JITE (Journal of Informatics and Telecommunication Engineering)

Available online <http://ojs.uma.ac.id/index.php/jite> DOI : 10.31289/jite.v3i2.3106

Face Recognition System to RPA Software Design and Implementation

Suwarno¹⁾ *, Kevin¹⁾

1) Prodi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam, Indonesia

*Corresponding Email: suwarno.liang@uib.ac.id

Abstrak

Tujuan dari penelitian ini adalah untuk membangun sebuah sistem pengenalan wajah, dan mengimplementasikan sistem tersebut terhadap sistem *Robotic Process Automation* (RPA) yang sudah ada. Sistem pengenalan wajah ini dikembangkan menggunakan bahasa pemrograman Python. Algoritma pengenalan wajah yang digunakan adalah OpenCV. OpenCV merupakan *library open-source* yang sudah dikembangkan dan berkedudukan kuat di dunia pengenalan wajah. Aplikasi di sisi pengguna berbasis *Desktop*, dan memerlukan koneksi internet yang stabil. Pengguna sistem ini dapat mendaftarkan wajah ke dalam sistem baik secara manual ataupun secara otomatis. Sistem ini mampu mendeteksi, dan mengekstrak informasi dari wajah tersebut dengan kecepatan rata-rata 500ms untuk setiap frame dengan tingkat akurasi ~98% jika nilai toleransi diatur pada angka 0.6. Sistem ini juga mampu mendaftarkan wajah baru yang dihadapinya.

Kata Kunci: Pengenalan wajah, Python, RPA, Desktop.

Abstract

The purpose of this research is to build a face recognition system, and implement it into an RPA (Robotic Process Automation) software to expand automation capabilities. The system is built using the Python programming language. The face recognition algorithm that is used is an open-source library that has been pre-trained and developed beforehand along with a library called OpenCV. The client side of the system is desktop based, and requires a stable internet connection. Users of the system are able to register faces into the system, and then later detect and extract information from them using only images of faces with an average speed of 500 ms for every frame, with an accuracy of ~98% with tolerance set at the default value of 0.6. The system is also capable of automatically registering any new faces that it encounters.

Keywords: Face Recognition, Python, RPA, Desktop.

How to Cite: Suwarno, Kevin. (2020). Face Recognition System to RPA Software Design and Implementation. *JITE (Journal Of Informatics And Telecommunication Engineering)*. 3 (2): 175-181

I. PENDAHULUAN

Pada zaman ini; dengan adanya *Big Data*, dan peningkatan penggunaan *Internet*; jumlah data baru yang dihasilkan secara global meningkat dalam kecepatan yang pesat. Berdasarkan sebuah artikel oleh Idrees, Alam, & Agarwal, (2018) perkiraan jumlah data yang dihasilkan setiap hari adalah 2,5 quintillion (25×10^{17}) byte; jumlah ini akan terus bertambah; terutama untuk negara-negara berkembang di Asia Tenggara menurut penelitian oleh Poushter, (2016). Banyak organisasi, dan perusahaan menggunakan teknologi informasi untuk menangani *Big Data* (Hashem et al., 2015), tetapi masih ada banyak organisasi dan perusahaan lain yang berjuang untuk mengikutinya; dan masih bekerja dengan metode tradisional untuk mengelola data. Metode tradisional ini melibatkan tugas berulang yang dilakukan oleh manusia secara manual; selain rentan terhadap kesalahan manusia, ada berbagai kelemahan lain seperti: harga, dan tidak selalu tersedia karena keterbatasan manusia. Mengganti pekerja manusia dengan bot yang digerakkan oleh AI menggunakan teknologi *Robotic Process Automation* (RPA) akan membebaskan pekerja manusia untuk melakukan pekerjaan yang hanya bisa digunakan oleh manusia; dan dengan demikian menurut Autor (2015) akan meningkatkan kualitas

kerja yang tersedia. Ada juga peningkatan permintaan dalam menggunakan alat RPA antar perusahaan pada akhir ini (van der Aalst, Bichler, & Heinzl, 2018).

Dalam penelitian ini, peneliti melakukan penelitian terapan terhadap *OpenCV library*, di mana peneliti menganalisis masalah dunia nyata dan kemudian melanjutkan dengan mengimplementasikan solusi untuk memecahkan masalah (Mani, 2016). Penelitian ini bertujuan untuk lebih memperluas kemampuan solusi RPA yang ada dengan menambahkan kemampuan pengenalan wajah karyawan sebuah perusahaan dan mengintegrasikannya ke dalam perangkat RPA.

II. PENDAHULUAN

Menurut Lacity, Willcocks, & Craig, (2015) RPA adalah “otomatisasi tugas yang sebelumnya dilakukan oleh manusia”. Tugas-tugas ini dapat berupa pergerakan data yang sederhana, sampai dengan tugas-tugas tingkat lanjut seperti membalas *email*. Sistem pengenalan wajah digunakan untuk mengenali, dan membuat katalog wajah-wajah orang yang berbeda, menyimpan data tersebut dan kemudian menggunakan informasi tersebut pada masa yang akan datang sebagai referensi. Sistem pengenalan wajah umumnya digunakan sebagai fitur di media sosial, dan juga untuk keperluan verifikasi di

media lainnya. Dengan bantuan algoritma *Machine Learning*, *Big Data*, dan teknologi baru yang muncul; algoritma pengenalan wajah menjadi lebih akurat dalam mendeteksi, dan mengenali wajah (Vinay et al., 2015). Penambahan sebuah sistem pengenalan wajah yang cepat dan akurat ke dalam RPA akan semakin memperluas daftar hal-hal yang dapat dicapai dengan menggunakan RPA. Sistem pengenalan wajah yang menggunakan RPA bergantung pada *dataset* besar untuk menjadi akurat; bahwa semakin besar dataset, semakin baik algoritma tersebut melakukan pengenalan wajah (Kemelmacher-Shlizerman, Seitz, Miller, & Brossard, E., 2015). Ini adalah alasan mengapa *dataset* yang digunakan oleh perusahaan seperti Google, dan Facebook adalah milik pribadi perusahaan, dan umumnya tidak terbuka untuk umum (Parkhi, Vedaldi, & Zisserman, 2015)

OpenCV dirancang untuk melakukan komputasi dengan lebih efisien dan fokus yang kuat pada aplikasi waktu nyata di bidang visi komputer (Suryatali & V.B.Dharmadhikari, 2015). *OpenCV* diperkirakan telah memiliki lebih dari 47 ribu orang di komunitas penggunanya, dan diperkirakan jumlah unduhan melebihi 14 juta; dengan penggunaannya mulai dari seni interaktif, hingga robotika tingkat lanjut (Sathyabama, 2017).

Pengembangan aplikasi pengenalan wajah saat ini juga banyak dikembangkan seperti halnya penelitian yang telah dilakukan oleh Patel, Kumar, Garg, & Kumar (2018) dengan membangun dan mengimplementasikan sistem absensi siswa berbasis pengenalan wajah menggunakan IoT. Salah satu masalah utama ketika menerapkan pengenalan wajah akurat secara *real-time* akan membutuhkan biaya komputasi yang tinggi, namun hal ini lain halnya dengan penelitian yang telah dilakukan oleh Bolotnikova, Demirel, & Anbarjafari (2017) berjudul "*Real-time Ensemble Based Face Recognition System for NAO Humanoids Using Local Binary Pattern*", mereka mencapai sistem pengenalan wajah dengan biaya komputasi yang rendah dengan terlebih dahulu melatih algoritma; dan dengan menyimpang dari metode konvensional yang biasanya digunakan.

III. METODE PENELITIAN

Penelitian yang dilakukan merupakan penelitian terapan. Penelitian terapan adalah metodologi penelitian di mana para peneliti akan menganalisis masalah dunia nyata dan kemudian coba untuk mengimplementasikan solusi untuk memecahkan masalah tersebut (Mani, 2016). Ruang lingkup penelitian ini adalah:

- 1) Sistem pengenalan wajah ditambahkan sebagai fitur yang dapat dipilih.
- 2)

Algoritma pengenalan wajah tidak dirancang dan dibangun dari awal, melainkan mengandalkan *library open-source* bahasa pemrograman *Python* yang bernama *face_recognition*. 3) Sistem pengenalan wajah menggunakan kamera yang terhubung ke komputer, atau laptop.

Gambar 1. Proses Desain

Pada gambar 1 di atas menunjukkan tahapan penelitian: tahap *planning* untuk mendefinisikan spesifikasi sistem pengenalan wajah, batasannya dan kebutuhannya dengan jelas dan bagaimana interaksinya dengan sistem yang sudah ada; kedua melakukan analisis batasan dan kebutuhan yang ada untuk dimasukkan ke dalam sistem; tahap desain dengan model

seperti pada gambar 2 dan gambar 3.

Gambar 2. Model proses scan wajah.

Gambar 3. Model untuk mengenali wajah dan menghasilkan informasi pemilik wajah

Tahapan desain dibagi menjadi dua bagian yaitu (1) menjelaskan proses *scan* wajah untuk merekam wajah baik secara manual dan otomatis seperti gambar 2; dan (2) menjelaskan proses mengenali wajah dan menghasilkan informasi dari si pemilik wajah tersebut dari basis data sistem yang sudah ada, seperti gambar 3. Tahap *development* untuk membangun sistem; seperti *pseudocode* yang ditunjukkan pada gambar 4 dan gambar 5.

```

1 tolerance = 0.6
2 cam = cv2.VideoCapture(0)
3 Results = []
4 process_frame = True
5 srv = 'http://192.168.0.1'
6 api = FaceAPI(srv, credentials)
7
8 while True:
9     frame = cam.read()
10    cv2.imwrite("face.jpg", frame)
11    # Only process every other frame of video to save time
12    if process_frame:
13        try:
14            with open("face.jpg", "rb") as file:
15                image_encoded = base64.b64encode(file.read())
16                Results = api.faceSearch(tolerance, image_encoded)
17            except:
18                print "No Connection to Server"
19            process_frame = not process_frame
20
21    if Results is not None:
22        for result in Results:
23            if (result['name'] == 'Unknown') and (process_frame is False):
24                try:
25                    img = Image.open("face.jpg")
26                    img2 = img.crop(box, result['coordinates'])
27                    img2.save("face_cropped.jpg")
28                    with open("face_cropped.jpg", "rb") as file:
29                        image_encoded = base64.b64encode(file.read())
30                    values = {
31                        u'face_image': image_encoded,
32                        u'name': 'Identified (Unknown)',
33                        u'email': '-'
34                    }
35                    api.addUnknownFace(values, True)
36                except:
37                    print "An error occurred"
38                    coordinates = result['coordinates']
39                    cv2.put_overlay(coordinates)
40                    cv2.imshow('Scanning for Faces', frame)
41
42    if Q_key is pressed:
43        break
44
45 cam.release()
46 cv2.destroyAllWindows()
47

```

Gambar 4. Implementasi algoritma scan wajah.

```

1 tolerance = 0.6
2 cam = cv2.VideoCapture(0)
3 Results = []
4 process_frame = True
5 srv = 'http://192.168.0.1'
6 api = FaceAPI(srv, credentials)
7 timer = 10
8
9 while timer > 0:
10    timer -= 1
11    frame = cam.read()
12    cv2.imwrite("face.jpg", frame)
13    # Only process every other frame of video to save time
14    if process_frame:
15        try:
16            api = FaceAPI(srv, credentials)
17            with open("face.jpg", "rb") as image_file:
18                image_encoded = base64.b64encode(image_file.read())
19                Results = api.faceSearch(tolerance, image_encoded)
20            except:
21                print "No Connection to Server"
22            process_frame = not process_frame
23
24    cv2.put_overlay(coordinates)
25    cv2.imshow('Scanning for Faces', frame)
26
27    if Q_key is pressed:
28        break
29 cam.release()
30 cv2.destroyAllWindows()
31 return Results
32

```

Gambar 5. Implementasi algoritma mengenali wajah dan menghasilkan informasi pemilik wajah.

Pada gambar 4 dan gambar 5 tertera nilai toleransi sebesar 0.6 (Geitgey, 2018) sebagai acuan dasar. Tahapan berikutnya adalah pengujian dengan foto wajah resolusi 320x240; dan satu dataset untuk satu orang terdiri atas 5 (lima) foto yang disimpan dalam basis data di server. Tahap

terakhir adalah mengimplementasikan sistem RPA ini.

IV. HASIL DAN PEMBAHASAN

Gambar 6 menunjukkan sisi klien sistem pengenalan wajah.

Gambar 6. Penanganan data di sisi klien sebelum dikirimkan ke server

Seperti yang diilustrasikan oleh Gambar 7; algoritma pengenalan wajah akan menggunakan *array* yang dihasilkan untuk membandingkan antara dua wajah. Dengan membandingkan antara dua elemen dari *array* kita akan mendapatkan nilai yang disebut dengan "*Distance*" yaitu jarak seberapa besar bedanya antara dua muka. Nilai jarak tersebut berkisar antara 0 hingga 1. Semakin tinggi nilai jaraknya; semakin sedikit kemiripan di antara kedua wajah, dan sebaliknya.

Gambar 7. Cara algoritma menggunakan *encoding*

Gambar 8 dan 9 menunjukkan jendela yang digunakan oleh pengguna ketika menggunakan sistem pengenalan wajah. Jendela tersebut menunjukkan video yang direkam oleh kamera komputer dan memasang *overlay* berupa nama pemilik wajah yang dideteksi bersama dengan nama pemilik wajah tersebut.

Gambar 8. Jendela skan muka (menyamping)

Gambar 9. Cara algoritma menggunakan *encoding*

V. SIMPULAN

Berdasarkan hasil desain dan implementasi sistem pengenalan wajah untuk perangkat lunak RPA, dapat ditarik

kesimpulan bahwa kecepatan algoritma pengenalan wajah bergantung pada jumlah data set, semakin banyak jumlah *dataset* maka akan semakin berkurang kecepatannya, namun sebaliknya tingkat akurasi akan semakin tinggi. Berdasarkan data yang telah dikumpulkan dapat disimpulkan bahwa *OpenCV library* dapat diterapkan pada kasus nyata untuk mengenali wajah dengan tingkat akurasi 98% dan nilai toleransi pada 0.6 sesuai dengan hasil *benchmarking* yang sudah dilakukan oleh pengembang *library open source*. Kecepatan sistem adalah 2 gambar setiap detik; berarti kecepatannya kurang lebih 500ms untuk setiap *frame* video yang ditangkap.

Penerapan sistem pengenalan wajah ini sebagai fitur di dalam RPA memberikan nilai tambah dikarenakan jumlah kasus penggunaan yang dapat diotomasi oleh perangkat lunak RPA ini meningkat. Pada penelitian berikutnya Peneliti akan melakukan studi perbandingan sejumlah *open source library* pengenalan wajah guna menganalisis *library* mana yang efektif dan dapat digunakan pada situasi dan parameter tertentu.

VI. UCAPAN TERIMA KASIH

Para Penulis ingin mengucapkan terima kasih kepada Tuhan Yang Maha Kuasa atas berkah-Nya untuk penyelesaian

penelitian ini. Hal ini tidak akan mungkin terjadi tanpa bantuan dan dukungan dari beberapa pihak yakni: Keluarga para Penulis, Prof. Handoko Karjantoro, CPA. Sebagai Rektor Universitas Internasional Batam, Bapak Tony Wibowo, S.Kom., MMSI sebagai Ketua Program Studi Sistem Informasi Universitas Internasional Batam.

DAFTAR PUSTAKA

- Autor, D. H. (2015). Why Are There Still So Many Jobs? The History and Future of Workplace Automation. *Journal of Economic Perspectives*, 29(3), 3–30.
- 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.
- Geitgey, Adam. (2018). Ageitgey/face_recognition. *Github repository*. Diunduh di https://github.com/ageitgey/face_recognition tanggal 28 Agustus 2019.
- Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Ullah Khan, S. (2015). The Rise of “Big Data” on Cloud Computing: Review and Open Research Issues. *Information Systems*, 47(1), 98–115.
- Idrees, S. M., Alam, M. A., & Agarwal, P. (2018). A Study of Big Data and Its Challenges. *International Journal of Information Technology*, (1), 1–6.
- Kemelmacher-Shlizerman, I., Seitz, S., Miller, D., & Brossard, E. (2015). The MegaFace Benchmark: 1 Million Faces for Recognition at Scale, (1), 4873–4882.
- Lacity, M. C., Willcocks, L. P., & Craig, A. (2015). Robotic Process Automation at Telefónica O2. *MIS Quarterly Executive*, 15(2), 1–19.
- Mani, R. (2016). a Study on Consumer Buying Behavior Towards Two Wheeler Bikes. *International Journal of Application or Innovation in Engineering & Management*, 5(August), 52–56.
- Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep Face Recognition. In *Proceedings of the British Machine Vision Conference 2015* (pp. 1–12).
- 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.
- Poushter, J. (2016). Smartphone Ownership and Internet Usage Continues to Climb in Emerging Economies. *Pew Research Center*, 1–45.
- Sathyabama, K. (2017). True Alarm Using Artificial Intelligence and Internet of Things. *International Journal of Creative Research Thoughts*, 5(4), 2688–2693.
- 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).
- van der Aalst, W. M. P., Bichler, M., & Heinzl, A. (2018). Robotic Process Automation. *Business and Information Systems Engineering*, 60(4), 269–272.
- Vinay, A., Shekhara, V. S., Rituparna, J., Aggrawal, T., Balasubramanya Murthy, K. N., & Natarajan, S. (2015). Cloud Based Big Data Analytics Framework for Face Recognition in Social Networks Using Machine Learning. *Procedia Computer Science*, 50(1), 623–630.