Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 9, August2021 :7002 - 7011

Research Article

Complete Solution to Fight Cyberattacks by feeding an Assembly Line by Fake Items

A.Aziz Naanani[a]*, Ismail Lagrat[b], Noureddine Masaif[c]

[a,c] Laboratory of Electronic Systems, Information Processing, Mechanics and Energy, Ibn Tofail University, Kenitra, Morocco
[b] Laboratory of Advanced Systems Engineering, National Schools of Applied Sciences, Ibn Tofail University, Kenitra, Morocco
Email: [a] aziz.naanani@gmail.comk

Abstract

This paper deals with the problem of the possible attack of an assembly line by an intruder with the aim of blocking or deteriorating the quality of production by sending false articles to the conveyors. The solution consists in setting up a monitoring system that visually detects fake items in real time and identifies them by means of artificial intelligence. Bait files are accessible to access the attacker's workstation and send essential information to the competent authorities fighting against cyber attacks.

Keywords: *cyberattacks, artificial intelligence, assembly line, opencv, python*

Introduction

Cyberattacks have become one of the mostly discussed topics around the globe due to the importance they have in our daily life. The online revolution of today’s connected world, have lead many researchers to think about ways to fight cyberthreats which can cause severe damage in all sectors. The increasing need of cyber security has motivated this work that revolves on dealing with the potential attacks of an assembly line via means of artificial intelligence. Additionally, this study is taken to another level by predicting possible attacks, rising this way the safety and security of the investigated assembly lines.

In 1913, Henry Ford proposed sequential manufacturing which consists of moving the parts to be assembled in front of workers to complete them and pass them to their successors, and so on until the final exit.

The advantage of this technique is to constantly improve the competence of the workers which saves time and performance.

Given this, assembly lines for mass production and sometimes small quantities quickly have become the essential element for industry in general and Industry 4.0 in this case.

Assembly lines have different shapes but in general it is a conveyor author of which specialized human or machine skills that make a single routine task during the period of work.

The major handicap of this technique is its great vulnerability because an error in a node penalizes the entire structure such as a chain where a link has dropped.

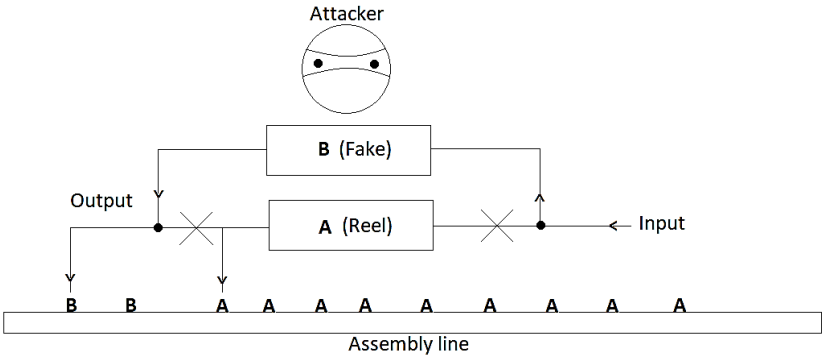
Because of its importance and extreme vulnerability, the purpose of this article is to present a complete solution which consists in:

- Real-time detection of fake items.
- Identification of the attacker.
- Prediction of possible attacks.

Different Scenarios and constraints can be taken into account, namely:

- An assembly line to which is sent by an act of vandalism instructions to feedwith false articles.

Figure.1 Assembly line

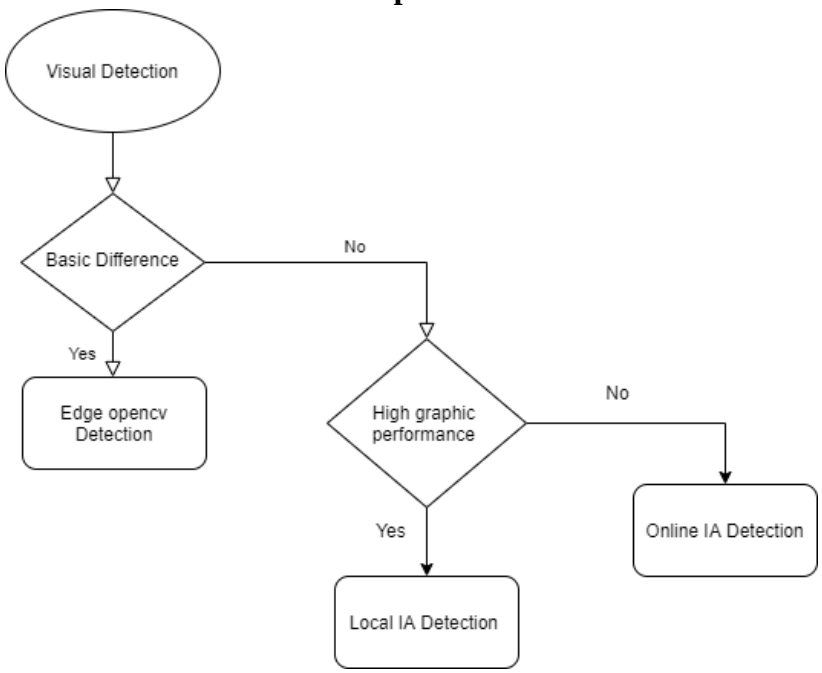


- ❖ The detection of the articles is done visually by camera and image processing.
- ❖ It is assumed that there are 2 types of discrepancy between genuine and fake articles. Very clear differences (of shape) and less clear deviations (of colors, shapes...)
- ❖ The attacker is supposed to be hiding behind a victim or has changed IP address.
- ❖ Through the type of attack and machine learning, we predict possible attacks.

Fake Item Detection Phase

In this part, we aim at detecting non-conforming items sent to the production line to avoid stopping the production as well as time and material losses due to corrective maintenance.
In this section, the detection is done visually, leading us to three cases:

Figure.2 Flowchart of detection techniques according to the degree of difficulty and technical possibilities.



2. 1. Small difference in article's form

In this subsection, it is assumed that the article's form is different, in this case:

We will use Opencv library and canny module which are not only widely used for automatic detection of particle size distribution [9], fruit detection [10], steganography in gray image processing [8], but also for color detection sometimes [17].
In this subsection, we will only use the already set tool; however, for deeper knowledge, we can use some improved algorithms [12],[18].

- ❖ We convert the Storytelling Image (Edge);
- ❖ We count the black and white pixels which do not change even if the object changes its orientation.

Principle:

- ❖ In the Python environment, the NUMPY - CV2 modules and IMUMUSS
- ❖ The angle of inclination is increased by 15 degrees by the CV2 and imutil tools, for instance.
- ❖ We count the black and white pixels and we make the report.

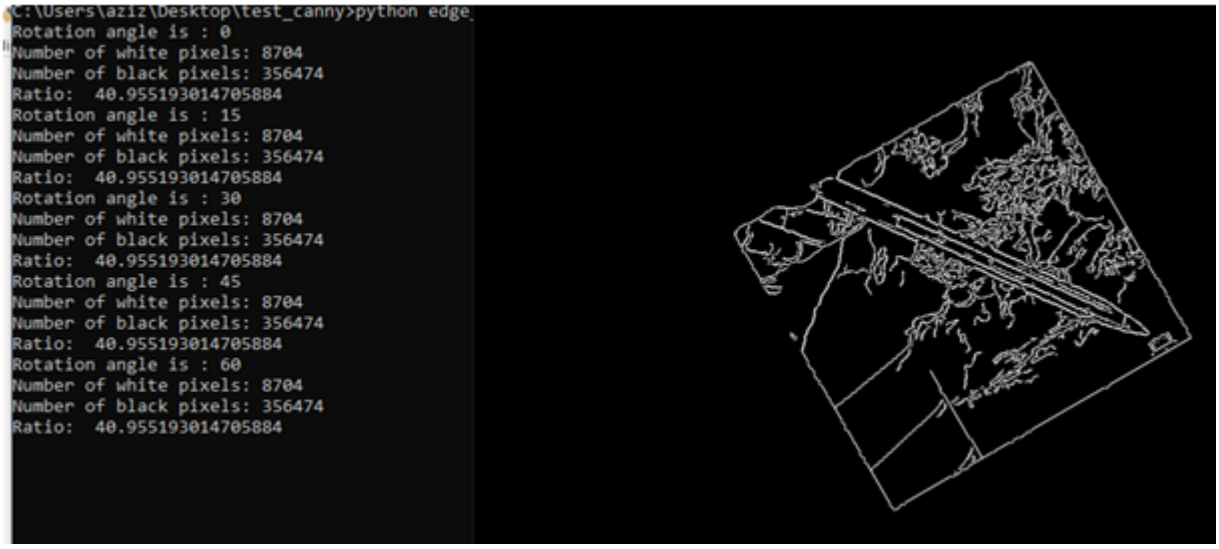
Results:

For an inclination of 15 deg, the pixel ratio is 40.955:

Figure.3 Black-white pixel ratio for a 15 degree inclination.

For an inclination of 60 deg, the pixel ratio is 40.955:

Figure.4 Black-white pixel ratio for a 60 degree inclination.



Conclusion:

Although the item changes its position in the chain, its identification is still precise, i.e., identifying the item takes place even with variation of its position!

2.2.Big difference in shape and color:

In case we have powerful machine resources, especially graphics, and it is none of our purposes to put the manufacturing process online, it is possible to do a processing by machine learning locally. To this end, we must use as development tools: Python with scikit-learn framework.

We will use real time object detection using machine learning [13][14], as well as CNN since it is highly used in AI for "Learned From Different Languages"[3], "Person-Related Categories in Photos"[6], "Cellular Neural Network" [2], "Handwritten character recognition"[5], "Eye Feature Detector" [5], "Electroencephalographic (EEG)" [7], "Image Character Recognition"[3] or "Forwarding error correcting encoder/decoders"[1].

For the process:

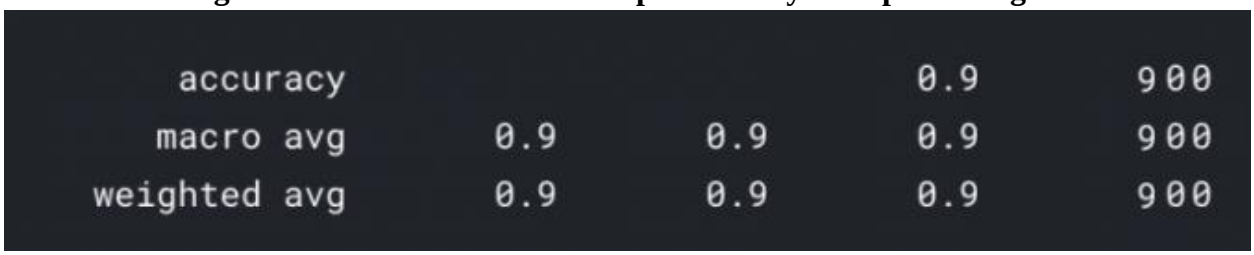
DataSet Building (Training and Test):We look for the data in the online dataset (ImageNet, MS COCO, COIL100...) otherwise python is used to extract images of a video.

For data processing:

- An estimator and its parameters are selected.
- To train the model: model.fit()
- To evaluate the model: model.score()
- To use the model: model.predict()

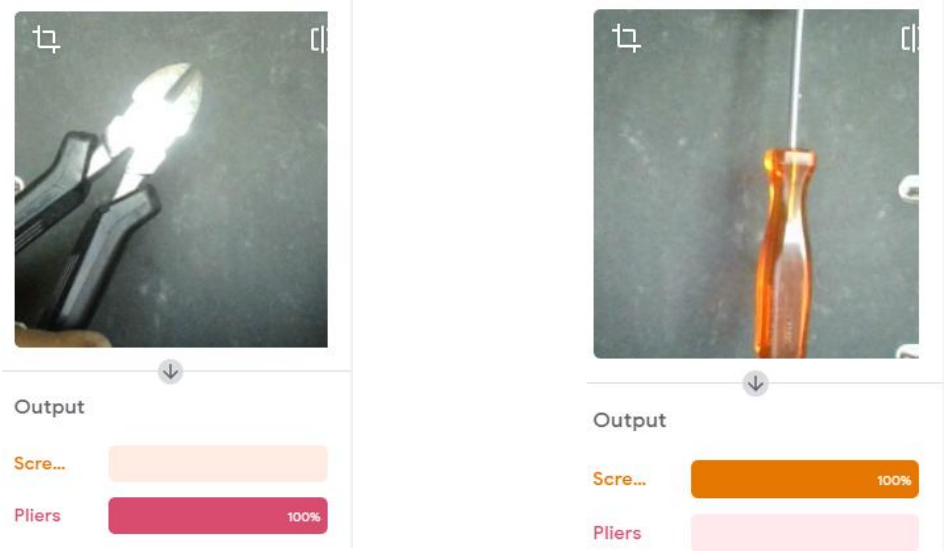
We have had a very good precision:

Figure.5 Results of the obtained precision by local processing



Note: This approach is precise but it is very demanding in terms of resources, especially graphics.In case of unavailable resources, one can solve the problem using Google Soft and Hardware resources paying the price of confidentiality.We used the tool: Google Teachablemachine.We do not need to install anything![6]
The camera is activated to make the classes and the training totally without having to code anything.Here are the results: to detect a plier and a screwdriver:

Figure.6 Results of the precision obtained by processing using the Google Teachablemachine tool



Conclusion:

For the dispensing of fake articles:

- If the cart is at the shape level, we can settle for the OpenCV tool;
- If the resources allow, it can be done by visual detection and the CNN;
- If resources are low (especially in graphic processing), the ML lines in line can be used.

Identification of the Attacker

The detection of false articles costs investment in time and material as well as in degradation or stop of the production. Consequently, it is wiser to solve the problem from its origin, hence the need to anticipate the attack by detecting the attacker in order to stop his acts. This objective is our major target in this part.
The attacker does not bother to change his IP address or the attacker is hidden behind a fake IP address. For the 1st case we use IP to geolocation converters (ex: geoip2), but we generally assume that the attacker is naturally disguised behind a fake IP address. The goal is to reach the attacker's machine and receive his identity.
Note: This operation must be done through the appropriate legal powers.
Assuming that the attacker is targeting images, here is the strategy to follow so as to reach our objective:

In the desired images we merge an informative program on identity and information essential for the skills of the responsible state on the security of goods and people.

Action Plan:

- Register in the registry to get started every time you start.
- Exercise keylogging.
- Send the information by email.

3.1. Registration in registry for launch at startup

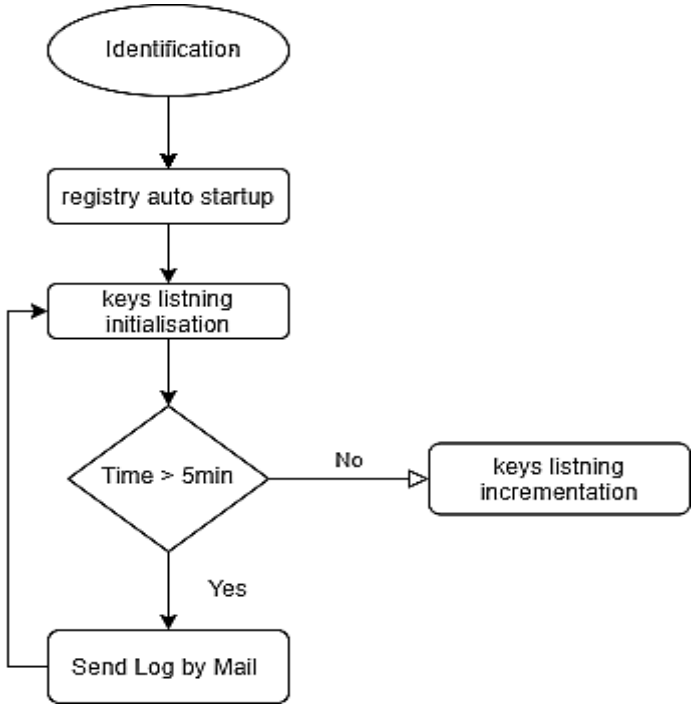
For a permanent and automatic tracking of the attacker without the event being triggered by the attacker, which can arouse his vigilance and which does not guarantee the continuity of the service, the tool (computer script) of identity detection must be launched spontaneously at each startup. For this to happen, it must be registered in the registry. The case of the tool Windows is discussed in this paper, but it is valid for any operating system.

- We start first by adding the path of the file in the value:« Software\Microsoft\Windows\CurrentVersion\Run »
- of the key :« HKEY_CURRENT_USER »
- Assuming that the file is c:\img.py

Code python :

```
importwinreg as reg
script = “c:\img.py”
key = HKEY_CURRENT_USER
key_value = "Software\Microsoft\Windows\CurrentVersion\Run"
open = reg.OpenKey(key,key_value,0,reg.KEY_ALL_ACCESS)
reg.SetValueEx(open,"script",0,reg.REG_SZ,script)
reg.CloseKey(open)
```

Figure.7 Identification flowchart



3.2.Recording events on machine

All the information of the attacker are through the keyboard and the mouse (passwords, local applications, searches on the net ...), the tool makes, in part, the recording of all the events (typed keys of the keyboard and clicks of mouse...) and saves them in a local text file. This technique is highly used to determine the master machine [11].
To hide the identity of the file, you can change its conventional extension (*.txt,*.doc...)
As mentioned on the graph above, we save during 5 min so that the text file is not very heavy and easily detectable and difficult to send by mail.
We use the module pynput.keyboard and threading in order to record the keystrokes and time in parallel by the threading module and its Timer() function.
The main function of recording is pynput.keyboard.Listener(on_press=process_key_press)
The main threading function is threading.Timer(seconds, report)
We increment the keyboard variables for 5 minutes.
We send by mail the incremented variable and we initialize it again and the cycle is restarted again.

3.3.Send the information by mail

The text file where the logs are recorded is sent as an attachment to the destination for processing, using the gmail smtp server and stmp protocol, which are used to send and receive plaintext messages, or php web-based e-learning [15]-[16]. In our case:
We use the smtplip and ssl module and its parameters:
port
context = ssl.create_default_context()
smtp server : login and password.
And the method " send " with its arguments : msg,from_addr,to_addrs

Conclusion

This paper allows large-scale detection of attacks in real time as well as tracing the attacker.Using Intel's Opencv tool, online and local artificial intelligence tools for image detection, we could identify items.Using device management APIs and the registry to identify the attacker through a pathfinder tool.In future works, studies continue to predict the origin of attacks based on historical data patterns.

References

[1] Alston, M. D., & Chau, P. M. (1990). A neural network architecture for the decoding of long constraint length convolutional codes. *1990 IJCNN International Joint Conference on Neural Networks*, 121-126 vol.1. https://doi.org/10.1109/IJCNN.1990.137556

[2] Babatunde, H., Folorunso, O., & Akinwale, A. (2010). *A Cellular Neural Network-Based Model for Edge Detection*. *5*(1), 8.

[3] Bai, J., Chen, Z., Feng, B., & Xu, B. (2014). Image character recognition using deep convolutional neural network learned from different languages. *2014 IEEE International Conference on Image Processing (ICIP)*, 2560-2564.

https://doi.org/10.1109/ICIP.2014.7025518

[4] Caihua Liu, Jie Liu, Fang Yu, Yalou Huang, & Jimeng Chen. (2013). Handwritten character recognition with sequential convolutional neural network. *2013 International Conference on Machine Learning and Cybernetics*, 291-296. https://doi.org/10.1109/ICMLC.2013.6890483

[5] Cao, L., Yan, Z., & Smith, J. R. (2015). Multi-facet Learning using Deep Convolutional Neural Network for Person-Related Categories in Photos. *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 567-570. https://doi.org/10.1145/2671188.2749356

[6] Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J., Jongejan, J., Pitaru, A., & Chen, A. (2020). Teachable Machine : Approachable Web-Based Tool for Exploring Machine Learning Classification. *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 1-8. https://doi.org/10.1145/3334480.3382839

[7] Cecotti, H., & Graeser, A. (2008). Convolutional Neural Network with embedded Fourier Transform for EEG classification. *2008 19th International Conference on Pattern Recognition*, 1-4. https://doi.org/10.1109/ICPR.2008.4761638

[8] Gaurav, K., & Ghanekar, U. (2018). Image steganography based on Canny edge detection, dilation operator and hybrid coding. *Journal of Information Security and Applications*, *41*, 41-51. https://doi.org/10.1016/j.jisa.2018.05.001

[9] Meng, Y., Zhang, Z., Yin, H., & Ma, T. (2018). Automatic detection of particle size distribution by image analysis based on local adaptive canny edge detection and modified circular Hough transform. *Micron*, *106*, 34-41. https://doi.org/10.1016/j.micron.2017.12.002

[10] Monir Rabby, M. K., Chowdhury, B., & Kim, J. H. (2018). A Modified Canny Edge Detection Algorithm for Fruit Detection & Classification. *2018 10th International Conference on Electrical and Computer Engineering (ICECE)*, 237-240. https://doi.org/10.1109/ICECE.2018.8636811

[11] Nyang, D., Mohaisen, A., & Kang, J. (2014). Keylogging-Resistant Visual Authentication Protocols. *IEEE Transactions on Mobile Computing*, *13*(11), 2566-2579. https://doi.org/10.1109/TMC.2014.2307331

[12] Rong, W., Li, Z., Zhang, W., & Sun, L. (s. d.). *An Improved Canny Edge Detection Algorithm*. 6.

[13] Saxena, M. R., Pathak, A., Singh, A. P., & Shukla, I. (2019a). REAL TIME OBJECT DETECTION USING MACHINE LEARINING AND OPENCV. . . *ISSN*, 4.

[14] Shitole, H. P., & Divekar, S. Y. (2019). *Secure Email Software using e-SMTP*. *06*(03), 6.

[15] Sudana, I. M., Qudus, N., & Prasetyo, S. E. (2019). Implementation of PHPMailer with SMTP protocol in the development of web-based e-learning prototype. *Journal of Physics: Conference Series*, *1321*, 032027. https://doi.org/10.1088/1742-6596/1321/3/032027

[16] Xin, G., Ke, C., & Xiaoguang, H. (2012). An improved Canny edge detection algorithm for color image. *IEEE 10th International Conference on Industrial Informatics*, 113-117. https://doi.org/10.1109/INDIN.2012.6301061

[17] Xu, Z., Baojie, X., & Guoxin, W. (2017). Canny edge detection based on Open CV. *2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, 53-56. https://doi.org/10.1109/ICEMI.2017.8265710.