PERANCANGAN SISTEM KEAMANAN RUMAH MENGGUNAKAN DETEKSI WAJAH BERBASIS MACHINE LEARNING MENGGUNAKAN TENSORFLOW

Designing a Home Security System Using Machine Learning Based Face Detection Using Tensorflow

Afrizal[1], Setia Juli Irzal Ismail [2], Gandeva Bayu Satrya[3]
[1,2,3] Universitas Telkom, Bandung
afrizal@students.telkomuniversity.ac.id[1], julismail@telkomuniversity.ac.id[2], gandevabs@telkomuniversity.ac.id[3]

Abstrak : Pengenalan wajah atau face recognition adalah sebuah teknologi yang mampu mengidentifikasi dan mengklasifikasi wajah dengan berbagai metode yang dapat di lakukan. Sistem ini di bangun pada pintu rumah yang berfungsi sebagai tambahan keamanan rumah, yang dapat mengirimkan notifikasi berupa pesan melalui aplikasi telegram apabila ada seseorang berada di depan pintu rumah baik itu wajah di kenali atau wajah tidak di kenali. Jika kamera menangkap objek, sistem akan mendeteksi apakah objek tersebut wajah atau bukan, dan jika sistem mendeteksi wajah, selanjutnya akan di lakukan proses verifikasi apakah wajah tersebut di kenali atau tidak dengan cara membandingkan data wajah dengan model database yang sudah di siapkan. Sistem di bangun menggunakan tensorflow berbasis machine learning dengan metode transfer learning, dan pengambilan objek menggunakan modul raspberry pi kamera v2 8MP.

Kata Kunci: Raspberry Pi, Machine Learning , Face Recognition, Transfer Learning, Telegram

Abstract : Face recognition is a technology that is able to identify faces with a variety of methods. This system is built on the door house which serves as additional home security. Also which can send notifications in the form of messages via the telegram application, if there is someone at the front door whether the face is recognized or the face is not recognized. If the camera captures the object, the system will identify whether the object has a face or not, and if the system detects faces, the next process will be carried out to determine whether the face is recognized or not by comparing face data with the database model database that has been prepared. This system was built using machine learning based on tensorflow with transfer learning methods, and object captured using the 8MP v2 camera raspberry pi module.

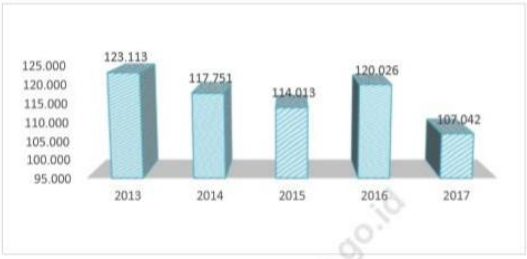
Keywords: Raspberry Pi, Machine Learning, Face Recognition, Transfer Learning, Telegram

.

1. Pendahuluan

Rumah adalah sebuah bangunan yang di gunakan sebagai tempat tinggal dan lain –lainnya seperti menyimpan harta benda atau barang – barang lainnya. Rutinitas sehari – hari memaksa pemilik rumah untuk meninggalkan rumah dan rumah pun menjadi kosong, contohnya pada saat bekerja atau berbelanja maka rumah di tinggalkan dalam keadaan kosong. Hal ini membuat rumah sangat rentan terjadi pembobolan dan pencurian harta benda, bahkan terkadang pembobolan terjadi pada siang hari dan kerugian mencapai hingga ratusan juta. Seperti yang di alami oleh seorang warga kabupaten Bekasi[1] yang rumahnya di bobol pada siang hari dan kerugian mencapai 500 juta.

Pada Desember tahun 2018 Badan Pusat Statistik (BPS) merilis statistik kriminal 2018 yang memuat beberapa informasi. Salah satunya adalah Jumlah kejadian kejahatan terhadap hak/milik tanpa penggunaan kekerasan, seperti pencurian rumah dengan cara membobol rumah tersebut selama periode 2013–2017[2], berikut adalah rangkuman data dari BPS yang di sajikan



Gambar 1 grafik kejahatan terhadap hak/milik tanpa penggunaan kekerasan periode 2013–2017

Berdasarkan gambar 1 grafik kejahatan terhadap hak/milik tanpa penggunaan kekerasan periode 2013-2017 bahwa kejahatan terhadap hak/milik tanpa penggunaan kekerasan tahun 2017 di banding tahun 2016 mengalami penurunan sebesar 12.984 kasus[2] Berdasarkan hal tersebut maka muncul sebuah inisiatif untuk menciptakan sebuah solusi bagi pemilik yang sering meninggalkan rumah dalam keadaan kosong agar dapat mengetahui siapa yang ada di pintu rumah. Dalam hal ini dirancanglah sebuah sistem keamanan rumah dengan cara deteksi wajah berbasis machine learning. Alat ini akan mendeteksi wajah orang yang berada di pintu rumah dankemudian akan mengirimkan notifikasi ke pemilik rumah tentang keberadaan orang tersebut, baik itu wajah yang di kenal atau tidak. Notifikasi berupa pesan menggunakan bot telegram, contoh notifikasi adalah sebagai berikut: jika wajah di kenali sebagai anak maka sistem ini akan mengirimkan notifikasi "Anak Terdeteksi", jika wajah yang tidak di kenali akan mengirimkan notifikasi " Wajah Tidak Di kenali Terdeteksi" .Untuk membangun sistem ini di gunakan library python bernama tensorflow yang berguna untuk membuat model machine learning, Model tersebut di buat menggunakan metode transfer learning dengan dataset berupa sampel wajah. Transfer learning adalah suatu metode yang yang menggunakan model yang sudah ada sebagai dasar untuk membuat model baru untuk menyelesaikan masalah yang lain dengan cara menjadi starting point, memodifikasi, dan mengupdate parameter yang di perlukan sehingga terbentuklah sebuah model baru[3].

2. Tinjauan Pustaka

Berikut ini adalah teori yang digunakan dalam penyusunan Proyek Akhir ini.

2.1 Machine Learning

Machine Learning (ML) adalah sebuah bidang keilmuan yang berfokus untuk mengembangkan sebuah sistem yang dapat belajar sendiri menggunakan data tanpa harus memprogram berkali - kali. sehingga program komputer bersifat dinamis bergantung dengan kelengkapan data yang di miliki[1]. Machine Learning sangat berguna untuk menyelesaikan kasus yang berubah–ubah dalam waktu tertentu. Misalnya aplikasi untuk mengenali wajah.

Gambar 2 Machine Learning

Machine Learning sudah sangat berkembang sehingga dapat di terapkan di bidang manapun. Misalnya , di bidang kesehatan machine learning dapat di gunakan untuk memprediksi penyakit[7], di bidang keamanan dapat di gunakan sebagai cctv pintar yang dapat mendeteksi maupun menghitung objek tertentu[8], di bidang otomotif machine learning di gunakan sebagai sistem autonomus.Autonomus adalah sebuah teknologi yang membuat mobil dapat berkendara sendiri layaknya di kendalikan oleh pengemudi tanpa harus ada pengemudi[9].

2.2 Transfer Learning

Transfer learning adalah suatu metode yang yang menggunakan model yang sudah ada sebagai dasar untuk membuat model baru untuk menyelesaikan masalah yang lain dengan cara menjadi starting point, memodifikasi, dan mengupdate parameter yang di perlukan sehingga terbentuk sebuah model baru[2]

2.3 Tensor Flow

Tensorflow adalah sebuah framework komputasional yang di buat oleh google yang berguna untuk meng-implementasikan dan membuat model machine learning atau deep learning[10]

2.4 Dataset

Dataset adalah kumpulan data / sampel yang di berguna untuk membuat model / untuk di train[2]. Dataset berfungsi sebagai data yang berguna sebagai bahan untuk mesin untuk belajar. Semakin banyak, akurat, dan bervariasi data, 9 maka semakin bagus pula output yang di hasilkan. Dalam mengumpulkan dataset harus di lakukan secara lengkap dan menyeluruh. Misalnya, untuk data set wajah maka harus memiliki sampel gambar wajah di berbagai kondisi maupun ekpresi wajah[3]. Dalam dataset di kenal istilah training set dan testing set. Training set adalah sekumpulan data yang di gunakan sebagai data / bahan latihan oleh mesin, sedangkan testing set adalah data – data yang di gunakan sebagai evaluasi dari model. Data testing set tentunya harus berbeda dengan data training set hal ini bertujuan untuk membuat sistem dapat mengenali suatu objek yang bahkan belum pernah di inputkan sebelumnya.

2.6 Supervised Learning

Dataset adalah kumpulan data / sampel yang di berguna untuk membuat model / untuk di train[2]. Dataset berfungsi sebagai data yang berguna sebagai bahan untuk mesin untuk belajar. Semakin banyak, akurat, dan bervariasi data, 9 maka semakin bagus pula output yang di hasilkan. Dalam mengumpulkan dataset harus di lakukan secara lengkap dan menyeluruh. Misalnya, untuk data set wajah maka harus memiliki sampel gambar wajah di berbagai kondisi maupun ekpresi wajah[3]. Dalam dataset di kenal istilah training set dan testing set. Training set adalah sekumpulan data yang di gunakan sebagai data / bahan latihan oleh mesin, sedangkan testing set adalah data – data yang di gunakan sebagai evaluasi dari model. Data testing set tentunya harus berbeda dengan data training set hal ini bertujuan untuk membuat sistem dapat mengenali suatu objek yang bahkan belum pernah di inputkan sebelumnya.

2.7 Single Shot Detector (SSD)

Single shot detector adalah sebuah metode untuk mengenali atau mendeteksi objek pada suatu gambar dengan mengunakan single deep neural network[11].SSD bekerja dengan cara merangkum perhitungan dalam satu jaringan sehingga membuat SSD relatih sederhana dan mudah di integrasikan ke dalam sistem serta memiliki akurasi yang baik berbanding rasio yang di butuhkan komputasi

2.8 Akurasi

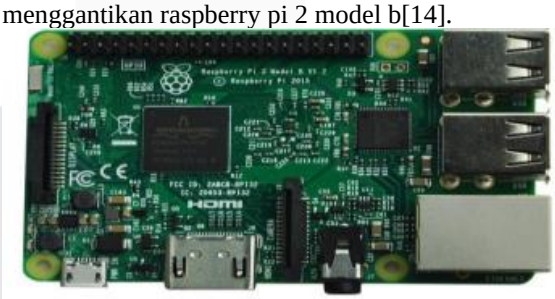
Akurasi dalam machine learning adalah suatu proses yang menghitung seberapa akurat sistem dalam mengukur suatu objek dari banyaknya data. Pada dasarnya akurasi di dapatkan dari nilai true positive (TP) di bagi dengan nilai true positive(TP) , true negative (TN) dan false positive (FP)[13].Untuk menghitung akurasi menggunakan rumus sebagai berikut.

2.9 Telegram Api

Aplikasi telegram menyediakan sebuah API yang berguna untuk membuat bot yang berguna untuk melakukan sesuatu fungsi yang sudah di atur sebelumnya, API ini di namakan dengan Telegram Bot API. Dalam hal ini bot hanya dapat mengirimkan ke pengguna telegram saja sehingga untuk menggunakan bot telegram user harus memiliki aplikasi telegram terlebih dahulu [12].

2.10 Raspberry Pi 3 Model B

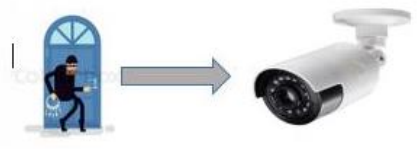
Raspberry pi (juga di kenal raspi) adalah sebuah SBC (Single Board Computer) dengan seukuran kartu kredit yang di lengkapi dengan wireless LAN and Bluetooth connectivity. Pada februari tahun 2016 raspberry pi 3 model b di ciptakan yang menggantikan raspberry pi 2 model b[14].



Gambar 5 Raspberry Pi Model B

3. Analisis dan Perancangan

3.1 Gambaran Sistem Saat Ini



Gambar 7 Gambaran Sistem Saat Ini

Gambar 3.1 Gambaran Sistem Saat Ini

Pada Gambar 7 di atas adalah gambaran sistem keamanan rumah pada umumnya saat ini. Pada umumnya cctv hanya merekam gambar dan tidak mengirimkan notifikasi kepada pemilik, oleh karena itu pemilik tidak mengetahui apabila ada orang yang sedang berada di depan pintu rumah sehingga pemilik rumah tidak dapat melakukan tindakan pencegahan apapun seperti menelpon tetangga. Pemilik rumah hanya dapat melihat tayangan ulang yang sudah di rekam sebelumnya.

3.2 Analisis Kebutuhan Sistem

Analisa kebutuhan fungsional dapat dilihat pada tabel berikut ini.

3.2.1 Fungsional

Tabel 5 Kebutuhan fungsionalitas

No	Kebutuhan Fungsionalitas
1	Mendeteksi objek berupa wajah
2	Dapat membedakan wajah di kenali atau tidak
3	Dapat mengirimkan Notifikasi melalui telegram
4	Menampilkan nama objek dan akurasi pada layar monitor

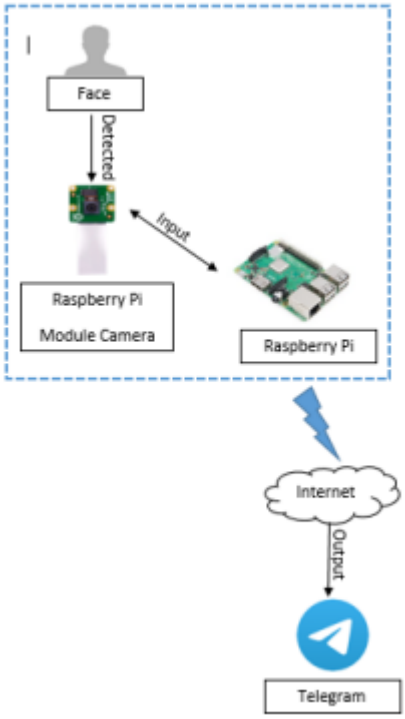
.

3.2.4 Non Fungsional

Tabel 6 Kebutuhan non fungsionalitas

No	Kebutuhan Non fungsionalitas
1	Sebuah *raspberry pi module camera*
2	Sebuah raspberry pi untuk menjalankan program
3	Sebuah monitor dan kabel HDMI
4	Bahasa pemograman python
5	Software notepad++ sebagai text editor
6	Sampel wajah sebagai dataset
7	Sebuah *handphone* untuk menerima notifikasi
8	Koneksi internet

3.3 Perancangan Sistem



Gambar 3.2 Gambaran Sistem Usulan

Penjelasan singkat Gambar 8 diatas:

1. Adaptor di gunakan untuk memberi daya kepada raspberry pi

2. Raspberry pi module camera di gunakan untuk mendeteksi wajah yang kemudian mengirimkan input berupa citra wajah.

3. Raspberry pi di gunakan untuk memproses data dari kamera kemudian mengirimkan notifikasi ke user.

4. *Telegram* di gunakan untuk mengirimkan sekaligus menerima notifikasi.

Gambar 9 Blok diagram pembuatan model

Gambar di atas adalah blok diagram pembuatan model yang terdiri dari 3 bagian,

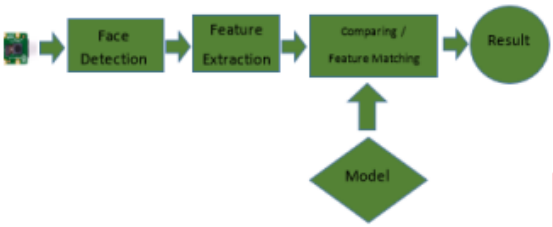
yaitu sebagai berikut :

1. Dataset berisi sekumpulan data yang berguna sebagai bahan latih mesin. Dalam dataset di kenal istilah training set dan testing set. Training set adalah sekumpulan data yang di gunakan sebagai data / bahan latihan oleh mesin, sedangkan testing set adalah data – data yang di gunakan sebagai evaluasi dari model

2. Learning algorithm adalah proses belajar mesin berdasarkan data yang di berikan. Pada pembuatan

proyek akhir ini proses learning termasuk ke dalam kategori supervised learning.

3. Output yang di keluarkan adalah model yang sudah terlatih. Setelah mesin mengalami proses belajar maka mesin akan di ujicoba menggunakan testing set. Model adalah hasil dari fase latih/ training dari data tertentu yang berisi algoritma yang siap untuk memprediksi data baru

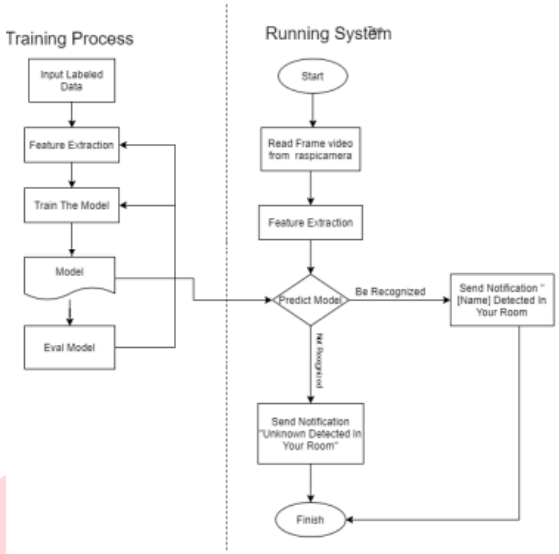


Gambar 10 Blok diagram sistem

Berdasarkan blok diagram di atas, proses kerja pada perancangan sistem ini di mulai oleh kamera. Kamera akan merekam dan mendeteksi wajah, apabila objek berupa wajah terdeteksi selanjutnya di lakukan proses feature extraction. Feature extraction adalah proses untuk mengambil nilai nilai penting yang merepresentasikan citra wajah tersebut. Selanjutnya sistem akan melakukan proses klasifikasi dengan cara melakukan template matching dengan model yang sudah di latih terlebih dahulu. Klasifikasi menggunakan single shot detector (SSD) yang menggunakan algoritma single deep neural network. Kemudian setelah di ketahui wajah di kenali atau tidak sistem akan meneruskan ke pemilik rumah via aplikasi telegram..

3.2 Cara Kerja

Sistem ini sangat di pengaruhi oleh kelengkapan dataset berupa sampel wajah dari berbagai sudut kamera, ekpresi maupun kondisi cahaya. Semakin lengkap dataset yang di miliki maka persentase keberhasilan membedakan objek akan menjadi semakin besar. Pada sistem ini kamera akan menangkap semua objek yang kemudian akan di kirimkan ke pc untuk diproses apakah objek di kenali atau tidak. Ketika objek berupa wajah terdeteksi melalui kamera, maka raspberry pi secara automatis melakukan pencocokan dengan database sistem, jika cocok maka akan mengirimkan notifikasi ke pemilik berupa pesan melalui aplikasi telegram. Jika objek di kenali sebagai anak maka akan mengirimkan notifikasi seperti berikut “Afrizal detected in your room”. Jika objek yang terdeteksi tidak dikenali maka akan mengirimkan notifikasi “Uknown detected in your room”. Adapun gambaran flowchart dari sistem yang akan diajukan adalah sebagai berikut:



Gambar 11 Flowchart Sistem Saat Ini

Penjelasan singkat Gambar 11 Gambar Blok Diagram Sistem diatas:

Pada saat training model :

1. User akan memberikan inputan berupa data yang sudah berlabel kepada

sistem. Data ini di kenal dengan istilah training set. Label di sini adalah “Afrizal”,”ari”,”tidak di kenali”

2. Selanjutnya sistem akan melakukan proses feature extraction di mana mengambil nilai - nilai penting dari data tersebut yang merepresentasikan data tersebut.

3. Selanjutnya sistem akan melakukan proses training/learning menggunakan algoritma tertentu dan akan menghasilkan model.20

4. Selanjutnya model akan di evaluasi menggunakan data yang belum pernah di lihat sebelumnya. Data ini di kenal dengan istilah testing set, lalu akan mengirimkan nilai hasil evaluasi ke dalam proses training sehingga akan

memperbarui informasi model.

Pada saat sistem berjalan :

1. Start untuk memulai sistem.

2. Selanjutnya sistem akan membaca frame demi frame yang di inputkan oleh raspicamera.Selanjutnya sistem akan mendeteksi wajah dan melakukan proses feature extraction.

3. Kemudian akan di lakukan proses template matching dengan model yang sudah di buat sebelumnya apakah wajah di kenali atau tidak

kemudian sistem akan memberikan notifikasi ke smartphone**.**

3.3 Spesifikasi Sistem

Tabel 7 Spesifikasi Hardware

No	Hardware	Fungsi
1	Raspberry pi module camera	Berfungsi untuk menangkap objek berupa wajah secara real time.
2	Raspberry pi 3 model B	Berfungsi untuk memproses data masukkan yang di terima dari kamera.
3	Monitor	Berfungsi sebagai user interface dengan sistem
4	Power Suply	Berfungsi sebagai catu daya untuk mengalirkan daya ke raspberry pi

Tabel 8 Spesifikasi Sofware

No	Software	Fungsi
1	Python	Berfungsi sebagai bahasa pemrograman untuk membuat sistem dan bot telegram.
2	Notepad++	Berfungsi sebagai text editor
3	Tensorflow	Berfungsi untuk membuat model untuk *machine learning* dan media untuk *training* dari *dataset*.
4	Open – cv	Berfungsi untuk mengaktifkan kamera
5	labelImg	Berfungsi untuk memberikan label pada sampel gambar dengan cara memberikan anotasi pada gambar dalam bentuk file XML
6	Telegram	Berfungsi sebagai client untuk menerima notifikasi

1. Studi literatur

Tahap ini di lakukan guna mencari tahu sistem apa yang di gunakan dan

cara kerja sistem.

2. Instalasi sistem operasi pada raspberry pi

Tahap ini di lakukan untuk menginstall sistem operasi pada raspberry pi. Sistem operasi yang di gunakan adalah raspbian yang di install menggunakan noobs.

3. Instalasi environment

Pada tahap ini di lakukan instalasi environment seperti menginstall beberapa library yang di gunakan, seperti: opencv-python, tensorflow,numpy, dan lain sebagainya.

4. Memberi label pada gambar

Langkah ini di lakukan untuk mengenalkan mesin citra wajah sehingga ketika masuk ke proses training algoritma dapat menentukan ini wajah siapa.

5. Konfigurasi training

Setelah semua gambar sudah di label maka selanjutnya data di training agar sistem dapat membedakan wajah. Proses training memerlukan waktu yang lama agar dapat mengeluarkan hasil yang optimal.

6. Pengujian

Ini adalah langkah terakhir di mana akan di lakukan pengujian dengan cara mendeteksi wajah secara real-time dan akan di lihat apakah sistem dapat membedakan wajah atau tidak.

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

4.2 Langkah Pengerjaan

Dalam membangun prototipe ini langkah – langkah yang di lakukan adalah sebagai berikut:

1. Studi Literatur

Tahap ini di lakukan guna mencari tahu sistem apa yang di gunakan dan bagaimana cara kerja sistem. Seperti pre trained model yang di gunakan, library – library python yang di gunakan dan perangkat apa saja yang di gunakan.

2. Pengumpulan sampel wajah

Untuk mengumpulkan sampel wajah saya merekam video selama 10-20 detik, kemudian dari video tersebut di ubah frame demi frame menjadi gambar. Semakin banyak data dan variasi dataset yang di gunakan maka akan semakin baik pula hasil yang di dapatkan. Pada pembuatan model ini dataset yang di ambil berupa video dengan 3 cara, yaitu: di rekam dari dekat, di rekam dari jauh, di rekam dari atas. Kemudian video tersebut di ubah frame demi frame menjadi gambar menggunakan script berikut:

```
import cv2
vidcap = cv2.VideoCapture('video3.mp4')
success,image = vidcap.read()
count = 240
while success:
  cv2.imwrite("frame%d.jpg" % count, image)
  success,image = vidcap.read()
  print('Read a new frame: ', success)
  count += 1
```

Dari script di atas frame demi frame video akan di ubah menjadi gambar. Untuk detail jumlah gambar yang di dapatkan dari video dapat di lihat pada tabel dibawah ini Jumlah gambar yang di gunakan dalam proses pengerjaan proyek akhir ini adalah 1206 dengan detail seperti di tujukan dari tabel 9 di atas. Gambar tersebut kemudian di bagi menjadi 2 data, yaitu: 1) data train yang berisi 80% dari keseluruhan gambar dan 2) data test yang berisi 20% gambar.

3. Membuat anotasi dari file gambar

Untuk membuat anotasi file saya menggunakan aplikasi labelimg yang saya dapatkan dari github. Anotasi file gambar berguna untuk memetakan gambar. Dalam pembuatan model ini yang di petakan adalah wajah, tahapan ini di lakukan pada semua gambar, hal ini di karenakan pada setiap gambar lokasi wajah dapat berbeda - beda. File anotasi tersebut berformat XML

Tabel 9 Detail jumlah gambar dataset

No	Nama Objek	Jumlah Gambar	Kondisi
1	Afrizal	283	Dekat dengan kamera
2	Afrizal	337	Jauh dari kamera
3	Afrizal	186	Dari atas
4	Ari	200	Dekat dengan kamera
5	Ari	200	Jauh dari kamera

Jumlah gambar yang di gunakan dalam proses pengerjaan proyek akhir ini

adalah 1206 dengan detail seperti di tujukan dari tabel 9 di atas. Gambar

tersebut kemudian di bagi menjadi 2 data, yaitu: 1) data train yang berisi

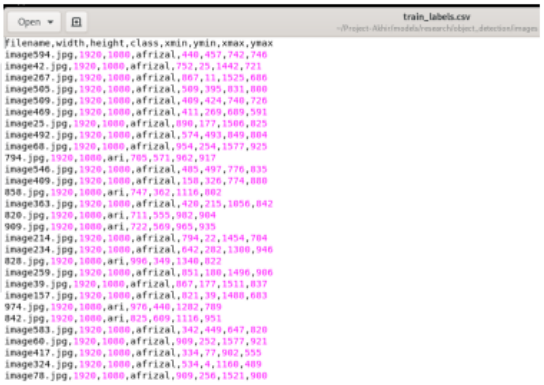
80% dari keseluruhan gambar dan 2) data test yang berisi 20% gambar.

3. Membuat anotasi dari file gambar

Untuk membuat anotasi file saya menggunakan aplikasi labelimg yang saya dapatkan dari github. Anotasi file gambar berguna untuk memetakan gambar. Dalam pembuatan model ini yang di petakan adalah wajah, tahapan ini di lakukan pada semua gambar, hal ini di karenakan pada setiap gambar lokasi wajah dapat berbeda - beda. File anotasi tersebut berformat XML

```
list_xml = []
for xml_file in glob.glob(path + '/*.xml'):
    tree = ET.parse(list_xml)
    root = tree.getroot()
    for member in root.findall('object'):
        value = (root.find('filename').text,
                 int(root.find('size')[0].text),
                 int(root.find('size')[1].text),
                 member[0].text,
                 int(member[4][0].text),
                 int(member[4][1].text),
                 int(member[4][2].text),
                 int(member[4][3].text)
                 )
        xml_list.append(value)
colom_baru = ['filename', 'width', 'height', 'class', 'xmin', 'ymin', 'xmax', 'ymax']
xml_df = pd.DataFrame(xml_list, columns=colom_baru)
return xml_df
```

Script di atas berguna untuk menggabungkan semua file anotasi yang berformat XML menjadi 1 file. di mana nilai yang di ambil adalah nama file, width, height, class, xmin, ymin, xmax, dan ymax.



Gambar 13 Hasil gabungan file XML

Gambar di atas adalah contoh isi dari file .csv. File ini berisikan semua informasi anotasi dari seluruh gambar yang di jadikan dataset.

4. Membuat label

Untuk membuat komputer dapat membedakan antara wajah si A dengan wajah lainnya pada perlu di lakukan

proses pelabelan. Masing masing wajah diberikan dengan id yang berbeda

Gambar 14 contoh label

Pada gambar di atas adalah contoh file untuk inisiasi id dari masing - masing wajah. Dan nantinya file tersebut akan terhubung ke file konfigurasi tensorflow. 5. Proses training Untuk menjalankan proses training, tensorflow sudah memberikan sebuah file bernama train.py yang berguna untuk menjalankan training agar dapat menghasilkan model dengan cara transfer learning dari 1 model ke sebuah model baru. Pada proyek akhir ini menggunakan basis model 28 ssdlite_mobilenet_v2_coco, model ini di gunakan karena resource pc yang di butuhkan cenderung kecil walaupun memakan waktu yang lama pada saat training.

Gambar 15 Nilai loss classification

Gambar di atas adalah hasil nilai loss classification pada saat training data. loss classification adalah nilai loss model untuk mengklasikasi objek Semakin rendah nilai loss yang di dapatkan maka model akan menjadi semakin bagus. loss function adalah fungsi untuk menghitung seberapa akurat model ketika proses training di lakukan tujuannya adalah meminimalisir nilai loss, akan tetapi hal ini bisa mengakibatkan overfitting pada model. overfitting adalah kondisi di mana model menjadi sangat kaku yang hanya berfokus pada data yang sudah ada sehingga tidak bisa melakukan prediksi dengan tepat dengan data baru yang serupa .

Gambar 16 Nilai Loss localization

Nilai loss localization adalah nilai loss yang melambangkan kinerja sistem dalam menentukan lokasi dari suatu objek. 29 6. Export hasil training menjadi model 7. Pembuatan source code untuk deteksi wajah Pada tahap ini di buat source code untuk mendeteksi wajah. Jika wajah terdeteksi maka akan masuk ke proses selanjutnya yaitu klasifikasi wajah apakah di kenali atau tidak. Source code dapat di lihat di bawah ini. 8. Pengujian Pengujian ini di lakukan untuk melihat kinerja sistem apakah sudah berjalan dengan baik atau belum. Ujicoba di lakukan dengan inputan beberapa gambar dan juga secara realtime. 9. Pengujian mengirimkan notifikasi Untuk mengirimkan notifikasi melalui telegram di gunakan library python bernama telepot, bot akan mengirimkan pesan ke id yang sudah di definisakan sebelumnya oleh user. Setiap user memiliki id yang berbeda – beda oleh karena itu jika ingin mengirimkan ke berbagai user maka id user harus di definisikan terlebih dahulu. Source code dapat di lihat di bawah ini

4.2.1 Pengujian

Nilai loss localization adalah nilai loss yang melambangkan kinerja sistem dalam menentukan lokasi dari suatu objek. 29 6. Export hasil training menjadi model 7. Pembuatan source code untuk deteksi wajah Pada tahap ini di buat source code untuk mendeteksi wajah. Jika wajah terdeteksi maka akan masuk ke proses selanjutnya yaitu klasifikasi wajah apakah di kenali atau tidak. Source code dapat di lihat di bawah ini. 8. Pengujian Pengujian ini di lakukan untuk melihat kinerja sistem apakah sudah berjalan dengan baik atau belum. Ujicoba di lakukan dengan inputan beberapa gambar dan juga secara realtime. 9.

Pengujian mengirimkan notifikasi Untuk mengirimkan notifikasi melalui telegram di gunakan library python bernama telepot, bot akan mengirimkan pesan ke id yang sudah di definisakan sebelumnya oleh user. Setiap user memiliki id yang berbeda – beda oleh karena itu jika ingin mengirimkan ke berbagai user maka id user harus di definisikan terlebih dahulu. Source code dapat di lihat di bawah ini

4.2.1.1 Tujuan Pengujian

Tujuan pengujian ini adalah untuk mengetahui kinerja sistem yang sudah di bangun apakah sudah berjalan dengan baik atau belum. Fungsionalitas sistem yang di bangun adalah dapat membedakan wajah yang sudah terdata dan yang belum terdata lalu mengirimkan notifikasi melalui bot telegram. Pengujian di lakukan dengan berbagai cara mulai dari merubah ekpresi wajah, memakai aksesoris seperti kacamata atau topi, dan mecoba berbagai jarak, mulai dari 1 meter hingga 2 meter.

4.2.1.2 Skenario Pengujian

1. Pengujian Terhadap Gambar

   Tujuan pengujian ini adalah untuk mengetahui kinerja sistem yang sudah di bangun apakah sudah berjalan dengan baik atau belum. Fungsionalitas sistem yang di bangun adalah dapat membedakan wajah yang sudah terdata dan yang belum terdata lalu mengirimkan notifikasi melalui bot telegram. Pengujian di lakukan dengan berbagai cara mulai dari merubah ekpresi wajah, memakai aksesoris seperti kacamata atau topi, dan mecoba berbagai jarak, mulai dari 1 meter hingga 2 meter.

2. Skenario Pengujian

   Skenario ini di lakukan dengan cara menggunakan kamera sebagai inputan, di sini kamera akan aktif dan sistem akan memproses data masukan dari kamera tersebut lalu akan mengirimkan notikasi melalui aplikasi telegram. Pada pengujian ini objek tidak memakai aksesoris apapun

4.2.2 Pengujian Menggunakan Aksesoris Secara Realtime

Skenario ini di lakukan dengan cara menggunakan kamera sebagai input, di sini kamera akan aktif dan sistem akan memproses data masukan dari kamera lalu akan mengirimkan notifikasi melalui aplikasi telegram. Pada pengujian ini objek memakai aksesoris baik itu kacamata atau topi. Sistem di katakan bekerja dengan baik apabila berhasil membedakan objek

4.2.3.1 Hasil Pengujian

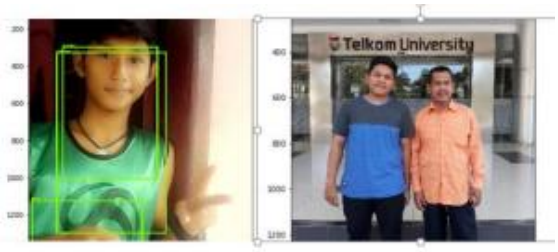
Berikut adalah hasil pengujian dari berbagai cara seperti yang sudah di jelaskan pada skenario di atas

Gambar 17 Sampel gambar yang di ujicoba

Gambar di atas menunjukkan beberapa sampel gambar yang akan di ujicoba secara bersamaan. Terdapat 11 gambar dengan 7 gambar dengan wajah yang di kenali dan 4 gambar dengan wajah yang tidak di kenali

Gambar 18 Hasil pengujian terhadap gambar berhasil

Pada gambar di atas sistem dapat membedakan wajah dengan baik pada kondisi pencahayaan tertentu dan juga dapat menampilkan persentasi keakuratan dalam melakukan prediksi wajah. Dari 11 gambar yang di ujicoba sistem berhasil melakukan prediksi 9 gambar dan gagal 2 gambar.



Gambar 19 Hasil pengujian terhadap gambar gagal

Gambar di atas adalah gambar yang gagal di prediksi oleh sistem. Pada gambar pertama sistem mengenali wajah tersebut dengan nama afrizal padahal wajah tersebut tidak di kenali. Dan pada gambar kedua sistem tidak dapat mengenali wajah padahal di gambar tersebut terdapat wajah yang di kenali yaitu afrizal

Dari pengujian ini di dapatkan hasil sebagai berikut :

Tabel 10 Hasil pengujian terhadap gambar 1 objek

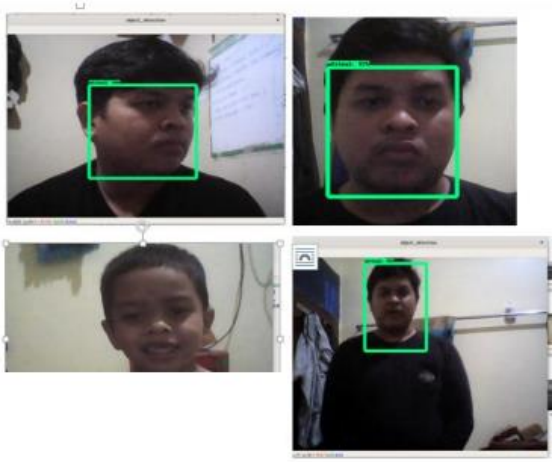
No	Nama file	Nama objek	Status Pemrosesan	Akurasi
1	Image1.jpg	Ari	Berhasil	77%
2	Image4.jpg	Afrizal	Berhasil	99%
3	Image5.jpg	Afrizal	Berhasil	95%
4	Image9.jpg	Afrizal	Berhasil	79%
5	Image11.jpg	Tidak di kenali	Tidak berhasil	x

Tabel di atas adalah rincian dari hasil ujicoba menggunakan gambar. Jika gambar yang di ujicoba mirip seperti gambar yang ada di dataset maka akurasi yang di peroleh akan sangat tinggi seperti di tunjukkan oleh nomor 2 dan nomor 3. Jika yang di ujicoba tidak pernah di masukkan ke dalam dataset maka hasil prediksi akan menurun seperti di tunjukkan oleh nomor 4.

No	Nama File	Nama Objek	Status Pemrosesan	Akurasi
1	Image2.jpg	Tidak di kenali	Berhasil	x
2	Image3.jpg	Tidak di kenali	Berhasil	x
3	Image6.jpg	Ari dan Afrizal	Berhasil	Ari 98%, Afrizal 94%
4	Image7.jpg	Ari dan Afrizal	berhasil	Ari 96%, Afrizal 90%
5	Image8.jpg	Tidak di kenali	berhasil	x
6	Image10.jpg	Afrizal dan tidak di kenali	Tidak berhasil	x

Pada tabel di atas adalah hasil ujicoba menggunakan gambar yang terdapat 2 objek atau lebih di dalamnya. Dari tabel di atas sistem gagal mengenali gambar nomor 6 di mana dalam gambar tersebut terdapat objek bernama Afrizal.

4.2.3 Pengujian Tanpa Aksesoris Secara Real Time

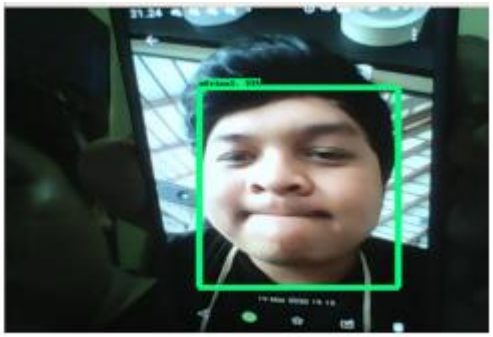


Pada gambar di atas adalah hasil pengujian tanpa menggunakan aksesoris. Dari hasil pengujian di dapatkan hasil bahwasanya sistem dapat mendeteksi objek walaupun objek tersebut tidak melihat kamera. Sistem dapat mengirimkan notifikasi ke user melalui aplikasi telegram apabila objek di kenali atau tidak. Notifikasi di kirimkan secara terus menerus hingga objek tidak terlihat lagi di kamera.

Gambar 21 Contoh notifikasi

Gambar di atas adalah contoh notifikasi yang di terima oleh user. Pengiriman notifikasi terdapat delay 3 detik setelah objek terdeteksi oleh kamera. Secara real time sistem ini memiliki kelemahan

yaitu hanya dapat memprediksi wajah baik itu gambar secara 3d maupun 2d sehingga sistem memiliki celah keamanan yaitu dapat di manipulasi dengan cara menampilkan wajah yang di kenali melalui handphone ke kamera



Gambar 22 Kelemahan sistem

Sistem ini memiliki kelemahan yaitu tidak dapat membedakan gambar 3d atau 2d sehingga sistem hanya akan mengenali apakah wajah tersebut di kenali atau tidak seperti yang di tunjukkan oleh gambar 16 di atas. Sistem akan langsung mengenali objek gambar tersebut. Untuk detail hasil ujicoba dapat di lihat pada tabel 12 di bawah ini.

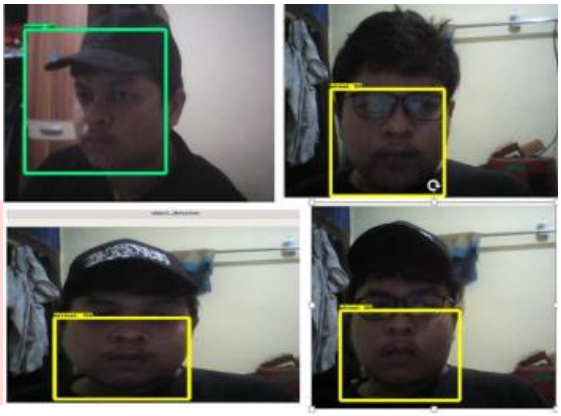
Tabel 12 Hasil Pengujian tanpa aksesoris secara real time

No	Nama Objek	Jarak	Rentang Akurasi	Delay Notifikasi	Status Pemrosesan
1	afrizal	30 cm	76% – 95%	3 detik	Berhasil
2	Afrizal	60 cm	70% – 95%	3 detik	Behasil
3	Afrizal	1 m	57% – 93%	3 detik	Berhasil
4	Afrizal	2 m	x	3 detik	Tidak Berhasil
5	Tidak di kenal	30 cm	80% – 96%	3 detik	berhasil
6	Tidak di kenal	60 cm	73% – 92%	3 detik	Berhasil
7	Tidak di kenal	1 m	70% – 89%	3 detik	Berhasil
8	Tidak di kenal	2 m	70% – 83%	3 detik	Berhasil

Berdasarkan tabel di atas menunjukkan bahwasanya akurasi prediksi sangat di pengaruhi oleh jarak objek ke kamera. Jika objek berada dekat dengan kamera maka akan meningkatkan hasil prediksi, sedangkan jika objek berada jauh dengan kamera akan menurunkan hasil prediksi. Dalam pengiriman notifikasi terdapat delay selama 3 detik sejak objek terlihat di kamera. Tanda x pada tabel berate akurasi tidak muncul pada saat sistem di jalankan.

4.2.2.1 Pengujian Menggunakan Aksesoris Real Time

Berikut adalah hasil pengujian secara real time dengan menggunakan aksesoris berupa topi dan kacamata



ıbar 23 Hasil pengujian menggunakan aksesoris secara real time

Pada gambar di atas adalah hasil pengujian dengan menggunakan aksesoris berupa kacamata dan topi. Dari hasil pengujian di dapatkan hasil bahwasanya sistem dapat mendeteksi objek walaupun objek tersebut memakai topi namun tingkat keberhasilan menurun dari sebelumnya. Pada saat objek memakai kacamata sistem mendapatkan akurasi sebesar 78%, menggunakan topi menurun menjadi 75%, menggunakan kacamata dan topi menurun menjadi 65%. Hasil detail dari pengujian ini dapat di lihat dari tabel 13 di bawah ini.

Tabel 13 Hasil Pengujian menggunakan aksesoris secara real time

No	Nama Objek	Aksesoris	Rentang Akurasi	Status Pemrosesan
1	Afrizal	Kacamata	77% – 85%	Berhasil
2	Afrizal	Topi	75% – 80%	Berhasil
3	Afrizal	Topi dan kacamata	60% – 73%	Berhasil
4	Tidak di kenali	Kacamata	83% – 90%	Berhasil

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dalam proses pengerjaan dan ujicoba proyek akhir ini di dapatkan kesimpulan sebagai berikut:

1. Kondisi cahaya dan jarak sangat mempengaruhi sistem dalam

memprediksi wajah di kenali atau tidak.

2. Aksesoris kepala mempengaruhi terhadap akurasi sistem dalam

memprediksi.

3. Terdapat delay selama 3 detik untuk mengirimkan notifikasi ke

telegram dari objek tertangkap kamera.

5.2 Saran

Untuk pengembangan sistem lebih lanjut pada penelitian ini, disarankan untuk.

1. Menggunakan raspberry pi versi lebih tinggi agar sistem dapat bekerja secara optimal dan lebih baik.
2. Menggunakan accelerator pada raspberry pi untuk mempercepat kinerja raspberry pi
3. Dapat mendeteksi 2 objek sekaligus secara real time dan dapat mengirimkan notifikasi ke user

REFERENSI

Y. Baştanlar and M. Ozuysal, Introduction to Machine Learning Second
Edition, vol. 1107. 2014.

[2] P. Jan and W. Gotama, “Pengenalan Pembelajaran Mesin dan Deep
Learning,” no. July, pp. 1–199, 2018.

[3] D. A. WIRAWAN, “PERANCANGAN DAN IMPLEMENTASI PROTOTIPE SISTEM
KEAMANAN RUMAH BERBASIS PENGENALAN WAJAH MENGGUNAKAN
METODE FISHERFACE DENGAN PUSAT KENDALI TELEGRAM PADA
RASPBERRY PI,” Karya Ilm. - Skripsi - Ref., p. 100, 2019.

[4] P. PURWANTO, “IMPLEMENTASI FACE IDENTIFICATION DAN FACE
RECOGNITION PADA KAMERA PENGAWAS SEBAGAI PENDETEKSI BAHAYA,”
p. 10, 2015.

[5] M. D. Cahyo, S. Sumaryo, R. Y. Nur, F. T. Elektro, and U. Telkom, “Kamera
Pengawas Berbasis Deteksi Pola Wajah Menggunakan Metode Principal
Component Analysis (Pca) Dan Artificial Neural Network (Ann) Camera
Surveillance Based on Face Pattern Recognition Using Principal Component
Analysis (Pca) and Artificial Neura,” vol. 5, no. 3, pp. 3927–3933, 2018.

[6] M. R. A. PANGESTU, “REAL TIME CCTV DETEKSI WAJAH DENGAN HAAR
CASCADE CLASSIFIERS,” pp. 1–8.

[7] B. Rifai, “Algoritma Neural Network Untuk Prediksi,” Techno Nusa Mandiri,
vol. IX, no. 1, pp. 1–9, 2013.

[8] V. Lempitsky and A. Zisserman, “Learning to count objects in images,” Adv.
Neural Inf. Process. Syst. 23 24th Annu. Conf. Neural Inf. Process. Syst. 2010,
NIPS 2010, pp. 1–9, 2010.

[9] “AUTONOMOUS CAR.” [Online]. Available:
https://socs.binus.ac.id/2018/12/06/autonomous-car/. [Accessed: 29-May2020].
40

[10] “TensorFlow Core.” [Online]. Available:
https://www.tensorflow.org/overview/. [Accessed: 28-Oct-2019].

[11] W. Liu et al., “SSD: Single shot multibox detector,” Lect. Notes Comput. Sci.
(including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol.
9905 LNCS, pp. 21–37, 2016.

[12] A. Sukusvieri, “Implementasi Metode Single Shot Detector (Ssd) Untuk
Pengenalan Wajah,” 2020.

[13] “Accuracy (error rate) Definition | DeepAI.” [Online]. Available:
https://deepai.org/machine-learning-glossary-and-terms/accuracy-errorrate. [Accessed: 01-Jul-2020].

[14] “Buy a Raspberry Pi 3 Model B – Raspberry Pi.” [Online]. Available:
https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. [Accessed:
28-May-2020].

[15] “Buy a Camera Module V2 – Raspberry Pi.” [Online]. Available:
https://www.raspberrypi.org/products/camera-module-v2/. [Accessed: 28-
May-2020]