# APPLICATION OF LONG-SHORT TERM MEMORIES (LSTM) FOR POSE RECOGNITION IN QUALITY CONTROL STATION

**BY**

**LE MY DUYEN**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF ENGINEERING (LOGISTICS AND SUPPLY CHAIN SYSTEMS ENGINEERING)**

**SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY**

**THAMMASAT UNIVERSITY**

**ACADEMIC YEAR 2022**

THAMMASAT UNIVERSITY

SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY

THESIS

BY

LE MY DUYEN

ENTITLED

APPLICATION OF LONG-SHORT TERM MEMORIES (LSTM) FOR POSE RECOGNITION IN QUALITY CONTROL STATION

was approved as partial fulfillment of the requirements for
the degree of Master of Engineering
(Logistics and Supply Chain Systems Engineering)

on July 3, 2023

Chairperson

(Associate Professor Charnchai Pluempitiwiriyawej, Ph.D.)

Member and Advisor

(Associate Professor Chawalit Jeenanunta, Ph.D.)

Member

(Assistant Professor Rujira Chaysiri, Ph.D.)

Director

(Professor Pruettha Nanakorn, D.Eng.)

| | |
|---|---|
| Thesis Title | APPLICATION OF LONG-SHORT TERM MEMORIES (LSTM) FOR POSE RECOGNITION IN QUALITY CONTROL STATION |
| Author | Le My Duyen |
| Degree | Master of Engineering (Logistics and Supply Chain Systems Engineering) |
| Faculty/University | Sirindhorn International Institute of Technology/ Thammasat University |
| Thesis Advisor | Associate Professor Chawalit Jeenanunta, Ph.D. |
| Academic Years | 2022 |

# ABSTRACT

The time-motion study aims to find the standard time and improve production efficiency in manufacturing. This paper applies the Deep Learning model to recognize worker motion elements to have an automatic and effective system to improve productivity. Human pose and motion recognition are used to reduce production management costs. Each pose's training and testing video dataset is collected from the shoe manufacturing firm's camera at the specific Quality Control station. We propose applying the Long-Short Term Memories (LSTM) model, one of the particular kinds of RNN, with a kinematic base. Mediapipe Pose is used to express the body poses under the kinematic type, representing the human body's shape before feeding it into the LSTM model to train. The experiment conducted three work elements: pickup, inspection, and storage. After combining the LSTM model with Mediapipe Pose, we can detect the labels of workers' activities through video-based input and promising application to real-time cameras. The intersection rule is proposed in this study to improve the "flickering effect" to improve the accuracy up to 99.88%, 94.86%, and 100% for the inspection, pickup, and storage actions, respectively.

**Keywords**: Deep Learning, LSTM, Mediapipe Pose, action recognition

# ACKNOWLEDGEMENTS

First, I would like to express my deepest appreciation to my advisor, Assoc. Prof. Dr. Chawalit Jeenanunta, for his feedback, encouragement, and dedication throughout my entire studying master's process. I would like to thank him for accepting and allowing me to study and develop myself in Thailand. I could not have finished my thesis without his invaluable guidance, support, and expertise.

I am grateful to the committee, Asst. Prof. Dr. Rujira Chaysiri and Assoc. Prof. Dr. Charnchai Pluempitiwiriyawej. Their knowledge, expertise, and willingness to share their insights and guidance have significantly enriched my understanding of the subject. I'd like to thank the study's participants, who kindly contributed their time and efforts to the data-gathering procedure. Their participation was critical in gaining reliable data and insights for this research.

I would also like to acknowledge the support and assistance received from my Vietnamese and international friends. Thank you for being my mental support and giving me lessons to become a better person. Although it is difficult for them to constantly be by my side, support, and love me, please know that your engagement and support have had a huge influence, and I am thankful for our relationship and your concern. I would like to acknowledge the financial support provided by SIIT. Their support has enabled me to carry out this research and acquire the necessary resources for writing the thesis.

I would like to express my heartfelt appreciation to my family for their unwavering support, understanding, and encouragement throughout my academic journey. Their love and belief in me have constantly motivated and inspired me. Finally, I want to thank myself for believing in me, never giving up, and always looking for the good and love in life. Thank you all for being part of this rewarding experience.

Le My Duyen

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF SYMBOLS/ABBREVIATIONS

| Symbols/Abbreviations | Terms |
|---|---|
| ML | Machine Learning |
| ANN | Artificial Neural Networks |
| QC | Quality Control |
| CNN | Convolutional Neural Network |
| LSTM | Long Short-term Memory |
| ConvLSTM | Convolutional Long Short-term Memory |
| ROI | Region of Interest |
| MQTT | Message Queuing Telemetry Transport |

# CHAPTER 1
# INTRODUCTION

In the past ten years, Machine Learning (ML) has been one of the biggest trends in the industry. The reason for using Machine Learning is that it can improve productivity and produce the most high-tech products such as speech recognition, translation through images, marketing recommendations, chess playing, or self-driving cars. Machine Learning is the computer science field that can perform commands from humans by learning the characters automatically from input data. The notion of Artificial Neural Networks (ANNs), which is inspired by the structure and functions of the human brain, is at the heart of Machine Learning. ANNs have transformed the area of machine learning by offering a framework for modeling complicated data linkages and patterns. As science progressed, however, a subset of ANN algorithms called Deep Learning developed as an outstanding alternative to regular ANNs. Deep Learning methods are frequently used interchangeably with ANNs because they can solve complex and large-scale challenges. The ability of Deep Learning to generate models with numerous layers of linked neurons allows for finding complicated patterns and representations inside data. Deep Learning has shown extraordinary performance in various fields, including computer vision, natural language processing, and speech recognition, by exploiting deep neural networks. The availability of massive amounts of data and advances in computer power have accelerated the use and success of Deep Learning approaches.

Using Deep Learning in Computer Vision has recently gained more attention to recognize and track human motion (Action Recognition), such as detecting no-face mask people in public, tracking human activities in sports, controlling product quality, etc. By detecting joint positions of the human body and training by Deep Learning, we can recognize human actions in images and videos.

Action recognition can be implemented after estimating human pose, but the dataset for training 2D human pose estimation needs more diversity for people's complicated motions in real work. For example, Max Planck Institute for Informatics (MPII) Human Pose Dataset (Andriluka et al., 2014) includes 410 human activities, such as sports, home activities, transportation, etc. Microsoft Common Objects in

Context (COCO) Dataset (Lin et al., 2014) provides 150,000 people with crucial points. PoseTrack (Andriluka et al., 2017) is mainly a video-based dataset with 276,000 body pose annotations. Most popular datasets provide daily general activities which could not apply to the industry environment. The fact that specific tasks are different is based on each industry's natural characteristics.

Traditional motion-time studies use a time-counter device to measure the total time of continuous tasks. The device is usually a stopwatch or an electric computer stopwatch. The process typically requires additional labor to monitor and count the time for each job element. The motion-time method could separate the job into small parts with the counted times; then, the job could be rearranged in a more effective order. However, the manual way to calculate the time could consume much time to observe the whole working process and money for hiring monitor labor. Due to the development of Deep Learning and Computer vision in the past decade, pose estimation could be the economical solution.

In the Quality Control (QC) station, most papers focus on classifying and detecting defective products. In the safety-shoes factory of this study, the quality control station concludes three main worker activities: pickup, inspection, and storage. After shoes are transported through conveyors from the Lasting phase to QC, the worker comes to pick up these shoes from the conveyors to the QC table. The second process is inspecting the shoes at the QC table, and after finishing, the worker will put the shoes in the storage area behind them. In the first step, we develop a specific dataset of a QC worker for three different actions, as mentioned above, and use the Mediapipe Pose model to detect the worker's keypoints of hands and pose. We use the Long-Short Term Memories (LSTM) in the second step to better recognize the human pose.

### 1.1 Statement of problem

The study motivation is the need for an automatic monitor system in the industry and the development of machine learning in the past ten years. The ineffective operation and management lead to higher production costs and longer leading time without releasing the reasons. The human pose solves the motion time problem to prevent wasted activities that cause the worker's energy and time. In the shoe manufacturing of

this study, the workers at the quality station perform the lasting checking activities, including pickup, inspection, and storage.

This research proposed the combination of Deep Learning and Mediapipe Pose to track the activities of workers through one camera and show the time of each activity for further analysis.

The research output is to show the working pose and total finishing time of each element by using Computer Vision through just one camera. The result is conducted automatically to save management time, production time at the Quality station, and total production cost.

### 1.2 Objective of the Thesis

The fundamental goal of this thesis is to design and build a Deep Learning model that is particularly customized for identifying worker motion time. This methodology provides significant insights into workflow efficiency and total production time by precisely recording and evaluating employees' motion time. The motion time data acquired will serve as the foundation for a complete study to identify and remove unnecessary or inefficient process operations. Patterns and trends can be discovered by evaluating the gathered data, revealing areas where time is being wasted, or procedures can be improved. This analysis will allow for informed process improvement decision-making, enabling the deployment of focused interventions to eliminate wasteful activities and, eventually, reduce total production time.

The successful development and deployment of the suggested Deep Learning model would be a vital tool for companies and organizations looking to enhance their operational efficiency. The insights gained from the motion time analysis can be used to influence process reengineering initiatives, resource allocation strategies, and training programs, resulting in higher productivity, cost savings, and market competitiveness.

This thesis aims to contribute to industrial optimization by creating a Deep Learning model capable of reliably recognizing worker motion time. This model provides possibilities for discovering and deleting inefficient actions, resulting in decreased overall production time and enhanced operational efficiency.

# CHAPTER 2
# REVIEW OF LITERATURE

Deep Learning is a subclass of machine learning that focuses on training multi-layer neural networks to learn hierarchical data representations. Because of its capacity to handle high-dimensional data and automatically extract significant characteristics, this technology has been widely used in various disciplines. In Computer Vision, Deep Learning has transformed image recognition, object detection, and image generation jobs. Convolutional Neural Networks (CNNs), a prominent Deep Learning model, have excelled at image classification tasks by automatically learning and encoding visual characteristics from raw pixel data (Krizhevsky, Sutskever & Hinton, 2012). Deep Learning has also significantly contributed to Natural Language Processing (NLP). Recurrent neural networks (RNNs) such as LSTM models have been widely used to analyze sequential data such as text and audio. With their specific memory cells, LSTM models can capture long-term relationships and contextual information in text, allowing for tasks such as language modeling, machine translation, and sentiment analysis (Hochreiter & Schmidhuber, 1997).

## 2.1 Application of Deep Learning to Computer Vision

### 2.2.1 Quality Control problem

Due to the vast potential of computer vision, many papers apply the Deep Learning model to production to tackle problems automatically. Most Quality Control papers that use Deep Learning mainly focus on image processing to classify “Good” or “Bad” products. Korkmaz and Barstugan (2020) proposed a model for an inverter quality control system with a robotic arm, conveyor, and Nvidia Jetson Nano card. The algorithm that achieves 99.90% accuracy aims to detect whether the braking resistors of the inverter are connected or not and to replace the control from people. Villalba-Diez et al. (2019) developed a Deep Neural Network (DNN) soft sensor that helps control the quality in the Printing Industry 4.0 by predicting mistakes and detecting defective products. In another application of Deep learning in the Food Industry by Banús, Boada, Xiberta, Toldrà and Bustins (2021), the author used different

convolutional neural network (CNN) architectures to inspect the sealing and closure of food packages automatically. Xing and Jia (2021) proposed an automatic detection method for workpiece surface defects using convolutional neural networks (CNNs). The model demonstrates real-time detection capabilities with a speed of 23 frames per second at an input image size of 416 × 416 × 3, making it suitable for real-time automatic detection of workpiece surface defects. Using production videos, Lu, Xu and Huang (2022) presented a deep-learning-based anomaly detection system for lace failure checking. Their research demonstrated the efficacy of the suggested framework on holes and broken yarn.

### 2.2.2 Action Recognition problem

Recent research has rapidly expanded to tackle Human Action problems by leveraging Deep learning in the Human Action Recognition field. To identify human motion, first, we use human body modeling to represent the shape of the human body. There are three types of human body modeling to define the shape of the human body: the kinematic, planar, and volumetric.

The kinematic model is the commonly used model which detects landmarks in the human body to illustrate the structure of the body. A direct kinematic object model is used as a neural network layer on a toy example and 3D Human Pose Estimation (HPE) (Zhou, Sun, Zhang, Liang & Wei, 2016). Isack et al. (2020) developed a lightweight end-to-end model for the HPE problem by leveraging the kinematic structures. Another example is OpenPose (Cao, Simon, Wei & Sheikh, 2017) which provides multi-person landmarks detection in real time.

After identifying the structure to represent the body shape, we need to use the technique to correctly locate the keypoints from the input (RGB images, videos, or other sensors). CNN is the commonly used architecture for estimating pose and detecting key points (e.g., Fan, Zheng, Lin & Wang, 2015; He, Zhang, Ren & Sun, 2016). For the single-person problem, the method primarily focuses on one person at a time; for more than one person, the system will crop to a single-person problem to solve sequentially. Toshev and Szegedy (2014) proposed a regression method that applied Deep Neural Networks (DNNs) called DeepPose, to have an impressive result in estimating the human pose. Using sensors to collect data input from walkers, Palermo, Moccia,

Migliorelli, Frontoni and Santos (2021) proposed a customized dataset of walkers to select 17 key points and then applied the Fully Convolutional Network (FCN) to detect the keypoint locations. Besides CNN, a "stacked hourglass" model was designed by Newell, Yang and Deng (2016), which repeated the bottom-up and top-down pipeline for adjusting the feature detection through each hourglass.

Deep Learning architectures are applied to recognize the activities of people after observing the body's key points, so Human Action Recognition is downstream of HPE. Sun, Ning, Zhao, Huang and He (2020) developed an automatic evaluation system to monitor the efficiency of worker activities. The recognition task was implemented in two stages: estimating human body joints using the poseGAN model and matching the teacher and worker videos with similar action matching. Another example of human recognition is a Multi-stream Deep Fusion Network (MDFN) to detect the driver's activities (Behera, Wharton, Keidel & Debnath, 2020). The author argued that the actions of a driver and the surrounding objects need to be considered simultaneously (e.g., phones, cosmetics, cups). The MDFN combines Deep CNN, body pose, and body-object interaction. To handle a full video length in one shot, Ng et al. (2015) proposed two methods: CNN and LSTM. The CNN model will be responsible for Spatiotemporal feature extraction, and the output of CNN will connect with the LSTM to support sequence prediction of the action in a longer video time from seconds to minutes. The further extension of CNN LSTM is the convolutional LSTM (ConvLSTM) network proposed by Shi et al. (2015). The difference between CNN LSTM and ConvLSTM is that the ConvLSTM directly used the CNN model as a reading input into LSTM, not the separate transition between CNN and LSTM. In our work, the LSTM model combined with the kinematic base is applied to recognize the three fundamental actions of workers (pick up, inspection, storage) in the Quality Control station through video-based input. We use the Mediapipe Pose to extract the keypoint features and feed them through the LSTM model to predict the worker's motion.

### 2.2.3 Motion time study

The motion time study was first published in 1937 (Barnes, 1949). The earlier works indicated the effect of separating each element of the whole working process

with time using motion and time study. Bon and Daim (2010) proposed that the time and motion study in the packaging station involved worker tasks. The authors stated that there was no standard time for the working process of workers, which impacted the profit. The results improved the cost and company production. The main instrument of the research is observations by stopwatch record and interview. Krenn (2011) applied science to business to set the standard criteria for time and work. The researcher wanted to convert the business from traditional oral to calculation and written approaches. The other motion and time study was applied to the construction sector, which can reduce idle time to 40.24% in the first observation week (Prakash et al., 2020). However, few research have looked at the potential of computer vision to improve the timing process. Ji et al. (2022) proposed a unique motion-time application that uses convolutional neural networks (CNNs) to automatically recognize the working element and its duration, therefore replacing the traditional way of measuring operation time.

# CHAPTER 3
# METHODOLOGY

In manufacturing, the actions of workers are complicated and inconsistent. There are several reasons to cause the long lead time at the quality station, and the management operation is hard to trace back. The LSTM model combined with the Mediapipe Pose in this paper can automatically reduce the management cost and avoid human mistakes to record the activities for further analysis (Figure 3.1).

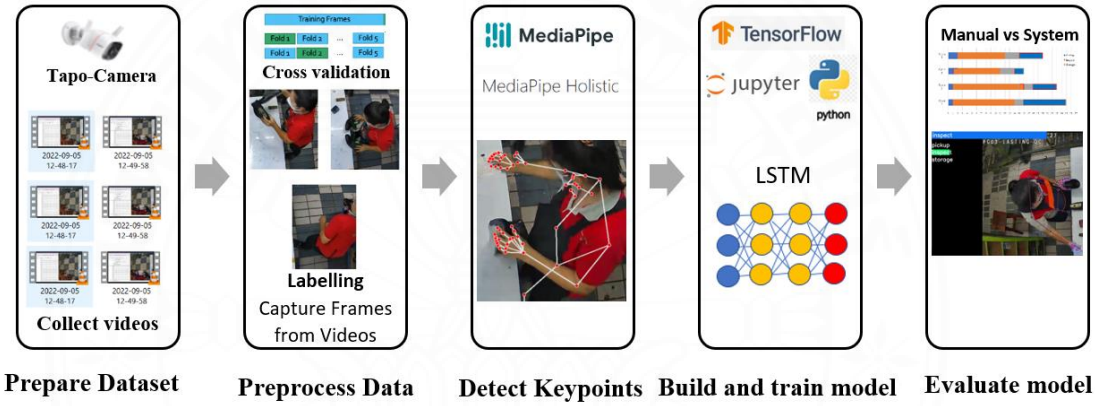


**Figure 3.1** LSTM model combined with Mediapipe Pose.

## 3.1 Dataset collection and Preprocess Data

We need to install a real-time camera at the quality station to keep track of the motions. The camera can be accessed and monitored online; we used the camera to record the real-time activities of the workers. The quality area has three parts: conveyors for delivering the shoes from the previous lasting state, a white table for inspecting the previous lasting state, and trays to temporarily store the shoes after finishing the quality checking. The workflow starts with picking the shoes on the conveyors, putting them on the checking table, performing the quality activities, and storing them in the trays after finishing checking. The workers use the stickers for the defective area on the bad shoe and use the red passing stamp for the good quality shoes. Therefore, we classified the quality task into three main activities: pickup, inspection, and storage.

To collect the training dataset for the LSTM model, we have to record the video and capture the frame in sequences. The total number of videos is 90 for three actions (30 clips for each activity). We created three folders for each action, and each folder contained 30 sub-folders for 30 clips. In each clip, there are 30 separated frames in sequence. There are three labels for each action and result visualization on the screen. Two workers are working at the QC table simultaneously; we need to crop the video into just one person problem to train.

In the second step, we decided to split 95% of the dataset for training and 5% for testing, 85 clips and five clips for training and testing, respectively. Due to the limitation of training clips, we continued to apply the 5-fold cross-validation technique. Eighty-five training clips were divided into five independent folds; each fold held one part for testing data, and the rest four parts were used for training repeatedly for the rest of 4 folds.

### 3.2 Using Mediapipe Pose to detect body keypoints

For the LSTM training step, we used the Mediapipe Pose framework (Lugaresi et al., 2019) to generate 543 landmarks, including pose landmarks, face landmarks, and hands landmarks). Before Mediapipe Pose, we got separate models for estimating hands, pose, and face pose. However, when we use several models for one input, the resolution is not fitting for each other. For instance, we install the pose model first; then it takes a frame with size 256x256 as input, then we continue to take the input to narrow down the area for the hands pose. The quality of the frame would be poor for detecting the hand keypoints on the cropped area. Therefore, Mediapipe Pose is proposed to treat several areas with proper frame resolution. The model first detected the pose landmarks and then estimated the regions of interest (ROI), two hands, and face. The final stage was to combine all the landmarks to have a full 543 keypoints. Each frame of 30 frames in one video consisted of 543 keypoints.

### 3.3 Build and Train model

For building and training model, the Long-Short-Term Memory architecture is designed with the input of worker's keypoints, then the Optuna technique is applied to finding the optimal hyperparameters for LSTM model. To prevent and improve the

accuracy of the LSTM in the production environment, the two more rules (sequence and intersection rule) are implemented in turn.

### 3.3.1 Long-Short-Term Memory (LSTM) model

The term LSTM can overcome the short-term memory disadvantage of traditional Recurrent Neural Networks (RNN). The RNN network can create a feedback loop that allows it to have a short memory of the previous states. LSTM is a type of RNN, but it was designed to handle long-term dependencies problems. The neural cell of the LSTM network has an additional state which constructs three gates: forget gate, input gate, and output gate. This state allows the LSTM to keep the information in the long term. The function of the forgetting gate is to decide which previous information should be kept in the network then the input gate continues to select the new information to be stored. The next step is to update the old cell to the new one; finally, we need to decide on the wanted part to be output at the output gate. For our problem, the input is 543 keypoints in each frame representing one pose. We seize 30 frames and store them in the sequence array to feed them into the LSTM model. The trained LSTM model gives the output under three probability scores from 0 to 1, standing for three actions. The prediction for 30 frames is the action that has the highest score. After that, the system takes the next new frame and eliminates the oldest frame in the last sequence sentence at the same time as starting the new prediction cycle. The LSTM is illustrated in Figure 3.2.

Because of the similarity of the pickup and storage actions, the accuracy of these two actions is not optimistic. To improve the problem, we proposed two new rules combined with the pure LSTM. The first rule is the sequence action rule.

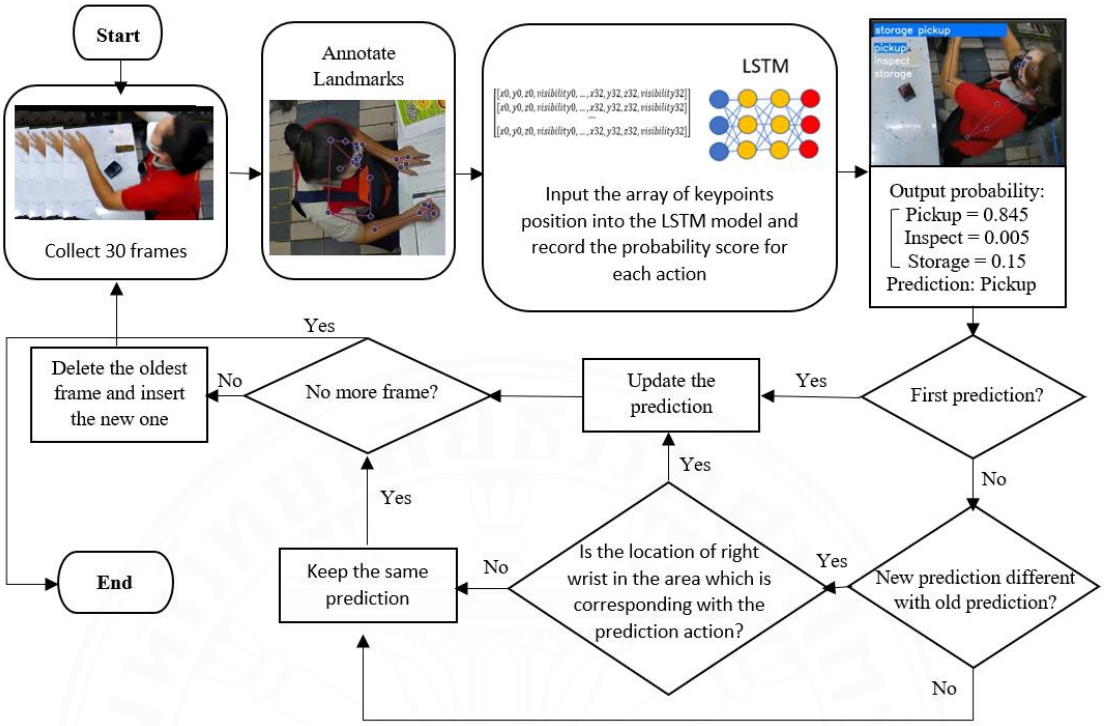


**Figure 3.2** Pure Long Short-term memory (LSTM).

### 3.3.2 Hyperparameter Optimization

Optuna is a powerful hyperparameter optimization tool in machine learning and deep learning research. Optuna's fundamental concept is to approach hyperparameter optimization as a sequential decision-making process. Optuna effectively explores the hyperparameter space using an intelligent sampling technique rather than painstakingly exploring the entire space. The Tree-structured Parzen Estimator (TPE) technique, which employs a Bayesian approach to describe the connection between hyperparameters and their corresponding goal values, is one of the sampling algorithms supplied by Optuna. Optuna runs a sequence of trials during the optimization process, with each trial corresponding to a different hyperparameter configuration. The objective function, which measured the model's performance, is assessed for each trial. Optuna automatically picks prospective hyperparameter values based on prior trial results to lead the search toward more optimum areas of the hyperparameter space. Optuna's adaptability allows for effectively exploring the hyperparameter space, saving computational resources and time.

Cross-validation is used to assess the performance of various hyperparameter combinations. In the context of hyperparameter optimization with Optuna, cross-validation is used again to assess the generalization performance of a particular hyperparameter configuration. As noted in section 3.1, the model is trained on a portion of the dataset and assessed on the remaining data, with the procedure repeated for each fold. This provides a more robust assessment of the model's performance and aids in preventing overfitting. In Table 3.1, there are four types of hyperparameters as inputs of Optuna. The task of Optuna is based on the provided searching space fine the optimal solution for each type of hyperparameter.

**Table 3.1** Hyperparameters Tuning Space for Optuna.

| Hyperparameters | Range Value |
|---|---|
| Learning rate | [1e-5, 1e-1] |
| Number of LSTM layers | [2, 4] |
| Number of Dense layers | [1, 4] |
| Number of nodes for each layer | [16, 128] |

### 3.3.3 LSTM combined with Sequence Rule

As the observation and workflow of the QC task, workers need to follow the order of action, starting from picking up shoes on the conveyor, moving to the QC table, and then storing them in the reservation space. To keep the prediction stable and prevent the switching between the pickup and storage in the prediction, we force the prediction updating to follow the action orderly. The input of the first 30 frames goes to the LSTM model; the first output is the action with the highest probability score. With the subsequent 30 frames, we have the second action prediction separately. The next step is to compare these two outputs; if they are the same, we do not have to update the prediction to the screen to avoid duplication; otherwise, we continue to check the second constraint. When the last two outputs are different, we continue to check the workflow logic to force the follow the sequence pickup-inspection-storage orderly in the circle (Figure 3.3).

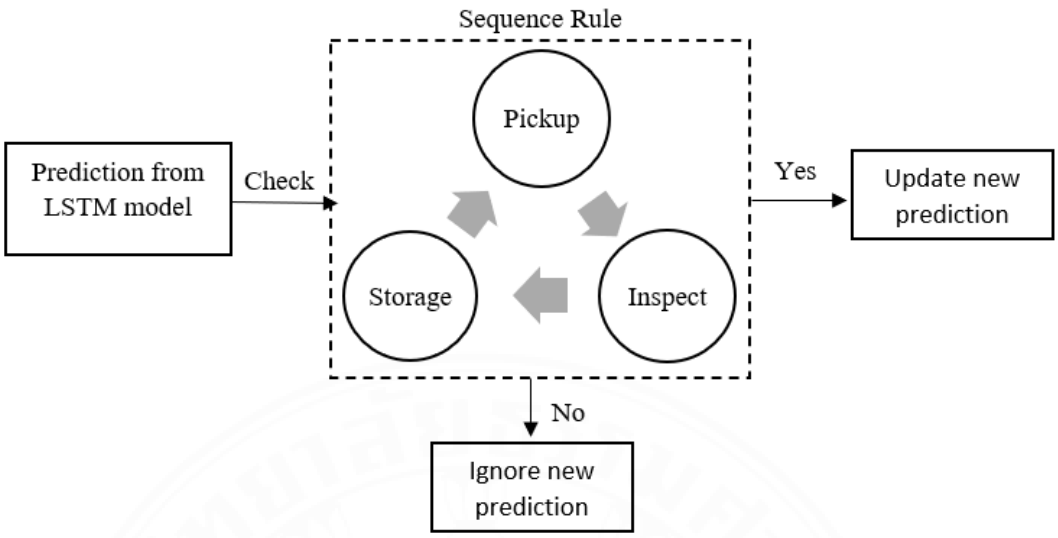


**Figure 3. 3** LSTM model with Sequence Rule.

The current prediction is inspection; for example, we check the last prediction. When the last prediction is pickup, we update and show the new result to say that the worker is performing inspection action after finishing the pickup stage. However, when the new prediction is storage, we keep the last prediction (pickup) the same without updating the new result. Because we assume that the worker could not store the shoes away right after the pickup stage without any checking, following the same logic, the action must update from pickup to inspection, then storage, and back to pickup in a circle.

### 3.3.4 LSTM combined with Intersection Rule

The drawback of the sequence rule is that when the LSTM fails to detect the new action, the whole process behind it is affected. The system waits until the following sequence rule in the next cycle. While waiting, it ignores all the correct predictions. The second rule ensures the system can perform as stable as the sequence rule and improves accuracy when the first rule ignores correct predictions. The working area could be divided into three main sub-areas for operating three main tasks. The top is reserved for pickup from a conveyor, the middle largest is for inspection, and the bottom is for storage. The original video from a real-time camera is 1280 pixels wide and 720 pixels height before cropping into a single person. Region of interest (ROI) is used to focus on a single person. The size of the ROI is 720 pixels in height and 580 pixels in width.

Based on the basic rule, we could draw two lines in the frame using OpenCV, as illustrated in Figure 3.4.

After applying ROI technique, the next step is to determine the position of the hand's keypoints with the drawn lines. There are 33 nodes for pose landmarks; the right wrist keypoint is the central considered node to compare with the lines. When the LSTM model gives the output to estimate the current pose, the model compares the current prediction with the previous one, as mentioned in the sequence rule part. After checking that the current one is different from the previous output, the coordinate of the right wrist would be considered next. The position of the right wrist coordinator would be combined with the line to check the prediction of the LSTM match with the working position of the workers. The fundamental of the intersection rule is to guarantee that the pose estimation matches the regulation area. To clarify this, we use the image coordinate system as a reverse graph with the x and y-axis. The original point (0,0) is in the frame's upper-left corner. The maximum numbers of x and y correspond to the width and tall of the image after cropping. Through observing the worker's activities, the location of the 2-line is as in Figure 3.4 The coordinate of the 2-line is $\left(0, \frac{3*y_{frame}}{16}\right)$ and $\left(0, \frac{2*y_{frame}}{3}\right)$. For the pose landmark, each node concludes x, y, z, and visibility. X, y is normalized from 0.0 to 1.0 by the wide and tall frame. Z implies the depth of the keypoints; the value of z would be more significant when the worker moves far from the camera. The visibility is from 0.0 to 1.0, which indicates the ability to detect landmarks in the worker's body. The coordinates of x and y are considered to locate the worker's position. When the LSTM prediction is pickup, and the x position of the right wrist keypoint is smaller than $\frac{3*y_{frame}}{16}$, then the sentence updates to pick up status. Accordingly, the x position of the right wrist is within $\frac{3*y_{frame}}{16}$ to $\frac{2*y_{frame}}{3}$ associated with the inspection result from the LSTM; the showing output is inspection. Finally, when the position of the right wrist is over $\frac{2*y_{frame}}{3}$, the system would update to the storage stage. The combination with LSTM is just for the pickup and inspection

process. The storage would be updated automatically when the worker moves below $\frac{2*y_{frame}}{3}$ positions without the LSTM prediction.

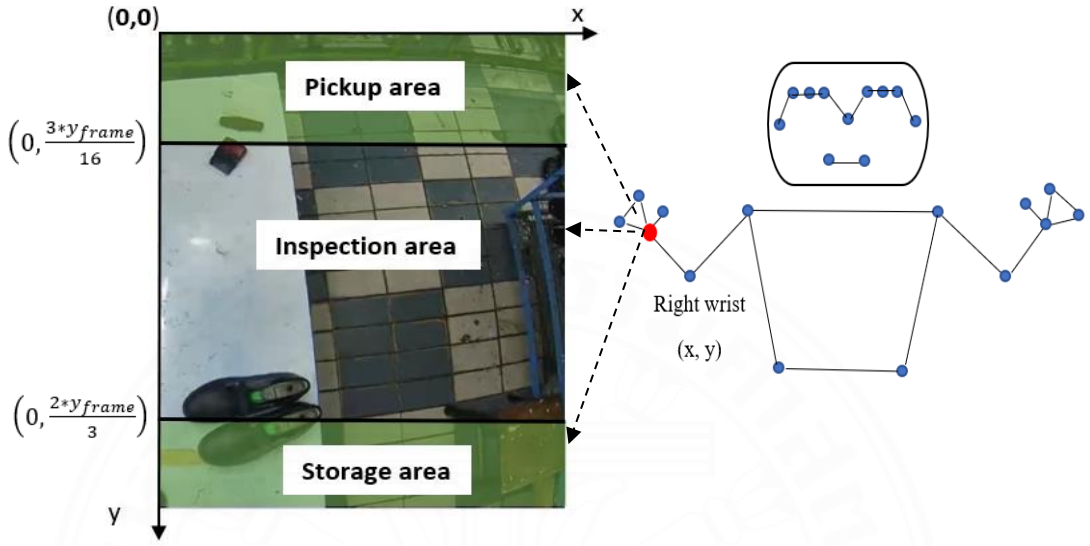


**Figure 3.4** The combination of LSTM, Mediapipe Pose and Intersection rule.

### 3.4 Evaluation Model Technique

There are two techniques which are used to evaluate the proposed models: offline video and real-time video.

#### 3.4.1 For offline evaluation

For offline evaluation in the video, the Tesseract OCR technique is applied to extract the time in the left top corner of the screen. OCR has been frequently used to extract text from photographs of documents (Blanke, Bryant & Hedges, 2012). OCR can capture each action's time and then compare it with the manual check (Figure 3.5).

Extracting numbers from a video involves utilizing Tesseract-OCR, an optical character recognition engine, in conjunction with OpenCV, a computer vision library. Figure 3.6 shows the example of one frame used for extracting the time. Initially, the OCR engine must be set up by specifying the path to the Tesseract-OCR executable. Subsequently, the video file is processed by the individual frames within the video. Each frame is sequentially retrieved until no more frames remain. OCR is then performed on the grayscale frame with a specified page segmentation mode. The resulting text is split into words, and numbers are extracted by checking if each word

is comprised solely of digits. The extracted numbers are stored in a list for output with the action name. Finally, the video file is released to free up system resources.

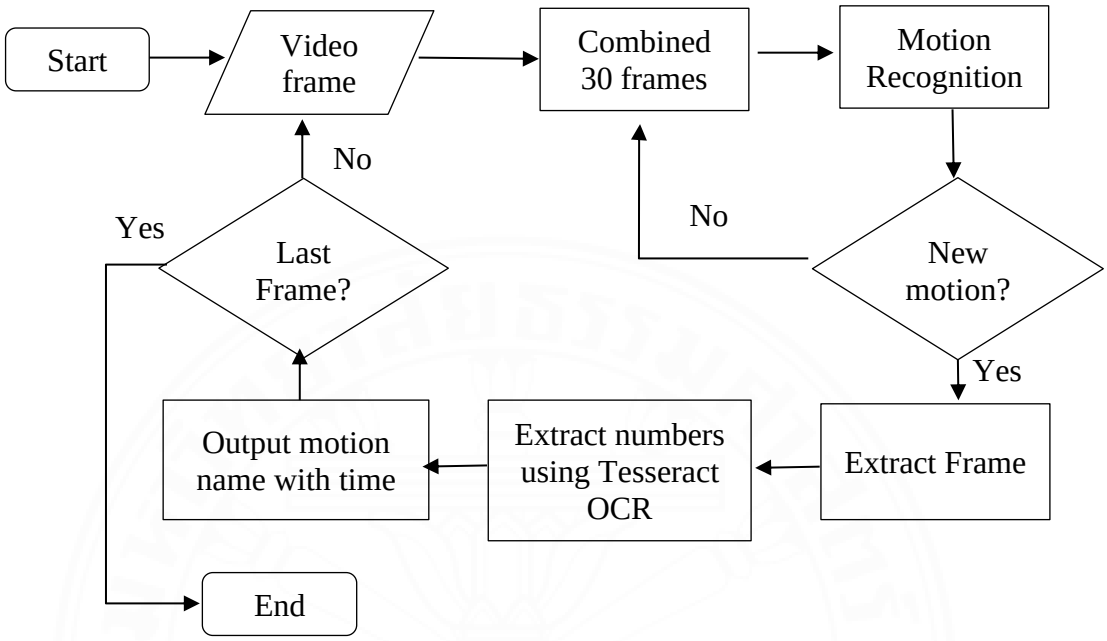


**Figure 3.5** Time extracting algorithm.

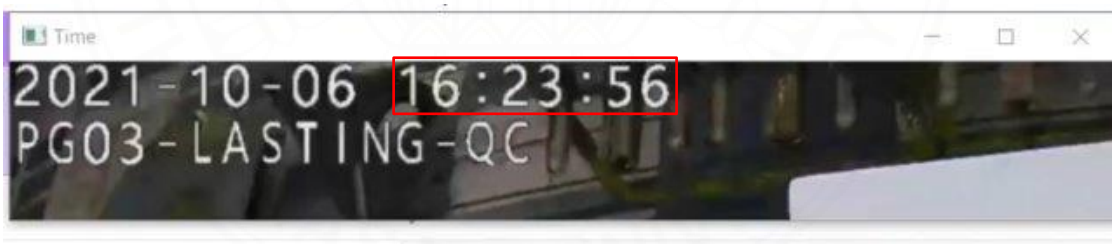


**Figure 3.6** Sample frame of time extraction method.

### 3.4.2 For online evaluation

MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol developed for efficient and dependable communication between devices in low bandwidth or high latency networks. An MQTT broker, also known as a MQTT message broker or a MQTT server, is a key component of the MQTT architecture that serves as a messaging intermediate between publishers and subscribers. The clients (subscribers) can receive the message which is published in real-time by the clients (publishers) through MQTT protocol under the topics they subscribed to. The

Our pose estimation model operates in real-time using a camera in a shoe manufacturing setting. Whenever the prediction model provides an output for pose estimation, we identify the current start time and associate it with the corresponding action. Subsequently, we publish the real-time working status using an MQTT broker. Upon receiving the messages, the MQTT broker distributes them to the subscribed clients or subscribers. When the subscribers subscribe to the relevant topic, they receive the working status along with the corresponding time, presented as a string. Figure shows the process of publishing and subscribing from a typical MQTT.

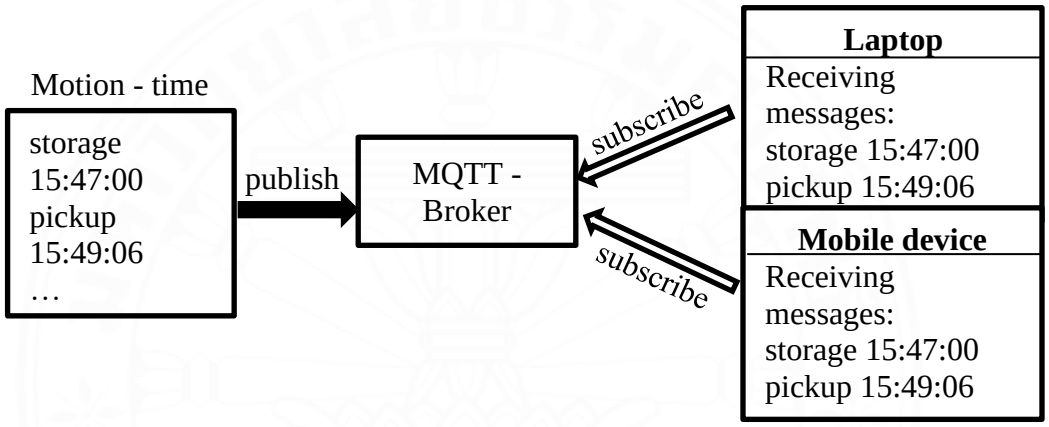


**Figure 3.7** Operation rule of MQTT protocol.

# CHAPTER 4
# RESULTS AND DISCUSSION

During the testing phase, the model generated prediction results at intervals of every 30 frames, ensuring a cumulative probability of 1 for each prediction. The action with the highest probability is selected to determine the predicted movement process, but only if it exceeds a predefined threshold of 0.8. Suppose the highest probability falls below the threshold, indicating uncertainty in predicting the worker's action. In that case, the system pauses and awaits additional features from the following 30 frames before proceeding with the task. The following Equation (4.1) and Equation (4.2) are employed to measure the accuracy of the model.

$$Accuracy = \frac{TN + TP}{TP + FP + TN + FN} \tag{4.1}$$

$$J(\Theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log\left(\hat{p}_k^{(i)}\right) \tag{4.2}$$

Equation (4.1) shows the accuracy measurement, which sums together the number of correct predictions (including true positives and true negatives) and divides by the total number of instances in the dataset. Equation (4.2) indicates the cross-entropy cost function. $y_k^{(i)}$ is the true label (either 0 or 1), and $\hat{p}_k^{(i)}$ is the predicted probability of the $i^{\text{th}}$ instance belongs to class *k*.

## 4.1 The accuracy of Pure LSTM

For searching the best combinations for the LSTM hyperparameters, Optuna has used the pruning technique to minimize the computational time. Pruning stops unpromising solutions by Median Pruning algorithms.

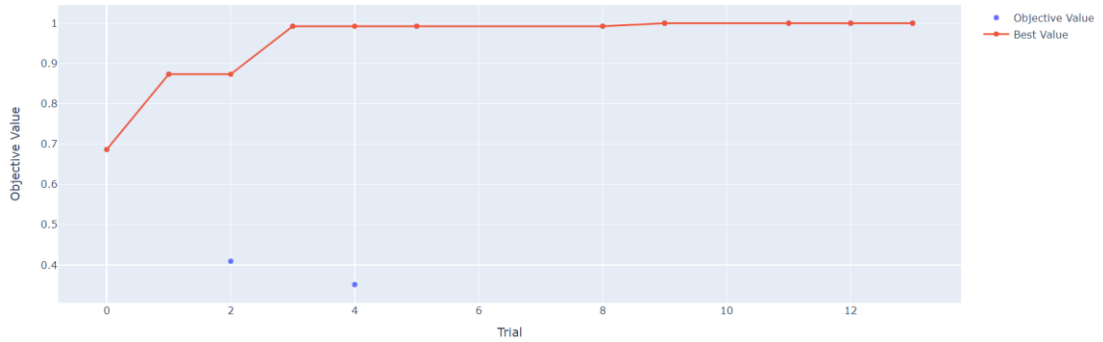


**Figure 4.1** Optimization History Plot.

The y-axis (Objective Value) on Figure 4.1 is the accuracy of the model. The pruning algorithm has stopped Trial 6, Trial 7, Trial 10, and Trial 14 in total 15 trials.

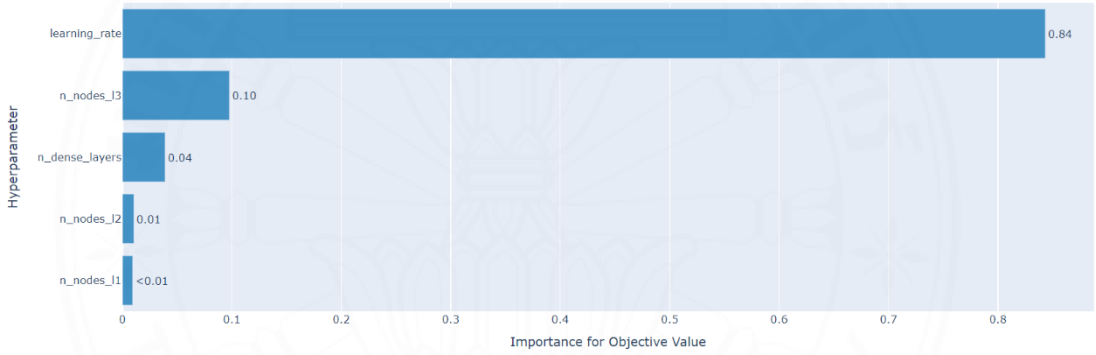


**Figure 4.2** Hyperparameter Importance.

In Figure 4.2, the algorithm's performance is primarily influenced by the learning rate, consistent with a widespread belief regarding neural nets. After running through 15 trials, the Optuna mechanism reported the best hyperparameters combination as follows: learning rate is 0.005; the number of hidden layers is 3; the number of nodes for layers 1, 2, and 3 is 64, 112, 80; the number of dense nodes for one layer is 121. After using Optuna for tuning, the number of parameters decreased from 596,675 (Table 4.1) to 201,655 (Table 4.2).

The best combination of hyperparameter is used to retrain the model with 5-fold cross validation. On the training test, the accuracy of the pure LSTM model is 98.51%, and the loss is 0.1181; on the testing set, the accuracy and loss are 97.06% and 0.3591, respectively. Figure 4.3 shows that one pickup video in the testing set was classified as inspection action.

**Table 4.1** Model summary before using Optuna.

| Layer (type) | Output Shape | Parameters |
|---|---|---|
| 1stm_6 (LSTM) | (None, 30, 64) | 442,112 |
| 1stm_7 (LSTM) | (None, 30, 128) | 98,816 |
| 1stm_8 (LSTM) | (None, 64) | 49,408 |
| dense_6 (Dense) | (None, 64) | 4,160 |
| dense_7 (Dense) | (None, 32) | 2,080 |
| dense_8 (Dense) | (None, 3) | 99 |
| Total params: | 596,675 | |

**Table 4.2** Model summary after using Optuna.

| Layer (type) | Output shape | Parameters |
|---|---|---|
| lstm_30 (LSTM) | (None, 30, 64) | 50,432 |
| lstm_31 (LSTM) | (None, 30, 112) | 79,296 |
| lstm_32 (LSTM) | (None, 80) | 61,760 |
| dense_20 (Dense) | (None, 121) | 9,801 |
| dense_21 (Dense) | (None, 3) | 366 |
| Total parameters: | 201,655 | |

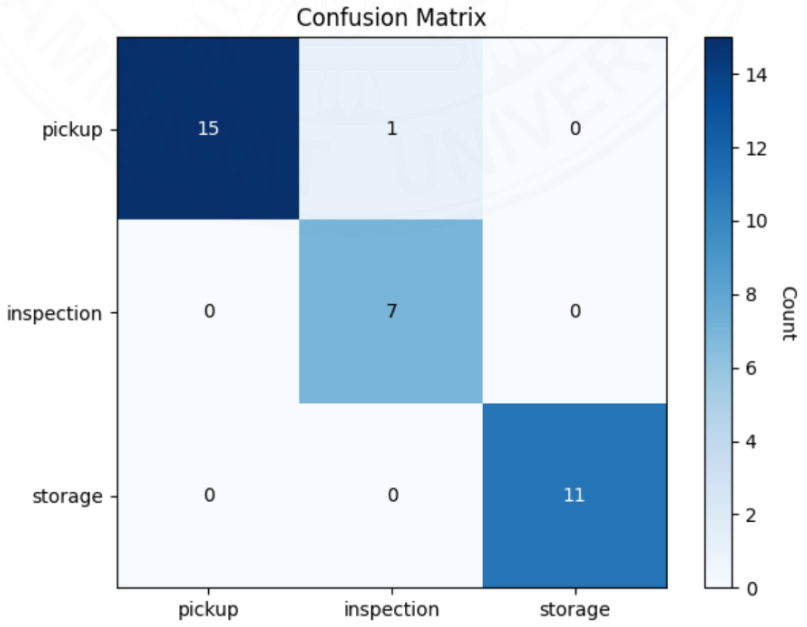


**Figure 4.3** Confusion Matrix for testing data.

### 4.2 The results of LSTM and Sequences rule

Due to the nature of the Sequence rule, the testing data must include only the motion which follows the order from pickup, inspection, then storage. The workflow cycle must be kept the same for the rest of the testing data set; otherwise, the system miscounts and skips the current action and waits until the motion in the following work cycle. A testing video with four work cycles measures the Sequence rule's accuracy.

**Table 4.3** Accuracy of Sequence rule.

| Cycle | Start | End | Manual | Sequence | Sequence (accuracy) |
|---|---|---|---|---|---|
| 1 | 16:23:18 | 16:23:20 | Pickup | Pickup | 100% |
| | 16:23:20 | 16:23:51 | Inspection | Inspection-Storage-Pickup | 61% |
| | 16:23:51 | 16:24:00 | Storage | Pickup | 0% |
| 2 | 16:24:00 | 16:24:02 | Pickup | Pickup | 100% |
| | 16:24:02 | 16:24:17 | Inspection | Inspection | 100% |
| | 16:24:17 | 16:24:25 | Storage | Storage-Pickup | 13% |
| 3 | 16:25:45 | 16:25:47 | Pickup | Inspection | 0% |
| | 16:25:47 | 16:25:56 | Inspection | Inspection | 100% |
| | 16:25:56 | 16:26:10 | Storage | Storage-Pickup | 7% |
| 4 | 16:26:10 | 16:26:12 | Pickup | Pickup | 100% |
| | 16:26:12 | 16:26:22 | Inspection | Inspection | 100% |

In Table 4.3, The inspection action in cycle 1 occurred with the "flickering effect." The detection of the system switched to the storage stage and pickup. Then for the storage stage in the first cycle, the sequence failed to detect because the current update is at pickup; based on the sequence rule, it cannot change from pickup to storage backward. Therefore, the system keeps the "Pickup" stage the same and waits until the inspection action for the following work cycle to update.

### 4.3 The results of LSTM and Intersection rule

To evaluate the accuracy of the LSTM model combined with the intersection model, we conducted a comparison between manual checks and the system's predictions using a 10-minute video clip that exclusively featured standard movements. The results of this comparison are illustrated in Table 4.4.

In the case of storage actions, both algorithms achieved the highest average prediction accuracy of 100%. However, when considering the overall accuracy, the LSTM combined with the Intersection Rule outperformed the Pure LSTM, showing a slight improvement from 97.17% to 98.13% (Table 4.4).

**Table 4.4** Accuracy comparation between Pure LSTM and Intersection rule.

| Action | Pure LSTM | LSTM and Intersection Rule |
|---|---|---|
| Pickup | 96.82% | 96.58% |
| Inspection | 96.06% | 97.80% |
| Storage | 98.63% | 100% |
| Total | 97.17% | 98.13% |

While the increase in accuracy may not be substantial, the integration of the intersection rule proved beneficial in mitigating the flicking effect that occurs during the transition between action stages. By incorporating the intersection rule, the system demonstrated better consistency and stability in accurately identifying and predicting the actions performed.

Figure 4.4 shows the prediction result of LSTM combined with the Intersection Rule through a 10-min video clip. The plot indicated that in cycle 34, there is an extreme outlier in storage action. After tracing back in the video, the reason is that the worker waited for the upcoming shoes from the conveyor. The idle motions could be analyzed from the study that way. However, the model has some minor “flickering effect” which occurred only 0.5 sec. For instance, at cycle 37, the system clicked 0.5 sec to inspect before correctly detecting the pickup at cycle 38.

## 4.4 The real-time application of proposed model

### 4.4.1 Intersection rule for real-time application

Due to the online assessment, the average FPS of the camera at the QC table is around 15. When the LSTM and Intersection rule model was applied through the SSH connection, the average FPS dropped slightly in Figure 4.5. The average FPS after applying the Intersection model is 13.59.

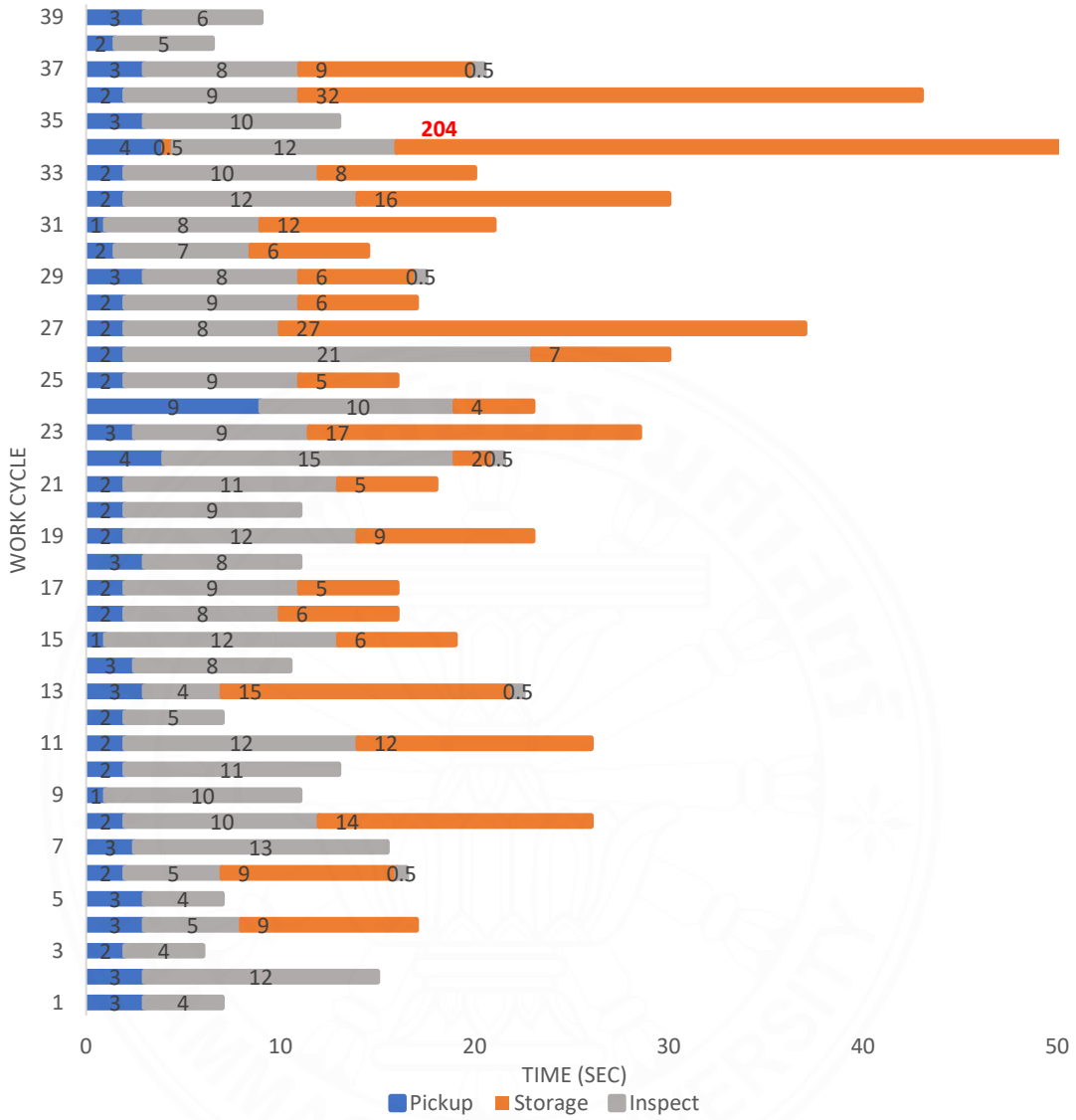


**Figure 4.4** Duration time for each work element using Intersection rule.

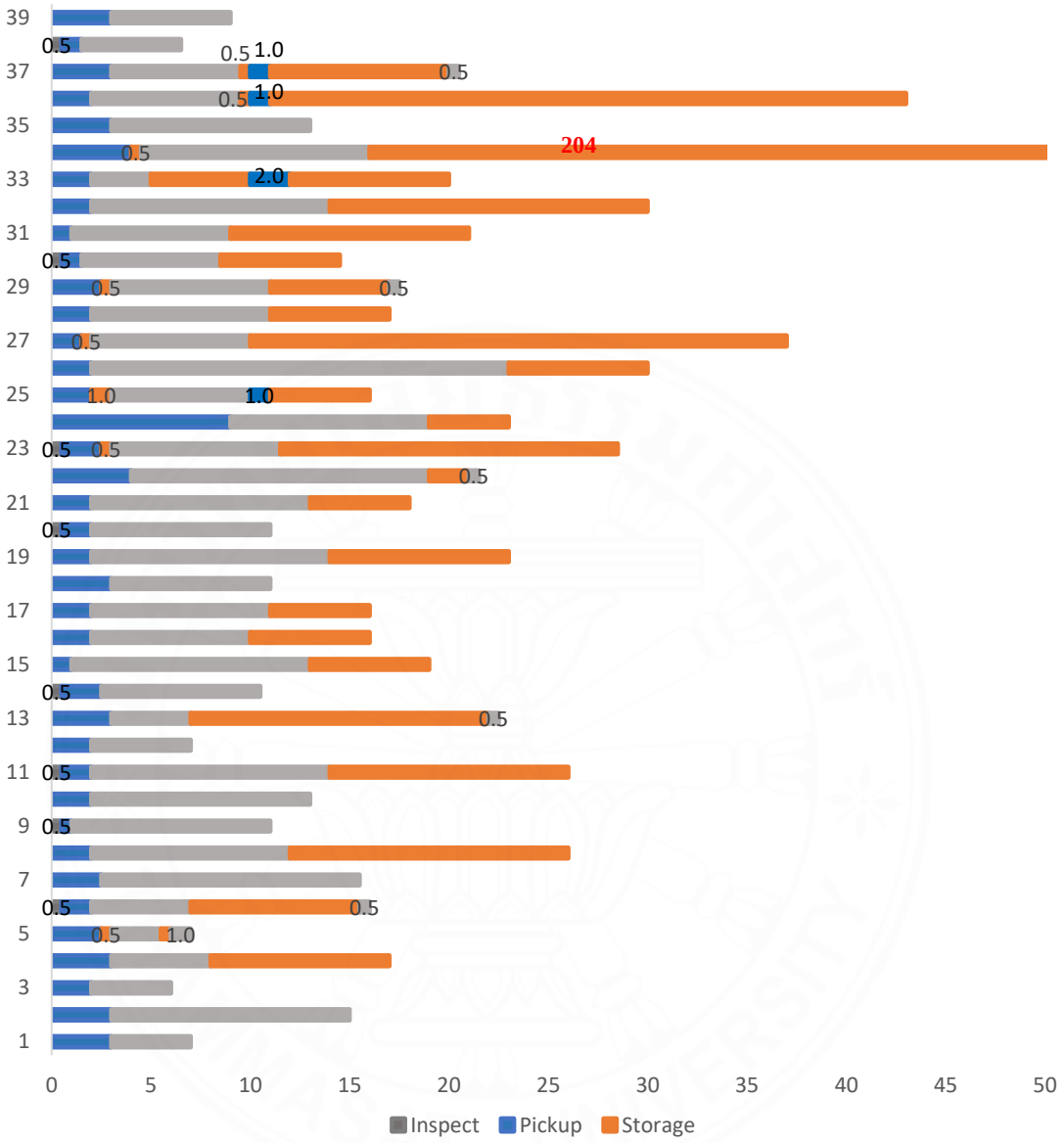


**Figure 4.5** Duration time for each work element using Pure LSTM.

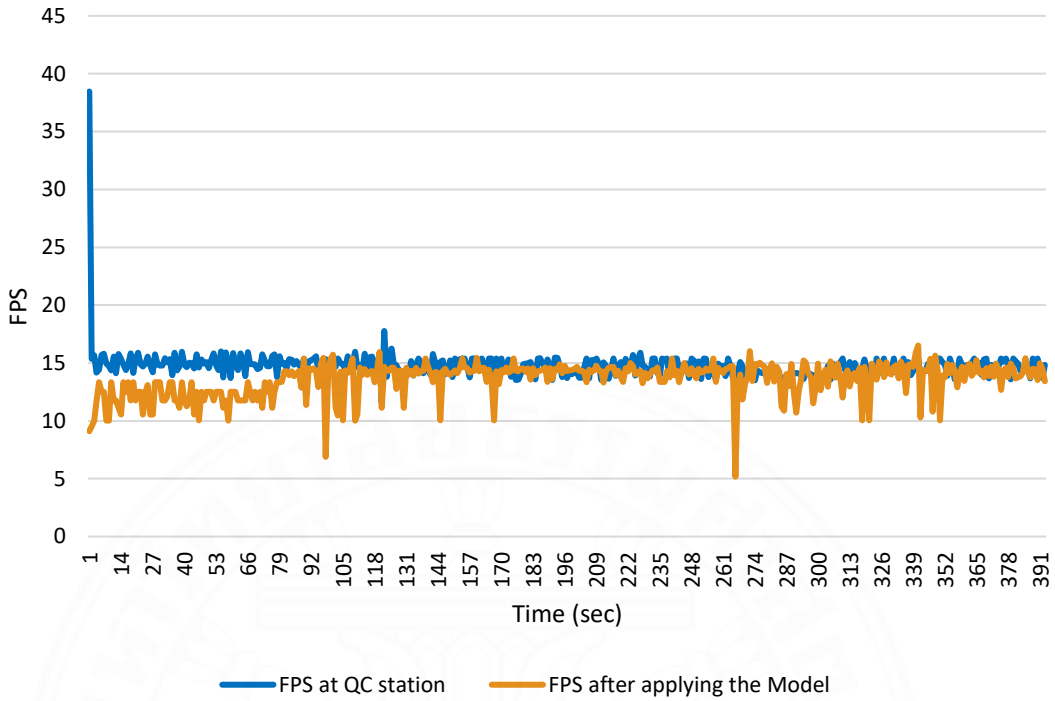


**Figure 4.6** Real-time FPS measurements at QC table.

Furthermore, the latency measured during the 10 min testing video clip for the intersection rule is around 0.07 seconds over 38,087 frames while utilizing a 2.40 GHz Core i5 - 9300H CPU running Windows 10 (Figure 4.7). In Figure 4.8 shows that latency of pure LSTM, the average of the pure LSTM is 0.02 seconds. The trade-off between stability and computational time. Although the accuracy between the pure LSTM and Intersection rule is not significantly different (97.17% and 98.13%), the “flickering effect” on the other hand should be put into consideration. In Figure 4.5 shows how the “flickering effect” impacted the stability of the model.

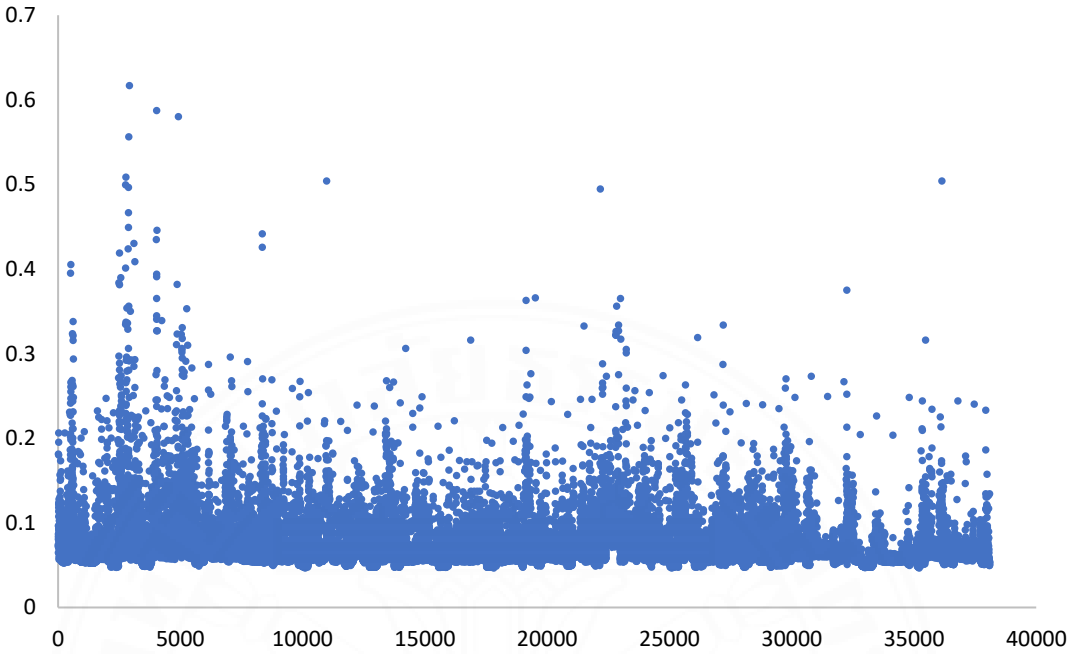


**Figure 4.7** The latency of Intersection rule for 10-min testing video.

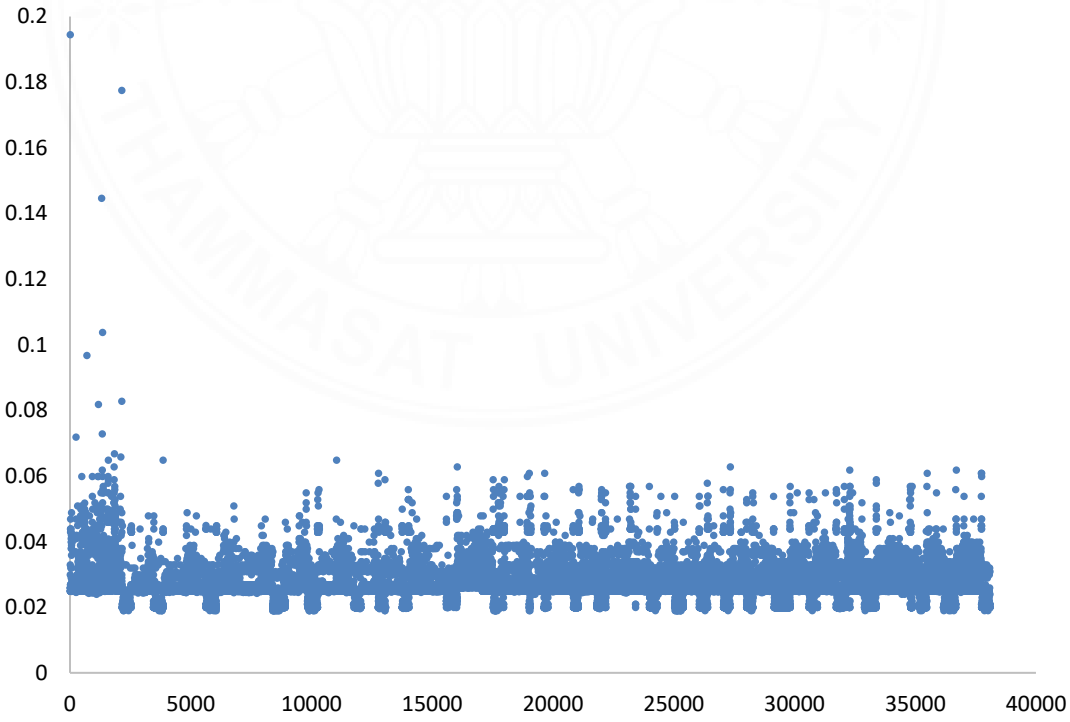


**Figure 4.8** The latency of Pure LSTM for 10-min testing video.

# CHAPTER 5
# CONCLUSIONS AND RECOMMENDATIONS

In this study, we proposed a novel approach that utilizes Mediapipe Pose and LSTM models to accurately recognize the actions of workers at a Quality Control station. Our evaluation of the pure LSTM model revealed impressive results, achieving an accuracy of 98.51% and a loss of 0.1181 on the training set. Similarly, on the testing set, the LSTM model achieved an accuracy of 97.06% and a loss of 0.3591. These findings demonstrate the effectiveness of LSTM in accurately classifying worker actions.

To address the issue of the "flickering effect" observed when workers transitioned between different actions, we incorporated the Intersection rule into the LSTM model. By doing so, we aimed to improve the model's stability and consistency during action recognition. The testing of the intersection rule combined with LSTM was conducted on a 10-minute video clip containing only the selected standard motions. The results were promising, with the combined algorithm achieving an average accuracy of 98.01% across all actions. This demonstrates the effectiveness of the intersection rule in maintaining accurate recognition even during transitional phases between actions.

The real-time applicability of the proposed algorithm was also evaluated, revealing a low latency of 0.07 seconds for the Intersection Rule and 0.02 seconds for the Pure LSTM model. This means that the system can provide real-time action recognition and timely feedback to control and analyze the efficiency of each functional element at the Quality Control station. The trade-off of accuracy and computation time must be considered for specific purposes. The integration of the MQTT protocol facilitated the publication of worker action statuses along with their corresponding timing, allowing for efficient monitoring and analysis of the production process.

However, it is essential to note that the reliability of the system's results may be limited to standard motions. In real-world production situations, workers may perform outlier motions or actions not covered by the trained model, which could impact the accuracy and stability of the system. To address this limitation, future work should focus on expanding the dataset to include a broader range of worker motions by using

multiple cameras. Additionally, considering the interaction of workers with objects such as shoes and conveyors could further enhance the accuracy and robustness of the system. Furthermore, the number of classes could be increased to include the recognition of idle movements performed by workers. This would allow for a more comprehensive analysis of the motion-time study, providing insights into the efficiency and productivity of workers during periods of inactivity.

In conclusion, our study has demonstrated the effectiveness of utilizing Mediapipe Holistic and LSTM models for worker action recognition at a Quality Control station. The proposed algorithm achieved high accuracy and low latency, showcasing its potential for real-time application. The incorporation of the Intersection rule further improved the system's stability during action transitions. However, future work should address the limitations of the system, such as the reliance on standard motions and the need for a more extensive and diverse dataset. By addressing these areas, we can enhance the accuracy and applicability of the proposed system, thereby contributing to the field of motion recognition and quality control in industrial settings.

# REFERENCES

Andriluka, M., Iqbal, U., Insafutdinov, E., Pishchulin, L., Milan, A., Gall, J., & Schiele, B. (2018). Posetrack: A benchmark for human pose estimation and tracking. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5167-5176).

Andriluka, M., Pishchulin, L., Gehler, P., & Schiele, B. (2014). 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 3686–3693. doi: https://doi.org/10.1109/CVPR.2014.471

Banús, N., Boada, I., Xiberta, P., Toldrà, P., & Bustins, N. (2021). Deep learning for the quality control of thermoforming food packages. *Scientific Reports*, *11*(1), 1-15.

Behera, A., Wharton, Z., Keidel, A., & Debnath, B. (2020). Deep cnn, body pose, and body-object interaction features for drivers' activity monitoring. *IEEE Transactions on Intelligent Transportation Systems*, *23*(3), 2874-2881.

Blanke, T., Bryant, M., & Hedges, M. (2012). Ocropodium: open source OCR for small-scale historical archives. *Journal of Information Science*, *38*(1), 76-86.

Bon, A. T., & Daim, D. (2010, April). Time motion study in determination of time standard in manpower process. In *3rd engineering conference on advancement in mechanical and manufacturing for sustainable environment*.

Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7291-7299).

Fan, X., Zheng, K., Lin, Y., & Wang, S. (2015). Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1347-1355).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

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

Isack, H., Haene, C., Keskin, C., Bouaziz, S., Boykov, Y., Izadi, S., & Khamis, S. (2020). Repose: Learning deep kinematic priors for fast human pose estimation. *arXiv preprint arXiv:2002.03933*.

Ji, J., Pannakkong, W., & Buddhakulsomsiri, J. (2022). A Computer Vision-Based Model for Automatic Motion Time Study. *Computers, Materials & Continua*, *73*(2).

Korkmaz, M., & Barstuğan, M. (2020). A Deep Learning-Based Quality Control Application. *Avrupa Bilim ve Teknoloji Dergisi*, 332-336..

Krenn, M. (2011). From scientific management to homemaking: Lillian M. Gilbreth's contributions to the development of management thought. *Management & Organizational History*, *6*(2), 145-161.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13* (pp. 740-755). Springer International Publishing.

Lu, B., Xu, D., & Huang, B. (2022). Deep-learning-based anomaly detection for lace defect inspection employing videos in production line. *Advanced Engineering Informatics*, *51*, 101471.

Lugaresi, C., & Tang, J. (2019). Hadon Nash, Chris Mc-Clanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, *4*, 15.

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14* (pp. 483-499). Springer International Publishing.

Palermo, M., Moccia, S., Migliorelli, L., Frontoni, E., & Santos, C. P. (2021). Real-time human pose estimation on a smart walker using convolutional neural networks. *Expert Systems with Applications*, *184*, 115498.

Prakash, C., Rao, B. P., Shetty, D. V., & Vaibhava, S. (2020, December). Application of time and motion study to increase the productivity and efficiency. In *Journal of Physics: Conference Series* (Vol. 1706, No. 1, p. 012126). IOP Publishing. doi: https://doi.org/10.1088/1742-6596/1706/1/012126

Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, *28*.

Sun, H., Ning, G., Zhao, Z., Huang, Z., & He, Z. (2020). Automated work efficiency analysis for smart manufacturing using human pose tracking and temporal action localization. *Journal of Visual Communication and Image Representation*, *73*, 102948.

Toshev, A., & Szegedy, C. (2014). Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1653-1660).

Villalba-Diez, J., Schmidt, D., Gevers, R., Ordieres-Meré, J., Buchwitz, M., & Wellbrock, W. (2019). Deep learning for industrial computer vision quality control in the printing industry 4.0. *Sensors*, *19*(18), 3987. doi: 10.3390/s19183987.

Xing, J., & Jia, M. (2021). A convolutional neural network-based method for workpiece surface defect detection. *Measurement*, *176*, 109185.

Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4694-4702).

Zheng, C., Wu, W., Chen, C., Yang, T., Zhu, S., Shen, J., ... & Shah, M. (2020). Deep learning-based human pose estimation: A survey. *arXiv preprint arXiv:2012.13392*.

Zhou, X., Sun, X., Zhang, W., Liang, S., & Wei, Y. (2016). Deep kinematic pose regression. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14* (pp. 186-201). Springer International Publishing.

# APPENDIX

# APPENDIX A

## PREDICTION RESULT OF INTERSECTION RULE

| Prediction No. | Start Time | Manual Check | Intersection Rule | Accuracy |
|---|---|---|---|---|
| 1 | 15:49:04 | pickup | pickup | 100% |
| 2 | 15:49:07 | inspect | inspect | 100% |
| 3 | 15:49:11 | pickup | pickup | 100% |
| 4 | 15:49:14 | inspect | inspect | 100% |
| 5 | 15:49:26 | pickup | pickup | 100% |
| 6 | 15:49:28 | inspect | inspect | 100% |
| 7 | 15:49:32 | pickup | pickup | 100% |
| 8 | 15:49:35 | inspect | inspect | 100% |
| 9 | 15:49:40 | storage | storage | 100% |
| 10 | 15:50:01 | pickup | pickup | 100% |
| 11 | 15:50:04 | inspect | inspect | 100% |
| 12 | 15:50:08 | pickup | pickup | 100% |
| 13 | 15:50:10 | inspect | inspect | 100% |
| 14 | 15:50:15 | storage | storage | 100% |
| 15 | 15:51:14 | pickup | inspect | 83% |
| 16 | 15:51:17 | inspect | inspect | 100% |
| 17 | 15:51:30 | pickup | pickup | 100% |
| 18 | 15:51:32 | inspect | inspect | 100% |
| 19 | 15:51:42 | storage | storage | 100% |
| 20 | 15:51:56 | pickup | pickup | 100% |
| 21 | 15:51:57 | inspect | inspect | 100% |
| 22 | 15:52:07 | pickup | pickup | 100% |
| 23 | 15:52:09 | inspect | inspect | 100% |
| 24 | 15:52:20 | pickup | pickup | 100% |
| 25 | 15:52:22 | inspect | inspect | 100% |
| 26 | 15:52:34 | storage | storage | 100% |
| 27 | 15:52:46 | pickup | pickup | 100% |
| 28 | 15:52:48 | inspect | inspect | 100% |
| 29 | 15:52:53 | pickup | pickup | 100% |
| 30 | 15:52:56 | inspect | inspect | 100% |
| 31 | 15:53:00 | storage | storage | 100% |
| 32 | 15:53:15 | pickup | inspect-pickup | 83% |
| 33 | 15:53:18 | inspect | inspect | 100% |
| 34 | 15:53:26 | pickup | pickup | 100% |
| 35 | 15:53:27 | inspect | inspect | 100% |
| 36 | 15:53:39 | storage | storage | 100% |
| 37 | 15:54:10 | pickup | pickup | 100% |
| 38 | 15:54:12 | inspect | inspect | 100% |

| | | | | |
|---|---|---|---|---|
| 39 | 15:54:20 | storage | storage | 100% |
| 40 | 15:54:26 | pickup | pickup | 100% |
| 41 | 15:54:28 | inspect | inspect | 100% |
| 42 | 15:54:37 | storage | storage | 100% |
| 43 | 15:54:42 | pickup | pickup | 100% |
| 44 | 15:54:45 | inspect | inspect | 100% |
| 45 | 15:54:53 | pickup | pickup | 100% |
| 46 | 15:54:55 | inspect | inspect | 100% |
| 47 | 15:55:07 | storage | storage | 100% |
| 48 | 15:55:16 | pickup | pickup | 100% |
| 49 | 15:55:18 | inspect | inspect | 100% |
| 50 | 15:55:27 | pickup | pickup | 100% |
| 51 | 15:55:29 | inspect | inspect | 100% |
| 52 | 15:55:40 | storage | inspect | 100% |
| 53 | 15:57:31 | pickup | pickup | 100% |
| 54 | 15:57:35 | inspect | inspect | 100% |
| 55 | 15:57:50 | pickup | storage-inspect pickup | 83% |
| 56 | 15:57:53 | inspect | inspect | 100% |
| 57 | 15:58:02 | storage | storage | 100% |
| 58 | 15:58:19 | pickup | pickup | 100% |
| 59 | 15:58:28 | inspect | inspect | 100% |
| 60 | 15:58:38 | storage | storage | 100% |
| 61 | 15:58:42 | pickup | pickup | 100% |
| 62 | 15:58:44 | inspect | inspect | 100% |
| 63 | 15:58:53 | storage | storage | 100% |
| 64 | 15:58:58 | inspect | inspect | 100% |
| 65 | 15:59:19 | pickup | pickup | 100% |
| 66 | 15:59:21 | inspect | inspect | 100% |
| 67 | 15:59:29 | storage | storage | 100% |
| 68 | 15:59:36 | pickup | pickup | 100% |
| 69 | 15:59:38 | inspect | inspect | 100% |
| 70 | 15:59:47 | storage | storage | 100% |
| 71 | 16:00:14 | pickup | pickup | 100% |
| 72 | 16:00:16 | inspect | inspect | 100% |
| 73 | 16:00:24 | storage | storage | 100% |
| 74 | 16:00:30 | pickup | pickup | 100% |
| 75 | 16:00:33 | inspect | inspect | 100% |
| 76 | 16:00:40 | storage | storage | 100% |
| 77 | 16:00:46 | pickup | inspect-pickup | 75% |
| 78 | 16:00:48 | inspect | inspect | 100% |
| 79 | 16:00:56 | storage | storage | 100% |

| | | | | |
|---|---|---|---|---|
| 80 | 16:01:02 | pickup | pickup | 100% |
| 81 | 16:01:03 | inspect | inspect | 100% |
| 82 | 16:01:15 | storage | storage | 100% |
| 83 | 16:01:27 | pickup | storage | 0% |
| 84 | 16:01:29 | inspect | inspect | 100% |
| 85 | 16:01:39 | storage | storage | 100% |
| 86 | 16:01:55 | pickup | pickup | 100% |
| 87 | 16:01:57 | inspect | inspect | 100% |
| 88 | 16:02:09 | storage | storage | 100% |
| 89 | 16:04:44 | pickup | pickup | 100% |
| 90 | 16:04:48 | inspect | storage-inspect | 95% |
| 91 | 16:04:58 | storage | storage | 100% |
| 92 | 16:08:22 | pickup | pickup | 100% |
| 93 | 16:08:25 | inspect | inspect | 100% |
| 94 | 16:08:34 | pickup | pickup | 100% |
| 95 | 16:08:36 | inspect | inspect | 100% |
| 96 | 16:08:44 | storage | storage | 100% |
| 97 | 16:09:16 | pickup | pickup | 100% |
| 98 | 16:09:19 | inspect | inspect | 100% |
| 99 | 16:09:27 | storage | storage | 100% |
| 100 | 16:09:36 | pickup | inspect | 75% |
| 101 | 16:09:36 | pickup | pickup | 100% |
| 102 | 16:09:38 | inspect | inspect | 100% |
| 103 | 16:09:43 | pickup | pickup | 100% |
| 104 | 16:09:46 | inspect | inspect | 100% |

# BIOGRAPHY

| | |
|---|---|
| Name | Le My Duyen |
| Education | 2019: Bachelor of Engineering (Logistics and Supply Chain Management) Ho Chi Minh City International University, Vietnam |

Publication

Duyen, L.M., Jeenanunta, C., Tunpan, A., & Sirimarnkit, N. (2022). A Mediapipe Holistic and a Long-Short Term Memories (LSTM) for Motion Study: Application in the Shoe Manufacturing. In *Proceedings of International Conference on Logistics and Industrial Engineering* (pp. 297-304).