task can be listed as follows:

- Picking or grasping part/tool;
- Positioning part;
- Assembly using a tool, such as tightening a screw or inserting a pin;
- Leaving assembly area with an empty hand.

Herein aims to develop a status recognition system for HRC mold assembly based on object and action recognition. The status recognition consists of two stages, as shown in **Figure 2**. In the first stage, herein recognizes a task by recognizing parts and tools. In the second stage, we recognize the status of a sub-task based on the executed action. **Figure 2** shows an example of the stages of the proposed status recognition model. Herein decomposes the task “Assemble location ring” into two sub-tasks: “Lift and place location ring” and “Insert and tighten the screw”. Herein recognizes the part during the first stage as location ring, and the screws and hex-key as the tools, in order to identify the sub-tasks. Then, herein recognizes the status based on the actions defined for the sub-task. During the execution of “insert and tighten screws”, the defined action sequence is “insert screws”, “tighten screws” with hex-key, then leave the assembly area. Status recognition plays an essential role in enabling the robot to identify the status of the manual task and execute the subsequent task in any future study.

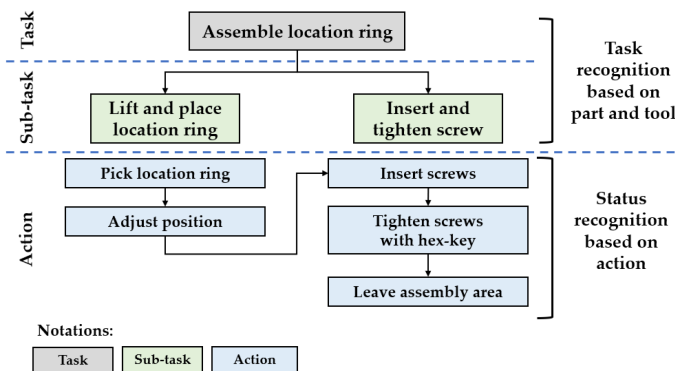


Figure 2. Decomposition of “Assemble location ring” task in proposed status recognition model.

3.2. Implementation of YOLOv5 and Transfer Learning

In this paper, we use the YOLOv5 model to develop a status recognition model because YOLOv5 has been proven to perform better in detection speed compared to R-CNN families [30][31]. We aim to implement status recognition in real-time task re-assignment and task execution in a future study. Therefore, the fast detection speed of the YOLO model is an important characteristic that enables us to recognize objects and actions in real-time during the assembly operation. Since we do not have a large assembly parts and tool images dataset, we implement a pre-trained YOLOv5 model instead of

building a model from scratch. In other words, we apply the transfer learning technique using a pre-trained YOLOv5 model to recognize assembly parts and tools based on small image datasets.

3.2.1. Data Collection and Processing

Herein focuses on a two-plate mold assembly operation, as shown in **Figure 2**. Herein need three image datasets to train the status recognition model: parts, tools, and hand actions. For the parts, herein categorized the mold parts into seven types based on the geometric shape. Besides this, herein collected images of tools, such as pins, screws, guide pins, sprue bushings, and location rings, from the internet. The mold assembly operation requires two types of tools, which are hammer and hex-key. Then, herein captured images from a YouTube video for actions representing the status of the sub-task during mold assembly [32]. After herein gathered the images, herein increased the number of images for training by rotating those images 90, 180, and 270 degrees. After collecting the images, herein used the LabelImg data annotation tool to label and create annotation files in the YOLO format [33]. Finally, we partitioned the dataset into training and testing sets containing 80% and 20% of the data, respectively. Herein then implemented k -fold cross-validation in the YOLOv5m model to evaluate the effects on model performance. Herein divided the datasets into five batches (i.e., $k = 5$), with 80% training datasets and 20% validation datasets for each fold.

3.2.2. Transfer Learning

Herein trained the models using the Windows 10 operating system and the Pytorch 1.7.0 framework with a single NVIDIA GeForce RTX2080Ti GPU. Herein used the YOLOv5 pre-trained models trained on the COCO dataset. We trained the datasets using YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x models to compare the performances of the models. Herein downloaded the weights obtained from the pre-trained model as the initial weights for training purposes. In the YOLOv5 model, the backbone acts as the feature extractor, and the head locates the bounding box and classifies the objects in each box. Therefore, herein froze the backbone to use the YOLOv5 models as a feature extractor and trained the head using the collected training datasets.

For YOLOv5 training parameter setting, we set the image size to 640×640 because the images collected from the internet have different sizes close to 640×640 pixels. Herein trained the models by employing different batch sizes and numbers of epochs with early stopping conditions. Herein obtained the best precision and weight from trial-and-error experiments by setting the batch size as 8 and using 600 epochs, with a learning rate = 0.01.

4. Conclusions

Herein presents the development of task and status recognition for an HRC mold assembly operation. The proposed recognition model consists of task recognition stages utilizing part and tool detection and status recognition, which identify the status of a task based on the human action.

Before developing the recognition model, herein decomposed the assembly operation into tasks, sub-tasks, and action. The sub-tasks contain information on parts and tools used. Then, herein decomposed the sub-tasks into a series of actions that defined the status of the task. Therefore, herein collected images of parts and tools, and defined actions to train the recognition model. Herein used a pre-trained YOLOv5 model to develop the model due to the limited dataset available. Herein selected pre-trained YOLOv5l without freezing layers to implement the task and status recognition because it showed the best performance based on accuracy and inference time among all YOLOv5 models. Besides this, smaller YOLOv5 models cannot detect all the statuses and parts because of the uneven number of images in each class. Herein re-trained the YOLOv5m model with only images of action classes and with a 5-fold cross-validation method. Then, herein combined the weights during the inference to investigate the detection ability. Herein found that the 5-fold cross-validation method improved the average mAP score and detection ability, but the inference time increased 2.4 times.

Herein are currently pursuing physical experiments using an HRC assembly cell testbed to further evaluate and verify the real-time practical implementations of this study. Herein focused on recognizing the manual task. However, it is necessary to recognize robot tasks so as to enable progress estimation and communication between humans and robots. Furthermore, we will expand the developed task recognition model to estimate the progress of the recognized task based on object tracking, and to estimate the completion time of the recognized task.

References

1. Barosz, P.; Golda, G.; Kampa, A. Efficiency analysis of manufacturing line with industrial robots and human operators. *Appl. Sci.* 2020, 10, 2862.
2. Khalid, A.; Kirisci, P.; Ghrairi, Z.; Thoben, K.D.; Pannek, J. A methodology to develop collaborative robotic cyber physical systems for production environments. *Logist. Res.* 2016, 9, 1–15.
3. Demir, K.A.; Döven, G.; Sezen, B. Industry 5.0 and human-robot co-working. *Procedia Comput. Sci.* 2019, 158, 688–695.
4. Maddikunta, P.K.R.; Pham, Q.V.; Prabadevi, B.; Deepa, N.; Dev, K.; Gadekallu, T.R.; Ruby, R.; Liyanage, M. Industry 5.0: A survey on enabling technologies and potential applications. *J. Ind. Inf. Integr.* 2021, 100257.
5. Krüger, J.; Lien, T.K.; Verl, A. Cooperation of human and machines in assembly lines. *CIRP Ann.* 2009, 58, 628–646.
6. Cherubini, A.; Passama, R.; Crosnier, A.; Lasnier, A.; Fraise, P. Collaborative manufacturing with physical human-robot interaction. *Robot Comput. Integr. Manuf.* 2016, 40, 1–3.
7. Makris, S.; Karagiannis, P.; Koukas, S.; Matthaiakis, A.S. Augmented reality system for operator support in human-robot collaborative assembly. *CIRP Ann.* 2016, 65, 61–64.
8. Müller, R.; Vette, M.; Mailahn, O. Process-oriented task assignment for assembly processes with human-robot interaction. *Procedia CIRP* 2016, 44, 210–215.
9. Ranz, F.; Komenda, T.; Reisinger, G.; Hold, P.; Hummel, V.; Sihn, W. A morphology of human robot collaboration systems for industrial assembly. *Procedia CIRP* 2018, 72, 99–104.
10. Casalino, A.; Cividini, F.; Zanchettin, A.M.; Piroddi, L.; Rocco, P. Human-robot collaborative assembly: A use-case application. *IFAC-PapersOnLine* 2018, 51, 194–199.
11. Liao, Y.Y.; Ryu, K. Task Allocation in human-robot collaboration (HRC) Based on task characteristics and agent capability for mold assembly. *Procedia Manuf.* 2020, 51, 179–186.
12. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karparthy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* 2015, 115, 211–252.
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 2012, 25, 1097–1105.
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
15. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representation (ICLR)*, San Diego, CA, USA, 7–9 May 2015.
16. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
17. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR 2014)*, Columbus, GA, USA, 23–28 June 2014; pp. 580–587.
18. Girshick, R. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
19. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* 2015, 28, 91–99.
20. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
21. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
22. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
23. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* 2018, arXiv:1804.02767.

24. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. arXiv 2020, arXiv:2004.10934.
25. Uzunovic, T.; Golubovic, E.; Tucakovic, Z.; Acikmese, Y.; Sabanovic, A. Task-based control and human activity recognition for human-robot collaboration. In Proceedings of the 44th Annual Conference of the IEEE Industrial Electronics Society (IECON 2018), Washington, DC, USA, 21–23 October 2018; pp. 5110–5115.
26. Chen, C.; Wang, T.; Li, D.; Hong, J. Repetitive assembly action recognition based on object detection and pose estimation. *J. Manuf. Syst.* 2020, 55, 325–333.
27. Wang, P.; Liu, H.; Wang, L.; Gao, R.X. Deep learning-based human motion recognition for predictive context-aware human-robot collaboration. *CIRP Ann.* 2018, 67, 17–20.
28. Wen, X.; Chen, H.; Hong, Q. Human assembly task recognition in human-robot collaboration based on 3D CNN. In Proceedings of the 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER 2019), Suzhou, China, 29 July–2 August 2019; pp. 1230–1234.
29. Chen, C.; Zhang, C.; Wang, T.; Li, D.; Guo, Y.; Zhao, Z.; Hong, J. Monitoring of assembly process using deep learning technology. *Sensors* 2020, 20, 4208.
30. Kim, J.A.; Sung, J.Y.; Park, S.H. Comparison of Faster-RCNN, YOLO, and SSD for real-time vehicle type recognition. In Proceedings of the International Conference on Consumer Electronics-Asia (ICCE 2020–Asia), Busan, Korea, 1–3 November 2020.
31. Yang, G.; Feng, W.; Jin, J.; Lei, Q.; Li, X.; Gui, G.; Wang, W. Face mask recognition system with YOLOV5 based on image recognition. In Proceedings of the 6th International Conference on Computer and Communications (ICCC 2020), Chengdu, China, 11–14 December 2020; pp. 1398–1404.
32. Cheng, S. Plastic mold assembly. Available online: <https://www.youtube.com/watch?v=laEWSU4oulw> (accessed on 31 January 2021).
33. GitHub. Labellmg. Available online: <https://github.com/tzutalin/labellmg.git> (accessed on 28 February 2021).

Retrieved from <https://encyclopedia.pub/entry/history/show/43112>