`

CHAPTER II
LITERATURE REVIEW

2.1 Literature Review

Based on research done by Amos, Ludwiczuk, & Satyanarayanan (2016) titled **"OpenFace: A General-Purpose Face Recognition Library with Mobile Applications"**, they have successfully created a face recognition library that is capable of detecting faces with a ~97% accuracy when compared against the Labelled Faces of the Wild (LFW) database, which according to Parkhi et al. (2015) "contains 13,233 images with 5,749 identities, and is the standard benchmark for automatic face verification". Even though OpenFace's training dataset is one order of magnitude smaller than what Facebook's DeepFace used; OpenFace can still competitively compete with DeepFace in terms of accuracy.

Based on an article by Patel, Kumar, Garg, & Kumar (2018) titled **"Face Recognition Based Smart Attendance System Using IOT"** by using open source libraries available online, and self-built hardware serving as input; they were able to create a fully working face recognition system that is not only accurate in detecting faces; but is also capable of generating an attendance list in the form of an Excel spreadsheet. The data used to generate the attendance list was prepared beforehand; stored in the form of a database. The data includes a picture of the student's face; along with the full name of the student; and contact information.

According to an article by Hu et al. (2015) titled **"When Face Recognition Meets with Deep Learning: an Evaluation of Convolutional Neural Networks for Face Recognition"** by using deep learning, specifically convoluted neural networks (CNN) and also other variants of CNNs; you can create a working face recognition system. The article also mentions ways to further improve the face recognition systems by using feature normalization, and dimensionality reduction.

One of the main concerns when implementing a real-time accurate face recognition is the high computational costs that you need, But thanks to work by Bolotnikova, Demirel, & Anbarjafari (2017) titled **"Real-time Ensemble Based Face Recognition System for NAO Humanoids Using Local Binary Pattern"** they achieved a low computational cost face recognition system by first training the algorithm; and by straying away from conventional methods that is usually used.

According to Anagnoste (2017), the top RPA vendors in 2017 is Automation Anywhere which is an RPA vendor from USA, followed by UiPath, and Blueprism. Out of all the three RPA vendors mentioned in the previous statement; none of them feature a fully functional face recognition system, and as discussed by Laurent, Chollet, & Herzberg (2015) in their article titled **"Intelligent Automation Entering the Business World"** one of the key points of intelligent automation is computer vision, and they specifically mention face recognition; and its application in identifying people in photographs.

Table 1 below is showing a short description of the literature works of other people that the writers took inspiration from:

Table 1 Literature review conclusion, and remarks

Author	**Year**	**Title**	**Remarks**
Patel, Kumar, Garg & Kumar	2018	Face Recognition Based Smart Attendance System Using IOT	A face recognition built with little to no costs using open source tools, and libraries. The work done by the authors also proves that common web cameras can be used to capture input reliably.
Bolotnikova, Demirel, & Anbarjafari	2017	Real-time ensemble based face recognition system for NAO humanoids using local binary pattern	The work done by the authors prove that a low computational cost face recognition system is possible.
Amos, Ludwiczuk & Satyanarayanan	2016	OpenFace: A General Purpose Face Recognition Library with Mobile Applications	A general purpose face recognition system that can be implemented to any existing system.
Hu et al	2015	When Face Recognition Meets with Deep Learning: an Evalutation of Convolutional Neural Networks for Face Recognition	The usage of deep learning, and convoluted neural networks to create a face recognition that not only works, but works better than conventional face recognition methods.
Laurent, Chollet & Herzberg	2015	Intelligent Automation Entering the Business World	The authors suggest adding more to intelligent functions to robotic process automation to further expand the range of use cases where RPA can be implemented.

In conclusion, based on the work done by the authors mentioned above; it is possible using open source resources to create a light-weight, and accurate face recognition system, and that it's also necessary to make RPA solutions smarter to further expand the capabilities of RPA.

2.2 Theoretical Base

2.3.1. Face Recognition

One of the few things that humans can do well when compared to computers is the ability to be able to recognize faces quickly, and accurately. Face recognition systems generally work by analyzing patterns detected in the facial contours of the human face. With the help of big data, and the Internet of Things (IoT); face recognition systems has seen major improvements in terms of availability, and viability (Taigman, Yang, & Ranzato, 2014).

There are three major approaches used in solving the problem of facial recognition in computing (Agrawal & Khatri, 2015):

1. Feature based approach

   In this approach, the facial features are detected, and then segmented before being put in as input data into a structural classifier.

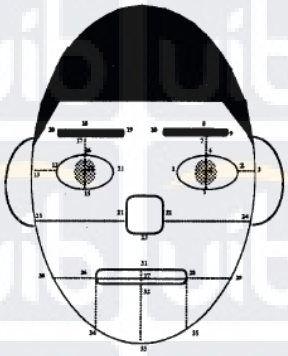

Figure 1 Facial features used in feature based face recognition, source: "Feature-based face recognition using mixture-distance" (1996).

2. Holistic approach

In this approach, the face is analyzed as a whole using statistical methods to extract the statistical characterization from the entire training sample of images.

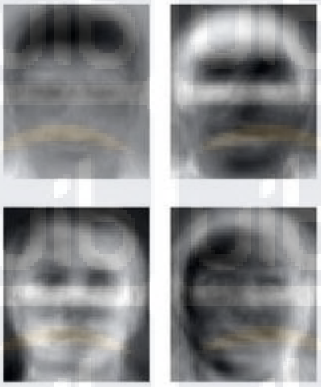

Figure 2 Eigen Faces, source: AT&T Laboratories

3. Hybrid approach

Which is a combination between the two mentioned approaches. The idea comes from how humans perceive the human face both as a whole, and also the local features.

There are 3 techniques that are used for face recognition in computing (Shyam & Singh, 2015):

1. Appearance based statistical techniques

There are numerous amounts of appearance based statistical techniques proposed in recent years that are used to identify individuals through their facial images. These techniques look at images of hundreds, to thousands of faces; and turn them into a usable feature maps that can then be used for face recognition purposes. Principal Component Analysis (PCA) is considered to be an appearance based statistical technique.

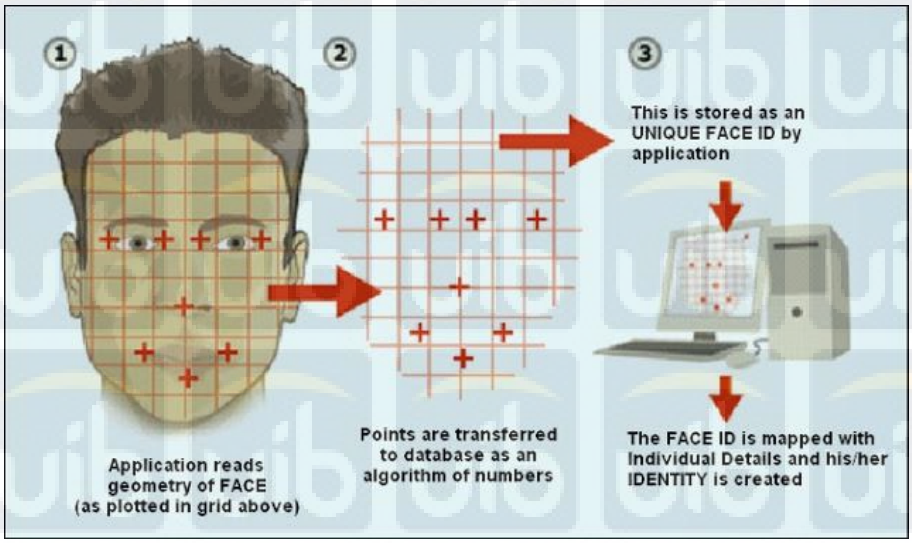

Figure 3 PCA for facial recognition, source: 4.bp.blogspot.com/-Gk3UWoJe5oA/Uqqa_EJgszI/AAAAAAAAAxY/HQV9gZpyGuo/s640/fig1.jpg

2. Texture based techniques

Texture based techniques look at the pixel values of an image directly to determine whether it contains a face or not, and then later can be used to identify faces. Local Binary Pattern (LBP) is a popular texture representation method used in face recognition systems.

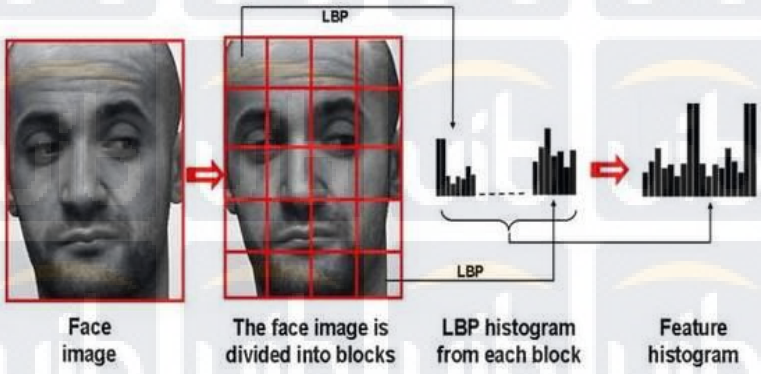

Figure 4 LBP histogram feature extraction, source: www.advancedsourcecode.com/images/lbp_facerecognition.jpg

3. Multimodal techniques

Multimodal techniques, are face recognition techniques that combine the information gathered from multiple sources. Combining the information cues gathered from different biometric sources using an effective fusion scheme can significantly increase model accuracy of a biometric system.

Face recognition systems that uses machine learning algorithms depend on large datasets to be accurate; the consensus is that the larger the dataset is, the better the algorithm will perform (Kemelmacher-Shlizerman, Seitz, Miller, & Brossard, 2015). This is the reason why datasets used by companies like Google, and Facebook are private property of the company, and is generally not open to the public (Parkhi et al., 2015). However, this has not stopped talented individuals from creating an open source face recognition system where the accuracy is comparable to those of Google's and Facebook's even though they are using smaller datasets.

Face recognition technology has never been more accessible to the public with open source libraries like OpenCV, and Dlib enabling individuals to create, and implement their own face recognition system. There's also the fact that the RPA market is constantly growing with multiple RPA vendors competing with each other across the globe. Therefore, it can be concluded that for GleeTrees Pte. Ltd. and its flagship product Gleematic to be able to compete with its competitors, a viable method would be to have something that other RPA vendors lack, and that is a fully functioning face recognition system.

2.3.2. Artificial Neural Network (ANN)

An ANN is a computing system loosely inspired from the human brain to process data and information, and includes neurons similar to those found in actual neural networks to process input data (Hemmat Esfe, Saedodin, Sina, Afrand, & Rostami, 2015).

According to Jordan & Mitchell (2015) there are three major learning paradigms that are used to train, and prepare ANNs:

1. Supervised learning

   In Supervised learning the ANN is provided input-output pairs (training sets) from existing data that has been labelled beforehand. The ANN is then to analyze the provided training data, and produce an inferred function. An optimal scenario would have the ANN then correctly determine the class label from data it has not seen before.

2. Unsupervised learning

   In Unsupervised learning the ANN is not provided any context before, or during training; and is then expected to provide a satisfactory output. Usually, clustering is used in unsupervised learning to group together data based on similarities that they have.

3. Reinforced learning

   In Reinforced learning, the ANN learns how to behave solely on the notion of a cumulative reward that it gets when it does something right.

2.3.3. Application Programming Interface (API)

An Application Programming Interface (API) is a set of routines, protocols, and tools for building software applications. An API serve as a mechanism for separating the interface from data processing, allowing for easier implementation of Applications, and also making the application easier to maintain in the long run (Hudson, 2015).

In a study conducted by Espinha, Zaidman, & Gross (2015) there are three major API implementations:

1. SOAP (Simple Object Access Protocol)

   An API that is performed through the sending of an XML document over an HTTP media to the server; clients request a WSDL file that defines the methods, and also data types that the API will expect. In a case study by Khan & Abbasi (2015) SOAP is defined as a platform independent API, and is the good API of choice when security is a concern; which is why some banking applications still use SOAP.

2. REST (Representational State Transfer)

   An API that relies on entities, and basic CRUD (Create, Read, Update, Delete) actions on those entities. An importance distinction between SOAP, and REST is the fact that while a WSDL equivalent exists in both; it is seldom used in REST. In a case study by Jolley, Bray, & Maiden (2017) they chose to implement a REST API to an existing database that contains molecular typing, and genome database. By Choosing REST users can delegate their access rights to third party tools easier.

3. RPC (Remote Procedural Call)

The RPC approach is similar to REST, but instead uses either JSON, or XML as the data format for requests. RPC provides the mechanism that allows one software system to invoke methods from another software system over a network. In a simulation model that uses two frameworks Lammel, Chraibi, Kemloh Wagoum, & Steffen (2015) used RPC as the API of choice to communicate between the two frameworks.

Other than implementations; there are also two major approaches to API versioning (Sohan, Anslow, & Maurer, 2015):

1. Single

APIs that force clients to migrate because there is only one version available, or that the older version would be removed fall into this category. Facebook's, and Twitter's REST API falls into this category. The advantage of adopting this method of API versioning is that it makes it easier to maintain.

2. Multiple

Multiple versions means that there are multiple versions of the same API in any given time, and are supported by the vendor for a longer duration. The Wordpress, and Stripe API falls into this category. By using this versioning method; it provides the API clients with more scheduling flexibility at the cost of the API provider's effort.

2.3.4. Robotic Process Automation (RPA)

Robotic Process Automation (RPA) is the use of software robots equipped with sufficient Artificial Intelligence (AI) to automate repetitive business processes that are usually done manually in a system (Aguirre & Rodriguez, 2017). RPA robots utilize computer vision, and autonomous software control to interact with existing systems; which is why a lot of businesses who use legacy systems can use RPA solutions in their business processes.

The concept of automating software has been around for a long time in the form of screen scraping; but RPA is considered to be a significant technological evolution of this technique in the sense that new software platforms, and technology such as artificial intelligence, and machine learning are being used (van der Aalst, Bichler, & Heinzl, 2018).

In a case study by Willcocks, Lacity, & Craig (2015) that looks into the implementation of RPA robots in a company where 25% of the processes are automated; there is a ROI (Return of Investment) of 200% over a span of a year. It's also mentioned that it only took two humans to oversee the work of 300 robots, which means that it takes little to no effort to maintain RPA robots after implementation is done.

2.3 Software Used

2.3.1. Sublime Text

Sublime Text is a text, and source code editor with a syntax highlighter that supports multiple programming, and markup languages (Kulvanit, Jones, Bosworth, & Wetzel, 2015). Sublime Text has a package manager where users can install plugins, typically community-built and maintained under free-software licenses. The original Sublime Text is developed by Jon Skinner, and Will Bond; and released on January the 18th, 2008 (Sumangali, Borra, & Suraj Mishra, 2017).

2.3.2. Python 2.7.15

According to Helmus & Collis (2016) Python is a high-level, and an interpreted programming language for general-purpose programming, well known for its expressive, concise, and easy to read syntax; created by Guido van Rossum and first released in 1991. The Python programming language has seen a large surge in popularity due to its readability, modularity, and large standard library (Muller et al., 2015).

2.3.3. Oracle VM VirtualBox

VirtualBox is a free open source software that provides a virtual machines, or a VM that can server as a complete substitute of a real machine in which the user can install a separate operating system, and software into it (Gusev & Swanson, 2015). VirtualBox will be used in the making of the project because of the client, and server side would be running on different operating systems, by using VirtualBox; the writers won't have to go get a separate computer to run an additional operating system.

2.3.4. Odoo v9 (OpenERP)

Odoo is an open source ERP software that offers a range of business applications totaling 4500 modules that form a complete suite of enterprise management applications targeting companies of all sizes (Kendengis & Santoso, 2018). There's also a marketplace for users to download modules to install in their Odoo server. Odoo also allows its users to create their own deployable modules.

2.4 Library Used

2.3.1. OpenCV

OpenCV is an open source computer vision library and is released under the BSD license, making it free for both academic, and commercial use (Palekar, Parab, Parikh, & Kamble, 2017). It has C++, Python and Java interfaces and supports multiple platforms. OpenCV was designed for computational efficiency and with a strong focus on real-time applications in the field of computer vision (Suryatali & V.B.Dharmadhikari, 2015). OpenCV is estimated to have more than 47 thousand people in its user community, and an estimated number of downloads exceeding 14 million; with its usage ranging from interactive art, to advanced robotics (Sathyabama, 2017).

2.3.2. Dlib

Dlib is a general purpose cross-platform open source software library written in the C++ programming language that contains machine learning algorithms and tools for creating complex software in C++ to solve real world problems (Ucar & Hsieh, 2018). Dlib is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge.

2.3.3. NumPy

NumPy is the fundamental package for scientific computing with Python (Pawlik, Segal, Sharp, & Petre, 2014). It contains among other things:

1. a powerful N-dimensional array object.
2. sophisticated (broadcasting) functions.
3. tools for integrating C/C++ and Fortran code.
4. useful linear algebra, Fourier transform, and random number capabilities.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases (Pasumarti & Sekhar, 2018). NumPy is licensed under the BSD license, enabling reuse with few restrictions.

2.3.4. xmlrpclib

xmlrpclib is a python module that allows the use of XML-RPC calls. XML-RPC is a remote procedure call (RPC) protocol which uses XML to encode its calls and HTTP as a transport mechanism (Persico, Montieri, & Pescape, 2016). In XML-RPC, a client performs an RPC by sending an HTTP request to a server that implements XML-RPC and receiving the HTTP response. A call can have multiple parameters and one result. The protocol defines a few data types for the parameters and result. Some of these data types are complex, i.e. nested. For example, you can have a parameter that is an array of five integers. The protocol does not allow HTTPS, but it is a common and obvious variation to use HTTPS in place of HTTP.

XML-RPC is designed to call methods from another software platform, while RESTful protocols use resource representations (documents). The practical difference is just that XML-RPC is much more structured, which means common library code can be used to implement clients and servers and there is less design and documentation work for a specific application protocol. One major technical difference between typical RESTful protocols and XML-RPC is that the RESTful protocol uses the HTTP URI for parameter information whereas with XML-RPC, the URI just identifies the server.

References

Agrawal, S., & Khatri, P. (2015). Facial Expression Detection Techniques: Based on Viola and Jones Algorithm and Principal Component Analysis. *International Conference on Advanced Computing and Communication Technologies, ACCT, April*(1), 108–112.

Aguirre, S., & Rodriguez, A. (2017). Automation of a Business Process Using Robotic Process Automation (RPA): A Case Study. *Communications in Computer and Information Science*, *742*(1), 66–71.

Amos, B., Ludwiczuk, B., & Satyanarayanan, M. (2016). OpenFace: A General-Purpose Face Recognition Library with Mobile Applications. *Technical Report CMU-CS, CMU School of Computer Science*, *16*(118), 1–18.

Anagnoste, S. (2017). Robotic Automation Process - The next major revolution in terms of back office operations improvement. *Proceedings of the International Conference on Business Excellence*, *11*(1).

Bolotnikova, A., Demirel, H., & Anbarjafari, G. (2017). Real-time Ensemble Based Face Recognition System for NAO Humanoids Using Local Binary Pattern. *Analog Integrated Circuits and Signal Processing*, *92*(3), 467–475.

Espinha, T., Zaidman, A., & Gross, H. G. (2015). Web API Growing Pains: Loosely Coupled Yet Strongly Tied. *Journal of Systems and Software*, *100*(1), 27–43.

Gusev, D. A., & Swanson, D. (2015). Tools to Use in an Information Technology Class – and Best of All They Are FREE! In *2015 ASCUE Proceedings* (pp. 52–61).

Helmus, J. J., & Collis, S. M. (2016). The Python ARM Radar Toolkit (Py-ART),

a Library for Working with Weather Radar Data in the Python Programming Language. *Journal of Open Research Software*, *4*. https://doi.org/10.5334/jors.119

Hemmat Esfe, M., Saedodin, S., Sina, N., Afrand, M., & Rostami, S. (2015). Designing an Artificial Neural Network to Predict Thermal Conductivity and Dynamic Viscosity of Ferromagnetic Nanofluid. *International Communications in Heat and Mass Transfer*, *68*(1), 50–57.

Hu, G., Yang, Y., Yi, D., Kittler, J., Christmas, W., Li, S. Z., & Hospedales, T. (2015). When Face Recognition Meets with Deep Learning: an Evaluation of Convolutional Neural Networks for Face Recognition. *Iccv*, (1), 142–150.

Hudson, A. (2015). Why Experimental Interfaces Should Include an Application Programming Interface Jennifer Windsor & Omar Rodriquez. *Scholarly and Research Communication*, *6*(2), 1–11.

Jolley, K. A., Bray, J. E., & Maiden, M. C. J. (2017). A RESTful application programming interface for the PubMLST molecular typing and genome databases. *Database : The Journal of Biological Databases and Curation*, *2017*(1), 1–7.

Jordan, M. I., & Mitchell, T. M. (2015). Machine Learning: Trends, Perspectives, and Prospects. *Science*, *349*(6245), 255–260.

Kemelmacher-Shlizerman, I., Seitz, S., Miller, D., & Brossard, E. (2015). The MegaFace Benchmark: 1 Million Faces for Recognition at Scale, (1), 4873–4882.

Kendengis, Y., & Santoso, L. W. (2018). Integration Between ERP Software and Business Intelligence in Odoo ERP: Case Study A Distribution Company.

Advances In Natural And Applied Sciences, *12*(4), 16–21. https://doi.org/10.22587/anas.2018.12.4.4

Khan, M. W., & Abbasi, E. (2015). Differentiating Parameters for Selecting Simple Object Access Protocol (SOAP) vs. Representational State Transfer (REST) Based Architecture. *Journal of Advances in Computer Networks*, *3*(1), 63–66.

Kulvanit, A., Jones, H. A., Bosworth, J. T., & Wetzel, J. M. (2015). JPMorgan Web Service Registry. *Major Qualifying Projects (All Years)*, *1*(January), 1–82.

Lammel, G., Chraibi, M., Kemloh Wagoum, A. U., & Steffen, B. (2015). Hybrid Multi and Inter-modal Transport Simulation: a Case Study on Large-Scale Evacuation Planning. *Journal of Applied Microbiology*, *119*(3), 1–17.

Laurent, P., Chollet, T., & Herzberg, E. (2015). Intelligent Automation Entering the Business World. *Inside*, (1), 66–73.

Muller, E., Bednar, J. A., Diesmann, M., Gewaltig, M.-O., Hines, M., & Davison, A. P. (2015). Python in Neuroscience. *Frontiers in Neuroinformatics*, *9*(April), 1–4.

Palekar, R. R., Parab, S. U., Parikh, D. P., & Kamble, V. N. (2017). Real time license plate detection using openCV and tesseract. *2017 International Conference on Communication and Signal Processing (ICCSP)*, *40055*(1), 2111–2115.

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep Face Recognition. In *Procedings of the British Machine Vision Conference 2015* (pp. 1–12).

Pasumarti, P., & Sekhar, P. P. (2018). Classroom Attendance Using Face

Detection and Raspberry-Pi. *International Research Journal for Engineering and Technology*, *5*(1), 167–171.

Patel, S., Kumar, P., Garg, S., & Kumar, R. (2018). Face Recognition Based Smart Attendance System Using IOT. *International Journal of Computer Sciences and Engineering*, (5), 871–877.

Pawlik, A., Segal, J., Sharp, H., & Petre, M. (2014). Crowdsourcing Scientific Software Documentation: a Case Study of The NumPy Documentation Project. *Computing in Science and Engineering*, (1), 1–12.

Persico, V., Montieri, A., & Pescape, A. (2016). CloudSurf: A platform for monitoring public-cloud networks. In *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow, RTSI 2016* (pp. 1–6).

Sathyabama, K. (2017). True Alarm Using Artificial Intelligence and Internet of Things. *International Journal of Creative Research Thoughts*, *5*(4), 2688–2693.

Shyam, R., & Singh, Y. N. (2015). Identifying Individuals Using Multimodal Face Recognition Techniques. *Procedia Computer Science*, *48*(1), 666–672.

Sohan, S. M., Anslow, C., & Maurer, F. (2015). A Case Study of Web API Evolution. In *Proceedings - 2015 IEEE World Congress on Services, SERVICES 2015* (pp. 245–252).

Sumangali, K., Borra, L., & Suraj Mishra, A. (2017). A Comprehensive review on the open source hackable text editor-ATOM. In *IOP Conference Series: Materials Science and Engineering* (Vol. 263, p. 042061). https://doi.org/10.1088/1757-899X/263/4/042061

Suryatali, A., & V.B.Dharmadhikari. (2015). Computer Vision Based Vehicle Detection for Toll Collection System Using Embedded Linux. In *International Conference on Circuit, Power and Computing Technologies [ICCPCT]* (pp. 1–7).

Taigman, Y., Yang, M., & Ranzato, M. A. A. (2014). Deepface: Closing the Gap to Human-Level Performance in Face Verification. *Proc. CVPR*, (1), 1701–1708.

Ucar, M., & Hsieh, S.-J. (2018). MAKER : Facial Feature Detection Library for Teaching Algorithm Basics in Python. *American Society for Engineering Education*, *1*(1), 1–11.

van der Aalst, W. M. P., Bichler, M., & Heinzl, A. (2018). Robotic Process Automation. *Business and Information Systems Engineering*, *60*(4), 269–272.

Willcocks, L., Lacity, M., & Craig, A. (2015). Robotic Process Automation: Mature Capabilities in the Energy Sector. *The Outsourcing Working Research Paper Series*, (October), 1–19.