Yüz Tanımaya Dayalı Kişi Bazlı Test Otomasyonu

Alphan Çamlı[1], Damla Gülen[1], Nihat Ük[1], and Anıl Gündoğdu[1]

Siemens A.S., Istanbul 34870, Turkey
{alphan.camli,damla.gulen,nihat.uk,anil.gundogdu}@siemens.com

Abstract. Sürekli entegrasyon modeli ile birlikte yazılım testlerinin otomatize edilmesi konusunda önemli yol katedilmiştir. Günümüzde farklı ürünler tek bir yazılım test otomasyon çerçevesi içerisinde test edilebilmektedir.Test otomasyon çerçevesini benimsemiş prosesler dağıtım zamanı ve kod kalitesi bakımından daha efektif olmaktadır. Ancak proje takımları güvenlik konusunun test sistemlerine entegre edilmesi konusunda zorluklar çekmektedir. Ayrıca, aynı test otomasyon çerçevesini kullanan farklı proje takımlarının farklı test senaryolarına sahip olması da endüstriyel test proseslerinde bir sorun olarak görülmektedir. Bu bağlamda kompleks otomatik test ortamlarında test senaryolarının gruplanıp modülarize edilmesi konusunda bir çözüm yolu bulunmalıdır. Eğer test senaryoları otomatik olarak kişilere özel olarak koşturulabilirse farklı proje grupları için farklı test senaryoları koşturulabileceği gibi aynı proje içerisinde iş bölümü yapılarak aynı ekipte çalışan farklı kişiler tarafından farklı test senaryoları da koşturulabilecektir. Bu çalışma bağlamında önerilen kişi bazlı test otomasyonu farklı endüstriyel projeler için Siemens özelinde kullanılan bir endüstriyel test otomasyon çerçevesi içerisinde gerçeklenip incelenmiştir.Sürekli entegrasyon döngüsü biyometrik yüz tanıma ile tetiklenerek otomatik testlerin kişilere özel olarak çalıştırılması hedeflenmiştir. Tetikleyici olarak biyometrik sinyal kullanıldığı için test senaryoları farklı kişiler için gruplanıp modülarize edilebilmekte ve sistemin genel güvenliğinde de gelişme görülmektedir.

Keywords: Çevik Yazılım Testi · Sürekli Entegrasyon · Test Otomasyon Çatısı · Kişi Bazlı Otomasyon · Güvenlik · Yüz Tanıma

Person Based Test Automation Using Facial Recognition

Alphan Çamlı[1], Damla Gülen[1], Nihat Ük[1], and Anıl Gündoğdu[1]

Siemens A.S., Istanbul 34870, Turkey
{alphan.camli,damla.gulen,nihat.uk,anil.gundogdu}@siemens.com

Abstract. Continuous integration has come a long way in making tests integral and automated. Nowadays, it is possible to integrate different products into a common test automation framework thanks to recent advances on automated testing. Processes adopting test automation framework seem to be more efficient in terms of deployment time and code quality. However, project teams are still struggling to get security adequately integrated into the test frameworks. Furthermore, having seperate test case suites for different project teams using the common test automation framework deemed as a problem especially in industrial test processes. Because there should be a way to modularize and group test cases in such a complex automated testing environment. Then, these test cases must be personalized according to different projects' members so that different test case scenarios can be run based on the person. In the scope of this study, the main objective is achieving person based test automation on Siemens specific industrial test automation framework. Continuous integration cycle on aforementioned test framework is triggered by bio inspired facial recognition trigger. Since, trigger is based on a biometric signal, test cases can be grouped according to user's identification while improving overall security of the framework.

Keywords: Agile Software Testing · Continuous Integration · Test Automation Framework · Person Based Automation · Security · Face Recognition

1 Introduction

In the innovation era, companies and individuals are always in competition nowadays. To achieve success in this competitive world, software development teams and projects need to be adapted to the rapid change. Requirements, needs and demands are now changing more frequently. In order to serve those highly alternating needs the agile manifesto [2] is presented in 2001. The agile methodology gives major priority to satisfying customer needs by providing early and continuous delivery of the minimum viable product. Agile software development brings out the importance of continuous integration in the software development lifecycle. In this manner, the cycles are shorter and the customer is more involved with the product. They investigate the deliverables and interfere early which reduces cost of many fixes. As early as a change is requested in the cycle, the easier it gets fixed. This brings up the importance of testing. Since frequent releases with small changes increases the test amount, the automation of the test cases also has high importance in this type of applications. In agile test automation pyramid presented by Mike Cohn, it is stated that most testing work is done under unit test. Every small piece of code needs testing and it is easier to fix the errors in this level of testing compared to Service and UI levels.

In agile teams, the members might be distributed to the different locations. Teams work remotely and they need to work productively with high communication among themselves. The common practice is to use a remote platform to run codes and tests on, in order to avoid any locality problems. Any change in the code is committed to the repository, build and run. After a successful build, a set of tests starts to run. At this point the idea is to focus on how well those tests can be separated, divided into groups and run securely.

At the start of the project, the main focus was security. A test system needs to be secure, well protected and the rules need to be well defined, clearly separated. Authentication systems usually work with username and password combinations which are known as credentials that the user knows. This is the most commonly used authentication method. However, this type of authentication starts to give its place to biometric authentication mechanisms, which authenticates users with their fingerprints, iris, face, voice or even with their behavior. Among these types of biometric authentication methods, the easiest and cheapest one to implement is face recognition in the industrial application field. It does not require any complex hardware such as iris scanner and its data is easier to collect and train. In this project, authentication based on face recognition is chosen with the same reasons that collaborated. In industrial testing environment, face recognition seems to be easier to implement, and only needs basic camera hardware which most of the test computers have it already. After getting authenticated by the face recognition algorithm, a test user can view and run the test cases that he/she is authorized to. In other words, the face recognition is used as trigger mechanism in the solution presented here.

Continuous Integration (CI) is a common software development practice. It motivates developers to commit their code early and often. The purpose of the practice is to eliminate the need for unpredictable and long integration effort before the release time by integrating the software with a constant frequency by different developers. CI needs higher degree of automation of code compiling and testing activities since it goes through the same cycles many times during a development of a product and these repetitive phases needs to get automated. CI enables more frequent releases and this improves the companies' ability to compete. Another benefit of CI is the early feedbacks which helps to fill the gaps between what the customer values and what the developers offer. Early and frequent feedbacks enable developers and testers to detect and fix the defects sooner [3].

In a CI process, developers' changes in the software trigger a set of automated activities. These activities aim to prove that the changes successfully integrate with the existing software. In this process, developers need to check in their changes into a central control system where the changes are monitored by the CI system. In the CI system there is a preset integrity criterion for the software and each increment in the code is tested against it and presents feedback related to the result. This feature enhances the quality of the product and notifies the developers early with the feedback. Continuous delivery is the extended case of CI where the software is deployed to production automatically after passing all the tests [4].

A simple workflow can be as follows. After an increment in the code is checked in to the version control, the CI system monitors the changes which results in triggering an automated workflow called CI build. CI build first compiles the source code. Then, CI system could run unit tests in order to verify the compiled code's quality. The result is reported by the CI system by e-mail to the developers who made changes in the increment. However, in practice more automated testing is required. To improve the quality in addition to static code analysis, automated integration and acceptance tests are performed by the CI system.

In order to achieve high test coverage in automated tests considered in the CI system, a lot of work is required to construct stable automated tests. Having unstable automated tests could lead to test results being ignored by developers. And consequently they might miss the points where a fix is required. Moreover, having manual tests instead of these automated ones will take huge amount of time and slow down the CI cycle [7].

This paper is organized as follows: Section 2 details the overall system overview. Section 3 explains deployed facial recognition algorithm for person identification where Section 4 presents test automation framework used for Siemens products without revealing the inner details. System integration is presented in Section 5, results for a sample project is given in Section 6. Section 7 is the conclusion part where we summarize our proposal.

2 System Overview

Typical continuous integration cycle begins with the check in of the source code to be tested. The check in of the source code and hence start of automated test scripts is triggered by a trigger which can be either predefined, or time scheduled or even can be done manually by a team member. In the case of manual trigger by a team member, a question arises, "Should any team member be able to execute all automated tests?". Traditionally, automated testing meant testing the end to end execution flow on the user interface level. However, agile test automation pyramid which is developed by Mike Cohn, suggests that test automation should be divided into three levels. Unit tests should form the base of this pyramid. The service level tests form the next layer and finally the user interface tests form the apex. In this concept, unit and service level tests form the major part of the test automation strategy. Hence, automated test scripts can be categorized under different test case suites. Furthermore, a test automation framework might be used for testing different products. Since every product has different functionalities, their test case suites can also be different. For example, one product might have cloud connectivity whereas another product has not. In this case, it doesn't seem plausible to run cloud test scripts on the product which has no cloud connectivity requirements. Considering a common test automation framework is used by every team, then giving different user rights to different team members seems logical. Here, different user rights would determine which test scripts are allowed to be executed.

There is also security aspect on the test automation. Currently, test computers in industrial test environments running automated test scripts are protected by simple means of user name and password authentication. However, an attacker may crack username and password in a likely scenario. Hence, forming an authentication scheme based on biometric features of defined team members would undeniably increase overall security of the test environment. Such system is hard to break in because authentication is based on a biometric feature which is hard to copy. In this paper, we provide a bio inspired continuous integration trigger based on face recognition.

It is possible to find different bio inspired identification methods in the literature. Bebis et al. [1] showed person identification based on fingerprints by using delaunay triangulation method. Iris recognition emerges as trending biometric recognition approach, and is becoming a very active research topic in both research and practical applications [6]. For speech based recognition, Reynolds and Rose [9] successfully utilized Gaussian mixture model in speaker recognition evaluations. However, all of the mentioned above identification methods require extra hardware such as fingerprint scanner, iris scanner or microphone which are uncommon for an industrial test automation environment. Since test PCs and laptops might have cameras which would be sufficient for a real time face recognition application, we decided to choose face recognition based identification for validating team members trying to trigger automated tests. The main idea here is to recognize team members based on facial features by comparing their face on test PC's camera to images with corresponding IDs stored in the system. If the face of a person who tries to trigger automated tests doesn't match with the one of the stored images, then the user is unknown and automated tests won't be started. If they match, then a query will be made to a local user database to get user's access rights and role. According to returned user access rights, user can start some or all of the tests in the test suite case.

In our proposed system, multi task cascaded convolutional networks [16] are used for face detection while an Inception Resnet [11] is used for ID classification. Every predefined team member has a corresponding ID and an image stored in the system. When a team member wants to trigger automated tests in the test environment, his/her face is tracked via test computer's camera. Multi task cascaded convolutional networks are employed for detecting whether a human face is present in front of the camera. After facial detection, facial features compared with every image stored in the system. For the purpose of face recognition, Inception Resnet model is trained as a classifier with the inclusion of center loss [13]. CASIA-Webface [15] dataset has been used for training. If user's face is matched with one of the stored images, then a query with corresponding user ID, which asks about given user's role, would be made to a local database. Then, automated test scripts would be triggered depending on the returned user role.

3 Face Recognition Algorithm

For the scope of the paper, it is necessary to implement a face recognition algorithm with a strong real time performance in practice. The application must try to find human face in front of a camera than try to match it with the stored images in the system using deep neural networks. Hence, face detection algorithm should run smoothly before face recognition. In the literature, there are various proposals for face detection and face alignment. Viola and Jones [12] proposes AdaBoost based on Haar-Like features for training cascaded classifiers, and shown that good performance with real time efficiency is achievable. Zu and Ramanan [17] proposes deformable part models (DPM) for face recognition with a good performance overall in expense of computational effort. Sun et al. [10] utilize convolutional neural networks for face recognition task. Currently, state-of-the-art techniques combine convolutional neural networks with deep learning methods. Yang et al. [14] utilize deep convolutional neural networks whereas Li et al. [5] train cascaded convolutional neural networks in order to decrease time cost.

As mentioned in "System Overview" chapter, multi task cascaded deep convolutional networks are utilized in the scope of the project for face detection and face alignment without compromising performance. Yandong et al. [14] showed that it is possible to achieve high accuracy up to 99 percent without compromising performance by combining convolutional neural networks with supervisory signal called as center loss. Large pose variations, visual variations of faces such as occlusions, extreme lightings result decrease in real world application performance. Face alignment is needed as preprocessing stage before dataset training for better performance. Hence, multi task cascaded deep convolutional networks [16] combined with center loss are implemented for the face alignment. Also, performance of a recognition algorithm varies with dataset to be trained. In that regard, training data becomes equivalently significant as the core algorithm.

CASIA-Webface [15] dataset has been chosen as training dataset. This training dataset consists of 494414 images containing 10575 identities. Training model is based on Inception Resnet architecture and the model is trained as a classifier combined with center loss.

4 Test Automation Framework

Industrial communication is the backbone of modern automation solutions. The communication networks and products involved allow totally integrated communication between the widest possible variety of automation components and devices. And our test automation framework is the name of an entire family of communications networks and products from Siemens. The various networks meet the widest possible range of performance and application requirements in automation engineering.

Our framework provides solutions for individual customer requirements in industrial communication. The communication networks and products are tested by the aforementioned test framework. On this basis, branch specific automation solutions can be implemented with comprehensive and highly integrated communication functions. The test framework simplifies the commissioning of automation systems regardless of the communication networks and products used.

In terms of their performance and range of functions, the communication networks and products of the framework can be represented in the form of an automation pyramid. The automation pyramid can be divided into three levels; field, cell and management level. The field level is where process or communication is handled. For this level, the framework offers PROFIBUS DP and the AS-Interface. At the cell level, the acquired process data is distributed to the various automation systems or PCs for operator control and monitoring. In this level, the communication networks Industrial Ethernet and PROFIBUS are used by framework. Higher-level management functions are handled, process data is saved, processed further, or used for analysis by the management level. For such tasks, Industrial Ethernet is suitable as the communication network.

Furthermore, our test framework includes many other interfaces such as OPC - which is a standard interface to communicate between numerous data sources-, including devices on a factory floor, laboratory equipment, test system fixtures and databases. Although, our test automation framework covers variety of projects and therefore test case suites, the one of the most commonly tested functionality is OPC standards and specifications. The OPC Foundation defined a set of standard interfaces that allow any client to access any OPC-compatible device using a protocol now referred to as Classic OPC (OPC COM). This protocol utilizes the Microsoft based COM/DCOM technology to provide standard specifications for data access (DA), historical data access (HDA) and alarms and events (AE). Although basing a protocol on this technology made sense in the 1990s, Classic OPC has several limitations because of this reliance on the Microsoft Windows platform, in the form of security issues and platform dependency.

OPC Unified Architecture (UA) is a new communication technology standard which was first released by the OPC Foundation in 2006 as an improvement upon its predecessor, Classic OPC. OPC UA includes all of the functionality found in Classic OPC. This is done by bringing together the different specifications of Classic OPC into a single entry point to a system offering current data access, alarms and events, combined with the history of both. Furthermore, OPC UA is based on a cross-platform, business-optimized Service-Oriented Architecture (SOA), which expands on the security and functionality found in Classic OPC, instead of the Microsoft-based COM/DCOM technology. OPC UA supports two protocols: a binary protocol that employs minimal resources, allowing for easy enablement through a firewall; and a Web Service protocol (SOAP) which uses standard HTTP/HTTPS ports. Because of the benefits of this new protocol, an increasing trend of industrial applications have adopted the UA protocol both in the traditional OPC-centric industrial automation space and emerging areas, such as energy.

Test automation triggering operation has been used for OPC interfaces. Along with other test cases, OPC part has been divided to three roles as admin, diagnose and standard for testing team to check COM, OPC UA, OPC DA, Alarms and Events (AE), Historical Data Access (HDA), Profinet, and Ethernet. In the conclusion section, exemplary results for a continuous integration cycle of an OPC project are presented.

5 System Integration

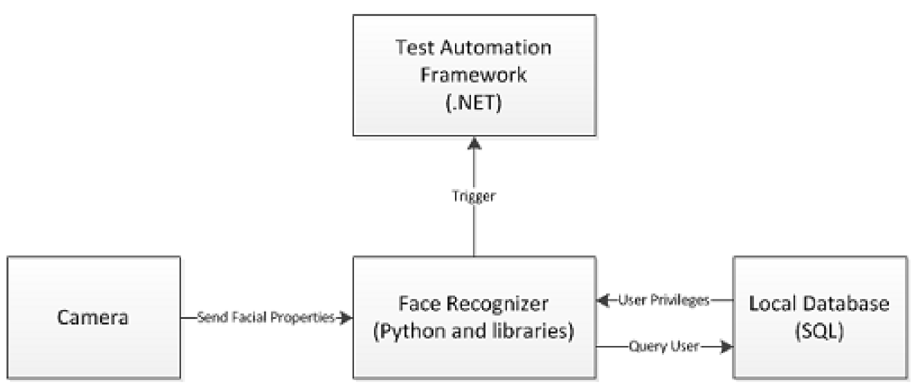

Fig. 1. Overall system schematic

Figure 1 depicts overall system schematic. Camera represents test PC's camera and takes test user's facial properties as input and sends them to face recognizer. Face recognizer algorithm is implemented with Python script language. Python language is chosen as the development language because it is very lightweight, easy to learn and has a good community support. Furthermore, Python SDK has open source support for image processing, machine intelligence, scientific computation libraries such as OpenCV, Pillow, TensorFlow, SciPy. Hence, face recognizer script has dependencies on aforementioned libraries. The script runs on test PC and takes facial properties from camera view via OpenCV library functions. Pillow and SciPy library functions such as resizing, cropping, convolution, filtering are used for processing captured images and implementing multi task cascaded convolutional network algorithm. The core concept at the heart of face recognition algorithm is training Inception Resnet model. Parkhi et al. [8] showed that training model as a classifier yields a much better performance. Therefore, we trained Inception Resnet model as a classifier considering the need for fast and efficient face recognition algorithm. Tensorflow library is used in classifier training. Also, CUDA extension is enabled while using Tensorflow considering exhaustive computational effort for training deep neural networks.

After test user's successful identification, an SQL query is made to the database for user's rights and role. The database is SQL based and it holds user ID, name, role, rights fields in the corresponding table. In this concept, different user roles such as admin, diagnose, standard, remote access, OPC UA, DA, COM, alarms and events (AE), historical data access (HDA), Profinet, Ethernet, S7-communication can run subset of test case suites in our test automation framework. The aforementioned test automation framework is running on .NET framework and it enables automated testing of entire SIMATIC family of industrial communication networks and products from Siemens. Therefore, this common automated test framework consists of different test case suites such as OPC, S7 communication, Profinet tests. And with our proposed solution, different project team members can trigger automated tests in their continuous integration cycle specific to them based on facial recognition. The proposed system also addresses security concerns in continuous development by suggesting a bio inspired continuous development trigger.

6 Person Based Automated Test Results

In this section, we run our proposed framework for an exemplary test scenarios. An OPC based product is chosen as example product under test, because of OPC' s wide popularity and trend towards it in the industrial automation community. We defined 6 user roles namely "Remote Access", "View Controller", "Server Controller", "Alarm Controller", "Diagnostic", and "Admin".

All of the roles are derived from Admin role. Admin has all rights for running each test cases. Remote Access role is created for only running the remote test cases which are under Profibus and Ethernet. View Controller role is created for View Controller people who are controlling OPC Unified Architecture (UA), Data Access (DA), and Alarm views in this system. And they will be responsible people for running view related test cases. Server Controller role is created for running server based test cases. These test cases are related with Classic OPC (OPC COM) and OPC Unified Architecture (OPC UA). Alarm Controller role is created for running Alarm test cases. Diagnostic role is created for only running the Diagnostic View test cases which are defined under Data Access (DA) and Historical Data Access (HDA). Then, we assigned these user roles to different test users with user name, user ID and access rights. Hence, our local database and users table is designed according to our test users information. If the user is verified by facial recognition, then an SQL query is made to this database, and corresponding test user role is returned. According to returned user role, test user can only run subset of all test scripts if any user role is assigned to verified facial recognition. Figure 2 shows which test cases can be run by the defined user roles.

	User Role					
Test Parts	**Remote Access**	**View Controller**	**Server Controller**	**Alarm Controller**	**Diagnostic**	**Admin**
COM			x			x
UA		x	x			x
DA		x			x	x
Alarm		x		x		x
HDA					x	x
Profibus	x					x
Ethernet	x					x

Fig. 2. User Roles for Exemplary OPC Test Suite

7 Conclusion

In this article, we proposed a person based continuous integration system which is triggered by bio inspired facial authentication for industrial test processes. Although, any real time person identification technique can be used, face recognition seems to be most suitable choice based on current industrial test environments. Siemens specific test automation framework is combined with face recognition implementation to address automated test security concerns along with possibility to run person based test cases. We demonstrated exemplary automated test results in a single continuous integration cycle with different users for a chosen OPC based project.

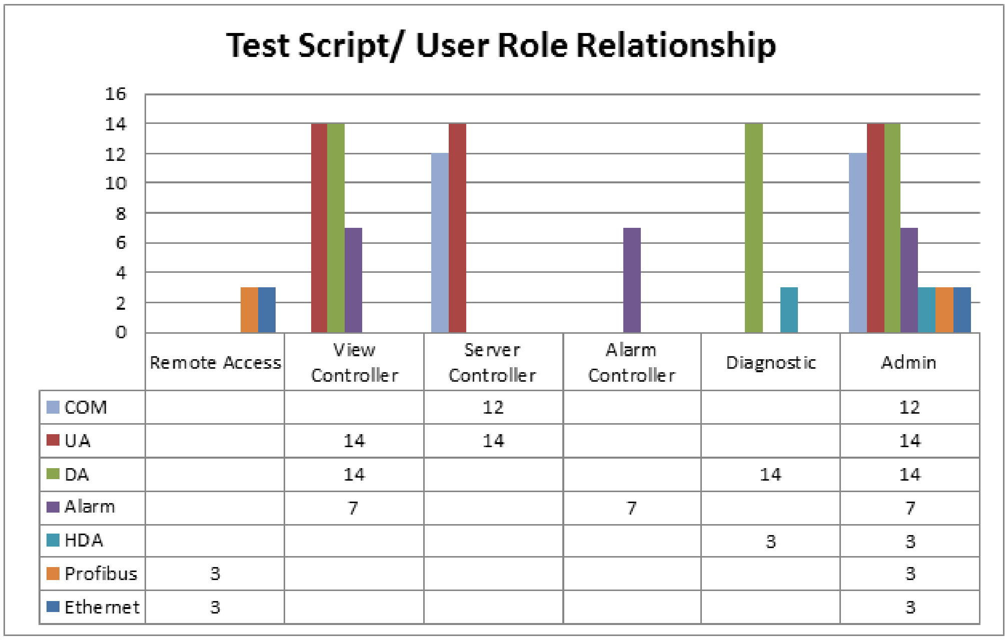

Fig. 3. Permitted test scripts for defined user roles

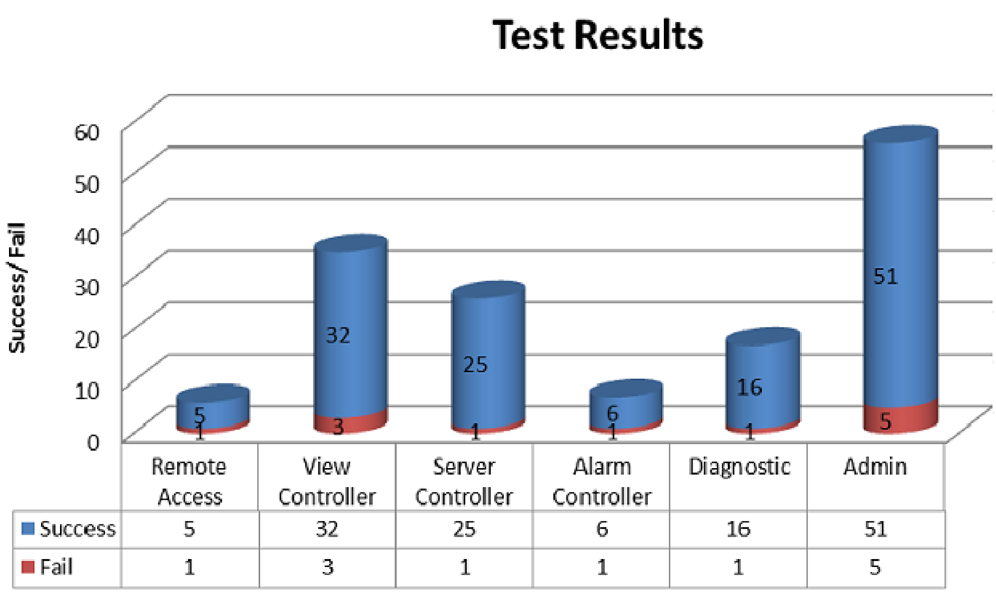

Fig. 4. Single CI cycle results for exemplary OPC test suite

References

1. Bebis, G., Deaconu, T., Georgiopoulos, M.: Fingerprint identification using delaunay triangulation. In: Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on. pp. 452–459. IEEE (1999)
2. Highsmith, J., Fowler, M.: The agile manifesto. Software Development Magazine **9**(8), 29–30 (2001)
3. Hukkanen, L.: Adopting Continuous Integration - A Case Study. Master's thesis, Aalto University (2015)
4. Jez Humble, D.F.: Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley Professional, New Jersey, NJ, 1st. edn. (2010)
5. Li, H., Lin, Z., Shen, X., Brandt, J., Hua, G.: A convolutional neural network cascade for face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5325–5334 (2015)
6. Ma, L., Tan, T., Wang, Y., Zhang, D.: Personal identification based on iris texture analysis. IEEE transactions on pattern analysis and machine intelligence **25**(12), 1519–1533 (2003)
7. Neely, S., Stolt, S.: Continuous delivery? easy! just change everything (well, maybe it is not that easy). In: Proceedings of the 2013 Agile Conference. pp. 121–128. AGILE '13, IEEE Computer Society, Washington, DC, USA (2013). https://doi.org/10.1109/AGILE.2013.17, http://dx.doi.org/10.1109/AGILE.2013.17
8. Parkhi, O.M., Vedaldi, A., Zisserman, A., et al.: Deep face recognition. In: BMVC. vol. 1, p. 6 (2015)
9. Reynolds, D.A., Rose, R.C.: Robust text-independent speaker identification using gaussian mixture speaker models. IEEE transactions on speech and audio processing **3**(1), 72–83 (1995)
10. Sun, Y., Chen, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. In: Advances in neural information processing systems. pp. 1988–1996 (2014)
11. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI. vol. 4, p. 12 (2017)
12. Viola, P., Jones, M.J.: Robust real-time face detection. International journal of computer vision **57**(2), 137–154 (2004)
13. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: European Conference on Computer Vision. pp. 499–515. Springer (2016)
14. Yang, S., Luo, P., Loy, C.C., Tang, X.: From facial parts responses to face detection: A deep learning approach. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3676–3684 (2015)
15. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. arXiv preprint arXiv:1411.7923 (2014)
16. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters **23**(10), 1499–1503 (2016)
17. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 2879–2886. IEEE (2012)