אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

Manual Operation Project

Students:

Deema Abu Hwaij, id: 211638879

Leen Wattad, id: 211324488

15.04.2024

Introduction:

In this project we will collect videos of manual operations, such as screwing, pushing, and measuring. Then train a model to detect these operations, test their accuracy and measure their performance.

The goal of this project is to develop a model system capable of detecting and measuring the performance of various manual operations, including but not limited to screwing, pushing, and measuring. The system will analyze video data and identify specific manual operations within a given context. Additionally, the system will quantify the performance of these operations based on defined metrics.

Operation detection generally refers to the process of identifying, monitoring, and analyzing operations or activities within a system, organization, or environment. This term is commonly used in various contexts, including: security, Industrial Processes, Financial Transactions, Healthcare. Overall, operation detection plays a crucial role in various fields by enabling organizations to identify and respond to abnormal or suspicious activities effectively. This helps enhance security, efficiency, and overall performance.

Deep learning offers a powerful solution for the challenges outlined in this project. With its ability to automatically learn and adapt from vast amounts of data, deep learning models can be trained to accurately detect and classify manual operations depicted in videos. By leveraging techniques such as convolutional neural networks (CNNs), deep learning algorithms can effectively learn the intricate patterns and features inherent in manual operations like screwing, pushing, and measuring.

Moreover, deep learning models can offer robustness and scalability, allowing them to handle various scenarios and adapt to different environmental conditions. Through continuous training and refinement, these models can improve their accuracy and performance over time, ensuring reliable operation detection even in complex and dynamic environments.

By integrating deep learning into the project's framework, it becomes possible to automate the process of operation detection and performance measurement from video data. This not only enhances efficiency but also opens up opportunities for real-time monitoring and analysis, facilitating prompt decision-making and intervention when necessary. In essence, deep learning serves as a pivotal technology in advancing the capabilities of systems aimed at detecting and measuring manual operations, thereby contributing to enhanced productivity and effectiveness across diverse industries and applications.

Methodology:

- **Dataset:** For this project, we collect a number of videos for each operation using the GoPro camera.
- **Hand Action (operations) Recognition Dataset Explanation:**

1. **Screw:** Screwing in and screwing out refer to the actions of inserting or removing a screw from a threaded hole or surface. Here's an explanation of each:

   • Screw-In: This refers to the process of inserting a screw by rotating it clockwise.

   • Screw-Out: This refers to the process of removing a screw from a threaded hole or surface by rotating it counterclockwise.

2. **Hammering:** When hammering, you use a hammer or mallet to strike an object with force, causing it to deform, shape, or move.
3. **Plug-Unplug:** the process of connecting or disconnecting an electrical appliance or device to a power source by inserting its plug into a compatible outlet.
4. **Open-Close:**

   • Open: Opening involves moving an object or structure from a closed position to an open position.

   • Close: Closing involves moving an object or structure from an open position to a closed or sealed position.

5. **Click:** The click operation involves pressing down and releasing a button or key on an input device, such as a mouse, keyboard, or touchscreen.
6. **Measure:** is the process of quantifying or determining the magnitude, extent, size, or quantity of something.
7. **Cut:** the action of dividing or separating an object or material into two or more pieces using a sharp tool or instrument.
8. **Tug:** encompasses both pushing and pulling actions, combining the efforts of exerting force in opposite directions.
9. **Cover-Uncover:** "Cover" and "uncover" are verbs that describe actions related to concealing or revealing something.

   • Cover: To cover something means to place or put something over or around it in order to conceal, protect, or hide it from view.

   • Uncover: To uncover something means to remove or reveal a covering or concealment from it, exposing it to view or making it visible.

10. **Attach-detach:** Attach" and "detach" are verbs that describe actions related to joining or separating objects or components from each other.

    • Attach: To attach something means to connect or fasten it to another object or surface.

• Detach: To detach something means to separate or disconnect it from another object or surface.

11. **Lift:** refers to the action of raising or elevating an object or oneself from a lower position to a higher position
12. **Round:** "Round in" and "Round out" are verbs that describe action related to turn or round the item.

- **The table below describing the operations:**

	Operation	Amount of videos	The total amount	Hand	Tool
1.	Screw in – electric screw in Screw out – electric screw out	63 63	126	1/2	Screwdriver, spanner, Wrench, Electric Screwdriver
2.	Measuring	42	42	2	Tap , Roller, Caliper
3.	Hammering in Hammering out	43 23	66	1/2	Hammer , Mallet
4.	Open Close	34 36	70	1/2	-
5.	Cutting	57	57	2	Wires , Scissors
6.	Plug Unplug	29 34	63	1/2	Wires
7.	Turning in Turning out	37 33	70	2	Spanner
8.	Tug – push Tug – pull	20 20	40	2	-
9.	Click	30	30	1/2	-
10.	Lift	29	29	2	-
11.	Cover Uncover	16 16	23	2	-
12.	Attach Detach	16 15	31	2	-

- **Link for the dataset:**
  https://drive.google.com/drive/folders/1u8XAdIdac4zSSWzQLYy-OYudQABeMHYI

- **Data Preparation:**
- data balance prior to training, the dataset underwent several preprocessing steps to standardize the data and prepare it for model input.
  We decide to make 100 videos for each operation class where in each video consist a 100 per frame.
  We made the data balance using these steps:
  1. **Augmentation:**
     To increase the diversity of our training data - to reach 100 videos for each operation - and improve the model's robustness, we apply data augmentation techniques such as rotation, random cropping, flipping, and changes in lighting conditions.
  2. **Frame Extraction:**
     We managed to reach 200 videos for each label, where each video is represented by 100-500 frames.

Feature Extractor:

- **In our model we used this technologies and frameworks related to computer vision and deep learning to extract features from the frame:**

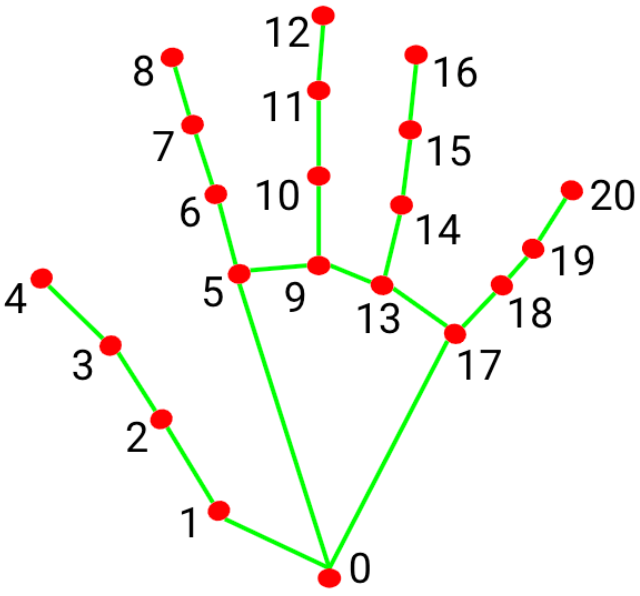
1. **CNN:** A Convolutional Neural Network (CNN) is a type of deep learning algorithm specifically designed for image processing and recognition tasks. Compared to alternative classification models, CNNs require less preprocessing as they can automatically learn hierarchical feature representations from raw input images. They excel at assigning importance to various objects and features within the images through convolutional layers, which apply filters to detect local patterns. By stacking multiple convolutional and pooling layers, CNNs can learn increasingly complex features, leading to high accuracy in tasks like image classification, object detection, and segmentation.

   ✓ **VGG:** The VGG model is a deep neural network that achieved state-of-the-art performance on the ImageNet Large Scale Visual Recognition Challenge in 2014, and has been widely used as a benchmark for image classification and object detection tasks. The VGG model uses max pooling layers to reduce the spatial resolution of the feature maps and increase the receptive field, which can improve its ability to recognize objects of varying scales and orientations. Our project is based on PyTorch which is a a machine learning library based on the Torch library, PyTorch provides a convenient environment for working with VGG models, allowing researchers and practitioners to leverage the power of this architecture for tasks such as image classification, feature extraction, and transfer learning.

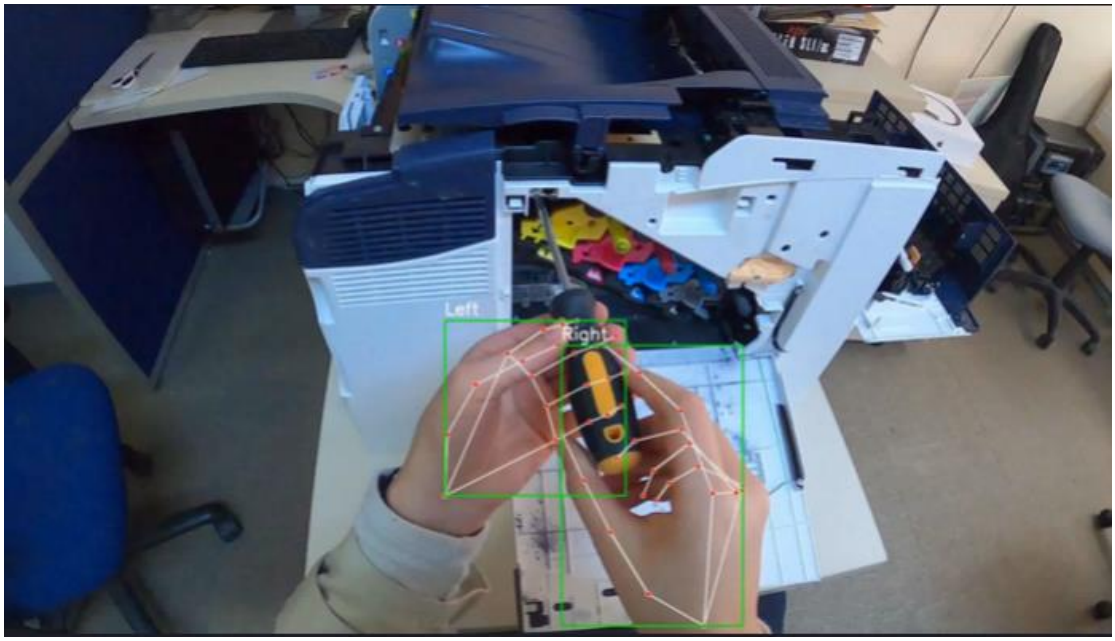
2. **YOLO:** You Only Look Once (YOLO) is a popular object detection algorithm in computer vision. The key idea behind YOLO is to divide the input image into a grid and predict bounding boxes and class probabilities directly for each grid cell. This approach allows YOLO to make predictions for multiple objects simultaneously and efficiently, resulting in real-time performance.
   We are using yolo for detecting the tool in each frame, implementing YOLO for tool detection in this project enhances its ability to precisely identify and track tools like screwdrivers, pushers, and measuring instruments in video frames. YOLO's efficiency in processing frames in real-time ensures rapid and accurate detection without compromising performance. By training YOLO on annotated data specific to manual operations, the system can effectively localize and classify tools, providing valuable insights into their usage and performance throughout the video footage. With YOLO's versatility and seamless integration, this project can leverage cutting-edge object detection capabilities to streamline operation detection and measurement, contributing to improved efficiency and performance evaluation across various industries.

Example for detecting the tool from our data:

3. **MediaPipe**: MediaPipe is an open-source framework for building pipelines to perform computer vision inference over arbitrary sensory data such as video or audio. It provides tools and pre-built components for tasks like object detection, face detection, hand tracking, pose estimation, and more.



Example for detecting the hands from our data:



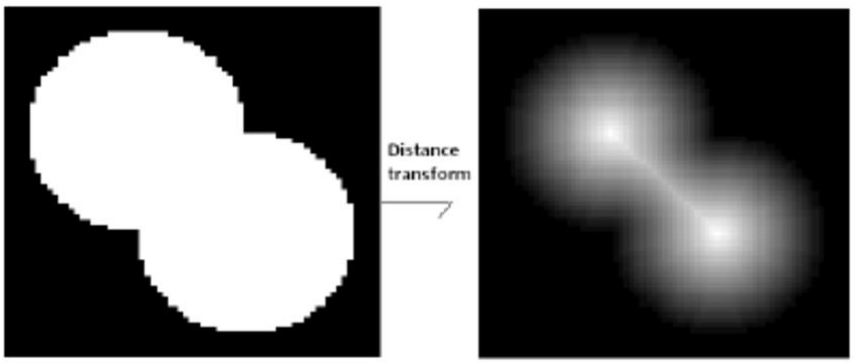
- Combination MediaPipe for hand tracking complements the tool detection aspect of this project by providing a comprehensive understanding of manual operations. MediaPipe offers robust hand tracking capabilities, enabling the system to accurately detect and track the movements of hands involved in the manual operations depicted in the videos. By integrating MediaPipe alongside YOLO for tool detection, the system can analyze not only the usage of tools but also the intricate interactions between hands and tools throughout the video footage. This allows for a more holistic approach to performance measurement, as it provides insights into the dexterity and coordination involved in executing manual tasks. Leveraging MediaPipe's hand tracking capabilities enhance the project's ability to quantify the performance of manual operations, ultimately contributing to a more thorough analysis and evaluation of operational efficiency and effectiveness.

- Once YOLO has detected tools and MediaPipe has tracked hand movements, we can calculate bounding ellipses for both the detected tools and the tracked hands. Here's how you can do it:
  For the detected tools (from YOLO), we can calculate the bounding ellipse based on the bounding box coordinates. Convert the bounding box into a set of points representing the four corners of the box, then fit an ellipse to these points using ellipse fitting algorithms such as the least squares method or the eigenvector-based method.
  For the tracked hands (from MediaPipe), we can calculate the bounding ellipse based on the hand landmarks provided. Use the landmarks representing the fingertips and palm positions to fit an ellipse that encapsulates the hand's shape and orientation. Again, employ ellipse fitting algorithms to calculate the parameters of the bounding ellipse.

- **Bounding ellipse:**

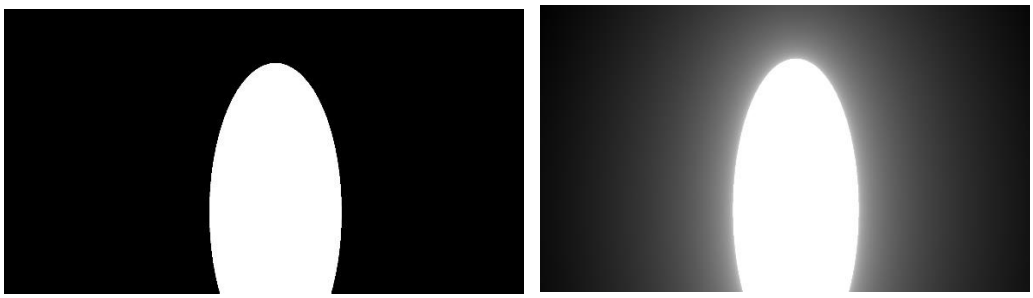
Detecting hand or manual action involves localizing the working hands, the applied tools. The action components allow for narrowing down the analysis to specific regions of interest (ROIs), excluding irrelevant background data. The pose of the hands, which includes the relative location of the joints, is crucial to interpreting hand actions.

We use this technic to enforce the model to learn from the relevant features not from the background (keep attention on the device and less focus on the background)



To analyze a video segment that is a sequence of frames hands engaged in an action, such as screw a bolt, the initial step involves pinpointing the hands, the tools being used, and the specific area of the device being manipulated - the action components typically situated nearby. Additionally, the surrounding background often contains elements relevant to understanding the action. Thus, we delineate the action region, denoted as R, encapsulating the action components within a bounding ellipse. This region R forms the saliency mask, exhibiting strong values within R and Distance transform.

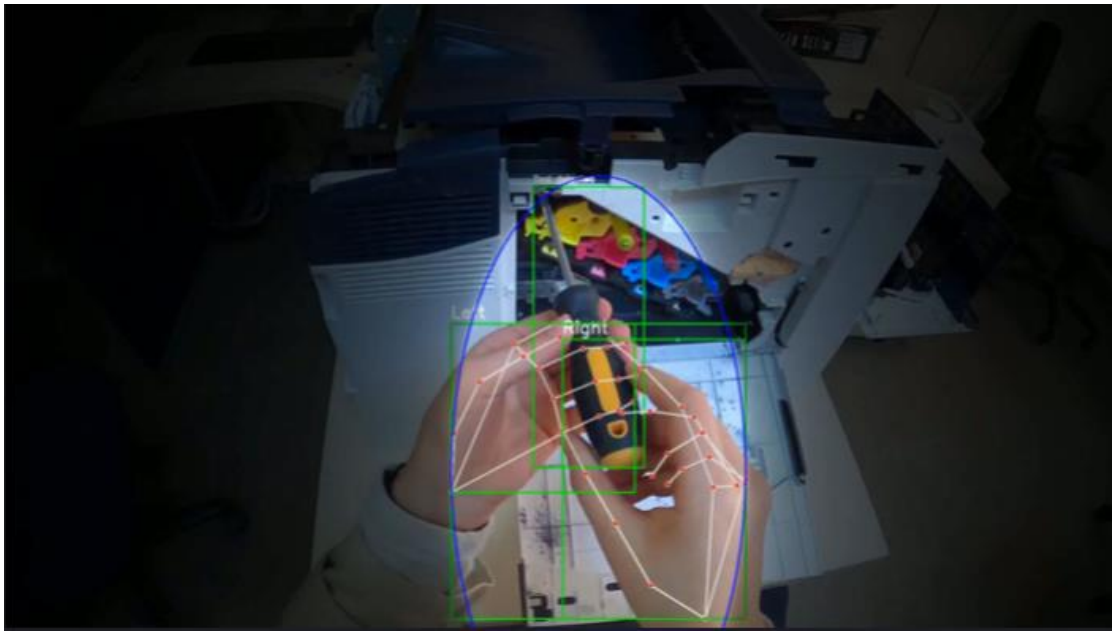
Example for the ellipse mask from our data:



This setup enables the learning model to concentrate on the hand gestures and the crucial components influencing the action. Leveraging a Transformers architecture, the model is designed for this purpose.

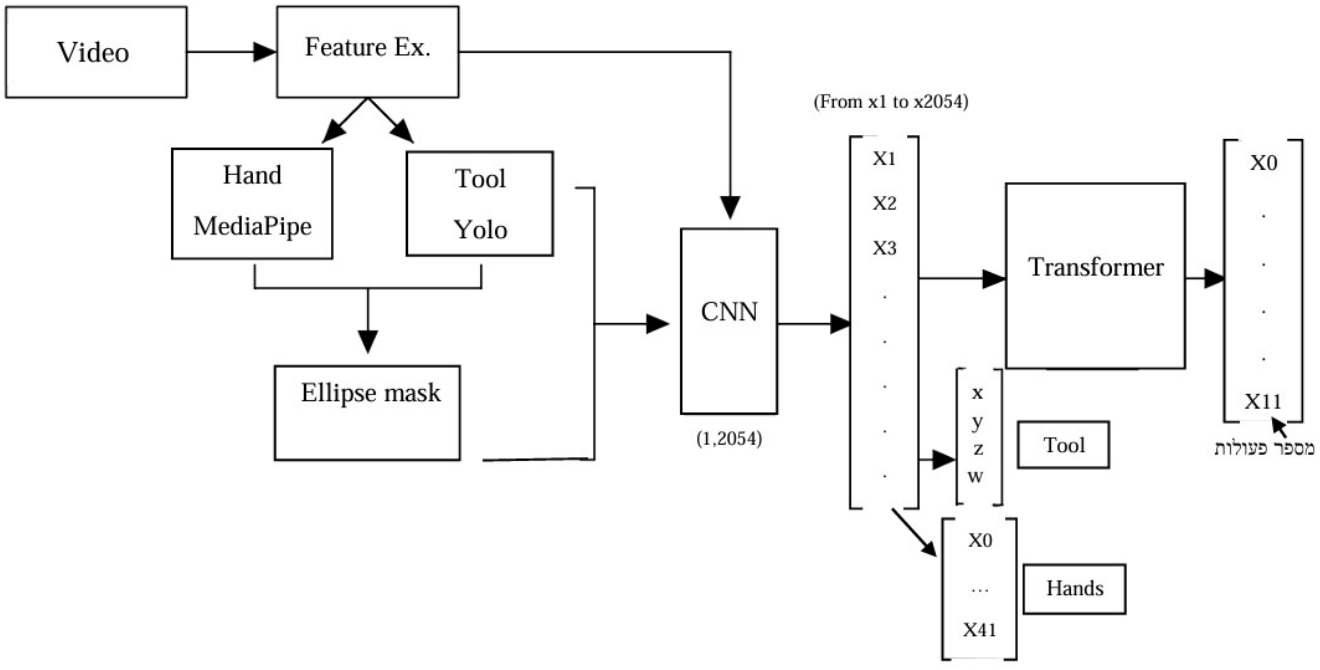
Our Feature Extraction (FE) module is employed to extract features from each frame. It amalgamates Convolutional Neural Network (CNN) features from the frame with skeletal features, facilitating a comprehensive analysis.

Example from our data after detecting the tool, hands and the bounding ellipse:



- **The model, and how it works:**

The action recognition model is based on the Transformers architecture, as it only uses the encoder part. We create a saliency map encompassing the working hands, the applied tool/s. An input frame is multiplied by its corresponding mask to guide a Convolutional Neural Network (CNN) to extract representative features from the region of interest. The sequence of CNN features, computed from the series of frames that define the action, is fed to a Transformer model. The model's output goes through a fully connected network that determines the action.



- ✓ **Transformers Architecture:**

The action recognition model in this project utilizes the Transformers architecture, specifically the encoder part. Transformers are known for their effectiveness in capturing long-range dependencies in sequences, making them suitable for tasks involving temporal data like video analysis. By leveraging the encoder part of the Transformer model, the system can process sequences of features extracted from video frames to understand temporal patterns in manual operations.

- ✓ **Saliency Map and Region of Interest (ROI):**

The system generates a saliency map to highlight regions of interest in the input frames, encompassing the working hands and the applied tools. This map serves as a mask that guides a Convolutional Neural Network (CNN) to focus on extracting features specifically from these regions, enhancing the model's ability to capture relevant information for action recognition.

- ✓ **Feature Extraction with CNN:**

The input frames are multiplied by their corresponding masks, directing a CNN to extract representative features from the highlighted regions of interest. CNNs are well-suited for image feature extraction tasks, and by focusing on the regions identified by the saliency map, the CNN can capture detailed information about the hands and tools involved in the manual operations.

- ✓ **Transformer Model:**

The sequence of CNN features, computed from the series of frames that define the action, is then fed to a Transformer model. This Transformer model processes the sequence of features, leveraging its ability to capture temporal dependencies across frames. By analyzing the temporal dynamics encoded in the sequence, the Transformer model can recognize patterns indicative of different manual operations.

- ✓ **Output Prediction:**

The output of the Transformer model is passed through a fully connected network, which serves as a classifier to determine the action being performed. This network maps the learned representations from the Transformer model to the corresponding action labels, providing the final prediction of the manual operation depicted in the input video sequence.

Experiment

- **Training details:**

In the section of evaluating the efficacy of our manual operation detection model, we conducted a series of experiments using the CNN. We split our dataset into training - 70%, testing 10% and Validation 20% subsets, ensuring that our model was evaluated on previously unseen data.

We managed to reach 100 videos for each label, where each video is represented by 100-500 frames. We train the model end-to-end for 300 epochs, continuously improving its representations and honing its ability to distinguish between different hand actions.

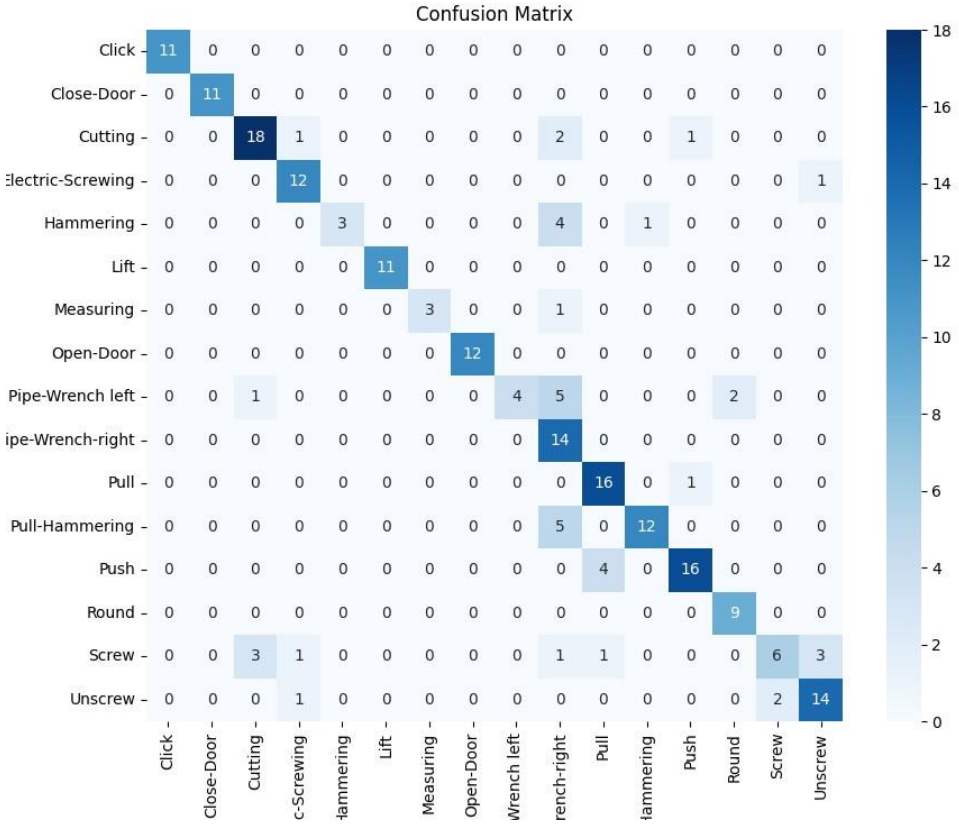
In evaluating the model's performance, we utilized metrics including accuracy, precision, and recall. These metrics offered a thorough assessment of the model's capacity to accurately discern various manual operations within video clips.

The findings were notably positive and encouraging. The model exhibited a commendable level of accuracy in identifying manual operations, as evidenced by precision and recall scores that showcased its adeptness in accurately categorizing positive instances. These favorable results not only underscore the efficacy of our methodology but also emphasize its practical potential in real-world scenarios where precise detection of manual operations holds significant importance.

Results

- **confusion matrix:**

In machine learning, classification is the process of categorizing a given set of data into different categories. In machine learning, to measure the performance of the classification model, we use the confusion matrix. Through this tutorial, understand the significance of the confusion matrix. A confusion matrix is a matrix that summarizes the performance of a machine learning model on a set of test data. It is a means of displaying the number of accurate and inaccurate instances based on the model's predictions. It is often used to measure the performance of classification models, which aim to predict a categorical label for each input instance.



The confusion matrix of the detection results provides another view of the performance of our model. It is a valuable tool for visualizing and systematically evaluating the model's predictions compared to the actual hand action labels. In addition, it provides an overview of the model's strengths and areas where it encountered difficulties. This in-depth analysis enabled us to pinpoint specific categories that were frequently misclassified, providing deeper insights into the model's behavior.

- **Link for the results:**

https://drive.google.com/drive/folders/1CRpOHyemoVhc3QwvWJVhUiFNfyu0-NKL

Conclusion

- ## Interpretation of Results and Model Analysis:

We've introduced an innovative dataset tailored to detect hand actions involved in manipulating mechanical devices, covering tasks like assembling and dismantling. Additionally, we present a novel action detection model leveraging Transformer-based architecture. Each entry in this dataset corresponds to a first-person-view video segment capturing hands engaged in specific actions, which may involve tool usage and device manipulation.

Our deep learning model extracts features from individual frames within the video segments, integrating position embedding, and then feeds these feature vectors into a Transformer Encoder. The resulting output vector undergoes further processing through a fully connected network to generate the final classification.

Our model is implemented and trained using the provided dataset, with experimental evaluations yielding promising results. Additionally, the model's performance provides confidence in its potential applicability to real-world scenarios where accurate detection of manual operations is crucial.

- ## Efficiency for the model:

We have evaluated our model’s performance using accuracy, precision, and recall metrics. The outcomes were encouraging. The model exhibited good accuracy in recognizing hand actions, with precision and recall scores indicating its adeptness in accurately categorizing positive instances. The accuracy result of our data is: Accuracy: 80.75% , Precision: 85.34%.

These results demonstrate the efficacy of our approach and emphasize its potential for practical applications where precise detection of manual operations holds great significance.