



Deteksi Non-Spoofing Wajah pada Video secara Real Time Menggunakan Faster R-CNN

Sunario Megawan, Wulan Sri Lestari*, Apriyanto Halim

Fakultas Informatika, Program Studi Teknik Informatika, Universitas Mikroskil, Medan, Indonesia

Jl. M.H Thamrin No. 140, 20212, Medan, Indonesia

Email: ¹sunario@mikroskil.ac.id, ^{2,*}wulan.lestari@mikroskil.ac.id, ³apriyanto.halim@mikroskil.ac.id

Email Penulis Korespondensi: wulan.lestari@mikroskil.ac.id

Submitted: 19/04/2022; Accepted: 29/04/2022; Published: 30/04/2022

Abstrak—Deteksi *non-spoofing* wajah merupakan pekerjaan penting yang digunakan untuk menjamin keamanan otentikasi dengan melakukan analisis terhadap wajah yang ditangkap. *Spoofing* wajah merupakan proses pemalsuan wajah oleh orang lain untuk mendapatkan akses secara ilegal ke dalam sistem biometrik yang dapat dilakukan dengan menampilkan video atau gambar wajah seseorang di layar monitor atau menggunakan gambar hasil cetakan. Ada berbagai bentuk serangan yang dapat dilakukan pada sistem otentikasi wajah berupa sketsa wajah, foto wajah, video wajah dan masker 3D dari wajah. Serangan tersebut dapat terjadi karena, foto dan video wajah dari pengguna sistem otentikasi wajah sangat mudah didapatkan melalui internet atau kamera. Untuk menyelesaikan masalah tersebut, maka pada penelitian ini diusulkan model deteksi *non-spoofing* wajah pada video menggunakan *Faster R-CNN*. Hasil yang didapatkan pada penelitian ini berupa model *Faster R-CNN* yang dapat mendeteksi *non-spoof* dan *spooof* wajah secara *real time* menggunakan *Raspberry Pi* sebagai kamera dengan *frame rate* 1 *fps*.

Kata Kunci: Wajah; Non-Spoofing; Otentikasi; Raspberry Pi; Faster R-CNN.

Abstract—Face non-spoofing detection is an important job used to ensure authentication security by performing an analysis of the captured faces. Face spoofing is the process of fake faces by other people to gain illegal access to the biometric system which can be done by displaying videos or images of someone's face on the monitor screen or using printed images. There are various forms of attacks that can be carried out on the face authentication system in the form of face sketches, face photos, face videos and 3D face masks. Such attacks can occur because photos and videos of faces from users of the facial authentication system are very easy to obtain via the internet or cameras. To solve this problem, in this research proposes a non-spoofing face detection model on video using *Faster R-CNN*. The results obtained in this study are the *Faster R-CNN* model that can detect non-spoof and spoof face in real time using the *Raspberry Pi* as a camera with a frame rate of 1 *fps*.

Keywords: Face; Non-Spoofing; Authentication; Raspberry Pi; Faster R-CNN.

1. PENDAHULUAN

Deteksi *non-spoofing* wajah merupakan pekerjaan penting yang digunakan untuk menjamin keamanan otentikasi dengan melakukan analisis terhadap wajah yang ditangkap [1]. *Spoofing* wajah merupakan proses pemalsuan wajah oleh orang lain untuk mendapatkan akses secara ilegal ke dalam sistem biometrik yang dapat dilakukan dengan menampilkan video atau gambar wajah seseorang di layar monitor atau menggunakan gambar hasil cetakan. Ada berbagai bentuk serangan yang dapat dilakukan pada sistem otentikasi wajah berupa sketsa wajah, foto wajah, video wajah dan masker 3D dari wajah [2]. Serangan tersebut dapat terjadi karena, foto dan video wajah dari pengguna sistem otentikasi wajah sangat mudah didapatkan melalui internet atau kamera [3]. Oleh sebab itu, perlu dikembangkan sebuah sistem yang dapat diterapkan pada proses otentikasi wajah untuk membedakan wajah asli (*non-spoof*) dan palsu (*spooof*).

Penelitian yang dilakukan Boulkenafet *et al.* berfokus kepada pendeteksian *spoofing* pada gambar yang dilihat berdasarkan tekstur dari warna pada gambar tersebut [4]. Hasil pada penelitian tersebut didapatkan efektivitas yang lebih baik untuk dua *dataset* yang digunakan. Dimana *dataset* pertama didapatkan secara langsung melalui internet, sedangkan *dataset* kedua merupakan data primer yang dibuat secara langsung. Namun, pada penelitian tersebut hanya didasarkan pada gambar bukan pada video. Nagpal dan Dubey menggunakan *Convolutional Neural Network* (CNN) untuk *non-spoofing* wajah pada video [5]. Hasil penelitian tersebut menunjukkan nilai akurasi yang tinggi yaitu 94.42%. Namun, penelitian yang telah dilakukan masih sebatas pada dataset umum yang sudah direkam sebelumnya. Sehingga pada penelitian ini, diusulkan pengembangan aplikasi *non-spoofing* wajah dengan menggunakan video secara *real time*.

Pada perkembangan teknologi saat ini, sangat dimungkinkan untuk melakukan penangkapan video secara *real time*. Hal tersebut dapat terjadi karena adanya teknologi seperti Andruino, *Raspberry Pi*, dan lainnya. *Raspberry Pi* merupakan komputer dengan ukuran seperti kartu yang dipasangkan pada monitor atau televisi dan dapat menggunakan mouse serta keyboard. Alat ini memiliki kemampuan untuk dapat menjalankan komputasi, dan menggunakan bahasa pemrograman Python [6].

Pada penelitian ini, metode yang diusulkan adalah *Faster Region based Convolutional Neural Network* (*Faster R-CNN*). *Faster R-CNN* merupakan salah satu metode yang paling unggul dalam bidang deteksi objek [1]. Ada banyak penelitian yang sudah menggunakan *Faster R-CNN* pada bidang deteksi objek seperti Zhong *et al* untuk mendeteksi teks [7], Benjdira *et al* untuk mendeteksi mobil pada video [8], Sunario dan Wulan untuk mendeteksi *spoofing* wajah pada gambar [9] dan lain-lain. Sehingga kontribusi yang diusulkan pada penelitian ini



adalah sebuah aplikasi yang dapat membedakan wajah asli (*non-spoof*) dan palsu (*spoof*) dengan implementasi *Faster R-CNN* yang akan diuji secara *real time* menggunakan *Raspberry Pi* sebagai kamera.

2. METODOLOGI PENELITIAN

2.1 Studi Literatur

Pada tahap ini, dilakukan proses pengumpulan bahan penelitian yang berhubungan dengan video, *spoofing image*, *Faster R-CNN* dan *Raspberry Pi* berupa jurnal, konferensi dan buku penunjang penelitian lainnya.

2.1.1 Video

Video merupakan media elektronik yang digunakan untuk merekam, menggandakan, mengulang kembali, menyiarkan dan menampilkan media visual yang dapat bergerak. Perkembangan awal dari video dimulai dari sistem mekanik dari televisi dan secara cepat berubah menjadi CRT (*Cathode Ray Tube*) dan sekarang dikenal dengan nama tampilan *flat* [10]. Jenis – jenis representasi pada video [10]:

1. Cerita merupakan video yang bertujuan untuk memaparkan suatu cerita.
2. Dokumenter merupakan video yang bertujuan untuk merekam sebuah kejadian atau peristiwa dalam kehidupan nyata.
3. Berita merupakan video yang bertujuan untuk memarkan sebuah berita.
4. Pembelajaran merupakan video yang digunakan dengan tujuan memberikan materi pembelajaran agar mudah dipahami dan dapat diputar kembali.
5. Presentasi merupakan video yang bertujuan untuk mengkomunikasi ide atau gagasan.

Format – format dalam video terbagi ke dalam beberapa bagian, yaitu format video analog *encoding* terbagi ke dalam NTSC, PAL, SECAM dan format video analog elektrik terbagi ke dalam RF, *Composite Video*, *Component Video*, S-Video, dan RGB [10].

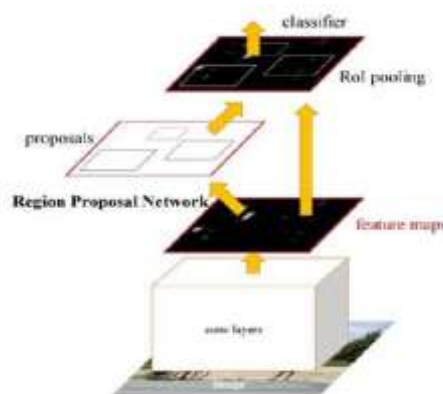
2.1.2 Spoofing Image

Spoofing image merupakan serangan yang ditujukan pada proses pengenalan wajah. Teknik ini dilakukan dengan menyamar menjadi seseorang dengan melakukan pemalsuan data dan mengambil keuntungan dari akses yang terlarang ke dalam sistem [11]. Oleh karena itu, diperlukan cara untuk mendeteksi terjadinya pemalsuan data pada gambar tersebut. Salah teknik yang digunakan untuk mendeteksi spoofing image adalah non-spoofing. Non-spoofing merupakan cara yang digunakan untuk mencegah terjadi penyerangan dengan menggunakan beberapa cara berupa mengekstrak terlebih dahulu ciri khusus dari pengguna yang asli dan lainnya [12].

2.1.3 Faster R-CNN

Faster R-CNN merupakan metode deteksi objek dalam bentuk generik berbasis wilayah R-CNN yang lebih cepat dalam mendeteksi objek. *Fast R-CNN* adalah metode yang tetap menggunakan kombinasi dari CNN dan RPN sebagai arsitektur utama. Metode ini menggunakan arsitektur dari R-CNN dengan penambahan pada bagian awal masukkan ditambahkan proses *convolutional* dan *max pooling layer* untuk menghasilkan *convfeature map*. Lalu, untuk setiap objek yang ditentukan dilakukan perhitungan RoI (*Region of Interest*) dan *pooling layer* untuk mengambil bagian dari *feature map*. Dari hasil ini nantinya akan dibagi ke dalam dua bagian, yang berupa bagian latar belakang dan bagian dari objek.

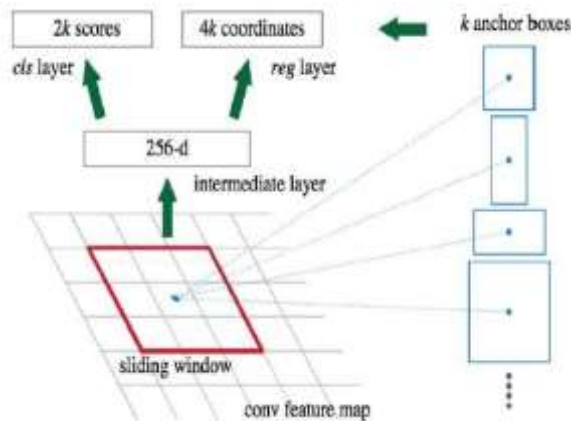
Pada bagian RoI *pooling layer* dilakukan pembagian wilayah sesuai dengan yang diinginkan misalnya 7x7. Setelah dilakukan pembagian wilayah, dilakukan proses *max-pooling* untuk menghasil bagian dari sel. Pada bagian lainnya dilakukan juga proses *convolution* untuk setiap piksel yang ada. Proses ini nantinya akan digabungkan kembali dengan proses RoI *pooling layer* dan dilakukan pengecekan manakah bagian yang merupakan objek dan mana bagian yang merupakan latar belakang [13]. Gambar 1 merupakan arsitektue *Faster R-CNN*.



Gambar 1. Arsitektur *Faster R-CNN* [14]

Faster R-CNN dibagi menjadi dua bagian penting [14]:

1. *Region Proposal Network* (RPN) berguna untuk mencari pada input gambar kemungkinan lokasi dari objek secara cepat. Posisi dari objek yang ada pada gambar memiliki kemungkinan objek dibatasi dari wilayah yang diketahui sebagai *Region of Interest* (RoI). RoI diperkenalkan oleh Ross Girshick [13] sebagai sebuah pendekatan deteksi objek menggunakan *deep learning*. Input yang digunakan pada layer. RoI adalah *feature maps* yang merupakan *output* dari CNN dengan *multiple convolution layers* dan *max pooling layers*. Sebuah $N \times N$ matriks dibuat dengan membagi ruang *feature map* menjadi RoI. Kolom pertama merupakan indeks dari citra dan sisa kolom lainnya merupakan koordinat dari RoI dimulai dari koordinat paling kiri atas hingga kiri bawah. RoI yang sudah ditentukan disebut sebagai *region proposal*. Gambar 2 merupakan proses RPN.



Gambar 2. Proses RPN [14]

Pada proses RPN, awalnya citra masukan diproses dalam jaringan saraf konvolusi untuk menghasilkan *feature map*. *Sliding window* ditempatkan pada setiap *feature map* dengan ukuran $N \times N$ sesuai dengan setiap *anchors sliding window* yang dibentuk. Setiap *anchor* akan memiliki titik pusat yang sama, namun memiliki *aspect ratios* dan *scaling factor* yang berbeda.

2. *Classifier* merupakan proses yang digunakan untuk mengklasifikasi RoI yang sudah diidentifikasi pada RPN kedalam kelas atau target yang sesuai. Teknik yang digunakan pada tahap ini adalah CNN.

2.1.3 Raspberry Pi

Raspberry Pi adalah sebuah komputer papan tunggal (*single-board computer*) atau SBC berukuran kartu kredit. *Raspberry Pi* telah dilengkapi dengan semua fungsi layaknya sebuah komputer lengkap, menggunakan SoC (*System-on-a-chip*) ARM yang dikemas dan diintegrasikan diatas PCB. Perangkat ini menggunakan kartu SD untuk booting dan penyimpanan jangka panjang [15].

Raspberry Pi memiliki dua model yaitu model A dan model B. Secara umum *Raspberry Pi* Model B, 512MB RAM. Perbedaan model A dan B terletak pada memory yang digunakan, Model A menggunakan memory 256 MB dan model B 512 MB. Selain itu model B juga sudah dilengkapi dengan ethernet port (kartu jaringan) yang tidak terdapat di model A. Desain *Raspberry Pi* didasarkan seputar SoC (System-on-a-chip) Broadcom BCM2835, yang telah menanamkan prosesor ARM1176JZF-S dengan 700 MHz, VideoCore IV GPU, dan 256 Megabyte RAM (model B). Penyimpanan data didisain tidak untuk menggunakan hard disk atau solid-state drive, melainkan mengandalkan kartu SD (SD memory card) untuk booting dan penyimpanan jangka panjang. Hardware *Raspberry Pi* tidak memiliki real-time clock, sehingga OS harus memanfaatkan timer jaringan server sebagai pengganti. Namun komputer yang mudah dikembangkan ini dapat ditambahkan dengan fungsi *real-time* (seperti DS1307) dan banyak lainnya, melalui saluran GPIO (*General-purpose input/output*) via antarmuka I²C (*Inter- Integrated Circuit*). *Raspberry Pi* bersifat open source (berbasis Linux), *Raspberry Pi* bisa dimodifikasi sesuai kebutuhan penggunaannya. Sistem operasi utama *Raspberry Pi* menggunakan Debian GNU/Linux dan bahasa pemrograman *Python*. Salah satu pengembang OS untuk *Raspberry Pi* telah meluncurkan sistem operasi yang dinamai Raspbian, Raspbian diklaim mampu memaksimalkan perangkat *Raspberry Pi*. Sistem operasi tersebut dibuat berbasis Debian yang merupakan salah satu distribusi Linux OS [15].

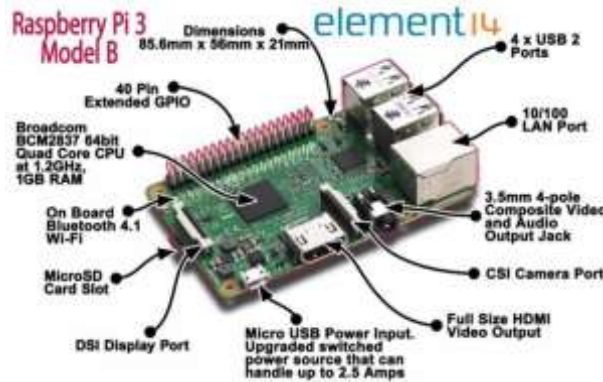
A. Raspberry Pi 3

Raspberry Pi 3 merupakan generasi ketiga dari keluarga *Raspberry Pi*. *Raspberry Pi 3* memiliki RAM 1GB dan grafis *Broadcom VideoCore IV* pada frekuensi *clock* yang lebih tinggi dari sebelumnya yang berjalan pada 250MHz. *Raspberry Pi 3* menggantikan *Raspberry Pi 2* model B pada bulan Februari 2016. Kelebihannya dibandingkan dengan *Raspberry Pi 2* adalah:

- I. A 1.2GHz 64-bit *quad-core* ARMv8 CPU
- II. 802.11n Wireless LAN
- III. Bluetooth 4.1

IV. *Bluetooth Low Energy (BLE)*

Sama seperti *Pi 2*, *Raspberry Pi 3* juga memiliki 4 USB port, 40 pin GPIO, *Full HDMI* port, Port Ethernet, *Combined 3.5mm audio jack and composite video*, *Camera interface (CSI)*, *Display interface (DSI)*, slot kartu *Micro SD* (Sistem tekan-tarik, berbeda dari yang sebelumnya ditekan-tekan), dan *VideoCore IV 3D graphics core*. *Raspberry Pi 3* memiliki factor bentuk identik dengan *Raspberry Pi 2* dan memiliki kompatibilitas lengkap dengan *Raspberry Pi 1* dan *2*. *Raspberry Pi 3* juga direkomendasikan untuk digunakan bagi mereka yang ingin menggunakan *Pi* dalam proyek-proyek yang membutuhkan daya yang sangat rendah. Gambar 3 merupakan Tampilan *Raspberry Pi 3 Model B*.



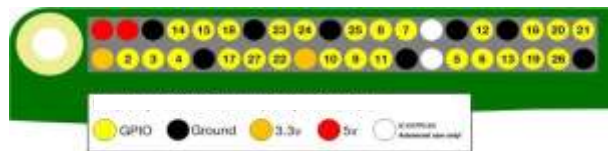
Gambar 3. Tampilan Raspberry Pi 3 Model B

B. *GPIO Raspberry Pi 3*

GPIO merupakan sederet pin yang terdiri dari 40 pin dengan berbagai fungsi. Salah satu fitur yang kuat dari *Raspberry Pi* adalah deretan GPIO (tujuan umum input / output) pin di sepanjang tepi atas pin board. These adalah antarmuka fisik antara *Pi* dan dunia luar. Pada tingkat yang paling sederhana, Anda dapat menganggap mereka sebagai switch yang Anda dapat mengaktifkan atau menonaktifkan (input) atau bahwa *Pi* dapat mengaktifkan atau menonaktifkan (output) [15].

Dari 40 pin, 26 pin GPIO dan yang lain adalah pin *power* atau *ground* (ditambah dua pin ID EEPROM yang tidak harus anda gunakan). Anda dapat memprogram pin untuk berinteraksi dengan cara yang menakjubkan dengan dunia nyata. Input tidak harus berasal dari saklar fisik; itu bisa menjadi masukan dari sensor atau sinyal dari komputer lain atau perangkat, misalnya. output juga dapat melakukan apa saja, dari menyalakan LED untuk mengirim sinyal atau data ke perangkat lain [15].

Jika *Raspberry Pi* adalah pada jaringan, Anda dapat mengontrol perangkat yang terhubung padanya dari mana saja (Tidak secara harfiah di mana saja, tentu saja. Anda perlu hal-hal seperti akses ke jaringan, jaringan yang mampu perangkat komputasi, dan listrik.) dan perangkat-perangkat dapat mengirim data kembali. Konektivitas dan kontrol dari perangkat fisik melalui internet adalah hal yang sangat kuat dan menarik, dan *Raspberry Pi* ideal untuk ini. GPIO *Raspberry Pi 3* dapat dilihat pada Gambar 4 [15].



Gambar 4. *Raspberry Pi* GPIO Pin

Penjelasan lebih lanjut mengenai fungsi masing-masing PIN GPIO pada *Raspberry Pi 3* adalah sebagai berikut:

Pin	Label	Pin	Label
01	3.3v	04	5v
02	GPIO12 (SDA2 - I2C)	05	Ground
03	GPIO13 (SCL2 - I2C)	06	GPIO14
04	Ground	07	GPIO15
05	GPIO16 (GPIO_06)	08	GPIO16
06	GPIO17 (GPIO_07)	09	Ground
07	GPIO18 (GPIO_08)	10	GPIO17
08	3.3v	11	GPIO18
09	GPIO19 (GPIO_09)	12	Ground
10	GPIO20 (GPIO_10)	13	GPIO19
11	GPIO21 (GPIO_11)	14	GPIO20
12	Ground	15	GPIO21
13	ID_00 (1.5k Ohm pull-up)	16	GPIO22
14	GPIO22 (GPIO_12)	17	GPIO23
15	GPIO23 (GPIO_13)	18	Ground
16	GPIO24 (GPIO_14)	19	GPIO24
17	GPIO25 (GPIO_15)	20	GPIO25
18	GPIO26 (GPIO_16)	21	GPIO26
19	GPIO27 (GPIO_17)	22	GPIO27
20	GPIO28 (GPIO_18)	23	GPIO28
21	GPIO29 (GPIO_19)	24	GPIO29
22	GPIO30 (GPIO_20)	25	GPIO30
23	GPIO31 (GPIO_21)	26	GPIO31
24	GPIO32 (GPIO_22)	27	GPIO32
25	GPIO33 (GPIO_23)	28	GPIO33
26	GPIO34 (GPIO_24)	29	GPIO34
27	GPIO35 (GPIO_25)	30	GPIO35
28	GPIO36 (GPIO_26)	31	GPIO36
29	GPIO37 (GPIO_27)	32	GPIO37
30	GPIO38 (GPIO_28)	33	GPIO38
31	GPIO39 (GPIO_29)	34	GPIO39
32	GPIO40 (GPIO_30)	35	GPIO40
33	GPIO41 (GPIO_31)	36	GPIO41
34	GPIO42 (GPIO_32)	37	GPIO42
35	GPIO43 (GPIO_33)	38	GPIO43
36	GPIO44 (GPIO_34)	39	GPIO44
37	GPIO45 (GPIO_35)	40	GPIO45
38	GPIO46 (GPIO_36)		
39	GPIO47 (GPIO_37)		
40	GPIO48 (GPIO_38)		

Gambar 5. *Raspberry Pi 3 Model B* GPIO 40 Pin Block Pinout

2.2 Pengumpulan Data

Data yang digunakan pada penelitian ini adalah data primer yang diambil dengan menggunakan alat berupa kamera dalam bentuk format video. Adapun kriteria video yang digunakan yaitu:

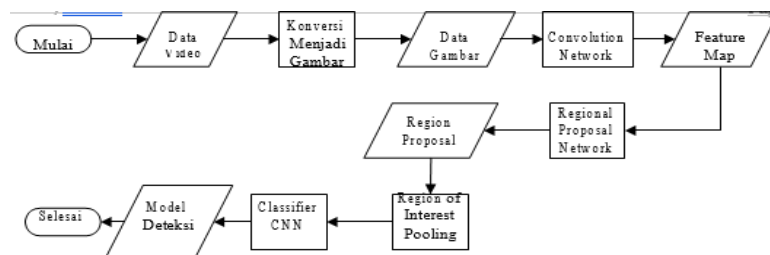
- Pengambilan *data testing* dilakukan secara *real time* menggunakan *Raspberry Pi*.
- Video *data training* terdiri dari 1 wajah dengan latar belakang tanpa gambar manusia dengan durasi masing-masing video 12 detik.

2.3 Pengembangan Alat Bantu

Tahap ini akan dihasilkan sistem yang menerapkan *Faster R-CNN* menggunakan bahasa pemrograman Python, Tensorflow dan Keras. Alat bantu yang dikembangkan digunakan untuk menjalankan proses *training* dalam deteksi wajah manusia pada video serta untuk mendeteksi *non-spoofing* dan *spoofing* wajah. *Raspberry Pi* digunakan sebagai *module* kamera untuk proses *testing* dalam mendeteksi *non-spoofing* dan *spoofing* wajah secara *real time*. Pada penelitian ini terdapat 2 proses penting yaitu tahap *training* dan tahap *testing*.

2.3.1 Tahap Training

Tahap *training* digunakan untuk membangun model implementasi algoritma *Faster R-CNN* dan data *training*. Gbr. 4 merupakan langkah-langkah yang dilakukan pada tahap *training*.



Gambar 6. Flowchart Tahap Training

2.3.2 Tahap Testing

Tahap *testing* digunakan untuk menguji model yang dihasilkan pada tahap *training*

3. HASIL DAN PEMBAHASAN

3.1 Hasil

Hasil yang dibahas pada bagian ini adalah spesifikasi perangkat yang digunakan, rincian dataset yang digunakan, proses *training* data dan *testing* data dalam mengimplementasikan *Faster R-CNN* untuk mendeteksi *non-spoof* wajah dan *spoof* wajah pada video secara *realtime* menggunakan *Raspberry Pi*.

3.1.1 Spesifikasi Perangkat Yang Digunakan

Untuk mengimplementasi teknik *Faster R-CNN* dengan data video maka dibutuhkan spesifikasi perangkat yang mendukung agar proses *training* dan *testing* dapat berjalan dengan baik dan lebih efisien. Tabel I. merupakan spesifikasi *hardware* dan *software* yang digunakan.

Tabel 1. Spesifikasi Perangkat yang Digunakan

Kriteria	Spesifikasi
Sistem Operasi	Windows 10 64-Bit
Processor	Amd Ryzen 3700x
Memory	32GB DDR4 PC 3200
Graphics	NVIDIA GTX 1660 TI 6GB DDR5
GPU Engine	NVIDIA Cuda Core 1536
Bahasa Pemrograman	Python
IDE Pemrograman	IDLE
Library	Tensorflow, Keras dan OpenCV
PC Camera	Raspberry Pi

3.1.2 Data Yang Digunakan

Data *training* yang digunakan pada penelitian ini adalah data *primer* yang diambil dengan menggunakan kamera dalam bentuk format video. Data *training* yang disediakan dalam bentuk video dengan 1 wajah menghadap depan dengan latar belakang gambar tanpa foto manusia. Adapun rincian data *training* yang digunakan dapat dilihat pada Tabel 2.

Tabel 2. Data Yang Digunakan

Kelas	Jumlah Video	Jumlah Frame/Video	Keterangan	Durasi
<i>Non-spoof</i>	4	293	2 video wanita dan 2 video laki- laki	12 Detik
<i>Spoof</i>	4	293	2 video wanita dan 2 video laki- laki	12 Detik

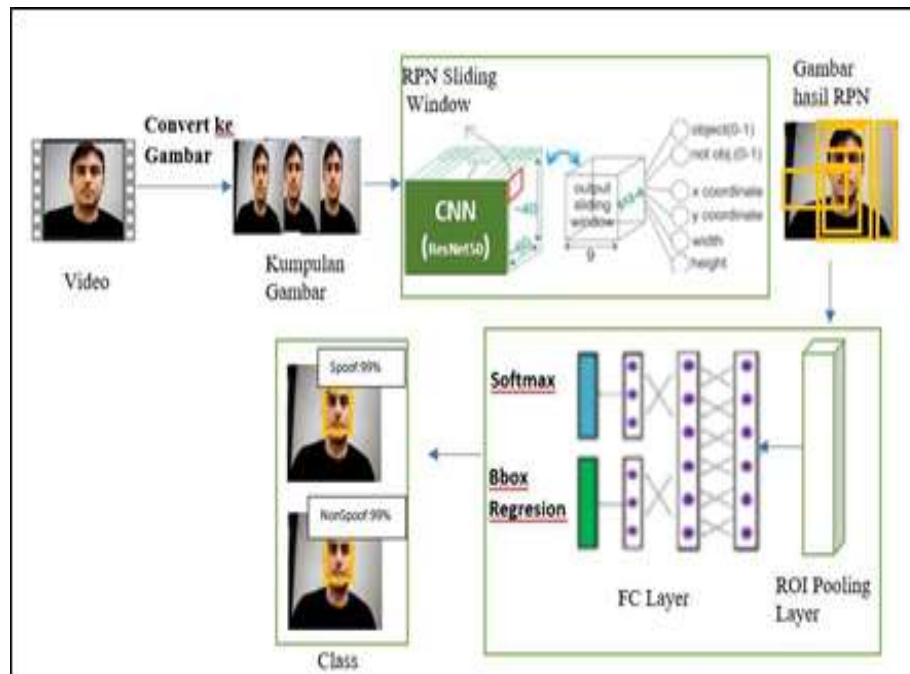
Data *testing* yang digunakan pada penelitian ini adalah gambar yang ada di *handphone* dan manusia yang langsung didetek secara *real time* menggunakan *Raspberry Pi*. Gambar 8 merupakan gambar yang digunakan untuk *testing*.



Gambar 7. Gambar Testing

3.1.3 Proses Training Data

Proses *training* digunakan untuk menghasilkan model yang akan dipakai pada saat proses *testing*. Gambar 8 merupakan arsitektur *Faster R-CNN* yang digunakan.



Gambar 8. Arsitektur *Faster R-CNN*

Proses yang dilakukan pada tahap *training* adalah sebagai berikut:

1. Input Video

Jumlah data video yang digunakan pada proses *training* terdiri dari 4 video *non-spoof* dan 4 video *spoof*.

2. Konversi Video

Pada tahap ini seluruh video dikonversi menjadi gambar menggunakan *library* OpenCV dan disimpan ke folder sesuai dengan jenis “*Class*” -nya masing-masing. Total gambar yang dihasilkan adalah 2344 yang terdiri dari 1172 gambar *Non-Spoof* dan 1172 gambar *Spoof*.

3. Training Data

Proses *training* merupakan proses yang bertujuan untuk melatih algoritma *Faster R-CNN* dalam mengenali *dataset* dan membentuk sebuah model berdasarkan pelatihan tersebut. Sebelum melakukan proses *training*, buat sebuah file *txt* yang berisi nama gambar, ukuran *bounding box* dan *class* seperti Gambar 9.

```
Data Training.txt - Notepad
File Edit Format View Help
DataPenelitian/RealImage/Mama1.jpg,374,147,517,300,non-spoof
DataPenelitian/RealImage/Mama2.jpg,375,147,517,299,non-spoof
DataPenelitian/RealImage/Mama3.jpg,372,147,518,303,non-spoof
DataPenelitian/RealImage/Mama4.jpg,375,148,519,302,non-spoof
DataPenelitian/RealImage/Mama5.jpg,374,148,517,301,non-spoof
DataPenelitian/RealImage/Mama6.jpg,374,148,518,302,non-spoof
DataPenelitian/RealImage/Mama7.jpg,374,148,518,302,non-spoof
DataPenelitian/RealImage/Mama8.jpg,373,147,519,303,non-spoof
DataPenelitian/RealImage/Mama9.jpg,374,148,517,301,non-spoof
DataPenelitian/RealImage/Mama10.jpg,374,148,517,301,non-spoof
DataPenelitian/RealImage/Mama11.jpg,374,148,518,302,non-spoof
DataPenelitian/RealImage/Mama12.jpg,373,147,518,302,non-spoof
DataPenelitian/RealImage/Mama13.jpg,375,147,518,300,non-spoof
DataPenelitian/RealImage/Mama14.jpg,374,146,519,301,non-spoof
DataPenelitian/RealImage/Mama15.jpg,373,146,519,302,non-spoof
DataPenelitian/RealImage/Mama16.jpg,375,146,518,299,non-spoof
DataPenelitian/RealImage/Mama17.jpg,373,146,518,301,non-spoof
DataPenelitian/RealImage/Mama18.jpg,375,146,519,300,non-spoof
DataPenelitian/RealImage/Mama19.jpg,374,146,519,301,non-spoof
DataPenelitian/RealImage/Mama20.jpg,374,147,518,301,non-spoof
DataPenelitian/RealImage/Mama21.jpg,375,147,519,301,non-spoof
DataPenelitian/RealImage/Mama22.jpg,375,148,519,302,non-spoof
DataPenelitian/RealImage/Mama23.jpg,375,147,519,301,non-spoof
DataPenelitian/RealImage/Mama24.jpg,375,147,519,301,non-spoof
DataPenelitian/RealImage/Mama25.jpg,375,147,520,302,non-spoof
DataPenelitian/RealImage/Mama26.jpg,375,148,518,301,non-spoof
DataPenelitian/RealImage/Mama27.jpg,375,148,518,301,non-spoof
DataPenelitian/RealImage/Mama28.jpg,376,149,517,300,non-spoof
```

Gambar 9. File Teks Data Training

Setelah file teks selesai, maka tentukan parameter yang akan digunakan. Pada penelitian ini untuk membangun model *Faster R-CNN* digunakan jumlah data latih sebanyak 2344 gambar. Arsitektur CNN yang digunakan adalah *Resnet50* yang merupakan model yang sudah di-*train* menggunakan *ImageNet Dataset* agar dapat menghasilkan *feature extraction* yang baik. Jumlah *Anchor* yang digunakan adalah 9 sesuai dengan default *Faster R-CNN*. *Anchor* merupakan bagian penting yang digunakan untuk menentukan bagian penting dari gambar (*region proposal*) yang akan dimasukkan ke *RPN*. *Optimizer* yang digunakan *Adam* dengan nilai *Learning Rate* sebesar 0.00001. Selain itu *Stochastic Gradient Descent* digunakan untuk mengoptimasi *convolution layer*, *RPN weight* dan *fully connected layer*. *Epoch length* yang digunakan 50.000 dengan jumlah *epoch* 71.

Berdasarkan tahap training yang telah dilakukan, nilai akurasi tertinggi ada pada epoch ke-68 dengan nilai 99.25%. Pada saat pembuatan model di tahap *training*, nilai *weight* akan diupdate setiap kali nilai akurasi pada *epoch* berjalan lebih besar daripada nilai akurasi *epoch* sebelumnya. Hasil akurasi *training* dapat dilihat pada gambar grafik di bawah ini.



Gambar 10. Grafik Akurasi Hasil Training

3.2 Pembahasan

Setelah melakukan proses *training* maka pada bagian ini akan dilakukan proses *testing* untuk menguji apakah model *training* dapat mendeteksi *non-spoof* dan *spoof* wajah secara *real time* menggunakan *Raspberry Pi*.

3.2.1 Proses Testing Data

Tahap ini dilakukan proses pengujian (*testing*) terhadap model yang sudah dibangun pada tahap *training*. Proses *testing* dilakukan dengan menggunakan *Raspberry Pi* sebagai kamera yang terkoneksi dengan model *training* yang sudah dibangun. Kamera kemudian diarahkan ke wajah manusia dan gambar wajah yang ada *handphone* secara *real-time* dengan pengaturan *frame* gambar 640x480 piksel dan 1fps. Berikut tahapan proses *testing* yang dilakukan:

1. Arahkan kamera Raspberry Pi ke wajah dan gambar wajah secara *real time*, sebagai contoh Gambar 11.



Gambar 11. Data *Testing*

2. Memanggil model *training* seperti yang terlihat pada potongan program Gambar 12.

```
# define the base network (resnet here, can be VGG, Inception, etc)
shared_layers = nn.nn_base(img_input, trainable=True)

# define the RPN, built on the base layers
num_anchors = len(C.anchor_box_scales) * len(C.anchor_box_ratios)
rpn_layers = nn.rpn(shared_layers, num_anchors)

classifier = nn.classifier(feature_map_input, roi_input, C.num_rois, nb_classes=
model_rpn = Model([img_input, rpn_layers])
model_classifier_only = Model([feature_map_input, roi_input], classifier)
model_classifier = Model([feature_map_input, roi_input], classifier)

print('Loading weights from {}'.format(C.model_path))
model_rpn.load_weights(C.model_path, by_name=True)
model_classifier.load_weights(C.model_path, by_name=True)

model_rpn.compile(optimizer='sgd', loss='mse')
model_classifier.compile(optimizer='sgd', loss='mse')

all_imgs = []
plasses = {}

hbox_threshold = 0.8

visualize = True

cap = cv2.VideoCapture('http://raspberrypi:8080/?action=stream')
q=0
```

Gambar 12. Pemanggilan Model *Training*

3. Proses selanjutnya adalah memasukkan gambar hasil konversi kedalam model untuk dilakukan deteksi wajah dan menentukan jenis kelas (*spoof* atau *non-spoof*) pada masing-masing gambar beserta dengan nilai akurasi. Gambar 14 merupakan hasil proses *testing* yang dilakukan.



Gambar 13. Hasil *Testing*

Berdasarkan hasil *testing* yang dilakukan maka model dapat mendeteksi dan mengklasifikasi wajah dengan baik. Pada saat *testing* diperoleh nilai akurasi sebesar 92% untuk deteksi wajah *non-spoof* dan 96% untuk deteksi wajah *spoof* secara *real time*.

4. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, maka dapat diperoleh kesimpulan bahwa model implementasi *Faster R-CNN* yang diusulkan dapat mendeteksi dan mengklasifikasi *non-spoof* dan *spoof* wajah pada video dengan akurat secara *real-time* menggunakan *Raspberry Pi* sebagai kamera dengan *frame rate* 1 *fps*.



REFERENCES

- [1] A. Liu, J. Wan, S. Escalera, H. J. Escalante and Z. Tan, “Multi-modal Face Anti-spoofing Attack Detection Challenge at CVPR2019,” IEEE, pp. 1601-1610, 2019.
- [2] S. Liu, P. C. Yuen, S. Zhang and G. Zhao, “3D Mask Face Anti-Spoofing with Remote Photoplethysmography,” Springer International Publishing AG, pp. 85-100, 2016.
- [3] X. Song, X. Zhao, L. Fang and T. Lin, “Discriminative Representation Combinations for Accurate Face Spoofing Detection,” Elsevier, vol. 85, pp. 220-231, 2019.
- [4] Z. Boulkenafet, J. Komulainen. J, and A. Hadid, “Face Spoofing Detection Using Colour Texture Analysis,” IEEE Transactions on Information Forensics and Security, vol. 11(8), 2016.
- [5] C. Nagpal and S. R. Dubey, “A Performance Evaluation of Convolutional Neural Networks for Face Anti Spoofing,” International Joint Conference on Neural Networks (IJCNN), 2019.
- [6] Raspberry Pi Web, “Raspberrypi.org,” 2012. [Online]. Available: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>. [Accessed 22 Februari 2020].
- [7] Z. Zhong, L. Sun and Q. Huo, “An Anchor-Free Region Proposal Network for Faster R-CNN based Text Detection Approaches,” International Journal on Document Analysis and Recognition (IJ DAR) vol. 22, pp. 315–327, 2019. [8]
- [8] B. Benjdira, T. Khurshed, A. Koubaa, A. Ammar, and K. Ouni, “Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3,” arXiv:1812.10968v1 [cs.RO], 2018.
- [9] M. Sunario and W. S. Lestari, “Deteksi Spoofing Wajah Menggunakan Faster R-CNN dengan Arsitektur Resnet50 pada Video,” *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol.9, pp.261-267, 2020.
- [10] H. Zettl, “Video Basics. 6 ed,” USA: Wadsworth, 2010.
- [11] J. Maatta, A. Hadid, and M. Pietikainen, “Face spoofing detection from single images using texture and local shape analysis,” *IET Digital Library*, 1(1), pp. 3-10, 2012.
- [12] A. Anjos, J. Komulainen, A. Hadid, and M. Pietikainen, “Face Anti-spoofing: Visual Approach. In: Handbook of Biometric Anti-Spoofing,” London: Springer, pp. 65-82, 2014.
- [13] R. Girshick, “Fast R-CNN. s.l.,” Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015.
- [14] S. M. Abbas and D. S. N. Singh, “Region-based Object Detection and Classification using Faster R-CNN,” Ghaziabad, 4th International Conference on Computational Intelligence & Communication Technology (CICT), 2018.
- [15] A. E. Putra, “Mengenal Raspberry Pi. [Online]”, 2012. Available at: <http://agfi.staff.ugm.ac.id/blog/index.php/2012/08/mengenal-raspberry-pi/> [Accessed 22 February 2019].