

Computer Vision-based Reader for analogue Energy/Water Meters in low-cost embedded System: a Case Study in an Office Building in Scotland

Guerra-Cabrera Adalberto^{1*}, Barbano Giulia¹, Tardioli Giovanni² and Mallya Udupi Girish²

¹Integrated Environmental Solutions LTD, Research and Development, Glasgow, United Kingdom

²Integrated Environmental Solutions LTD, Research and Development, Dublin, Ireland

Abstract. Implementation of cost-effective energy conservation measures (ECMs) is expected to generate up to 18% of carbon emissions reductions in office buildings. In order to determine adequate ECMs for a specific building, operational data is required. However, buildings generally lack operational data in the form of time series that can limit a breath of analysis required for determining adequate ECMs. Energy time-series data is commonly lacking in the UK due to uneven availability of smart meters (heat, gas, water), security restrictions in Energy Information Systems (EIS) and building management systems (BMS), restrictions and costs associated for automated reporting from utility companies, etc. This work presents a non-intrusive computer vision-based reader to generate energy readings at 10-minute resolution using a Raspberry-Pi, a traditional webcam and an LED light. OpenCV, an open source computer vision library, is used to detect and interpret numeric values from a heat meter, which are in turn uploaded to a cloud-based energy platform to create a complete operational data set enabling detailed analytics, fault detection and diagnostics (FDD) and model calibration. A case study of an office building in Scotland is presented. The building has a heat meter with no remote access capabilities. The accuracy of the method, i.e. the ability of the script to accurately derive the rate of change between readings, resulted on a 92% percent during a test done for 100 samples. Recommendations for accuracy improvements are included in the conclusions.

1 Introduction

Buildings consume 40% of the total energy used globally and are responsible of 30% of the total CO₂ emissions [1]. In the UK, buildings account for 37% of the total annual greenhouse gas emissions [2]. Research shows that building stock can greatly reduce their energy demand by implementing energy conservation measures (ECMs) [3], [4] and the Intergovernmental Panel on Climate Change (IPCC) report suggests that buildings in the public and commercial sector could achieve an 18% reduction in carbon emissions through no or low cost ECMs [5]. For the particular case of the UK, it has been shown that energy efficient operations, an example of a low-cost ECM, can achieve up to 34% savings in office buildings [6].

In order to identify and implement ECMs, operational data from the studied building is required for detailed analysis, fault detection and diagnostics (FDD) and the creation of building energy models (BEM) that be used for evaluating cost-effective ECMs. Operational data refers to measured time-series data collected from the systems installed in the building. Operational data is usually collected by smart meters (AMR), data-loggers or building management systems (BMS) [7].

With the spread use of automated electricity meters, building management systems (BMS), Internet of things (IoT) sensors, and cloud databases, a new generation of applications becomes available to understand how a building uses energy, and reduce energy use while keeping occupants comfortable. Applications enabled by a complete operational data set include, but are not limited to:

- Predictions of building energy load and peak demands, in particular from heating, ventilation and air conditioning (HVAC) systems and other high energy consuming systems [8].

- Rule mining. Building operation rules can be extracted to determine associations and correlations in data that are not evident for facility managers [9].

- Calibrated building energy models. Examine the relation between energy loads and building components such as walls, windows, lighting, heating, ventilation, etc. to detect potential design and operation problems. Software such as the IES Virtual Environment (IES-VE) [10] and Energy Plus [11] are examples of tools that can be used to calibrate building energy models.

- Savings estimations of installed ECMs, process also known as Measurement and Verification 2.0 [12]–[14]

- Fault detection and preventive maintenance. Operational data can help to detect faults in buildings,

* Corresponding author: adalberto.cabrera@iesve.com

sometimes before they occur. Examples of applications are documented in [15], [16].

Other applications include economic analysis, energy fraud detection and model predictive control [17]. All these applications are enabled only when a complete operational dataset is available, as energy-related data is a basic requirement.

A complete data set should include: outdoor weather variables, indoor comfort and air quality data, HVAC information regarding status of the systems alongside with analogue measurements, and energy data in the form of time series. In table 1 the typical source of these types of data is presented. It is worth noticing that energy data in the form of time-series is often the most challenging to capture from typical buildings in the UK. Hence, a low-cost, non-intrusive alternative to obtain this type of data is presented in this work.

Table 1. Operational data categories and typical sources.

Type of data	Source
Outdoor weather	Local weather station, Third-party weather services, government weather stations
Indoor comfort/ air quality	Building management systems, stand-alone sensors
HVAC	Building management system, short-term measurements
Energy data	Smart meters, short-term measurements

It is important that highlight that most of the building stock in the UK is equipped with non-energy efficient building management systems (or has no BMS at all) [6] meaning that a large proportion of operational data is not available for detailed analysis, FDD and BEMs, which limits the implementation of cost-effective ECMs. Additionally, time-series energy information is not always accessible in near real time due to the lack of smart meters for energy sources (heat, gas, water), security restrictions in Energy Information Systems (EIS) and building management systems (BMS), restrictions and costs associated for automated reporting from utility companies, limitations in the installation of newer meters due to lack of ownership in leased spaces, restrictions due to specific regulations of a country and the fact that meters must comply with safety regulations.

In this paper, we propose the implementation of a computer vision-based reader for analogue energy/water meters running in a low-cost embedded system that can turn building information (e.g. energy meters) into operational data. The data can be later analysed in detail alongside existing operational data, enabling a more comprehensive cost-effective ECMs estimation process. Notice that this technique relies on the pre-existence of meters of any time and relies on the assumption that these are calibrated.

To achieve this goal requires the combined implementation of computer vision in the edge and building data analytics, two very active research topics

which together can facilitate the acquisition of operational data in virtually any building. Improved data privacy can be considered as a side benefit of this approach given that no image/video actually leaves the edge device (i.e. a Raspberry Pi) but only the relevant bits information. Additionally, this solution overcomes the challenge of collecting private high frequency readings from energy suppliers, which is a difficult and sometimes impossible task, due to the current lack of legislation regarding free data access for commercial buildings in the UK [18].

Finally, the proposed approach can be used as short-term (e.g. two continuous weeks) data collection technique with negligible up-front costs to determine if a building is a good candidate for a deeper energy auditing, hence reducing energy efficiency investment risks.

In the following subsection, a brief description of the concept of computer vision in the edge is presented. In section 2, the proposed methodology is described; section 3 presents a case study done in an office building in Scotland and in section 4 we draw conclusions on the current approach and present future work.

1.1 Computer vision in the edge

Computer vision consists of the transformation of image on video data into a new representation with the objective of achieve a goal. The most common type of computer vision applications are classification, detection and segmentation. Classification determines the “label” or “class” an image belongs to, for instance, decide whether a picture contains a “cat” or a “dog” based usually on some sort of probability per possible class. Detection consists in determining if an object appears in certain areas within an image, often detection is deployed using bounding boxes based on a probability threshold. Finally, segmentation classifies each pixel of a picture to determine, for instance, whether a pixel corresponds to a certain material. Computer vision tasks are usually not generalisable, meaning that in most cases a tailor-made solution is required for each specific problem.

OpenCV is an open source computer vision library available from <http://opencv.org>. It was developed in 1999 with the intent of accelerating computer vision research. OpenCV is written in optimised C++, designed for computational efficiency focusing on real-time applications. It contains over 500 functions covering image processing, security, camera calibration and robotics. OpenCV is free and the code can be used in commercial or research applications [19].

In recent times, the concept of Edge AI, or artificial intelligence on the edge, has emerged, to mean placing AI workloads as close to the edge of the system where the data is created. Edge AI enables (nearly) local processing minimising network use, reducing latency and enabling real-time decision making for applications that require it [20]. The use of vision processing units (VPU) and Image processing units (IPUs) are expected to increase performance on small devices (e.g. Raspberry

Pi) to enable faster AI on edge applications. Applications of Edge AI include self-driving cars, robotics, animal tracking, and security video surveillance.

In the context of the current work, Edge AI means that relevant building energy information can be generated continuously from video or photo streams such as current energy consumption. Virtually any energy meter can be read and converted into a string of text that can be then exported to a cloud database. Additional applications of this approach can be developed for other analogue variations that affect energy use in the building such as window status, doors position and occupancy detection without compromising the privacy of building users.

In figure 1, we present an example of a webcam connected to Raspberry Pi where the object detection and classification script is executed, then a single value (0 == Closed, 1 == Opened) alongside a timestamp is securely sent to a cloud database where it can be visualised.



Fig. 1. Example of application of computer vision in the edge, using a script that is able to detect a window in the image and determine whether it is opened or closed (upper part), then the information is turned into time-series data (bottom part).

1.2 Cloud analytics platform: iSCAN

iSCAN (intelligent Control and Analysis) is a cloud-based analytics platform developed by Integrated Environmental Solutions, Ltd [21]. iSCAN allows users to centralise any time-series data from different BMS systems, utility meters, sensors and portable data loggers in one platform. Users can then organise and analyse this data to gain hidden insights to improve building or portfolio operation.

iSCAN includes a graphic user interface (GUI) which is used to set up incoming data, post processing and analysis of the data through flexible plot visualisations, and create alerts to detect data issues and anomalies. iSCAN allows to manage several data sets at the same time (e.g. different type of sensors in different buildings) while maintaining them separate through granular user access permissions.

Once the data has been set up it can be accessed via API, allowing its use in a variety of other applications (e.g. detailed analytics, alarm notifications, building energy modelling, etc.). API access is secured by tokens linked to authorised project users. Figure 2 shows an example of how data can be visualised in iSCAN for two different data streams, also known as channels.

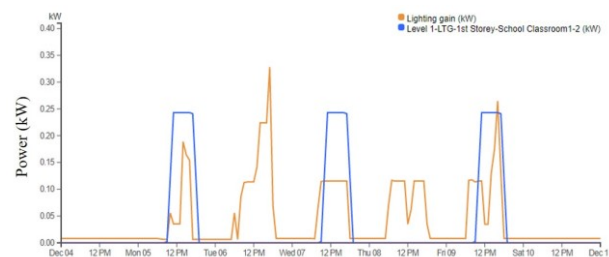


Fig. 2. Example of how data can be visualised in iSCAN for two different data streams, also known as channels.

2 Methodology

The process follows the same approach as the connection to iSCAN of any sensor that is installed in the building. In figure 3, we present a basic diagram explaining the process. The data source is an analogue screen, and conversion to time series requires interpreting a seven-segment display for specific time intervals. An image is captured by a webcam and interpreted by a computer vision algorithm specialised on the specific interpretation task. Then the interpreted value is paired to a time-stamp of the moment of the capture of the image. A JSON file with a SenML structure is created and pushed to iSCAN through the use of an API that requires a user-unique token for security reasons. Once the data is in iSCAN it is pre-processed before it becomes available for display alongside data collected from the other building sensors.

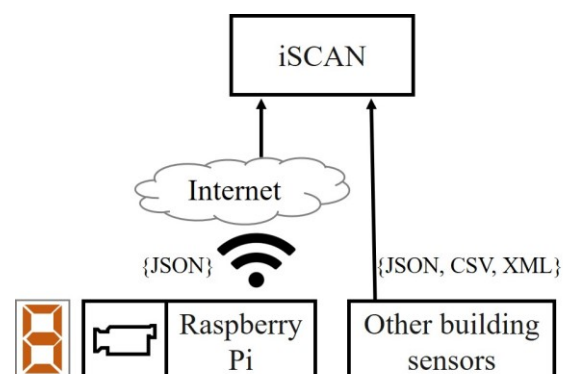


Fig. 3. Basic schematic representation of the proposed solution in the context iSCAN and other building sensors.

In the following subsections, the computer vision script for seven-segment screen digit interpreter, the data export and the pre-processing steps are documented.

2.1. Computer vision in the edge

A 1080P@30fps webcam is connected to a Raspberry Pi 4 equipped with a Broadcom BCM2711 Processor Quad-core A72 and 5GB LPDDR4 SDRAM. We chose this device due to its Wi-Fi connectivity and relatively simple use through the Raspbian OS [22]. The actual hardware implemented is presented in figure 4.



Fig. 4. Raspberry Pi 4 and standard webcams used for the experiment.

A seven-segment screen digit screen reader script has been created for this example. The script is written in Python 3.7 and uses OpenCV libraries both to detect digits and identify them. The process is explained in the following paragraphs.

Image is captured in 600 seconds (10 minutes) intervals and cropped to display only the section of the picture where the meter is located. This process is explicitly indicated in the script. Then cropped picture is rotated automatically by detecting the inclination of the main lines (cv2.HoughLinesP function) of the cropped pictures, figure 5.

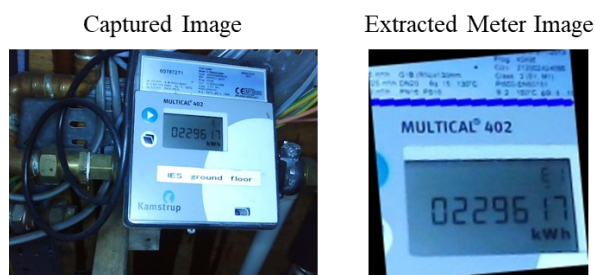


Fig. 5. Sample image cropped and rotated to extract the meter region.

After extracting the meter region from the original acquired image, a series of pre-processing steps are carried out to segment the LCD region. Given the meter image as input, the pre-processing script converts it from RGB to greyscale, then performs the following broad steps on the greyscale meter image.

1. Extracting candidate regions of interest. This involves improving the image contrast (histogram equalisation), followed by some morphological processing to remove the digit pixels and make the LCD region a dark homogeneous region.

Then, a thresholding step (OTSU) separates this dark LCD region (along with other small dark regions) from the background which is bright. The output is a binary image that indicates that indicates the locations of all candidate regions of interest.

2. Extracting the largest region of interest from the candidates. This involves running a connected-component analysis on the binary image from step (1) to generate a label image. Then, the area of each component is calculated; only the largest component is retained in the label image. This label image is the output of this step.
3. Improving the contrast of the digits against the LCD background in the greyscale meter image. This involves a morphological processing step to first correct uneven illumination, followed by a contrast-limited adaptive histogram equalisation (CLAHE) step to improve local contrast. The output of this is a greyscale meter image with improved contrast.
4. Generating an image with only the LCD region in it. This simply involves using the label image output of step (2) as a mask on the greyscale meter image output of (3).

Sample result after this process is presented in figure 6.

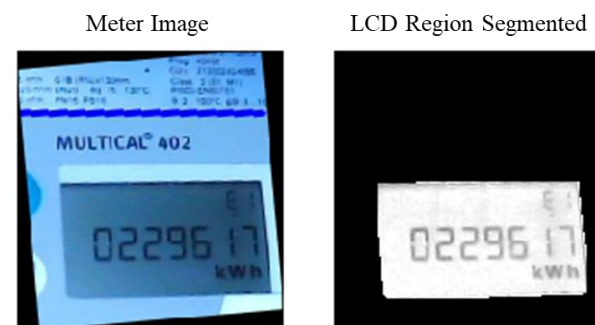


Fig. 6. Meter image pre-processed to segment and enhance the LCD region.

Once the LCD region is segmented, the following broad sequence of steps are performed to recognise and localise digits.

1. For every digit, a “template” image is loaded and slid across the input image to find matches, using the template matching algorithm. This is done using the normalised correlation coefficient as the similarity measure. Matches lower than a specified threshold are discarded. Duplicate matches (i.e., the same digit detected at the same location with slight shifts) are also discarded. This step generates a list of locations in the form of bounding boxes, for every digit.
2. Then, overlaps across digits are checked (e.g., a “9” in the input image would match the template for 9, but it would also match the template for 3 because of the same segments

lighting up). The digit for which the similarity measure is the highest is retained, and the others are discarded. This results in a list of locations in the form of bounding boxes, for every digit, with each location being unique.

Finally, a sorting step sorts the list of bounding boxes so that they are arranged in the left-to-right sequence. From this sorted list, the final reading is generated as shown in figure 7.

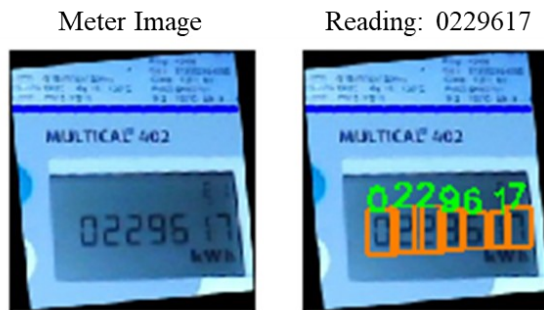


Fig. 7. Reading generated after digit recognition and localisation.

2.2 Export values data cloud platform (iSCAN)

Once the value of the meter has been extracted from the image, a second script is used to post-process and to upload the data in the form of SenML. Given that the raw data consist on energy usage in kWh, the post processing consists on calculating the positive difference between sample values divided by their time difference (e.g. 10 minutes) and the rate is converted to the building sample time (i.e. the sample time for which all building data is being displayed, e.g. 30 minutes). Then, calculated rates are averaged and converted to units per hour. It means that the script calculates the average rate of change over one hour, so the final units displayed are kW even if the source data is kWh.

An illustrative example is presented in figure 8, the line represents rate between values in units per quarter hour. The contribution of the value is the gradient of the orange line. If the right-hand point's value is left than the left-hand point's value, then it is treated as the counter having been reset to zero. Hence, the second value (7, 4) has a positive difference of 4 compared to the first value. The difference in time is 6 minutes. 4 units / 6 minutes is equal 10 units / 15 minutes, hence the first bar has a height of 10. The average height of the bars in the first quarter of the hour is 12 (10 and 14), hence the average rate is 12 units / 15 minutes or 48 units per hour. 48 is the value that is visualised.

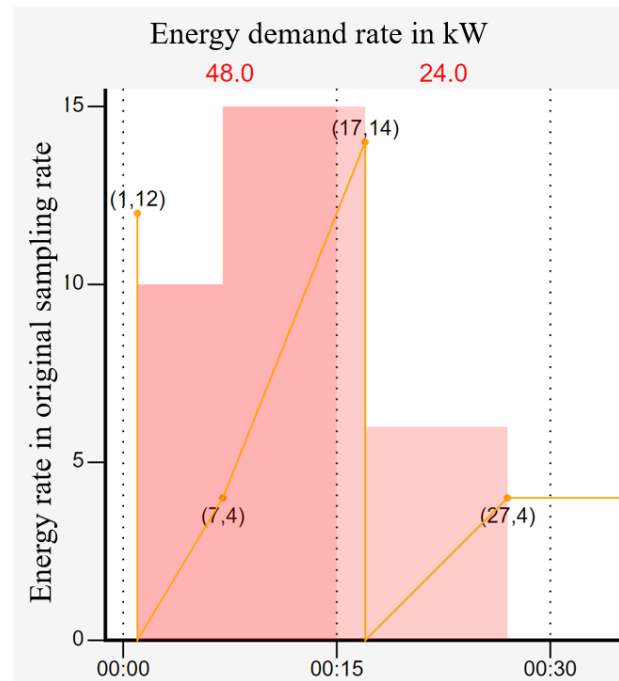


Fig. 8. Energy data (cumulative) into power (gradient) data example.

Notice that while this calculation is ideal for heat meters where peak instantaneous demand is limited by the boiler capacity, bespoke post processing might be required for other types of measurements.

Data is exported in the form of time-series data following a Sensor Measurement Lists (SenML) structure in JSON (JavaScript Object Notation) file format [23]. For authentication, the script requires a project URL, a building name and a token linked to a user that has access to the project. The upload script requires the use of a library called “scan_api” which has been developed to enable other python libraries to interact to other applications such as computer vision and data analytics. The relevant part of the script used to upload the values is presented below. Notice that meter value() is a function that returns a post processed value in kW.

```
1. while True:
2.     data = {
3.         'bt': dt.datetime.now().times
         tamp(),# timestamp
4.         'n': 'heatingGroundFloor', #
         unique ImportReference
5.         'v': processed_value() # (Ret
         urns post processed power value)
6.     }
7.     print(data)
8.     send_data(ds,data)
9.     time.sleep(600) # wait for 10 minutes
```

3 Case study: Office building

Integrated Environmental Solutions LTD (IES) headquarters in Glasgow UK is used as case study. The Helix building is an office type of building with a floor

area of 2900m². It was constructed post 2000s and has been historically used by two companies. The building has natural ventilation and heating is provided and a biomass (main) and natural gas (back up) boiler. The building can host around 180 people. A picture of the building is presented in figure 9. Additionally, main areas of the studied building are specified in figure 10.



Fig. 9. The Helix building. An office type of building located in Glasgow, Scotland used as case study.

The building is controlled by two thermostats. The current temperature is 24, with a night setback of 15 C for both of them (boiler is actually never switched off). Currently, the building has 14 indoor environmental sensors at desk level plus 5 ceiling-level additional sensors. Information is polled with a 5-minute resolution every 10 minutes. Additional measurements include relative humidity, CO₂ concentration in the indoor air, motion and lighting levels (lux). A weather station, located on the rooftop of the building, senses data including dry-bulb temperature, relative humidity, and solar radiation.

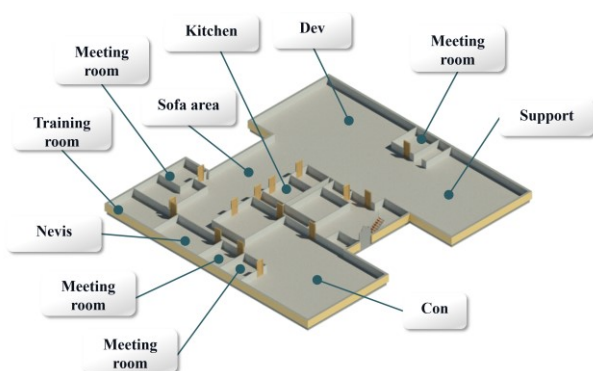


Fig. 10. Main areas of the office building used as case study, each of these areas is equipped with indoor environmental sensors.

With regards to energy information, electricity is sub metered using live current meters; two heat meters are installed, one for each floor. The heat meter values are currently manually recorded in a spreadsheet

alongside with photographic evidence on a monthly basis.

In this case study, a camera was mounted on a nearby structure as well as the raspberry pi. The camera was set up in order to point to the ground floor heat meter with a sampling interval of 10 minutes. The Raspberry was connected to the local Wi-Fi network so it was possible to upload directly the values directly to iSCAN. The arrangement of the camera near the heat meter is presented in figure 11.



Fig. 11. Heat meter arrangement in the Helix building.

A 17-hour period, consisting on 100 readings with 10-minute intervals, was used as test. For this example, observations done during a week indicated that the maximum value change during a 10-minute interval, cannot be higher than 7, meaning that the outermost right digit (the one representing the unit) is the only relevant value. The accuracy was defined as the number of instances where the last digit was read correctly divided by the total number of samples. Hence, the accuracy metric represents the ability of the script to accurately derive the rate of change between readings, which resulted on a 92% percent during a test done for 100 samples. It means that 92 out of 100 values were predicted correctly. Actions such as improved lighting and more representative digit templates are expected to increase the accuracy. The script has yet to be tested in multiple lighting conditions and camera positions, however the implemented automated thresholding is a step toward this goal.

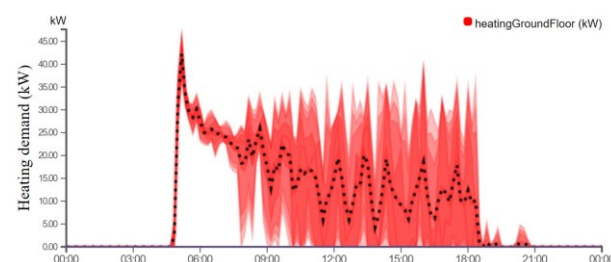


Fig. 12. Heating demand during weekdays (Mon to Fri) in the Helix building using an overlay plot in iSCAN. The black dotted line represents the mean value over the all the selected days.

Once data is uploaded, it can be plotted for its analysis detailed in iSCAN. Figure 12 is an example of an overlay plot that shows the use trends over a 5-day period during workdays (Monday to Friday).

Additionally, heating data can be compared directly with indoor air temperature values, as presented in figure 13. It can be noticed that there is a correlation between the heat demand and the increase in temperature. Also, it can be seen that the cool-off rate varies for some of the rooms.

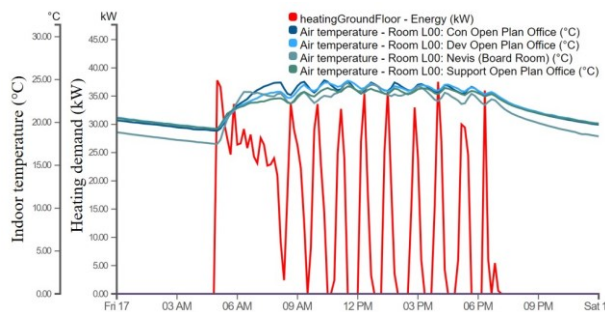


Fig. 13. Heat demand during a specific day plotted alongside air temperature data in various areas in the Helix building.

Finally, daily heating demand can be also calculated. It can be seen that the building consumes around 224 kWh per day and that this demand has an inverse correlation with the mean external air temperature, as expected, figure 14.

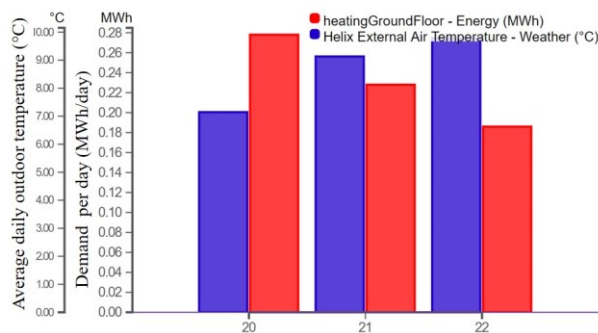


Fig. 14. Total heating demand alongside average outdoor temperature for three days in the Helix building.

3 Conclusions and future work

In this paper we propose the implementation of a computer vision-based reader for analogue energy/water meters, via a low-cost embedded system that can turn analogue building information (e.g. analogue meters) into operational data. Benefits of this approach include increased privacy, reduced risk for energy efficiency investments, and access to otherwise unavailable data. A case study in an office building in Scotland is presented. The accuracy of the method, i.e. the ability of the script to accurately derive the rate of change between readings, resulted on a 92% percent during a test done for 100 samples. Actions such as improved lighting and more representative digit templates are expected to increase

the accuracy. The script has yet to be tested in multiple lighting conditions and camera positions.

This paper has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 847053.

References

1. M. W. Ahmad, M. Mourshed, D. Mundow, M. Sisinni, and Y. Rezgui, "Building energy metering and environmental monitoring - A state-of-the-art review and directions for future research," *Energy and Buildings*. 2016.
2. Committee on Climate Change, "Progress reducing emissions from buildings," in *Meeting Carbon Budgets – 2014 Progress Report to Parliament*, 2013.
3. C. Change, *Building a low-carbon economy – the UK 's contribution to tackling climate change*. 2008.
4. CIBSE GUIDE F, "CIBSE Guide F: Energy efficiency in buildings," *Energy Effic. Build. Chart. Inst. Build. Serv. Eng. London*, 2nd Ed., 2004.
5. IPCC, *Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. 2014.
6. A. Lewry, "Energy management and building controls," 2016.
7. Electrical and Mechanical Services Department, "Technical Guidelines on Retro-commissioning," 2018.
8. Z. Li, "An empirical study of knowledge discovery on daily electrical peak load using decision tree," in *Advanced Materials Research*, 2012.
9. Z. Yu, F. Haghighat, B. C. M. Fung, and L. Zhou, "A novel methodology for knowledge discovery through mining associations between building operational data," *Energy Build.*, 2012.
10. IES, "Integrated Environmental Solutions Limited," *Main website*, 2018. .
11. D. B. Crawley, C. O. Pedersen, L. K. Lawrie, and F. C. Winkelmann, "Energy plus: Energy simulation program," *ASHRAE J.*, 2000.
12. J. Grandersona, S. Touzani, S. Fernandes, and C. Taylor, "Application of automated measurement and verification to utility energy efficiency program data," *Energy Build.*, 2017.
13. S. Touzani, B. Ravache, E. Crowe, and J. Granderson, "Statistical change detection of building energy consumption: Applications to savings estimation," *Energy Build.*, vol. 185, pp. 123–136, 2019.
14. C. V. Gallagher, K. Leahy, P. O'Donovan, K.

- Bruton, and D. T. J. O'Sullivan, "Development and application of a machine learning supported methodology for measurement and verification (M&V) 2.0," *Energy Build.*, 2018.
15. D. Chakraborty and H. Elzarka, "Early detection of faults in HVAC systems using an XGBoost model with a dynamic threshold," *Energy Build.*, 2019.
 16. A. Capozzoli, F. Lauro, and I. Khan, "Fault detection analysis using data mining techniques for a cluster of smart office buildings," *Expert Syst. Appl.*, 2015.
 17. M. Molina-Solana, M. Ros, M. D. Ruiz, J. Gómez-Romero, and M. J. Martín-Bautista, "Data science for building energy management: A review," *Renewable and Sustainable Energy Reviews*. 2017.
 18. Department for Business Energy & Industrial Strategy, "SMART METERING IMPLEMENTATION PROGRAMME Review of the Data Access and Privacy Framework," no. November, 2018.
 19. G. Bradski and A. Kaehler, *Learning OpenCV*. 2008.
 20. Y. Ai, M. Peng, and K. Zhang, "Edge computing technologies for Internet of Things: a primer," *Digit. Commun. Networks*, 2018.
 21. Integrated Environmental Solutions Limited, "iSCAN: A Single Platform to Optimise the Operational Performance of Buildings," 2019. [Online]. Available: <https://www.iesve.com/icl/iscan>.
 22. A. Kurniawan and A. Kurniawan, "Introduction to Raspberry Pi," in *Raspbian OS Programming with the Raspberry Pi*, 2019.
 23. C. Jeanings, Z. Shelby, J. Arkko, A. Keranen, and C. Bormann, "Sensor Measurement Lists (SenML)," 2018.