# A Novel approach to Increase Productivity in the Industry Using Wearable Devices and Artificial Intelligence

Shafkat Waheed
Research Assistant
*Electrical and
Computer Engineering
North South University
shafkat.waheed@gmail.com*

B. M. Raihanul Haque
Research Assistant
*Electrical and
Computer Engineering
North South University
raihanul.haque@northsouth.edu*

Dr. Mohammad Ashrafuzzaman Khan
Assistant Professor
*Electrical and
Computer Engineering
North South University
mohammad.khan02@northsouth.edu*

***Abstract*—This paper proposes a general approach to increase the productivity in day to day work flow of the people who are engaged in monotonous task in the industry.This papers explores various deep learning techniques like convolutional neural network and Wavelet Analysis to extract information from wearable devices such as eSensor and Camera.This paper further explores how to give meaningful feedback using Recurrent Neural Network to maximize worker productivity through out the day, Some feedback like worker schedule,stress level and the method of working is suggested in this paper that would increases the total work flow in the industry.**

## 1. Introduction

Since the invention of wearable devices, more and more wearable devices are being used to solve day to day problems [?].The the future is IOT devices and providing smart solution through it.IOT devices are being applied in home monitoring, health monitoring and improving human experiences intensively[?]. These small devices of broad spectrum are changing the way one interacts forever[?].

Human beings are only able to make decision and optimize their day to day activities using the six sense they posses[?].IOT devices allowed us to go beyond our five senses, this added dimensionality changed the way one makes decision.These devices worked as a catalyst to provide more information than what one could collect using their biological senses, these information with the help of machine learning and A.I enabled drastic optimization on every sector that feeds on data.Wearable devices like fitbit, smart watch are changing the whole scenario of data harvesting and decision making [?]. The world is changing, due to small optimization provided by these IOT devices and A.I. Therefore understanding the application of such devices has opened new doors of research.

Manufacturing industry is no stranger to Iot devices and A.I[?].Germany was the first to understand the potential of optimizing manufacturing process using IoT devices and Artificial Intelligence[?].They were able to change the whole scene of manufacturing with the integration of small devices in everyday production.Sensors like accelerometer, gyroscope, heat detector, light detector and vibration detector increased the dimension of standard information one could garner or gather[?]. Information of such volume crafted the way for machine learning and A.I to effectively optimize the work flow, industrial production and efficiency.

We used the data collected from wearable device like E-Sensor and Surveillance Camera to detect complex monotonous work performed by workers in an industry setting.

## 2. Motivation

In the past few years machine learning and deep learning are making tremendous progress in the field of science and technology.Neural networks and statistical models allowed us to analyze data and extract intelligence from it.Now, with the help of such technology we are able to trace and detect human activity. Such advancement provides us with scope of optimization of monotonous complex human work. The detection of such complex work though possible but not yet implementable in the industrial sector. Among the works done in this field most researcher used various sensor data to map the human activity or used images from camera to marginalize a model that can classify human activity but the curse of accuracy still plagues the model. This halts the process of implementing this model in real world. As, we wished to optimize monotonous work of garments industry we developed a model that integrates data from two sources of origin one from sensor and another from surveillance camera which uses Neural Network to accurately detects human work, furthermore The model is accurate enough to be applied in garments industry for activity optimization.

## 3. Related Works

Wearable devices can be like a watch,spectacles or headphones but to collect data of head and mouth related activities a small device on ear can be used that contains various sensor like accelerometer and gyroscope. eSensor is designed to collect such data.Three aspects helped the design decision: the physical dimension of the eSense printed circuit board to maintain the aesthetics and comfort, the minimization of signal interference from adjacent sensors, and the maximization of battery life to offer the primary functional service[5]. eSense can be effectively used to monitor head- and mouth-related behavioral activities including speaking, eating, drinking, shaking, and nodding, as well as a set of whole-body movements. Moreover, with eSense conversational activity monitoring capabilities, social interactions can be quantified that to further help treat different mental health conditions and provide well-being feedback[4].

This paper [1] showed how a data collected from phone can be used to predict simple human activity with the help of traditional machine learning techniques.Firstly they collected the data from a mobile phone, the data was accelerometer data of three axis. The collected data was pre-processed and supervised classifier techniques were was used to map the data to its designated classification.

The proposed model [2] used data collected from RGB camera and Image of field depth to detect human activity for surveillance.The model took frames of images with the help of kinet and used them to detect human activity with the help of Support Vector Machines.But the model only can detect activity of short time period.

The paper [3] suggested a different approach of detecting instead of using only one source of data he integrated data collected from image and sensors.Further, he used CapsNet and lstm to extract intelligence form two sets of data and merged them at the end to accurately classify the data into nine activity of short time span.

The idea [4] is to read bio-acoustic signal from smartwatches and other wearable that resides with body. A customized kernel has been built to carry out the task where data is being sampled with the rate of up to 4 KHz. Newly generated data, which is barely distorted, is used to define hand gestures, input modalities, motor-powered object detection and so on. The combined system is called ViBand, which contains, accelerometer found in smartwatches, domains of examples, user studies and examples demonstrating applicability.

AJ Piergiovanni and Co. [5] discussed an approach to classify human activity using temporal attention filters. Any high-level activity can be sub-divided into multiple small events also known as sub-events or temporal events and they can vary in terms of duration. From a video feed, with the help of temporal filters coincided with segment based CNN and recurrent LSTM, the system learn these small events which corresponds to certain activity.

Quentin Mascret et. al. [6] proposed that instead of using supervised machine learning techniques and deep learning methodologies on extracted data features new models can be implemented on raw dataset. Therefore, raw dataset have been collected from customized motion sensors placed around able-bodied humans, which have embedded in them are inertial measurement unit (IMU) sensors and Surface Electromyography (sEMG). After that, Spherecial normalization is used for the purpose of pre-processing, normalization and augmented with that are support vector machine and radius basis function (RBF-SVM) to classify the activity.

We used a similar approach of classification where we collected data from surveillance camera and accelerometer.Further, we converted the accelerometer data into scalogram with the help of wavelet. We integrated the data on timestamp and applied two different CNN to extract intelligence to detect activity of small time span.Our aim was to develop a model that could detect complex work for optimization of monotonous work done on industry.Therefore, we defined a time window for complex work as complex work can be composed of activity of small time span so, various small activity detected on the defined time window of complex work can be used to determine the complex.

## 4. Proposed Methods

To carry out the task few machine learning models have been considered. Each model has their own role to play and also generate results collectively. From end to end, two neural networks i.e. Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) have been used along with a Fourier transformation strategy called wavelet analysis. eSense data and images will be analyzed by these suggested methodology to produce comprehensive and meaningful results.

### 4.1. CNN

One of the benefits of using this network is, unlike other algorithms, less pre-processing is required which makes it much more suitable for image analysis. The other convenient factor is it can capture spatial and temporal dependencies while reducing number of parameters and reusing weights to understand a sophisticated image. The dimension of the input image is denoted as such, $height \times breadth \times channels(RGB)$ and after performing convolutional operation with a kernel or a filter a new matrix having convolved features is generated. Whether there is one channel or are multiple channels the kernel strides in such a way which creates squashed one-depth channel convoluted feature output. The first layer extracts low level features and the following layers extracts high level features which creates a network to understand a image like a human would do. However, the deimensionality reduction process is carried out by valid padding and in order to preserve the same dimension or increase it same padding is incorporated. Pooling is of two type i.e. max and average and this technique is brought to extract dominant feature and decrement the spatial size which allows to use low computational power. Max pooling is preferable since it is noise suppressant. After going through all these processes, the obtained values are flattened into a column vector which is then fed to a conventional feed-forward neural network along with backpropagation techniques.

### 4.2. Wavelet analysis

Wavelet transformation, unlike any other Fourier transformation methodology, has the ability to compress an image efficiently. By managing factors like shifting and scaling it can decompose an image to multiple lower resolution image. THe waves have features like varying frequency, limited duration and zero average value. This is also eligible to remove high frequency noise from a dataset. The implementation of wavelets revolves around implementing two different transformation and incorporating one threshold function. Two transformations are wavelet transformation and inverse wavelet transformation. The wavelet transformation is achieved by the following formula,

$$C(\tau.s) = \frac{1}{s^{\frac{1}{2}}} \int_t f(t)\psi^*(\frac{t-\tau}{s})dt$$

Above is the formula for continuous wavelet transformation where, $\tau$ and s are transition parameter and scale parameter respectively, $\frac{1}{s^{\frac{1}{2}}}$ is normalization constant and $\int_t f(t)\psi^*(\frac{t-\tau}{s})dt$ is the mother wavelet. Inverse operation is carried out by the following function:

$$f(t) = \frac{1}{s^{\frac{1}{2}}} \int_\tau \int_s C(\tau.s)\psi(\frac{t-\tau}{s})d\tau ds$$

Discrete wavelet transformation is bit straight-forward than this and, to state the obvious, free from integral operation. The formula for discrete wavelet transformation is, $a_{jk} = \sum_t f(t)\psi^*_{jk}(t)$ and inverse discrete wavelet transformation can be written as such, $f(t) = \sum_k \sum_j a_{jk}\psi_{jk}(t)$. These formulas provide simultaneous localization in time and scale, sparsity, adaptability and linear time complexity which allow noise filtering, image compression, image fusion, recognition, image matching and retrieval efficiently. Finally, an additional threshold function can be represented as such to improve the proposed model, $threshold = alpha * noise * \sqrt{data\_size}$ and in this formula alpha is a constant and noise = absolute median value.

### 4.3. RNN

Both CNN and RNN have fundamental similarities which is sharing parameters. RNN has the ability to generate future information based on its past. A general NN remembers things during training and while RNN does the same, additionally, it remembers stuffs from previous inputs during producing outputs. Also, unlike NN, RNN can tackle unlimited number of inputs (not fixed initially) and these input vectors are manipulated by the weights of the inputs and hidden state vectors. Thus, this can give rise to one or more output vectors. Since, no fixed input is fed into this model there cannot be any fixed weight for individual input. Thus weights are being shared by each input and to maintain versatility and depth hidden state vectors come into action creating link between two inputs. This parameter sharing strategy makes it different than conventional NN. Furthermore, to have multi-level abstraction and representation any of the four following methods can be tried; (a) have more hidden states, (b) have more non-linear hidden layers and lay them between input and hidden state, (c) have more depth within hidden states and (d) have more depth in between hidden states and output layer. These techniques can also be found in Bidirectional RNN, Recursive neural netwrok, Encoder Decoder Sequence to Sequence RNN and last but not least in LSTM with slight variation.

## 5. Methodology

The objective here is to generate a description of a complex work that is retrieved from a video feed. To accomplish that, initially a video is captured containing the process of making an poached egg. The process is consists of eight steps where two particular steps are identical to another two. The duration of each step is 10 to 12 seconds long so that over-fitting can be avoided. At first, image were extracted from these video feeds. Then captions were generated and finally, a graph was created where each graph represents each steps.

### 5.1. Image extraction

To extract the images, one of the most popular library was used known as OpenCv. Depending on the sequence of steps frames were generated and sorted. Total number of 650 frames were extracted from all the videos. Figure 1 represents one of the frames that was used.

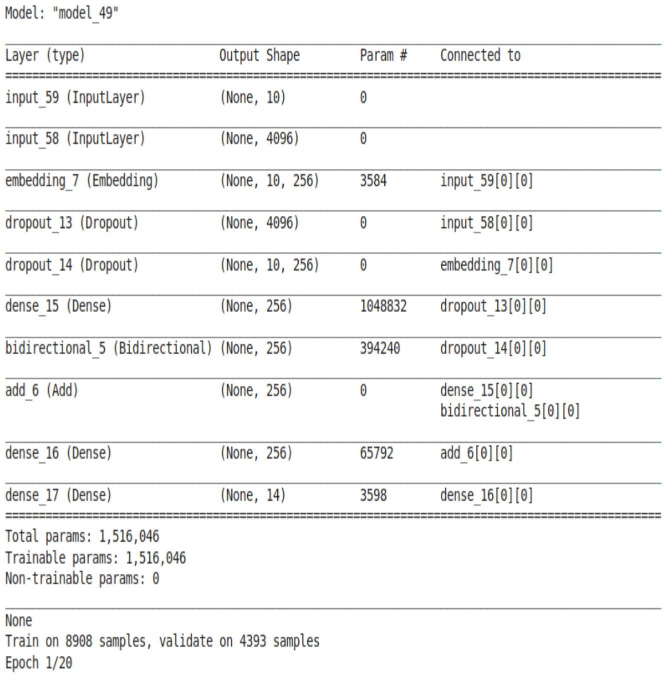
Figure 1: A frame of the complex work showing egg is being hold by a person

### 5.2. Image captioning

At the beginning, unique ids were created for each image and against each id two captions were assigned. Since, a bunch of images represents a particular phase captions generated for those images are same. Now, to extract the features from images VGG model is called upon which is a pre-trained model. VGG causes faster computation and less memory consumption. The focus here revolves around returning the dictionary which contains image features of internal layers and save it to a file.

The captions that were generated earlier were loaded from a file and at the same time unique frame identifier is returned based on the description. To alleviate the difficulty working with the description, they were tokenized and cleaned. Tokeniztion is a procedure where description turns into individual words or vocabulary and cleaning process involved making the description in lowercase and remove redundant words, numbers, punctuation marks and articles. This newly created dictionary is then saved to a new file to be loaded later on.

At the beginning of our training, photo features were loaded from the file that was previously created. Along with that, description was loaded ase well. In order to map description to unique integer value tokenization was performed on the training set before feeding it to the model. Two string, ‘startseq‘ and ‘endseq‘, were used to mark the start and end of an caption. The sequence of words were generate based on the parameters i.e. highest length of the sequence, tokenizer and both descriptions (image and text).

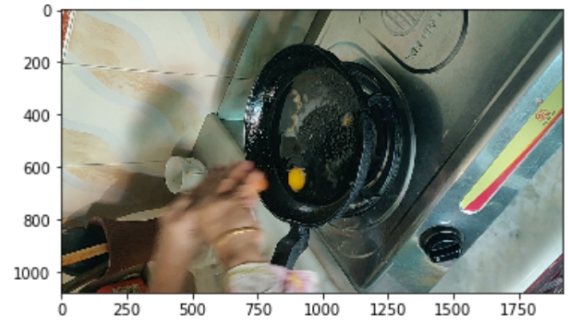


```
Model: "model_49"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_59 (InputLayer)           (None, 10)           0
__________________________________________________________________________________________________
input_58 (InputLayer)           (None, 4096)         0
__________________________________________________________________________________________________
embedding_7 (Embedding)         (None, 10, 256)      3584        input_59[0][0]
__________________________________________________________________________________________________
dropout_13 (Dropout)            (None, 4096)         0           input_58[0][0]
__________________________________________________________________________________________________
dropout_14 (Dropout)            (None, 10, 256)      0           embedding_7[0][0]
__________________________________________________________________________________________________
dense_15 (Dense)                (None, 256)          1048832     dropout_13[0][0]
__________________________________________________________________________________________________
bidirectional_5 (Bidirectional) (None, 256)          394240      dropout_14[0][0]
__________________________________________________________________________________________________
add_6 (Add)                     (None, 256)          0           dense_15[0][0]
                                                                 bidirectional_5[0][0]
__________________________________________________________________________________________________
dense_16 (Dense)                (None, 256)          65792       add_6[0][0]
__________________________________________________________________________________________________
dense_17 (Dense)                (None, 14)           3598        dense_16[0][0]
==================================================================================================
Total params: 1,516,046
Trainable params: 1,516,046
Non-trainable params: 0
__________________________________________________________________________________________________
None
Train on 8908 samples, validate on 4393 samples
Epoch 1/20
```

Figure 2: Overview of the model after first epoch

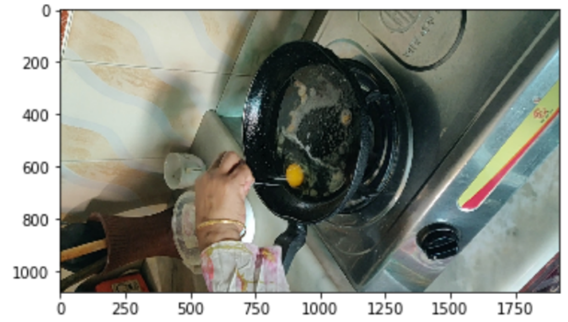
As for the model, Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) were implemented. The text, which was encoded as integers, was fed to one part of the model and the image is fed to another part. This will create a probabilistic distribution of the caption to be matched with an image. The structure of the model is shown in Figure 2.

The evaluation was done in several steps. Initially, a test dataset was formed containing photos. Then, the model was called recursively to generate captions against the test datset. Then, a mapping between the caption and image was observed to see if the model can successfully generate the description. Figure 3 and Figure 4 illustrates how well captions were generated for new dataset.



```
startseq egg on pan endseq
```

Figure 3: Successful prediction of an image from test dataset

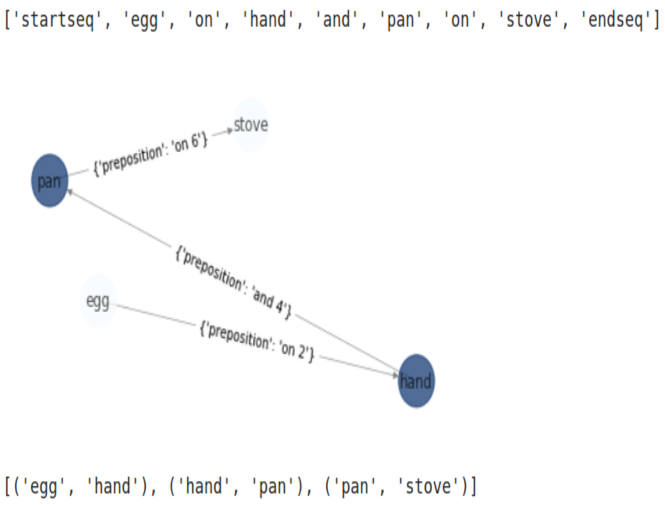


```
startseq egg on pan and hand over pan endseq
```

Figure 4: Successful prediction of another image from test dataset

### 5.3. Graph generation

Before generating the graph, the descriptions were split into noun and preposition. Hence, we made a list of preposition and a dictionary containing pairs of nouns to be connected by any of the preposition from the list. Each node of the graph represents noun and each directed edges the preposition. The number associated with preposition indicates timestamps or sequence of a work. For all the similar captions there is only graph mapped to it, illustrating a certain steps. Figure 5 demonstrate the outlook of the graph for a certain caption.



```
['startseq', 'egg', 'on', 'hand', 'and', 'pan', 'on', 'stove', 'endseq']
```

```
[('egg', 'hand'), ('hand', 'pan'), ('pan', 'stove')]
```

Figure 5: Directed graph of a caption

## 6. Result

In the methodology section, we have mentioned an event or action (poached egg making) and how we extracted the data, generated captions and produce graph of the corresponding image. To test the efficiency of our model, we used a YouTube video (https://www.youtube.com/watch?time_continue=57&v=wr8GJytDjZo&feature=emb_title).

The video demonstrates a man, organizing tools (wrench, hook etc.) as part of maintenance work. Using the same process, we have generated graphs of the corresponding images as shown in Figure 6. One thing to note here is that in some of the images i.e. $c_1$, $e_1$ and $i_1$ noise is present (graphical tick mark). This is because we used a video that is not recorded by us. In real life cases this type of noises can easily be avoided.

Figure 6 represents that, in all the 9 cases the graphs have been generated successfully. Usually, in real life scenario many incident takes place withing a certain frame and it is possible to generate captions for all of them. However, this could be redundant and we only generated graph captions for those which are relevant to the event. For instance, if the man putting a hook on the bar then the captions will be like hook, on ,bar with arrows in between.

Unlike our poached egg example, the nodes of opposite ends do not represent noun only but any other types of word. For example, in $b_2$ the caption is perfectly, on ,bar. This shows the efficiency of graph generation in fashion which can describe an event properly. Likewise, the middle word can be a verb instead of a preposition as seen in $h_2$.

## 7. Conclusion

The conclusion goes here.

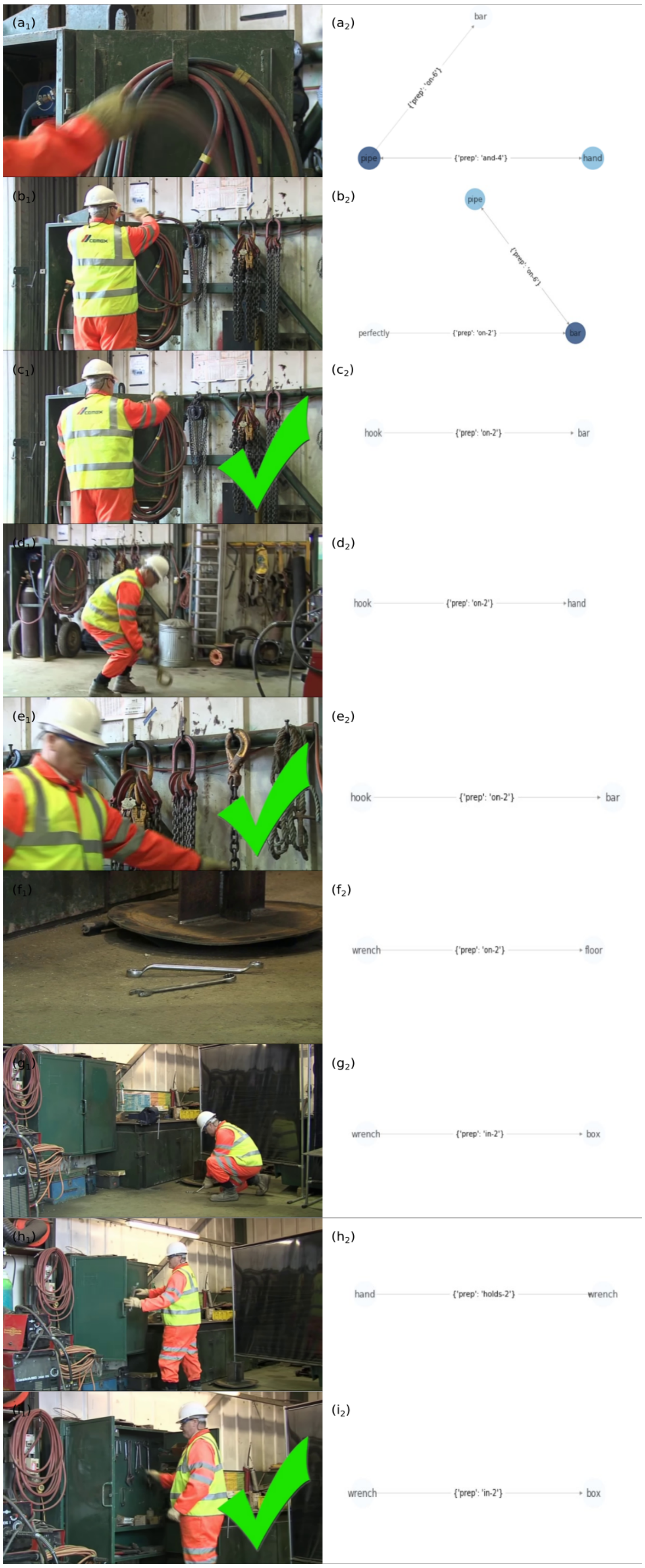


Figure 6: Generated graphs using the frames of a video

## Acknowledgments

## References

[1] Ahmed Taha, Hala Zayed and El-Sayed M. El-Horbarty ”Human Activity Recognition for Surveillance Applications,” The 7th International Conference on Information Technology. Devices, May 2015. DOI: 10.15849/icit.2015.0103.

[2] Kavita V. Bhaltilak, Harleen Kaur and Cherry Khosla ”Human Motion Analysis with the Help of Video Surveillance: A Review,” International Journal of Computer Science and Information Technologies. Devices, Vol. 5 (5), 2014. ISSN: 0975-9646.

[3] Yantao Lu and Senem Velipasalar ”Autonomous Human Activity Classification from Ego-vision Camera and Accelerometer Data,” EPIC@CVPR2019 workshop, Cited as: arXiv:1905.13533. Devices, 28 May 2019.

[4] Gierad Laput, Robert Xiao and Chris Harrison, ”ViBand: High-Fidelity Bio-Acoustic Sensing Using Commodity Smartwatch Accelerometers,” The 29th Annual Symposium. Devices, October 2016 DOI: 10.1145/2984511.2984582.

[5] AJ Piergiovanni, Chenyou Fan and Michael S. Ryoo ”Learning Latent Sub-events in Activity Videos Using Temporal Attention Filters,” Journal reference: AAAI 2017, Cited as: arXiv:1605.08140. Devices, version-3, 26 Dec 2016.

[6] Quentin Mascret, Mathieu Bielmann, Cheikh-Latyr Fall, Laurent J. Bouyer and Benoit Gosselin ”Real-Time Human Physical Activity Recognition with Low Latency Prediction Feedback Using Raw IMU Data,” IEEE Engineering in Medicine and Biology Society. Conference 2018:239-242. Devices, July 2018. DOI: 10.1109/EMBC.2018.8512252.