

## METHOD FOR REAL TIME FACE RECOGNITION APPLICATION IN UNMANNED AERIAL VEHICLES

Romualdas JUREVIČIUS<sup>1</sup>, Nikolaj GORANIN<sup>1</sup>, Justinas JANULEVIČIUS<sup>1</sup>, Justas NUGARAS<sup>2</sup>,  
Ivan SUZDALEV<sup>2\*</sup>, Aleksandr LAPUSINSKIJ<sup>2</sup>

<sup>1</sup>*Faculty of Fundamental Sciences, Vilnius Gediminas Technical University, Vilnius, Lithuania*

<sup>2</sup>*Antanas Gustaitis Aviation Institute, Vilnius Gediminas Technical University, Vilnius, Lithuania*

Received 14 June 2019; accepted 19 June 2019

**Abstract.** Newly evolving threats to public safety and security, related to attacks in public spaces, are catching the attention of both law enforcement and the general public. Such threats range from the emotional misbehaviour of sports fans in sports venues to well-planned terrorist attacks. Moreover, tools are needed to assist in the search for wanted persons. Static solutions, such as closed circuit television (CCTV), exist, but there is a need for a highly-portable, on-demand solution. Unmanned aerial vehicles (UAVs) have evolved drastically over the past decade. Developments are observed not only with regards to flight mechanisms and extended flight times but also in the imaging and image stabilization capabilities. Although different methods for facial recognition have existed for some time, dealing with imaging from a moving source to detect the faces in the crowd and compare them to an existing face database is a scientific problem that requires a complex solution. This paper deals with real-time face recognition in the crowd using unmanned aerial vehicles. Face recognition was performed using OpenCV and Dlib libraries.

**Keywords:** unmanned aerial vehicle, drone, face recognition, real-time analysis, monitoring.

### Introduction

The potential of using unmanned aerial vehicles (UAVs) as a means to improve security measures is obvious: mobility and agility combined with the stabilization and data transfer of existing technology makes for the optimal solution in situations where previously much more complex and heavy equipment would have been required. Recent technological developments have allowed previously unavailable technological compounds, resulting in a breakthrough of new technologies. One of which is UAVs used as a tool for real-time face recognition in the crowd.

Naturally, collecting such information faces ethical issues, but, if aligned with ethical norms and legislation, it can be highly beneficial with the least damage to privacy and other issues. Security challenges described in (He, Chan & Guizani, 2017) suggest that, although public safety plays an important role in maintaining a stable and secure environment in society, facilitating UAVs for this cause raises new security issues to be considered. This paper deals with real-time face recognition in the crowd using unmanned aerial vehicles.

### 1. Previous research

The idea of face recognition has a well-developed scientific basis. Although early research examples date back to as early as the 1970s, the main research focus on face recognition dates back to the 1990s. It includes solutions to a series of complex problems: human face detection in a complex background (Yang & Huang, 1994), training support vector machines for face detection (Osuna, Freund & Girosit, 1997), and a neural network-based face detection approach (Rowley, Baluja & Kanade, 1998). A systematic approach to understanding the methods for face recognition is provided by (Hjelmas & Low, 2001). The overview provides a clear perspective on the available options that include feature-based approaches, including low-level analysis, feature analysis, and active shape models, and image-based approaches that are based on linear subspace methods, neural networks and statistical approaches.

With the development of face detection algorithms and the advent of affordable mass-produced digital imaging equipment that could be used in daily tasks, it became obvious that data captured by CCTV cameras would be much more useful if certain attributes of the information

\*Corresponding author. E-mail: [ivan.suzdalev@vgtu.lt](mailto:ivan.suzdalev@vgtu.lt)

could be added to the footage automatically as meta-data. This idea is supported by approaches found in the patent (U.S. Patent No. 15/486,109, 2018) and research papers (Panta, Roman-Jimenez & Sedes, 2018). In this sense, a new technological opportunity arises – it is possible to store semantic information rather than raw data.

If the ability for face detection is combined with a more portable device, enabling the positioning of the imaging equipment to an exact location, where the monitoring is required, new, highly agile technology can be conceived. The latest developments in unmanned aerial vehicles offer exactly that – they are very portable, can be positioned virtually anywhere in a three-dimensional space and have phenomenal stability required for sufficient-quality imaging. Some recent application examples are photogrammetry of crime scenes (Edelman & Aalders, 2018) and personal security systems using drones (U.S. Patent No. 15/051,166, 2017).

## 2. The architecture of the system

In light of the circumstances presented in the introduction of this paper, the aim of this research is to provide a method for real-time face recognition for UAVs. To achieve this goal and to build the system, the following issues are to be resolved in this research study.

1. Aerial imaging – focusing on flight positioning adapted to face recognition requirements. Skewing or distorting the graphical information of the face reduces the accuracy of face detection and identification abilities. Flight positioning is investigated with regards to the following parameters: speed and the horizontal and vertical angles between the face and the camera. The influence of the vertical and horizontal angles is investigated as presented in Figure 1.

2. Media transmission – to ensure real-time or very close to real-time analysis, the media acquired by the UAV has to be efficiently transmitted with sufficient quality.

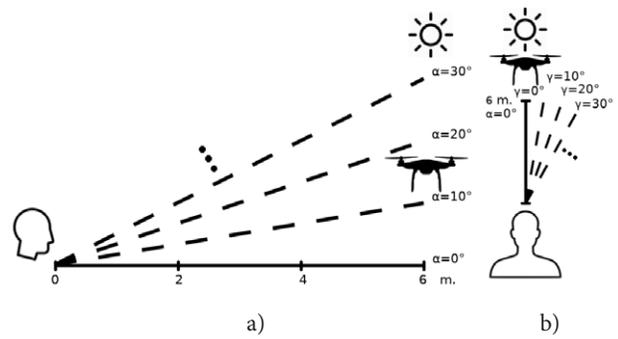


Figure 1. The model for assessment of the influence of the a) vertical and b) horizontal angle between the face and the camera

Media transmission is one of the key components to the successful operation of the system, as it has a direct influence on the timeliness and quality of the media used for face detection and identification

3. Face detection – a method to recognize facial features and identify objects as faces within the required environment. In ideal conditions (e.g. white background and one face looking forward), the issue seems non-significant, but, when it comes to crowds and a lot of background noise, underexposure or overexposure of images, a robust method has to be used for face detection;

4. Face identification – after the detection phase, the face information has to be transformed in order to correctly identify a person. This means that skewness, tilt and shifting have to be corrected, lighting conditions optimized and certain portions of graphical information filtered to initiate the identification.

The proposed architecture of the system is built using readily available components combined with custom solutions, required for this application. The conceptual architectural layout is presented in Figure 2.

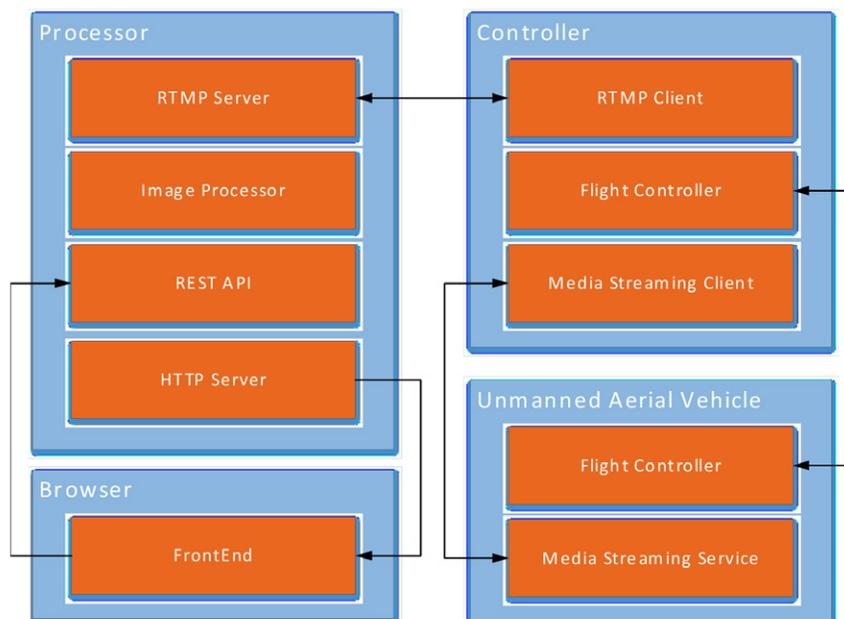


Figure 2. The proposed architecture of the system

## 2. The technical realization of the system

For the technical realization, a market available solution – a quadcopter unmanned aerial vehicle – has been used as the source for imaging. Recent developments in the product ensure that the imaging provides the best stability (Patent No. US 2014/0037278 A1, 2014) and flight positioning (Patent No. US 2017/0123425 A1, 2017).

A Raspberry Pi3 Model B (Raspberry Pi Foundation, 2018a) microcomputer with an Raspberry Pi cam v2 camera was used for the filming and media transmission (Raspberry Pi Foundation, 2018b). The main characteristics of the Raspberry pi3 microcomputer are quad-core 1.2 GHz 64-bit ARM A53 processor, 1 GB LPDDR2-900 RAM, supported Wi-Fi 802.11n network, and integrated VideoCore 4 graphics processor used for image processing. The camera module used supports resolutions up to 4k; however, due to technical capabilities, the microcomputer was able to film and transmit media at a maximum resolution of 1080p at 30 frames per second. During trialing the media was transmitted from the Raspberry Pi system to a processing computer via Wi-Fi using the RTMP and applying for the FFmpeg programme. It is notable that the transmission of media via a network using the Raspberry Pi microcomputer results in delays in the system operation, the transmitted media is delayed by 3–4 s. Such delays are permissible in the processing chain, however, such a view would not be reliable for vehicle control.

In the system, the OpenCV (OpenCV, 2018a) package was used for image processing. The main functions needed for this system are image size adjustment, cutting and changing colour. OpenCV denotes images as multidimensional matrixes. The main object describing an image is Mat. It includes the information on image size, method of storage and a reference to the pixel matrix. The pixel matrix can be multi-layered, with the number and value of layers determined by the method of storage. Therefore, in essence, the OpenCV frame object is close to raw data and is not compressed. Each processed frame has to be converted into a Mat object. Since face recognition is the core of this system, it is a critical step.

The recognition was carried out in five stages (Figure 3).

Before identifying a face, its place in the image had to be detected. Face detectors were used to identify the face location. Without detecting the faces in the image, it was impossible to use them for recognition. It was also important to detect as many faces in the crowd as possible. Additionally, during this step, it was key to minimise the false-positive responses, cases when an object is erroneously detected as a face. The main criteria for the face detection algorithm used for this system were speed and accuracy.

Since a high-resolution image was used, a high number of faces in the crowd was detected, and, as a result, the speed of the algorithm was very important. There were many additional details present in the filmed media, so it was necessary for the algorithm to not detect false responses, as this would lead to a superfluous comparison of faces and calculation of results. The simplest Viola-Jones algorithm (Wang, 2014) was not applicable to the system, because, despite being very fast, it is not accurate enough and is very susceptible to disturbance.

The OpenCV library, used for image processing, employs two different face detection algorithms: one is based on Haar cascades (OpenCV, 2018b), and the other on Local binary patterns (LBP) cvlbp. The algorithm based on Haar cascades is similar to the classic Viola-Jones but has more layers of features, thus inheriting the same problems. The LBP based algorithm is faster but is also less accurate. A good speed/accuracy ration is shown by face detector dlib, found in the Dlib package. Dlib uses a classic The histogram of oriented gradients (HOG) based detector connected to a direct classifier, an image pyramid, using the moving frame detection scheme. To compare the Dlib detector with OpenCV detectors, the characteristics of the former are better, as it captures fewer false-positive faces (Dlib, 2018a). The face detection algorithm used by Dlib has sufficient speed and accuracy; therefore, it was used in the system that was designed.

Once the face was detected in an image, it was oriented during the next step. Orienting the faces minimised the impact of face position variation on face identification. Since faces in a crowd can be oriented in many directions, this step is essential and allows obtaining better identification results. The Dlib library has a trained model which detects 68 semantic face points. On the basis of these points, it is possible to orient the face using affine transformations. The model is quite fast, and, as such, will not cause great delays in the system and will improve the face recognition process. Finally, a face recognition module was required. The main available open-source packages carrying out face recognition algorithms are OpenFace (OpenFace, 2019) and Dlib (Dlib, 2018b). The Dlib package was used in the system since the OpenFace package is not updated regularly and uses many of the functions present in the Dlib package.

The system applies a trained resnet\_v1 model (ResNet, 2019). The Rasnet model expresses facial features as a 128-dimension vector, and faces are compared using a vector difference. This model has an accuracy of 99.38, which is sufficient for the system application aims. Based on the system requirements and elements described previously, the system architecture was designed. The me-

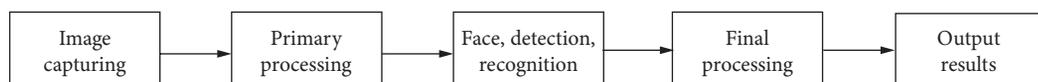


Figure 3. Stages of face recognition

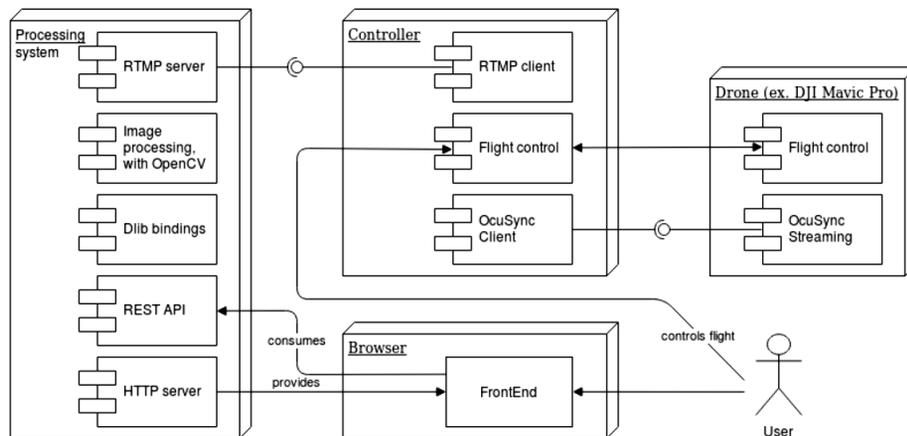


Figure 4. Diagram of the experimental system

dia stream, required for UAV control, was streamed to the control panel, using a DJI Mavic Pro UAV; media data was transmitted via the OcuSync system (OcuSync, 2018). Media data from the operator control panel was re-transmitted to the RTMP server (Figure 4). The main processing system was connected to the UAV into a single network via Wi-Fi or wired link depending on the UAV system that was employed. The main processing system consisted of the RTMP server module, image processing module, Dlib bindings, REST API interface and an HTTP server module.

Face detection, primary processing and identification operations were carried out in the server. The images were compared with the available images of human faces. This database consist of 13233 faces. Data was kept in the SQL database PostgreSQL. The OpenCV was used for image processing in the system server, whereas machine learning algorithms were applied using the Dlib library. The operations requiring the most resources from the system were transferred to the server, as a result, it worked at quite a sufficient speed. Had the images been processed in the UAV, firstly, there would not have been enough computational resources; secondly, the energy reserve necessary for the flight would be quickly depleted. Current data transfer technologies allow transferring image processing, without introducing major delays into the system.

Go programming language was applied for the back-end part of the system (Go, 2019). Programmes written in Go language are compilable. As a result, they are very fast and have appropriate libraries for network request control, so do not need a separate request control server (Nginx or Apache). In the server, the programme was connected with other elements of the system using the per REST JSON interface. A web-based programme was used for the user interface. It was written in Javascript using Vue.js library (VueJs, 2019). This one-page programme interacted with the server via a REST interface. The web based programme was the main tool of the operator working with the system.

### 3. System testing

During the first test, the dependency of Euclidean distance on aircraft filming distance and vertical angle was determined (Figure 5).

It is evident from the linear regression curve that the vertical angle does not have a significant impact on face recognition. This is due to the orientation of faces before transferring them to the recognition algorithm. The chain orienting mechanism works efficiently and can compensate for changes due to face orientation. Conversely, face detection posed much greater problems.

When the unmanned vehicle climbed higher than at a 30-degree angle, the face detection system was not able to detect the location of faces. After assessing the impact of flight distance, it is evident that a shorter distance has a positive impact on recognition, and the Euclidean distance is shorter when filming is done closer. Consequently, to obtain the best recognition results, it is necessary to either film from as close as possible or to use lenses with greater focal length, which, however, were not available and could not be employed by the UAV used for testing.

That the line orienting step was successful was proven by the results of a trial during which the impact of horizontal deviation on the Euclidean distance between the

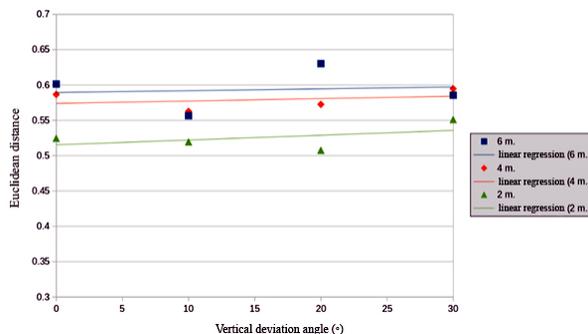


Figure 5. Euclidean distance at a vertical deviation angle

compared and filmed face images was assessed. The results of this test are illustrated in Figure 6.

As shown by the linear regression curve, horizontal deviation did not significantly distort the face and the Euclidean distance remained stable.

It has to be stressed that once a 35-degree angle is passed, the face system is unable to detect faces; also, past this value, it was not possible to determine the deviation. Faces were not detected successfully using the chosen HOG algorithm if the deviation from the face was greater than 35 degrees horizontally, or greater than 30 degrees vertically from normal of face. This means that to obtain the optimal performance of the system, the UAV must be positioned within these minimums. Up to these deviation values, the face is successfully oriented using the orienting algorithm.

The findings of the impact of illumination on face recognition show that critical illumination conditions, both high and low, cause notable difficulties for the recognition process. If the illumination was too high, the images were affected by shadows, which in turn affected the facial expression output obtained via the recognition algorithm. With illumination too low, the UAV’s camera was not able to adapt when correcting the lens, sensitivity and exposure. The image would become too dark, making it much more difficult to identify images in the processing chain. Faces were not detected in dark images. It is worth noting that, with the illumination being too low, the UAV used for testing becomes unstable, its optical position control sensors stop working, and it is dangerous to use it under such conditions.

The test to determine the dependence of system delays on aircraft distance yielded results that were expected – delays increase with increasing UAV distance. When measuring delays, it was noticed that as the delays got longer when the UAV moved further away, more interference was observed in the stream. Interference damaged part of the frames so badly that they could not be used for recognition. Finally, the system was tested for its main purpose – identifying people in a crowd. These tests were done with a higher number of human faces, thus simulating a crowd. The crowd testing showed that the impact of the number of faces on system speed is not significant.

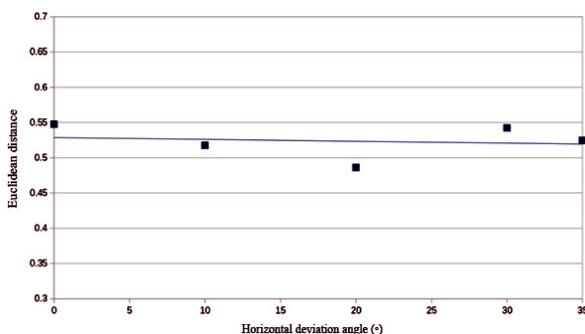


Figure 6. Euclidean distance at a horizontal deviation angle

During the tests, the number of frames per second remains stable, at 11 frames per second, with every 10th frame used for recognition. This is possible since face detection, the stage that requires the most resources and takes the longest, does not depend on the number of images. In average, face detection, while processing a 1920×1080 px frame, took around 320 ms, the orientation of one face – 1–2 ms, computation of its expression vectors – 60 ms, and the computation of the Euclidean distance – 1–2  $\mu$ s. The speed of the system would be affected more significantly if the faces were processed more often or the number of them was extremely high – 30–50 faces and more. During the last test, the optimal value of the weighted Euclidean distance was assessed, in order to successfully categorise the people in the test crowd. In the crowd, 12 people were detected, half of whom were denoted as suspects. It must be stressed that the weighted value test is rather subjective under the given conditions and the persons in the image; however, it still allows making a preliminary assessment of its optimal value. The findings of this test are provided in Figure 7. As indicated by the results, the optimal threshold value is within the limits of 0.5–0.6.

In the case of many similar faces and when a specific person has to be identified, the threshold value can be lowered in order to minimise the allowable difference among faces. If the threshold value is too low, the recognition becomes problematic, since the face in the image will have to be very similar, and, with changing conditions concerning face orientation, illumination, etc., it is very unlikely. Increasing the threshold value poses a danger of merging with another person’s face, so the threshold value should not be increased too much. It has been determined that the system operates best when the UAV’s vertical and horizontal deviation angles do not exceed 30 degrees from the face. The face orientation algorithm is able to compensate deviations within these limits successfully. Also, for the system to work well, the average illumination has to be within 100 to 5000 lx.

If the illumination is greater, system operation is affected by shadows and disturbances, and if it is lower – the face contrast decreases, it becomes too dark. It is necessary to maintain the UAV as close to the operator as possible if the delay time is important. When the UAV moves

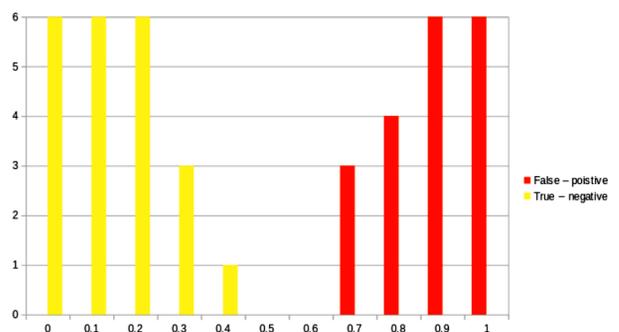


Figure 7. True-negative and False-positive according to the Euclidean distance

further away, delays increase. However, if delays of up to several seconds are acceptable, it can fly even 500 m from the operator. For face categorisation in the crowd with the system, it is most efficient to use 0.5–0.6 Euclidean distance threshold values (Figure 7). If needed, the threshold values can be lowered when more false - positive faces are detected, especially when the faces in the frame belong to persons who are related.

## Conclusions

1. The designed system was tested and it was determined that it can successfully recognise persons in a crowd.
2. The UAV flight conditions ensuring the most optimal operation of the system have been determined. The system operates at its best when the UAV vertical and horizontal deviation angles from the face do not exceed 30 degrees.
3. The face orienting algorithm is able to successfully compensate deviations within these limits. The most optimal operation of the system is achieved when the average illumination is within 100–5000 lx. To maintain the least delay in the streaming, a short distance should be maintained from the operator to the UAV, as increased distance from the operator results in greater delays, which range from 0.5 s with the UAV close to the operator to 2.2 s at a distance of 500 m.
4. Threshold values of the Euclidean distance at 0.5–0.6 should be used for face categorisation in the crowd using the system, thus ensuring the lowest number of false-positive and true-negative detections.
5. Good optics and/or cameras with a good resolution have to be used for face recognition, otherwise, the UAV with the recognition system has to be low and close to the object of interest.

## References

- Buyse, M., Blavier, M., Buyse, M. & Marchand, M. (2017). *U.S. Patent No. 15/051,166*.
- Dlib. (2018a). *Dlib 18.6 released: Make your own object detector*. Retrieved from <http://blog.dlib.net/2014/02/dlib-186-released-make-your-own-object.html>
- Dlib. (2018b). *Modern C++ toolkit containing machine learning algorithms*. Retrieved from <http://dlib.net/>
- Edelman, G. J., & Aalders, M. C. (2018). Photogrammetry using visible, infrared, hyperspectral and thermal imaging of crime scenes. *Forensic Science International*, 292, 181-189. <https://doi.org/10.1016/j.forsciint.2018.09.025>
- Go. (2019). The Go Programming Language. Retrieved from <https://golang.org/>
- He, D., Chan, S., & Guizani, M. (2017). Drone-assisted public safety networks: The security aspect. *IEEE Communications Magazine*, 55(8), 218-223. <https://doi.org/10.1109/MCOM.2017.1600799CM>
- Hjelmas, E. & Low, B. K. (2001). Face detection: A survey. *Computer Vision and Image Understanding*, 83, 236-274. <https://doi.org/10.1006/cviu.2001.0921>
- Lee, S., & Kim, T. (2018). *U.S. Patent No. 15/486,109*. <https://doi.org/10.1016/j.cca.2018.07.032>
- OcuSync. (2018). OcuSync DJI OcuSync Air System. Retrieved from <https://store.dji.com/product/dji-ocusync-air-system>
- OpenCV. (2018a). Open Source Computer Vision Library, OpenCV team. Retrieved from <https://opencv.org/>
- OpenCV. (2018b). OpenCV Cascade Classifier. Retrieved from [https://docs.opencv.org/3.4/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html)
- OpenFace. (2019). OpenFace: Free and open source face recognition with deep neural networks. Retrieved from <https://cmusatyalab.github.io/openface/>
- Osuna, E., Freund, R. & Girosit, F. (1997). Training support vector machines: an application to face detection. *Computer Vision and Pattern Recognition* (pp. 130-136). IEEE.
- Panta, F. J., Roman-Jimenez, G., & Sedes, F. (2018). Modelling metadata of CCTV systems and Indoor Location Sensors for automatic filtering of relevant video content. In *2018 12th International Conference on Research Challenges in Information Science* (pp. 1-9). IEEE. <https://doi.org/10.1109/RChIS.2018.8406677>
- Raspberry Pi Foundation. (2018a). *Raspberry Pi3 Model B*. Retrieved from <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- Raspberry Pi Foundation. (2018b). *Camera module v2*. Retrieved from <https://www.raspberrypi.org/products/camera-module-v2/>
- ResNet. (2019). *Understanding and implementing architectures of ResNet and ResNeXt for state-of-the-art image classification: From Microsoft to Facebook*. Retrieved from <https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624>
- Rowley, H. A., Baluja, S., & Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1), 23-38. <https://doi.org/10.1109/34.655647>
- VueJs. (2019). VueJs: The Progressive JavaScript Framework. Retrieved from <https://vuejs.org/>
- Wang, T. (2014). *Patent No. US 2014/0037278 A1*. <https://doi.org/10.1016/j.trprot.2014.09.001>
- Wang, Y.-Q. (2014). An analysis of the Viola-Jones face detection algorithm. *Image Processing on Line*, 4, 128-148. <https://doi.org/10.5201/ipol.2014.104>
- Yang, G., & Huang, T. (1994). Human face detection in a complex background. *Pattern Recognition*, 27(1), 53-63. [https://doi.org/10.1016/0031-3203\(94\)90017-5](https://doi.org/10.1016/0031-3203(94)90017-5)
- Zhao, C. (2017). *Patent No. US 2017/0123425 A1*.