# Image-Based Hand Tools and Accessories Recognition by ResNet50

Chomtip Pornpanomchai

*Faculty of Information and communication technology, Mahidol University, Phuthamonthon 4 road, Salaya, Nakhorn Pathom, Thailand, 73170, E-mail: chomtip.por@mahidol.ac.th*

**Abstract**: The objective of this research is to create a computer system which can recognize various kinds of hand-tools by using only a single image. The developed system is called "Hand-tool and accessory image recognition system or (HTAIRS)". The system consists of 4 main modules, namely: 1) dataset training, 2) image acquisition, 3) image recognition, and 4) result presentation modules. The system employs the convolutional neural networks (CNN) called "ResNet50", which is a toolbox in MATLAB software. The developed system creates its own dataset called "Hand Tools Dataset", which consists of 165 different video clips in 110 hand tool categories and 600 images each. The HTAIRS separates 500 images for training dataset and 100 images for evaluating the system. The accuracy of the training system is 99.30% and the accuracy that of the evaluating is also 99.30%. The system's average access time are is 0.8549 Seconds per image.

**Keywords:** Convolutional Neural Network, Hand Tools and Accessories, Image Processing Pattern Recognition, Resnet50.

## 1. INTRODUCTION

It is very difficult to find handy men to repair people house during the Covid-19 situation or to recover a house after a big flooding. Some people try to repair their house by themselves but some of them cannot do it. Because they do not have a knowhow and skill to operate the hand tools or power tools. Moreover, miss-operation hand tools will cause three kinds of damage, namely: 1) users injure, 2) hand tool broken and 3) both users and hand tools damage. The first sample is some people use pliers to hit a nail and miss to hit the nail but hit their finger. The second sample is people use a wood cutting saw to cut metal. The consequence is to blunt a wood cutting saw. The third sample is people use the wrong size of cement grinding wheel to cut a concrete wall. It will cause a serious injure to them or both users and a cement-grinding wheel will injure and crack. Therefore, people need to recognize various hand tools, learn how to use them and practice their skill to operate the hand-tools.

### 1.1 Hand Tools Classifications

Hand tools are classified in five groups, namely: 1) laying out the tools, 2) striking tools, 3) cutting tools, 4) holding tools and 5) sharpening and grinding tools [1]. Each of them has the following details.

**1.1.1. Laying Out the Tools** – these kind of hand tools use for measure an object size, namely: a tape measure, a machinist square, a level, as shown in Figure 1 (a) – 1 (c), respectively.

**1.1.2. Striking Tools** – there are many kinds of hammers for striking objects, such as claw hammer, straight peen hammer and sledges hammer, as shown in Figure 1 (d) - 1 (f), respectively.

**1.1.3. Cutting and Drilling Tools** – These tools use for cutting material or make a material hole such as hand saw uses for cutting metal, a wireless drill use for making a hole, pipe cutter pliers for cutting PVC pipe and scissors to cut the material, as shown in Figure 1 (g) – 1 (j), respectively.

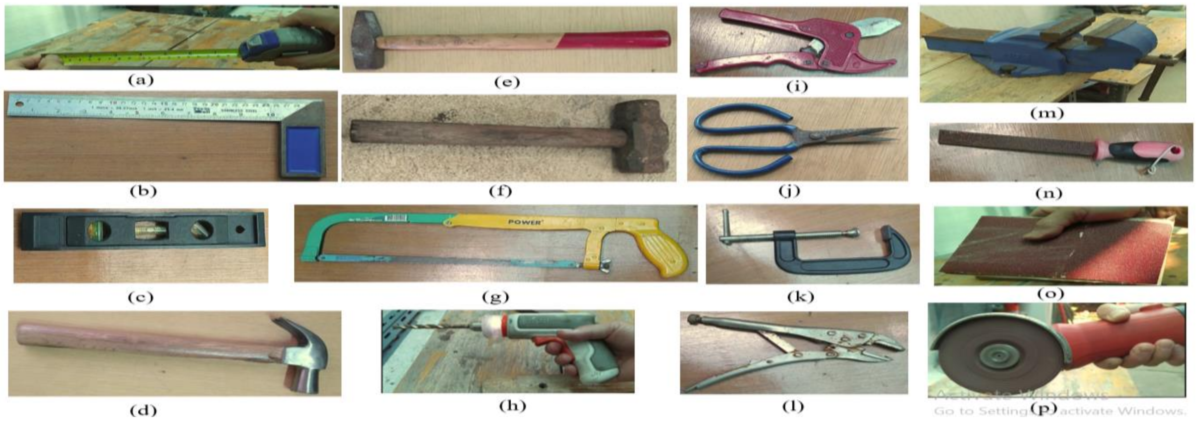


**Figure 1**: Images of some hand tools in the HTAIRS dataset, which are: (a) a tape measure, (b) a machinist square, (c) a level, (d) a claw hammer, (e) a straight peen hammer, (f) a sledges hammer, (g) a hand saw, (h) a wireless drill, (i) pipe cutter pliers, (j) scissors, (k) a C-camp, (l) a locking pliers, (m) a vise, (n) a file, (o) a sandy paper and (p) a grinder.

**1.1.4. Holding Tools** – These tools use for holding the objects, such as a C-camp, a locking pliers and a vise, as shown in Figure 1 (k) – (m), respectively.

**1.1.5. Sharpening and Grinding Tools** – the sample of these kind of tools are files, sandy paper and grinder, as shown in Figure 1(n) – 1 (p), respectively.

### 1.2 Hand Tools Recognition Purpose

Many researchers conducted the experiment to recognize various kind of tools in three main purposes, which are 1) the safety purpose, 2) the robotic purpose and 3) the hand tools usages purpose. Each has the following briefly details.

#### 1.2.1. Safety Purpose

Kurnaz et al. [2] employed many kinds of convolutional neural networks (CNN) to detect 49 different hand tools for the safety use-case with the mean average precision (mAP) of 30.6, 34.1, 22.7, 28.0, 27.4 and 27.1 for fast R-CNN, cascade R-CNN, RetinaNET, YOLOv3, RepPoint and FreeAnchor, respectively. Yang et al. [3] developed the personal protection equipment (PPE) for tools and protection equipment pair checking system. The classification for PPE protection every parts of worker body, which are head with bump caps, eyes with safety goggles, hearing with ear plugs, breathing with respirators, hand with gloves, foot safety toed shoes, skin coats and body with safety vest. Oka et al. [4] presented how to recognize 24 kinds of dental instruments in 1425 images with correctly and operate safely by using a convolution neural network. The average precision rates of the system are 89.7%

#### 1.2.2. Robotic Purpose

Mancini et al. [5] illustrated how robot operated nine different hand tools for three different categories with 4,500 RGB images, which are hammer, pliers and screwdriver. The experiments conducted on Kinect and webcam by using AlexNet with the accuracy around 90.0%. Pavlasek et al. [6] showed robots working in human tabletop environment for recognizing and localizing hand tools in uncluttered and cluttered configuration. The experiment conducted on eight hand tools instances: hammer, clamp, box cutter, flash light, screwdriver, longnose pliers and 2 instances of lineman's pliers. Qin et al. [7] presented a framework of learning-key point representation of tool-based manipulation (KETO) for self-supervised robot's arm interactions without explicit human annotations. The KETO tasks have three operations, which are hammering, pushing and reaching and each task requires different strategy of grasping and manipulation for achieving the task goal.

### 1.2.3. Usages Purpose

Nakamura and Nagai [8] demonstrated how to understanding the hand tools appearance, function and usage. The system dataset consists of nine hand tools, which are scissors, pen, pliers, tweezers, cutters, stapler, glue, scotch tape and vinyl tape. The system employed the scale invariant feature transform (SIFT) to find the inference of the function of all hand tools with the accuracy around 98.0 %. Stasiak [9] used fast orthogonal neural network (FONN) to recognize 10 different hand tools, which are brush, clamp, hook, knife, pencil, pliers, scissors, screwdriver, spanner and string. Three experiments conducted on 50 images of STaR dataset, 50 images of STaR-light dataset and 100 images of STsR-light and STaR-light-30 dataset with the recognition rate of 100%, 79.6% and 63.5%, respectively. Zhu et al. [10] presented a framework for task-oriented object modeling, learning and recognition. The experiments conducted on 32 kind hand tools, 24 household objects and 16 natural stones. The classification accuracy of the system is 89.30%. Zamora-Hernandez et al. [11] proposed two manufacturing processes, which are 1) recognizing 24 categories with 80,000 images of hand tools and accessories and 2) identification of basic actions in assembly processes. The sensitivity of the developed system is around 98.00%. Shilkrot et al. [12] showed how to retrieve hand with hand tool image matching with the video performing similar operations in the system dataset. The system conduct with 13 different tools, in 39 video files and each recorded in 150 frames. The average precision rate of the system around 0.9563.

According to the previous hand tools recognition researches, many researchers employed many type of convolutional neural networks (CNN) to recognize the hand tools namely: R-CNN, YOLO, RetinaNet, RepPoint, FreeAnchor, AlexNet, ResNet [2][4,5][11,12]. The details of the convolutional neural networks are describe in the next section.

## 1.3. Convolutional Neural Networks (CNN)

Normally, the convolutional neural networks consists of three main components, which are feature extraction, classification and output classification component, as shown in Figure 2. The operations of the CNN have the following steps. First, getting an input image with, then a feature extraction component is repeated in the convolution and pooling layer until it extracts all the input image features. Second, the full connection component uses all features for neural network input layer and classifies an input image to show to an output layer. Finally, the output classification component displays the most similar between the input image and image in the dataset.

Based on the previous hand tools recognition researches, some researchers showed the results in the video clip [12]. The intension of showing the video clip output is to demonstrate how to operate the hand tools correctly. The hand tool image-to-video retrieval is a simple and easy technique for people to query video clips because it is easy for them to take a picture with their mobile phone and use that mobile phone to query the most similar between an input image and a video clip in the dataset. The contribution of this research is to develop a computer system, which can get a hand tool input image and recognize it, then present how to operate the hand tool by using video presentation. The architecture and methods of HTAIRS consist of the following details.

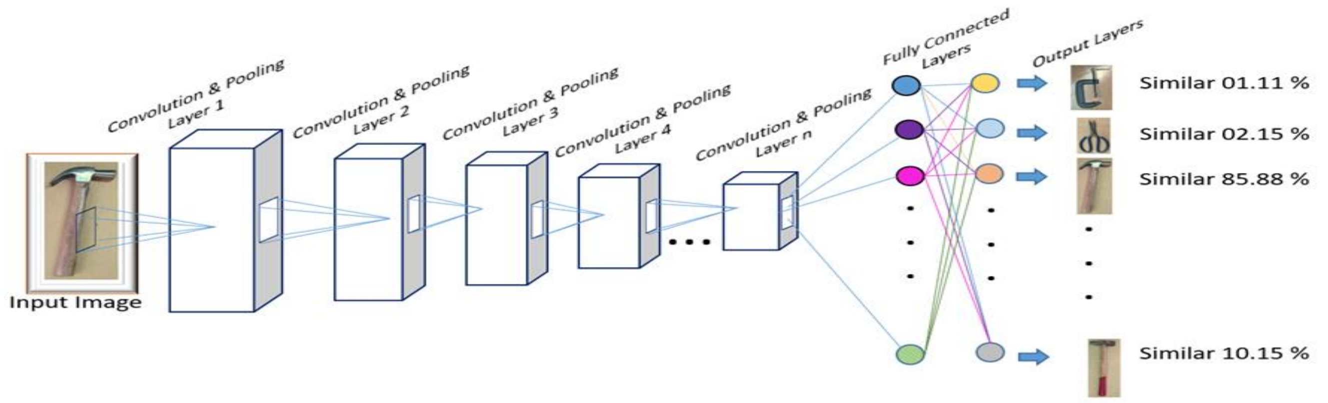


**Figure 2**: The convolutional neural network architecture

## 2. MATERIEL AND METHODS

The system was developed on the following computer hardware and software. The Intel(R) Core (™) i7-1195G7 CPU @ 2.90 GHz with 16 GB RAM was used as the central processing unit and Windows 11 was the operating system. MATLAB R2020b with license number 40598465 was the developing software. The digital cameras used in this research were Xiaomi, Redmi 8 to take all the video clips. To avoid the copyright problem, this research creates its own hand tools dataset, which is called "HTAIRS dataset".

### 2.1 Physical Diagram

The HTAIRS physical diagram starts with a user capturing an hand tool image by using a simple mobile phone camera. Then the image is submitted to the computer system for recognizing and the HTAIRS retrieves the video clip by matching an image with all video frames in a training-dataset. Finally, the system demonstrates the video to present how to operate the recognition hand tools, as shown in Figure 3.

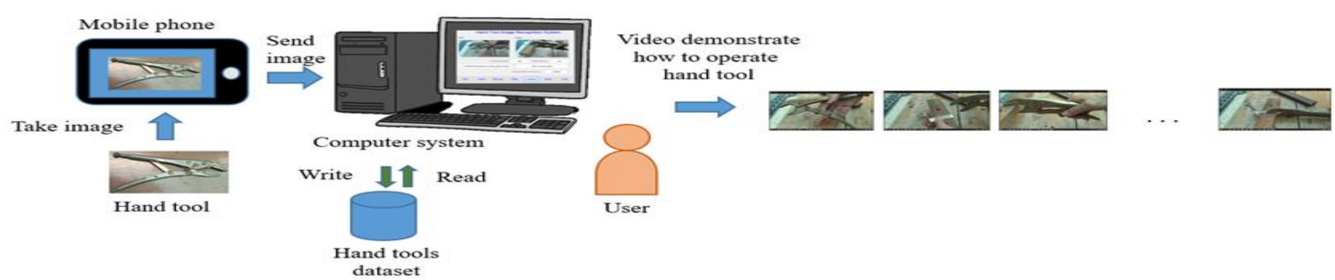


**Figure 3**: The HTAIRS physical diagram

### 2.2. System Structure Chart

The HTAIRS structure chart consists of four main modules, namely: 1) hand tools and accessories dataset creation module, 2) image acquisition module, 3) image recognition module, and 4) result presentation module, as shown in Figure 4. The hand tools and accessories dataset creation module consists of three sub-modules, which are extraction video frames, training dataset and testing dataset sub-modules. The image acquisition module used to receive a query image. The hand tools recognition module used to match hand tool image with hand tools video files in the dataset. Finally, the result presentation module contains three sub-modules, which are a playing video, a pausing video and a stopping video sub-module. Each HTAIRS module has the following details.

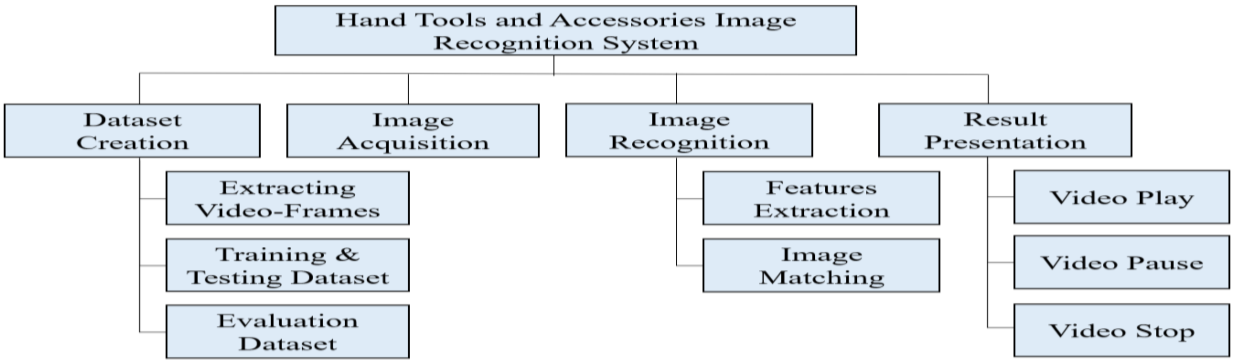


**Figure 4:** The HTAIRS structure chart

### 2.2.1 Dataset Creation Module

Regarding the HTAIRS research, the HTAIRS employs ResNet50 to train and test dataset. The ResNet50 is famous and commonly used to implement image recognition without human supervision [13]. The HTAIRS recorded 165 video clips to present how to operate 165 hand tools. The video clip was taken in the full HD system, with the resolution of 1920 * 1080 * 3 (pixel-width * pixel-height * plane) in the .MP4 file format. Normally, the video clip contains 32 frames/second. There are 1,920 video frames if the system captures every frame in a one-minute (32 * 60 = 1,920).

#### 2.2.1.1 Video Frame Extraction Sub-Module

The HTAIRS employed 165 video clips, with 600 frames each. Totally, the HTAIRS conducted the experiment with 99,000 video frames, which are saved in .JPG file format. In 165 tool categories, the HTAIRS randomly selected 500 odd frame number for a training dataset and also randomly selected 100 even frame number for an un-training dataset (evaluating dataset).

#### 2.2.1.2 Training and Testing Sub-Module

For the 82,500 video frames (165 * 500), the system separated them into two parts, which were a training part and a testing part. For the training part, 80% of the video frames (82,500 * 0.8 = 66,000 frames) were randomly selected while the remaining 20% (82,500 * 0.2 = 16,500frames) were for the testing part.

#### 2.2.1.3. Evaluating sub module

The HTAIRS employed 16,000 video frames (165 * 100) for evaluating the ResNet50. All the evaluation video frames are stored in the Excel file. The HTAIRS employed both MATLAB and Excel software to evaluate the ResNet50 performance.

### 2.2.2. Image Acquisition Module

There are two ways to retrieve a video clip by using a single image, which are directly taking an image from user mobile phones and selecting a video frame from the dataset. The system takes pictures with mobile phone cameras, with the full HD image size of 1920 * 1080 * 3 (pixel * pixel *plane) in three planes. The ResNet50 resizes the full HD image to the ResNet50 suitable size, which is 224 *224 * 3 pixels. The HTAIRS uses ResNet50 image-size to retrieve the most similar video clips for the answer.

### 2.2.3 Recognition Module

Based-on the system structure chart in Figure 4, the hand tools recognition module consists of two sub-modules, which are feature extraction sub-module and matching sub-module. Each sub-module has the following details.

#### 2.2.3.1 Feature Extraction Sub-Module

The HTAIRS employs the ResNet50 to recognize hand tools image. The structure of the ResNet50 in the HTAIRS contains three components, namely, 1) feature extraction component, 2) classification connection component and 3) output component, as shown in Figure 5. The HTAIRS starts with receiving the input image from users. After that, it sends an input image to the feature extraction component. The feature extraction component consists of two sub-components, which are convolutional layer and pooling layer sub-components. The convolutional layer extracts all of the important features from an image, while the pooling layer reduces the dimension from the convolutional layer. The ResNet50 repeats the convolutional and pooling layers for 50 loops to extract all features from the input image.

#### 2.2.3.2 Matching Sub-Module

The classification component consists of a full connection neural-network. The classification component matches the most similar between an input image and video frames from the dataset and then sends the matching result to the output component. Finally, the output component displays the retrieval video clip to the user.

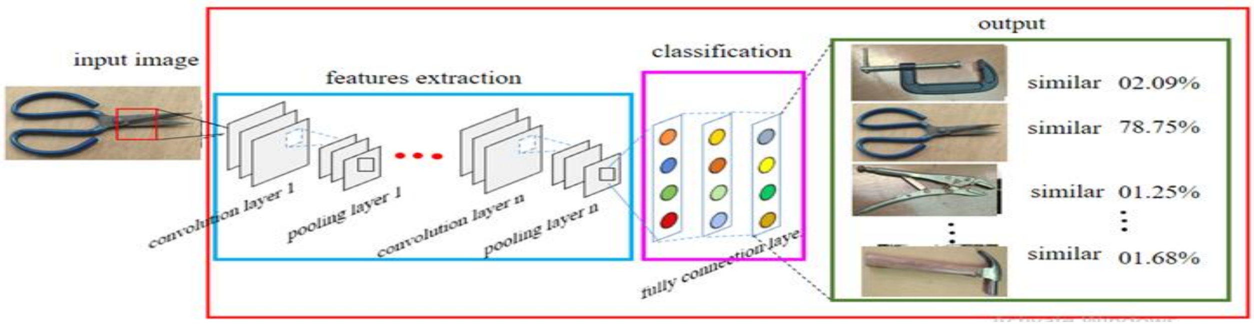


**Figure 5**: The architecture of ResNet50 in this research


**Figure 6**: The HTAIRS graphic user interface

### 2.2.4 Result presentation module

The HTAIRS shows query-results via the graphic user interface (GUI), which is composed of two display-graphic windows, five display-text boxes and seven push buttons, as shown in Figure 5. Each component in the GUI has the following details.

The two display-graphic windows have the following details:

1. In the label 1 of Figure 6, the display-graphic window for showing the input image.
2. In the label 2 of Figure 6, the display-graphic window for showing the retrieval video clip.
3. The five display-text boxes have the following details:
4. In the label 3 of Figure 6, the display of the frame number
5. In the label 4 of Figure 6, the display of total frame number of video clips.
6. In the label 5 of Figure 6, the display of the input path and filename box.
7. In the label 6 of Figure 6, the display of the retrieval video filename box.
8. The seven push buttons have the following details:
9. In the label 8 of Figure 6, the get image button for getting the input image.
10. In the label 9 of Figure 6, the clear button for clearing all the HTAIRS values.
11. In the label 10 of Figure 6, the recognize button for matching an input image with all video frames in the dataset.
12. In the label 11 of Figure 6, the play button for playing a video clip from the current frame.
13. In the label 12 of Figure 6, the pause button for pausing a video clip from the current frame.
14. In the label 13 of Figure 6, the stop button for stopping a video clip.
15. In the label 14 of Figure 6, the exit button for exiting the system.

## 3. RESULTS AND DISCUSSIONS

### 3.1. Experimental Results

The HTAIRS employed 82,500 images of 165 hand toos and accessories categories to train the dataset and used the same hand toos and accessories categories of 16,500 images for evaluating of the ResNet50. The training time for HTAIRS dataset was 5,237.4 seconds or 1 hour 27 minutes and 17.4 seconds. The confusion matrix for training the ResNet50 had the true positive (TP) of 81,942 false positive (FP) of 558 false negative (FN) of 558 and true negative (TN) of 13,530,000, as shown in Table 1. Moreover, the confusion matrix for cross-validation of the ResNet50 had the true positive (TP) of 16,385 false positive (FP) of 115, false negative (FN) of 115 and true negative (TN) of 2,706,000, as shown in Table 2. The average access time to recognize an image is 1.57 seconds / image.

**Table 1: The confusion matrix for training the ResNet50**

| | | Actual class | |
|---|---|---|---|
| | | Positive | Negative |
| Prediction Class | Positive | 81,942 (TP) | 558 (FP) |
| | Negative | 558 (FN) | 13,530,000 (TN) |

Remarks: TP = true positive; FP = false positive; FN = false negative; TN = true negative

**Table 2**: The confusion matrix for ResNet50 evaluation

| | | Actual class | |
|---|---|---|---|
| | | Positive | Negative |
| Prediction Class | Positive | 16,385 (TP) | 115 (FP) |
| | Negative | 115 (FN) | 2,706,000 (TN) |

Remarks: TP = true positive; FP = false positive; FN = false negative; TN = true negative

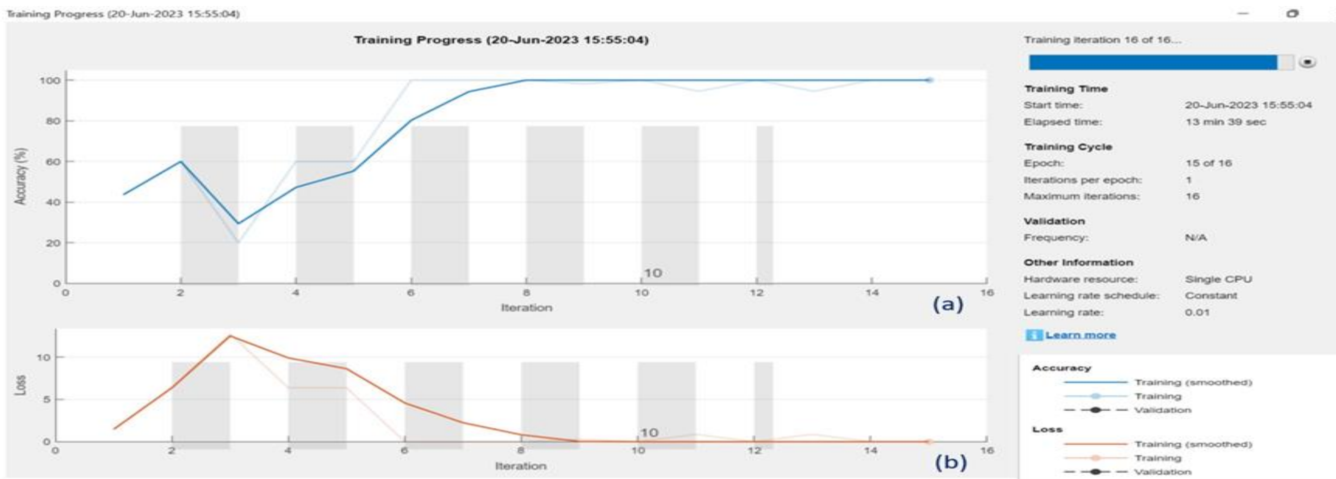


**Figure 7**: (a) The accuracy graph and (b) loss graph for training ResNet50 in the HTAIRS

The accuracy graph and loss graph for training the ResNet50 are shown in Fig. 7 (a) and 7 (b), respectively. The setting parameters for training the ResNet50 have an epoch per iteration, 0.01 learning rate and 16 maximum epochs. Based on the experiment results, the correct recognition result and incorrect recognition result are illustrated on Fig. 8 (a) and 8 (b), respectively.

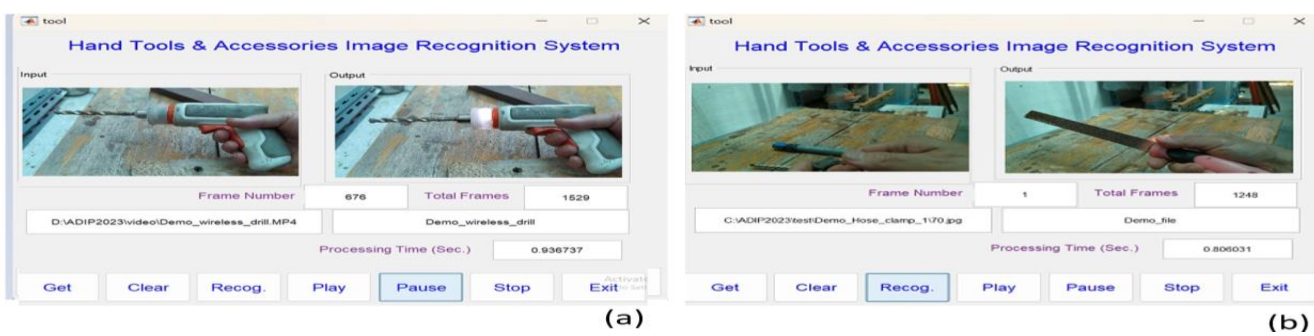


**Figure 8**: The GUI samples of (a) correct recognition result and (b) incorrect recognition result

### 3.2. Discussion

This section discusses two topics about the comparison of performance among the various convolutional neural networks and comparison of performance among previous researches. Each topic has the following details.

#### 3.2.1 Comparison of Performance Between Convolutional Neural Networks

The HTAIRS employed MATLAB multi-classes confusion-matrix function called “confusion.getMatrix” to calculate the six statistical values for measuring the HTAIRS performance, which were accuracy, precision, recall, F1-score, All statistical values were calculated from the confusion matrix, which had true positive (TP), false positive (FP), true negative (TN) and false negative (FN). Each statistical value has the following details [14].

#### 3.2.1.1 Accuracy Statistical Value

An accuracy value is one of the most common measurements of matching performance and it is defined as the ratio between the correct number of images matching and the total number of images, as shown in Equation 3.1.

Accuracy=((TP+TN))/((TP+FP+FN+TN) ) (3.1)

#### 3.2.1.2 Precision Statistical Value

A precision value is an ability of an image to identify only the recognize image within a dataset. The precision value is found by using the number of true positives divided by the number of true positives plus the number of false positives, as shown in Equation 3.2.

Precision=TP/((TP+FP) ) (3.2)

#### 3.2.1.3 Recall Statistical Value

A recall value is an ability of an image to find all the recognition images within a dataset. The recall value is defined as the number of true positives divided by the number of true positives plus the number of false negatives, as shown in Equation 3.3.

Recall=TP/((TP+FN) ) (3.3)

#### 3.2.1.4 F1-Score Statistical Value

The F1-score considers both the precision and the recall of the test to compute the score. The F1 score can be interpreted as a weighted average of the precision and recall, as shown in Equation 3.4.

F1-score=(2*Precision*Recall)/((Precision + Recall) ) (3.4)

This research was conducted on seven different convolution al neural networks types, namely: Alex Net, Google Net, InceptionV3, ResNet18, ResNet50, ResNet101 and VGG16, all with the same hand tool images dataset. These are seven different convolution al neural networks types were supported by the MATLAB 2020 version. The comparisons of accuracy, precision, recall, F1-score and training times are shown in Table 3. The VGG16 consumes the shortest time while the ResNet101 consumes the longest time for training the dataset. The ResNet101 gives the lowest statistical measurements and the ResNet50 gives the highest statistical measurements, namely: accuracy, precision, recall and F1-score. Therefore, this research employed the ResNet50 for the HTAIRS.

**Table 3**: Performance and training time comparison

| CNN model | Accuracy | Precision | Recall | F1-Score | Training Time (Sec.) |
|---|---|---|---|---|---|
| Alex Net | 0.7919 | 0.8144 | 0.7919 | 0.8029 | 1,468.90 |
| Google Net | 0.9928 | 0.9925 | 0.9928 | 0.9926 | 5,676.20 |
| Inception V.3 | 0.9930 | 0.9918 | 0.9930 | 0.9924 | 7,694.70 |
| ResNet18 | 0.9918 | 0.9916 | 0.9918 | 0.9916 | 3,372.00 |
| ResNet50 | 0.9930 | 0.9930 | 0.9930 | 0.9930 | 5,237.40 |
| ResNet101 | 0.6605 | 0.6605 | 0.6605 | 0.6606 | 8,279.70 |
| VGG 16 | 0.7250 | 0.7250 | 0.7250 | 0.7250 | 1,353.70 |

### 3.3. Comparison Performance Among Previous Researches

There are many researchers conducted many experiment on hand tool image recognition by many methods, as shown in Table 4. All previous researches employed different dataset size and hand tool categories with different techniques. The state-of-art for the image-based computer peripheral and accessory recognition researches have the accuracy between 80.00 to 99.30%.

**Table 4**: Performance comparison among previous researches

| Author | Method | Tool types | Number of Images | Precision % |
|---|---|---|---|---|
| Kurnaz [2] | Faster RetinaNet | 49 | 2,788 | 85.00 |
| Oka [4] | Yolo V7 | 24 | 1,425 | 80.80 |
| Manici [5] | Alex Net | 9 | 4,500 | 60.00 |
| Stasiak [9] | KNN | 10 | 150 | 85.00 |
| Hernandez [11] | Yolo | 24 | 8,000 | 98.00 |
| This Research | ResNet50 | 110 | 99,000 | 99.30 |

## CONCLUSIONS

The HTAIRS fulfills the objective of this research, which is to develop a computer system to recognize computer peripheral and accessory image. The HTAIRS dataset consists of 20,800 images, which are 104 computer peripheral and accessory categories and each category has 110 images. The accuracy of the system are 99.30% for training dataset and 99.30% of evaluation. The average access time of the system are 0.8549 seconds per image. The HTAIRS also compared the performance with various kind of CNN and the experiment results showed the state-of-art of this research.

## REFERENCES


[1] Jain S. What are the types of hand tools and how are they important? [Internet]. 2019. [cited 2023 May 31]. Available from https://www.moglix.com/blog/what-are-the-types-of-hand-tools-and-how-are-they-important/

[2] Kurnazl FC, Hocaogl B, Yılmaz1 MK, Sulo I, Kalkan S. ALET (automated labeling of equipment and tools): A dataset for tool detection and human worker safety detection. In: Bartoli, A., Fusiello, A. (eds) Computer Vision – ECCV 2020 Workshops. ECCV 2020. Lecture Notes in Computer Science, vol 12538. Springer, Cham. pp. 371–386.

[3] Yang X, Yu Y, Shirowahan S, Sepasgozar S, Li H. Automated PPE-tool pair check system for construction safety using smart IoT. Journal of Building Engineering. 2020; 32: 1-12.

[4] Oka S, Nozaki K, Hayashi, M. An efficient annotation method for image recognition of dental instruments. Science reports, 2023; 13 (169): 1-10.

[5] Mancini M, Karaoguz H, Ricci E, Jensfelt P, Caputo1 B. Kitting in the wild through online domain adaptation. Proceedings of the 2018 IEEE/RSJ International conference on intelligent robots and systems (IROS), 2018 Oct. 1-5. Madrid, Spain. IEEE explore; pp. 1103-1109.

[6] Pavlasek J, Lewis S, Desingh K, Jenkins OC. Parts-based articulated object localization in clutter using belief propagation. Proceedings of the 2020 IEEE/RSJ International conference on intelligent robots and systems (IROS), 2020 October 24 – 2021 January 2021, Las Vegas, USA. IEEE explore; pp. 1-8.

[7] Qin Z, Fang K, Zhu Y, Fei-Fei L, SavareseS. KETO: Learning keypoint representations for tool manipulation. Proceedings of the 2020 IEEE International conference on robotics and automation (ICRA), 2020 May 31 – 2020 August 31, Paris, France. IEEE explore; pp. 1-8.

[8] Nakamura T, Nagai, T. Object concept modeling based on the relationship among appearance, usage and functions. Proceedings of the 2020 IEEE/RSJ International conference on intelligent robots and systems (IROS), 2010 October 18 - 22, Taipei, Taiwan. IEEE explore; pp. 5410-5415.

[9] Stasiak B. FONN-based affine-invariant image recognition. In: Choraś, R.S. (eds) Image Processing and Communications Challenges 2. Advances in Intelligent and Soft Computing, vol 84. Springer, Berlin, Heidelberg. 1-8.

[10] Zhu Y, Zhao Y, Zhu S.C. Understanding tools: task-oriented object modeling, learning and recognition. Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015 June 07 - 12, Boston, MA, USA. IEEE explore; pp. 2855 - 2864.

[11] Zamora-Hernándeza M, Castro-Vargas J, Azorin-Lopez J, Garcia-Rodriguez J. Deep learning-based visual control assistant for assembly in Industry 4.0. Computers in Industry. 2021; 131: 1-15.

[12] Shilkrot R, Narasimhaswamy S, Vazir s, Hoai M. WorkingHands: a hand-tool assembly dataset for image segmentation and activity mining. Proceedings of the British Machine Vision Conference, 2019 September 9-12, Cardiff, UK. Cardiff University; pp.1-12.

[13] Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma, O, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of big data. 2021: 8(53): 1-74.

[14] Amalia L, Alejandro C, Alejandro M, Ana H. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition. 2019: 91: 216-231


DOI: https://doi.org/10.15379/ijmst.v10i2.3193