Enabling Real-Time Quality Inspection in Smart Manufacturing Through Wearable Smart Devices and Deep Learning

Sarivan, Ioan-Matei; Greiner, Johannes; Díez Alvarez, Daniel; Euteneuer, Felix; Reichenbach, Matthias; Madsen, Ole; Bøgh, Simon

Published in:
Procedia Manufacturing

DOI (link to publication from Publisher):
10.1016/j.promfg.2020.10.053

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 51 (2020) 373–380

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021)
15-18 June 2021, Athens, Greece.

Enabling Real-Time Quality Inspection in Smart Manufacturing Through Wearable Smart Devices and Deep Learning

I.-M. Sarivan[a,b,*], Johannes N. Greiner[a], D. Díez Álvarez[a], F. Euteneuer[a], M. Reichenbach[a], O. Madsen[b], S. Bøgh[b]

[a]*Mercedes-Benz AG, HPC F150, 71059 Sindelfingen, Germany*
[b]*Robotics & Automation Group, Aalborg University, Fibigerstræde 16, 9220 Aalborg Øst, Denmark*

Abstract

In this paper, we present a novel method for utilising wearable devices with *Convolutional Neural Networks* (CNN) trained on acoustic and accelerometer signals in smart manufacturing environments in order to provide real-time quality inspection during manual operations. We show through our framework how recorded or streamed sound and accelerometer data gathered from a wrist-attached device can classify certain user actions as successful or unsuccessful. The classification is designed with a Deep CNN model trained on Mel-frequency Cepstral Coefficients (MFCC) from the acoustic input signals. The wearable device provides feedback on three different modalities: audio, visual and haptic; thus ensuring the worker's awareness at all time. We validate our findings through deployments of the complete AI-enabled device in production facilities of Mercedes-Benz AG. From the conducted experiments it is concluded that the use of acoustic and accelerometer data is valuable to train a classifier with the purpose of action examination during industrial assembly operations, and provides an intuitive interface for ensuring continued and improved quality inspection.

Keywords: Smart Wearable Devices; Deep Learning; CNN; MFCC; Sound Classification; Smart Manufacturing; Quality Inspection.

1. Introduction

Quality inspection is a very important process that deems a product ready to be dispatched to the customer or not. This is usually a post-production process that involves certain methods to test the functionality and the performance of the manufactured item ensuring it is functioning inside operating parameters. There are however situations, when a certain item is composed of multiple components assembled by manual assembly processes. With the currently available methods, the item can only be quality inspected at the end of the manufacturing process. If, by any reason, the assembly turns out not to meet the set quality standard, it needs to be disassembled, fixed and then reassembled, generating therefore extra effort for the company.

Current markets demand higher grades of variety and quality across products, thus pushing manufacturing companies to implement highly flexible production lines. The human worker however is still an essential part of a highly flexible manufacturing system and therefore, of product quality. To ensure the set quality level when it comes to manual assembling operations, two workers are normally required: one who performs the operation and one that checks for the mechanical integrity of the assembly. However, this is a time-consuming method especially when the worker performs assembly operations while located in difficult to access locations (like inside the car's body).

The hereby paper investigates a novel method of performing quality inspection on-spot, by making use of a wearable device (Huawei Watch 2, Wear OS 2). Having the smart watch on the wrist, the device is meant to assist the worker when performing assembly tasks by gathering acceleration and audio data that are processed using deep neural networks for classifying the actions as successful or unsuccessful. The classification result is then provided to the worker through the display, the speaker and the haptic feedback capabilities of the smartwatch. Depending on the classification result, the worker can take immediate ad-

* Corresponding author. Tel.: +45 50 15 21 47;
E-mail address: s.i.matei96@gmail.com (I.-M. Sarivan).

2351-9789
10.1016/j.promfg.2020.10.053

justment actions or send the product further on the production line. It is desired to obtain a system that generates zero false positives and that prompts the worker to check the assembled components if the action is classified as unsuccessful.

By adhering to Industry 4.0, the smartwatch system can be integrated within a Manufacturing Execution System (MES), if desired. The available input-output interfaces of the smartwatch make it possible for the worker to easily receive and send task-related data to a MES, without the need of abandoning the task at hand in order to operate an external interface. Providing this functionality to the worker is an effort of digitalizing manual manufacturing processes with the purpose of monitoring the status of the items on the production lines and assist the worker when or where necessary. The presented system is also meant to favour flexible manufacturing by involving human workers, but achieve a constant quality level similar to robotised processes.

The hereby presented project is based on a patent filed in October 2019 [7] and owned by Mercedes-Benz AG. The development of the project took place at ARENA2036, Stuttgart. Access to production lines was provided by Mercedes-Benz AG for investigating the current manufacturing processes and necessary data gathering for development and test of the solution.

2. Related work

By having in focus the ISO 9000 standard for quality management, two of the standard's seven principles are made relevant in the hereby paper: engagement of people and continuous improvement. It is desired to digitalize manual processes in the effort of improving flexibility by engaging human workers on the production line and achieving a high and constant level of quality across products. As A. De Carolis et al. [4] state in the conclusion of their paper about a digitalization maturity model, the investment in digital technologies starts at the level of strategic processes. One solution for digitalizing the manual processes can be by tracking the motion of the worker's arms. As B. Sugandi et al. [16] present in their paper a stereo camera for tracking arms and hand gestures, the technology is mature enough even to control an industrial robot just through hand movements.

In inaccessible locations, processing sound signals is an alternative to using a camera, when the part which is being worked on is occluded. As shown by G. Fedorko et al. [5], sounds can be tracked and mapped on a video feed in order to detect irregularities on heavy industrial assemblies. This system is designed to detect and accurately establish the source of sounds or vibrations using a microphone array. The array is backed up by a camera that provides visual output to the user. The sound sources are displayed on the fed images in order for the user to locate the sound or vibration sources.

In the extend of using acoustic signals for determining the completion of an action, other signal sources can be investigated. I. Suarez et al. [15] present a method for activity recognition using only the accelerometer of a smartphone, starting from the incentive that the best results for recognising an activity is through sensor fusion, but it affects the battery life of the device. Wearable devices are increasing in popularity, as B. Cvetković et al. [3] state in their paper about real-time monitoring with a wrist band and a smartphone. Smartwatches are equipped with the necessary sensors to capture both acceleration and acoustic signals. Using both signal sources and performing sensor fusion with them, E. Garcia-Geja et al. [6] are able to perform classification on seven daily activities like floor sweeping, eating etc.

Once the data is gathered, in a quality related application it is important to have a reliable method for classification. To perform real time quality inspection on riveting, R. Mueller et al. [10] are implementing artificial intelligence (AI) enabled by neural networks to classify the scanned part as good or bad. They further state that the AI classifier based on images reaches an accuracy of 97%. AI proves to be reliable in healthcare applications too. J. Rubin et al. [13] use Deep Convolutional Neural Networks (DCNNs) to classify phonocardiograms as normal, abnormal or uncertain. The acoustic signals gathered from the heart are transformed into images using Mel Frequency Cepstral Coefficients (MFCC) in order to make use of the latest breakthroughs in AI for image analysis. MFCC is considered the most successful method for sound processing when it comes to speech recognition when confronted with variations such as noise since their introduction in 1980 by Davis and Mermelstein [9]. The MFC coefficients extraction method is used by H. Seddik et al. for speaker recognition using neural network classifiers [14]. K. J. Piczak [11] is providing details about DCNN implementation and architecture in their paper about environmental sound classification. They conclude the method to be successful even with a low amount of data available for training.

Based on the presented related work, a system was designed for classifying manual assembly actions, therefore digitalizing a manual process. This was done with the purpose of raising the worker's confidence in their actions and providing them with a reliable on-spot quality inspection tool. The paper presents implementation details of this solution and is structured as follows: section 3 provides an insight into the wearable-enabled quality inspection framework. Implementation details are given regarding the WearOS app developed to gather acoustic and acceleration data from a smartwatch while the worker performs the task. Data preprocessing is made using MFCC and action classification using DCNNs. Section 5 focuses on the testing of the system and other reliability related matters. Further on, section 6 brings a discussion based on the presented results leading to section 7 containing information about planned future work.

3. QCApp framework

To obtain a system that can be used by the worker without any movement or workspace restrictions and provide on-spot quality inspection, a Huawei Watch 2 was chosen for gathering sound and acceleration data while the worker is performing certain actions. The smartwatch is connected to a Wi-Fi network and communicates with a computer that runs AI classification algorithms. The smartwatch sends the recorded data to the com-

puter. After the computer finishes processing the data, it returns the result to the smartwatch which provides either negative or positive feedback to the worker, depending on the result. A visual representation of the system's framework can be observed in Figure 1.

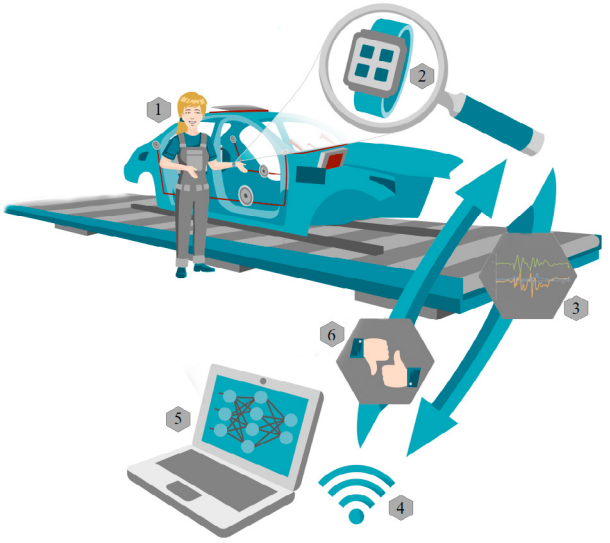

Fig. 1. Framework structure: 1. The worker is performing the assembly action; 2. The smartwatch is gathering data while the action is performed; 3. Data is sent to the computer for processing; 4. The connection between the smartwatch and the computer is made through the local wireless network; 5. The received data is classified using AI on the computer; 6. The classification result is sent to the smartwatch and the feedback for the performed action is given to the worker.

3.1. The assembly action

The action being subject to on-spot quality inspection performed by the worker is marked with "1" in Figure 1. The actions must be short in nature and generate specific audio and vibration signals. For the project presented hereby, the action for connecting wire plugs into sockets (Figure 2) was selected for developing the system and test it on. The moment the worker is connecting a plug, the generated sound (click!) is picked up by the microphone. The movement of the hand and the reactive force at the moment of connection is picked up by the accelerometer of the smartwatch.

3.2. QCApp

A specially developed WearOS application runs on the smartwatch (marked with "2" in Figure 1), named QCApp. QCApp is developed in Android Studio and provides a simple graphical user interface for starting and stopping the quality inspection process. Its back-end handles the data recording, data sending to the computer for processing, classification result and feedback delivery to worker. The WearOS application can be observed in Figure 3. Figure 4 contains a flow representation of the quality inspection process that is performed by the worker when using QCApp.

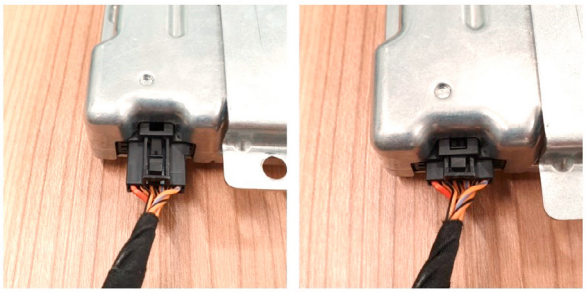

Fig. 2. Example of plugs that are connected to sockets by workers on the production line. On the left side: image of the plug being disconnected. On the right side: image of the plug being connected.

Fig. 3. Main view of QCApp running on a WearOS device. It contains a button for starting and stopping the on-spot quality inspection process: "Record"; information about the connection with the computer: "Client Disconnected"; information about how many audio and acceleration are stored in the memory of the watch: "Recordings stored: 0".

The audio and acceleration data recorded while the worker is performing the action are stored in a waveform audio file (.wav) and in a comma separated values file (.csv), respectively. The files are archived together after the worker finishes the task and stops the recording (after step 3 in Figure 4). The archive is then sent to the computer via Wi-Fi using the TCP/IP protocol. In the same manner, the feedback is received by the worker through the smartwatch as observed in steps 4a and 4b in Figure 4. The visual feedback that can be received by the worker can also be observed in Figure 4: green screen for positive feedback, red screen for negative feedback. Depending on the feedback, the worker either proceeds with the same action on the next item, or repeats the action on the same item until the action is successful and the received feedback is positive.

To ensure the functionality of QCApp, five threads are running in the back-end to ensure the TCP/IP communication with

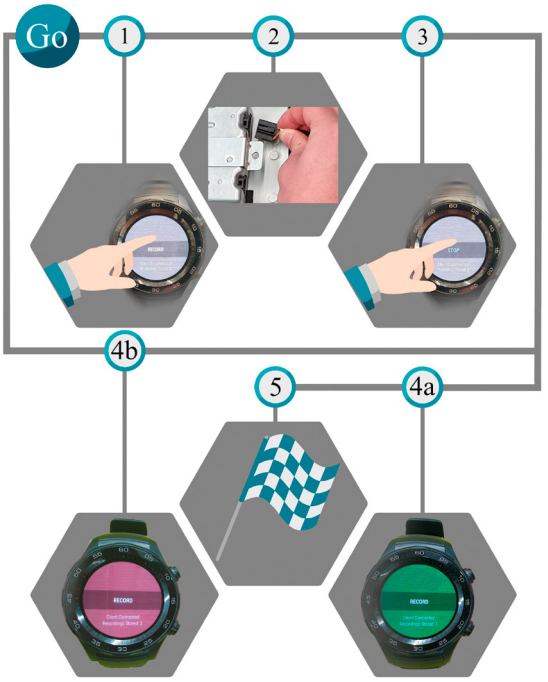

Fig. 4. Steps followed by the worker to perform on-spot quality inspection using QCApp running on the smartwatch. Step 1: The worker touches the screen to start the audio and acceleration recording of their action. Step 2: The worker performs the action to be quality checked. Step 3: The worker stops the recording. Step 4a: Positive feedback is received by the worker via the smartwatch display, speaker and haptically. Step 4b: If the action was not successful, negative feedback is received and the worker needs to repeat the action. Step 5: The action is successfully completed, the item can be passed further on the production line and the worker can proceed with the next item.

the computer, the audio signal recording, the archiving of the audio and acceleration files, and the feedback delivered to the worker. A thread based approach was chosen for developing the back end of the app due to the blocking nature of the TCP/IP sockets, audio recording and other processes that need to be executed concurrently rather then sequentially. The threads are sharing Boolean and string variables that allow inter-thread communication. To avoid any resource sharing related problems, the threads are synchronised using locks.

The main thread running in QCApp is the thread that handles the main activity of the app. The main activity manages what the worker sees on the smartwatch screen together with the input from the user or from the acceleration sensor. The user input is the push action of the ”RECORD” button. While a recording is active, the text of the button becomes ”STOP” (step 3 in Figure 4). When a recording is active, the main thread starts to record data from the accelerometer while the audio thread starts to record data from the microphone. Once the recording is stopped, the data is saved in their respective files.

The main operations performed by the other three threads are mainly related to files archiving, TCP/IP communication and feedback for the worker. The creation of audio and acceleration files (.wav and .csv) triggers the archiving process of these files. Once the archive containing the audio and acceleration files is sent to the computer, the archive is deleted. These operations are executed in a loop that ends when QCApp is stopped.

The TCP/IP thread handles the connection with the computer via Wi-Fi (marked with ”4” in Figure 1). Once the thread is started with the QCApp, it awaits incoming connection from a computer. On the smartwatch side, a TCP/IP server is implemented, while on the computer side there is a client. This approach was chosen as it is more accessible to modify the IP address of the smartwatch on the computer side if needed. If a client computer is connected to the smartwatch and the archive file is ready, the archive is transmitted to the computer. Before sending the archive to the computer, the size of this archive is sent. This is part of a file integrity process executed on the computer side that ensures the audio and acceleration file have not been altered during transmission. After the archive is received by the computer, the TCP/IP thread awaits the feedback for the action performed by the worker.

The feedback thread handles the delivery of the feedback to the worker via the smartwatch display, speaker and haptic functionality (marked with ”6” in Figure 1). The main reason this process is handled in a different thread is because of the blocking nature of the haptic feedback delivery, as a delay method is necessary which otherwise would halt the rest of the app execution. If the feedback is positive, the display gains the colour green as seen in step 4a in Figure 4, if negative, the display gains the colour red as seen in step 4b in Figure 4. The neutral state of the app is displayed to the worker as shown in Figure 3. The positive visual feedback is strengthened by an approval sound and a short haptic feedback, while the negative one has a disapproval sound and a long haptic feedback.

3.3. Audio and acceleration signals

The recorded audio and acceleration signals are the input for the AI action classification system which runs on the computer side of the framework, that can be observed in Figure 1. Plots of the typical acceleration and audio signals recorded by the QCApp and then sent to the computer, can be observed in Figure 5. The accelerometer which the Huawei Watch 2 is equipped with records data on three channels, each representing a movement axis of the smartwatch: x, y and z.

As it can be observed in the plot containing the acceleration signals (marked with ”3” in Figure 1), the ”click” of the plug is detected after about 100 milliseconds after the recording has started. The time spot of the click detected by the acceleration sensor matches the one in the plot for the audio recording. The ”click” sound is obvious in the plot of the audio signal.

The second plot with acceleration signals in Figure 5, showcases a failed attempt to connect a plug into a socket. Just before 100 milliseconds have passed after the start of the recording, a vibration is picked up as it can be seen in the plots. However, the audio signal remains flat, indicating that the connection was not successful.

Providing two signal sources rather than just one for classifying a connection action, it increases the reliability of the system significantly. It is important to mention that both the accel-

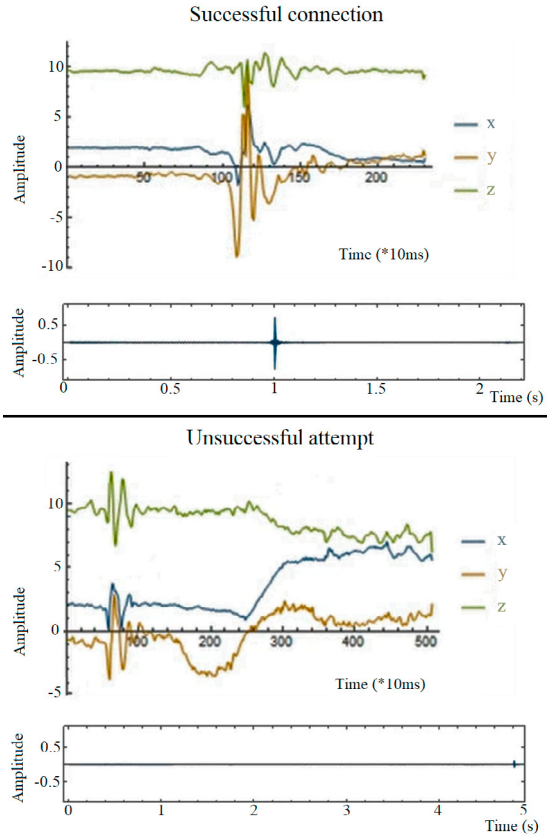

Fig. 5. Plots of the acceleration and audio data gathered while attempting plug connection actions. The top part represents plots of the acceleration and audio signals while a successful plug connection is detected. The bottom part contains plots made while recording an unsuccessful attempt to connect the plug into the socket.

eration sensor and the microphone are subject to various noise sources that can resemble the desired signal generated when a successful connection is performed. Therefore, by the rule of logical AND, the eventuality of a false positive generated by various noises is eliminated. It is desired to have no false positives generated by the system, as the worker needs to have full confidence on the positive result of the system once they pass the item further on the production line. While false negatives will only result in a double check of the system, a false positive may endanger the quality and performance of the product being output on the production line.

4. Deep learning model for classification

4.1. Data preprocessing - MFCC

Before performing the classification of the action, the computer receives the archive containing the recordings of the acceleration and audio data from the smartwatch. This is made possible using a TCP/IP client implemented using the Java programming language. The client connects to the server running on the smartwatch and then it waits for the archive to be received. After that, the recordings are dearchived and fed into the classification program that runs the classification CNNs. Once the action is classified as successful or unsuccessful, the result is sent to the smartwatch and the feedback process begins as displayed in steps 4a and 4b in Figure 4.

As stated in section 2, it is possible to repurpose CNNs so they can fit the action classification for plug connections. It is however necessary to transform the time domain signals gathered by the microphone and accelerometer into a form that the first convolutional layer of the CNN can interpret. This is done by generating an image using the MFCC [9]. The MFCC generated images are used to train, validate and test models for the data. As the approaches are similar in both the acceleration and audio case, only the audio case will be presented in what follows.

A number of 200 audio and acceleration recordings were gathered on the production lines at Mercedes-Benz AG while the worker performs a successful connection and the same number while they perform an unsuccessful connection. These 200 recordings were used to train, validate and test the AI model for classification.

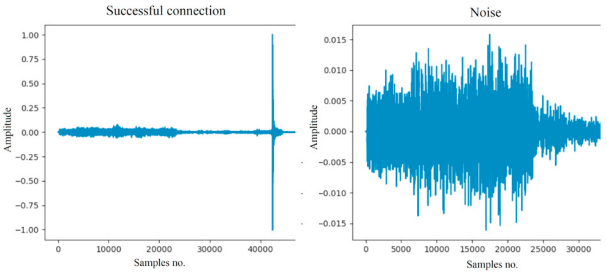

Fig. 6. Plots in time domain of audio signals recorded at 44.1 kHz sample rate for training the AI model. On the left hand side, plot of the audio signal recorded while performing a successful plug connection. On the right hand side, plot of noisy audio signal. To be observed the difference in amplitude range between the two samples.

The MFCC enhances the focus of high energy components in the signal like the ”click!” sound of a successful connection which can be observed in the plot on the left hand side in Figure 6 after the 40.000 samples (approx one second) point. This approach is necessary to generate features that can be fed into the DCNN by discarding the low-energy components like noise. By visually inspecting Figure 7 and Figure 8, it is enough to notice the difference between recordings containing ”click!” and recordings not containing ”click!”. As observed in the figures, it is intended to obtain a certain consistency in the features available in the images (in this case, it can be clearly noticeable on the top of the image, where the contrast between high and low energy components is visible across horizontal stripes).

4.2. Action classification - CNN

Having the time domain audio and acceleration signals under a form from which features can be extracted by the convolutional layer of the implemented CNN, it is possible to repurpose

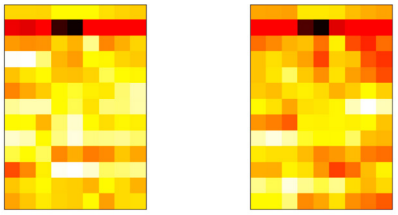

Fig. 7. Images generated using the MFC coefficients of audio signal recordings while performing successful connections. It can be noticed the dark stripes caused by the high amplitude signal generated by the ”click” sound during plug connection in the top part of the image, which are consistent across successful connection recordings.

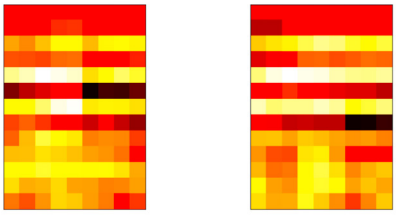

Fig. 8. Images generated using the MFC coefficients of audio signal recordings while performing unsuccessful connections.

implementations already available for image classification using Keras [8]. The implementation hereby presented is based on typical neural network architectures for binary classifications. It is important that the MFCC generated images all have the same size; 89x520. 200 samples were used for training the audio and acceleration models each. 180 samples were used for training, 18 for validating and two for testing.

It is determined that after only ten training epochs the trained CNN model is accurate enough to provide a reliable classification of the MFCC data. As it can be observed in Figure 10, the accuracy of the model over the training data is very good, with the validation accuracy reaching over 90%.

There is a level of concern regarding the overfitting of the model on the training data set. However, after the testing of the model, it is deemed to be accurate enough to make reliable classifications. The testing results can be observed in Figure 11. For testing the model, the test samples are used together with two random training samples (for successful and unsuccessful actions). The classification in Figure 11 is correct.

As mentioned before, the Keras library is used to design the CNN model used for performing classification on tasks. The CNN is built as linear stack of layers, each layer feeding data into the next one as it can be observed in Figure 9. The first layer of the network is a convolutional layer. The input for this layer is the 89x520x3 tensor representing the value for each pixel in the MFCC image. The value 3 indicates the number of channels that gives the colour and brightness of the pixel (RGB). The convolutional layers have filters that contain weights used in element-wise multiplications with the pixel values of the input image. The output of the element wise multiplications are feature maps [2]. Every convolutional layer used in this application has a filter of size of 3x3. As for the first convolutional layer it is desired to keep the dimensions of the image, therefore the padding is automatically adapted to ensure that the output has the same dimensionality as the input.

The weights of the filter are randomly initialised and then updated during the training session in order to detect specific features in the image. As explained in subsection 4.1, by using the MFCC, it is intended to obtain images with accentuated features that makes the determination of the weights as fast and precise as possible. A remarkable feature in the MFCC images, as observed in Figure 7 and in Figure 8 are the stripes in the top-middle region of the images which appear to have an accentuated contrast in comparison to the rest of the image. It is safe to assume that as a result of convolution operations between the filter and the receptive area of the image, the values obtained while scanning the ”successful signal” are higher for the top part of the image than the values obtained while scanning the ”unsuccessful signal” in the same region. The brightness across the horizontal stripes in the images generated from ”successful signals”, have a consistent location and accentuated contrast on the frequency band of the ”click”, while the brightness is more evenly distributed in the case of an ”unsuccessful signal” with no ”click” or a failed ”click”. This is likely the classification criteria that the CNN learns during training.

The activation function used for convolution ensures the desirable output by performing non-linear element-wise operations, turning any negative value into 0. The used activation function is the rectified linear unit (ReLU). The reason for ReLU being used as an activation function is speeding up training through the simple nature of the operation: 0 if the value is negative, value is kept as it is if the value is positive. Thus easier training process and better performance are obtained.

A max pooling layer is implemented for reducing the dimensionality of the feature maps. This layer uses a window of defined size (2x2 in this implementation) slided over the rectified feature map taking the max value in each region. The reason for using pooling is that once a feature is found in the feature map, its location is not important so what is around it can be discarded. As observed in figure Figure 9, the fourth layer from left to right is a max pooling layer reducing the height and width of the tensors by a factor of two. The stride factor for the layer is set to 2, having as consequence the disregarding of the 89th column in the input feature maps and removing the influence of padding from the convolutional layers.

The convolutional part of the neural network contains four groups, each containing two convolutional layers and one max pooling layer for dimensionality reduction. The resulted tensor from the convolution is passed to a flattenning layer that reshapes it into a tensor of 4069 single elements. This is the first layer of the fully connected part of the neural network that contains two more dense layers of size 256. The dropout technique is applied to these two layers with a deactivation factor set to 0.5. This means that the weights of 50% of the neurons randomly chosen in the layer are set to 0 each step, while training. Dropout is implemented to prevent overfitting the model on the training data by adding randomness to it. In other terms, this technique is used to increase the robustness of the model. No dropout is implemented in the convolutional part of the NN as

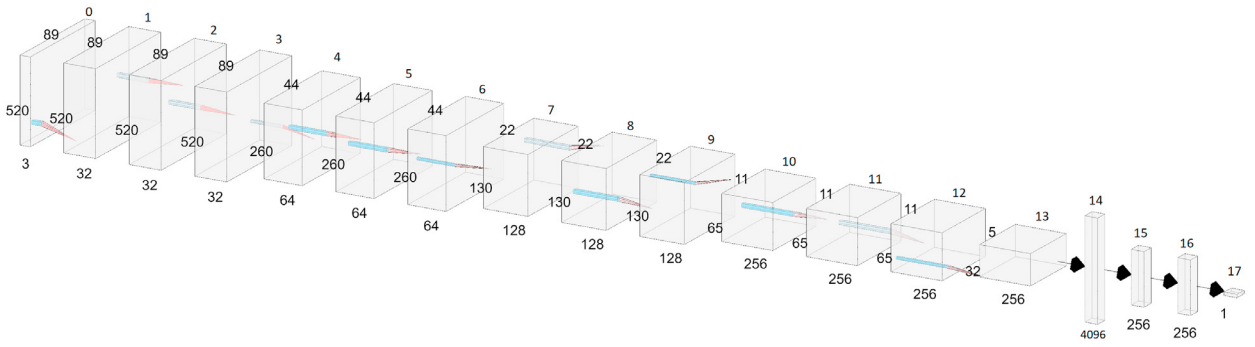

Fig. 9. Arcthitecture of the neural network. There are four distinct groups of layers for reducing the dimensionality of the input containing two convolutional layers and one max pooling layer. Convolutional layers 1 and 2 together with the max pooling layer 3 make up the first group that reduces the height and width of the input from 89x520 to 44x260. The fourth group (layers 10-12) outputs 256 feature maps of size 13x32. The tensor is reshaped into a vector of 4096 elements (layer 14). Layers 14-16 are fully connected. The output is a single binary value that indicates the classification result: successful or unsuccessful

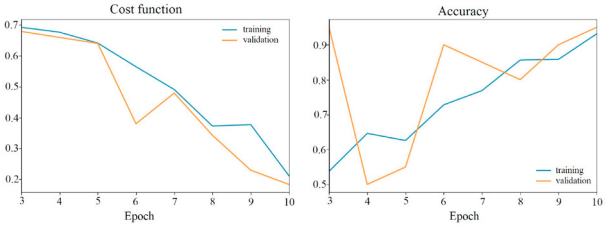

Fig. 10. Computed loss and accuracy of the model during training over ten epochs

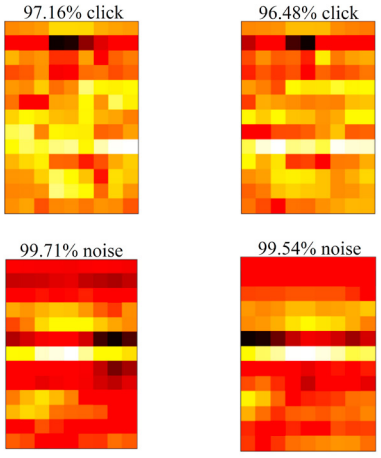

Fig. 11. Correct classification of the MFCC generated images using the Keras trained CNN model

there is no need to penalise time in favour of accuracy when performing training. As it will be presented in what follows, the obtained accuracy is satisfactory. The overall architecture and further design details can be observed in Figure 9.

During training, the objective of the model is to minimise the value of the binary crossentropy loss function, as observed in Figure 10, on the left side. This measures how far away the predicted result of the neural network is from the expected result. The weights for the convolutional filters are iteratively updated through backpropagation. After determining the weights of the filters these are saved in a .json file and in a .h5 file to be used for classifying the incoming data from the smartwatch.

5. Experiments and results

5.1. User testing

Experiments and tests were conducted to verify the quality and performance of the proposed framework. The system was tested in production environments where the environment is susceptible to acoustic and vibration noise. The experiments implied the following:

- The smartwatch running QCApp is tested by five different workers
- Each worker attempts 50 successful connections attempts
- Each worker attempts 50 unsuccessful connection attempts (however they expect fit)
- The data recording does not last for more than five seconds for an operation

Given that a mandatory requirement is to have zero false positives (in case of a failed connection, the smartwatch should not provide positive feedback to the worker under any circumstances), the success of the framework was measured to be 96%.

6. Discussion and conclusion

The presented framework demonstrates a novel approach for on-spot real time quality inspection when performing certain manual operations on production lines. It was shown that by utilising the input methods available on a smartwatch, it is possible to gather relevant data that can be processed and fed into

a CNN classification model, thus generating feedback to the worker that can take action accordingly.

Though a result of 96% success regarding the classification of operations is satisfactory for an initial proof-of-concept, future work will improve and refine the technique in order to ensure full reliability on the system. A few proposed ways for further development in this regard are:

- Improve the training data pre-processing by enveloping and thresholding the signals in the success data set so only the actual "click!" will remain in the training data set
- Increase the number of training samples
- Increase the number of validation samples
- Add more relevant signal sources.

However, given a result of over 95% accuracy while testing the model with test data as described in subsection 4.2, the cause of having a 96% overall success may also be due to the fact that the connection between the smartwatch and the computer on which the Convolutional Neural Networks are running is made through WLAN. Combined with the fact that the environment where the framework was tested contains a large number of Wi-Fi networks [12], the problem can just as well be generated by the corruption of audio and acceleration files while transferred. The Wi-Fi performance can also affect the real-time capability of the system adding a latency of 1-5 seconds for receiving the feedback on the smartwatch. A solution for this would be to have a full implementation on the smartwatch for classifying the actions, while having only the training process on a computer.

The framework presented in this paper makes use of a Keras implementation for classifying the worker's action as successful or not, through Convolutional Neural Networks. Two other implementations were conducted at the same time, one using the PyTorch machine learning library and one using Wolfram Mathematica. The discussion is raised therefore upon which implementation is best in terms of classification reliability. Another aspect is, which implementation is better to be transferred on the smartwatch in order to remove the need of performing data transfers via Wi-Fi.

7. Future work

Given the big potential of the wearable quality inspection system, future work and research is in place to extend the framework's capabilities in terms of reliability and user experience. Plans are currently in place for investigating a new signal source for increasing the reliability of the system. By using the BIOX Armband, arm and hand motions can be tracked and therefore predict what action the user is going to perform [1]. This is highly relevant also because the smartwatch currently needs user input for starting the recording of signals when the action is going to be performed. By integrating the BIOX armband in the system presented so far, the quality of the user experience is also expected to increase.

Acknowledgements

This work was supported by Mercedes-Benz AG, ARENA2036 e.V. (Stuttgart, Germany) and Department of Materials and Production, Aalborg University (Aalborg, Denmark).

References

[1] Biox, 2020. About Us. https://www.bioxgroup.dk/products-biox-armband/.
[2] Chollet, F., 2016. Building powerful image classification models using very little data. Keras Blog .
[3] Cvetković, B., Szeklicki, R., Janko, V., Lutomski, P., Luštrek, M., 2018. Real-time activity monitoring with a wristband and a smartphone. Information Fusion 43, 77 – 93. doi:https://doi.org/10.1016/j.inffus.2017.05.004.
[4] De Carolis, A., Macchi, M., Negri, E., Terzi, S., 2017. Guiding manufacturing companies towards digitalization a methodology for supporting manufacturing companies in defining their digitalization roadmap, in: 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), pp. 487–495. doi:10.1109/ICE.2017.8279925.
[5] Fedorko, G., Liptai, P., Molnár, V., 2018. Proposal of the methodology for noise sources identification and analysis of continuous transport systems using an acoustic camera. Engineering Failure Analysis" 83, 30 – 46. doi:https://doi.org/10.1016/j.engfailanal.2017.09.011.
[6] Garcia-Ceja, E., Galván-Tejada, C.E., Brena, R., 2018. Multi-view stacking for activity recognition with sound and accelerometer data. Information Fusion 40, 45 – 56. doi:https://doi.org/10.1016/j.inffus.2017.06.004.
[7] Greiner, J., Euteneuer, F., Schreiber, M., 2019. Qualitätsüberprüfung manueller Steckermontage per externer Sensorik und Künstlicher Intelligenz. [(unveröffentlichte) Patentanmeldung: DE 102019006810].
[8] Gulli, A., Pal, S., 2017. Deep learning with Keras. Packt Publishing Ltd.
[9] Lyons, J., 2020. Mel Frequency Cepstral Coefficient (MFCC) tutorial.
[10] Mueller, R., Vette, M., Masiak, T., Duppe, B., Schulz, A., 2019. Intelligent real time inspection of rivet quality supported by human-robot-collaboration, in: SAE Technical Paper, SAE International. doi:10.4271/2019-01-1886.
[11] Piczak, K.J., 2015. Environmental sound classification with convolutional neural networks, in: 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6. doi:10.1109/MLSP.2015.7324337.
[12] R.S. Mogensen, I. Rodriguez, G.B.A.F.R.M.S.M.T.R.T.K.G.P., Barbera, S., Fall 2019. Implementation and trial evaluation of a wireless manufacturing execution system for industry 4.0. IEEE Vehicular Technology Conference (VTC) .
[13] Rubin, J., Abreu, R., Ganguli, A., Nelaturi, S., Matei, I., Sricharan, K., 2016. Classifying heart sound recordings using deep convolutional neural networks and mel-frequency cepstral coefficients, in: 2016 Computing in Cardiology Conference (CinC), pp. 813–816.
[14] Seddik, H., Rahmouni, A., Sayadi, M., 2004. Text independent speaker recognition using the mel frequency cepstral coefficients and a neural network classifier, in: First International Symposium on Control, Communications and Signal Processing, 2004., pp. 631–634. doi:10.1109/ISCCSP.2004.1296479.
[15] Suarez, I., Jahn, A., Anderson, C., David, K., 2015. Improved activity recognition by using enriched acceleration data, in: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Association for Computing Machinery, New York, NY, USA. p. 1011–1015. doi:10.1145/2750858.2805844.
[16] Sugandi, B., Toar, H., Alfianto, A., 2018. Hand tracking-based motion control for robot arm using stereo camera, in: 2018 International Conference on Applied Engineering (ICAE), pp. 1–5. doi:10.1109/INCAE.2018.8579360.