

TESIS

**ANALISIS KOMPARASI CONVOLUTIONAL NEURAL NETWORK
(CNN) DENGAN SUPPORT VECTOR MACHINE (SVM) UNTUK
KLASIFIKASI CITRA PENGGUNA MASKER**



Disusun oleh:

Nama : Arham Rahim
NIM : 19.51.1175
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2021

TESIS

**ANALISIS KOMPARASI CONVOLUTIONAL NEURAL NETWORK
(CNN) DENGAN SUPPORT VECTOR MACHINE (SVM) UNTUK
KLASIFIKASI CITRA PENGGUNA MASKER**

**CONVOLUTIONAL NEURAL NETWORK (CNN) COMPARISON
ANALYSIS WITH SUPPORT VECTOR MACHINE (SVM) FOR MASK
USER IMAGE CLASSIFICATION**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Arham Rahim
NIM : 19.51.1175
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA**

2021

HALAMAN PENGESAHAN

**ANALISIS KOMPARASI CONVOLUTIONAL NEURAL NETWORK (CNN)
DENGAN SUPPORT VECTOR MACHINE (SVM) UNTUK KLASIFIKASI CITRA
PENGGUNA MASKER**

**CONVOLUTIONAL NEURAL NETWORK (CNN) COMPARISON ANALYSIS
WITH SUPPORT VECTOR MACHINE (SVM) FOR MASK USER IMAGE
CLASSIFICATION**

Dipersiapkan dan Disusun oleh

Arham Rahim

19.51.1175

Telah Ditujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Kamis, 6 Mei 2021

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 6 Mei 2021

Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

ANALISIS KOMPARASI CONVOLUTIONAL NEURAL NETWORK (CNN) DENGAN SUPPORT VECTOR MACHINE (SVM) UNTUK KLASIFIKASI CITRA PENGGUNA MASKER

CONVOLUTIONAL NEURAL NETWORK (CNN) COMPARISON ANALYSIS WITH SUPPORT VECTOR MACHINE (SVM) FOR MASK USER IMAGE CLASSIFICATION

Dipersiapkan dan Disusun oleh

Arham Rahim

19.51.1175

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Kamis, 6 Mei 2021

Pembimbing Utama

Dr. Kusriani, M.Kom
NIK. 190302106

Pembimbing Pendamping

Dr. Arief Setyanto, S.Si., M.T
NIK. 190302036

Anggota Tim Penguji

Emha Taufik Luthfi, S.T., M.Kom.
NIK. 190302125

Dr. Andi Sunyoto, M.Kom
NIK. 190302052

Dr. Kusriani, M.Kom
NIK. 190302106

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 6 Mei 2021
Direktur Program Pascasarjana

Dr. Kusriani, M.Kom.
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Arham Rahim
NIM : 19.51.1175
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:

Analisis Komparasi Convolutional Neural Network (CNN) dengan Support Vector Machine (SVM) untuk Klasifikasi Citra Pengguna Masker

Dosen Pembimbing Utama : Dr. Kusriani, M.Kom
Dosen Pembimbing Pendamping : Emha Lutfi Taufik S.Kom.M.T

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 06 Mei 2021
Yang Menyatakan,



Arham Rahim

HALAMAN PERSEMBAHAN

Pertama dan paling utama, saya ucapkan puji syukur kepada Tuhan Yesus Kristus yang telah memberikan kemudahan dan kelancaran dalam proses pembuatan tesis ini. Tesis ini saya persembahkan untuk :

- 1 Kedua orang tua (Bapak Usman M. Kasim dan Ibu Rahmatia) yang senantiasa memberikan semangat dan doa, semoga selalu dalam lindungan Allah Subhanahu Wa Ta'ala.
- 2 Ibu Dr. Kusriani M.Kom dan Bapak Emha Taufik Luthfi, S.T., M.Kom. yang telah memberikan bimbingan aktif selama pelaksanaan penelitian, semoga mendapatkan banyak keberkahan dan dilancarkan segala urusannya.
- 3 Keluarga besar yang selalu mendukung dan memberikan semangat tanpa henti serta juga yang selalu ada dalam keadaan apapun. Serta semua pihak yang tidak bisa penulis sebutkan satu persatu.
- 4 Keluarga besar kelas MT122-A yang selalu membantu dan memberi saran demi kelancaran pengerjaan tesis, semoga selalu semangat dan sukses.

HALAMAN MOTTO

“Baik jadi orang penting, tapi lebih penting jadi orang baik”



KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadirat Allah SWT, karena atas berkat dan anugrah-Nya sehingga penulis dapat menyelesaikan Tesis ini yang berjudul "**Anallsis Komparasi Convolutional Neural Network (CNN) dengan Support Vector Machine (SVM) untuk Klasifikasi Citra Pengguna Masker**". Tesis ini disusun sebagai salah satu syarat kelulusan dalam jenjang perkuliahan Strata 2 Universitas Amikom Yogyakarta.

Dalam menyusun tesis ini mulai dari persiapan hingga tahap penyelesaian senantiasa disertai dengan berbagai hambatan, berkat kesabaran dan keimanan yang keras dari penulis serta bantuan dari berbagai pihak baik secara langsung maupun tidak langsung, baik moral maupun material akhirnya tesis ini dapat selesai.

Oleh karena itu sepantasnya jika penulis dengan kerendahan hati menghantarkan banyak terimakasih dan penghargaan sebesar-besarnya kepada :

1. Bapak Prof. Dr. M. Suyanto, MM selaku rektor Universitas Amikom Yogyakarta.
2. Ibu Dr. Kusriani M.Kom dan Bapak Emha Taufik Luthfi, S.T., M.Kom selaku dosen pembimbing yang telah dengan sabar dalam memberikan masukan, saran, bantuan, dan bimbingan dalam menyelesaikan naskah tesis ini
3. Ibu Dr. Kusriani, M.Kom selaku Direktur Program Pascasarjana Universitas Amikom Yogyakarta.

4. Kedua Orang tua dan saudara yang tak pernah lelah dalam memberikan support dan doanya.
5. Dosen Amikom Yogyakarta yang telah memberikan banyak ilmu dan pengalaman.

Semua pihak yang telah membantu penyelesaian tesis ini yang tentunya sangat berharga dan tidak bisa disebutkan satu persatu. Penulis juga meminta maaf apabila dalam penyusunan tesis ini masih banyak kekurangan dan masih jauh untuk memberikan kata sempurna. Penulis juga dengan senang hati menerima kritik dan saran.

Akhirnya, penulis berharap tesis ini dapat memberikan manfaat bagi mahasiswa. semoga Allah SWT, senantiasa melimpahkan berkat-Nya kepada kita semua sehingga tesis ini dapat memberikan manfaat kepada kita. Amin

Yogyakarta, 6 Mei 2021

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS	v
HALAMAN PERSEMBAHAN	vi
HALAMAN MOTTO	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xiv
INTISARI.....	xvi
<i>ABSTRACT</i>	xvii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah.....	5
1.3. Batasan Masalah	6
1.4. Tujuan Penelitian	6
1.5. Manfaat Penelitian	7
BAB II TINJAUAN PUSTAKA.....	8
2.1. Tinjauan Pustaka	8
2.2. Keaslian Penelitian.....	11

2.3. Landasan Teori.....	17
BAB III METODE PENELITIAN	27
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	27
3.2. Metode Pengumpulan Data.....	27
3.3. Metode Analisis Data.....	28
3.4. Alur Penelitian	28
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	32
4.1. Membangun <i>Dataset</i>	32
4.1.1. Pengumpulan Data	32
4.1.2. <i>Preprocessing</i> Data.....	34
4.1.3. Pengelompokan Data.....	36
4.2. Analisis Data.....	37
4.2.1. <i>Convolutional Neural Network</i> (CNN).....	39
4.2.2. <i>Support Vector Machine</i> (SVM).....	51
4.2.3. Skenario Percobaan.....	53
4.2.4. Hasil <i>Training</i> dan <i>Testing Data</i>	53
4.3. Analisis Hasil Penelitian.....	66
4.3.1. Perbandingan Hasil <i>Training</i> dan <i>Validation</i>	67
4.3.2. Perbandingan Hasil <i>Testing</i>	68
4.3.3. Perbandingan Waktu Proses.....	69
4.3.4. Kelemahan Sistem.....	70

BAB V PENUTUP.....	73
5.1. Kesimpulan	73
5.2. Saran	74
DAFTAR PUSTAKA	76



DAFTAR TABEL

Tabel 2.1. Matriks literatur review dan posisi penelitian	11
Tabel 2.2. <i>Confusion matrix</i>	25
Tabel 4.1. Skenario percobaan	53
Tabel 4.2. Hasil Percobaan.....	66
Tabel 4.3 Waktu komputasi data uji	69



DAFTAR GAMBAR

Gambar 2.1. Sistem kordinat yang mewakili citra	17
Gambar 2.2. Operasi konvolusi.....	20
Gambar 2.3. Operasi <i>max pooling</i>	21
Gambar 4.1. <i>Cropping</i> gambar	33
Gambar 4.2. Proses pelabelan dataset	33
Gambar 4.3. Sampel dataset <i>mask</i> dan <i>nonmask</i>	34
Gambar 4.4. Hasil <i>Preprocessing</i>	36
Gambar 4.5. Proses <i>convolution</i>	41
Gambar 4.6. Pergerakan proses <i>convolution</i>	42
Gambar 4.7. Proses <i>Pooling</i>	44
Gambar 4.8. Proses <i>Flatten</i>	45
Gambar 4.9. Arsitektur VGG-16.....	46
Gambar 4.10. Arsitektur ResNet50.....	47
Gambar 4.11. Arsitektur MobileNet	48
Gambar 4.12. <i>Layer</i> VGG16 Sebelum <i>Fine Tuning</i>	49
Gambar 4.13. <i>Layer</i> VGG16 Setelah <i>Fine Tuning</i>	51
Gambar 4.14. Grafik <i>traing and validation accuracy</i> skenario 1	54
Gambar 4.15. Garafik <i>traing and validation loss</i> skenario 1	54
Gambar 4.16. Garafik <i>confusion matrix</i> pengujian skenario 1	55
Gambar 4.17. Grafik <i>traing and validation accuracy</i> skenario 2	56
Gambar 4.18. Grafik <i>traing and validation loss</i> skenario 2.....	57

Gambar 4.19. Garafik <i>confusion matrix</i> pengujian skenario 2	57
Gambar 4.20. Grafik <i>traing and validation accuracy</i> skenario 3	59
Gambar 4.21. Grafik <i>traing and validation loss</i> skenario 3.....	59
Gambar 4.22. Garafik <i>confusion matrix</i> pengujian skenario 2	60
Gambar 4.23. Garafik <i>confusion matrix</i> pengujian skenario 4	62
Gambar 4.24. Garafik <i>confusion matrix</i> pengujian skenario 5	63
Gambar 4.25. Garafik <i>confusion matrix</i> pengujian skenario 6	65
Gambar 4.26. Hasil <i>training and validation</i>	68
Gambar 4.27. Hasil <i>testing</i>	68
Gambar 4.28. Waktu Proses.....	70



INTISARI

Salah satu alternatif pencegahan penularan COVID-19, pada 6 April 2020 *World Health Organization* (WHO) mengeluarkan anjuran penggunaan masker di tengah masyarakat tetapi ketidak patuhan sebagian masyarakat terhadap protokol kesehatan sebagai upaya pencegahan penularan COVID-19, menjadi keresahan bagi masyarakat secara umum. Sehingga memerlukan inovasi berupa sistem yang dapat melakukan pengawasan. *machine learning* dapat digunakan sebagai sistem pengawasan dengan membangun sebuah model yang dapat melakukan klasifikasi citra wajah pengguna masker dan tidak menggunakan masker. Dalam penelitian ini, *Convolutional Neural Network* (CNN) akan dibandingkan kinerja dan waktu komputasi per gambar dengan kombinasi *Convolutional Neural Network* (CNN) dan *Support Vector Machine* (SVM) dengan menggunakan 1700 gambar yang terbagi menjadi 850 gambar wajah yang menggunakan masker dan 850 gambar wajah yang tidak menggunakan masker. Untuk mendapatkan model terbaik, penelitian ini menggunakan tiga arsitektur yang telah teruji yaitu VGG-16, ResNet50 dan MobileNet. Penelitian ini menyimpulkan CNN dengan menggunakan ResNet50 mendapatkan nilai kinerja tertinggi dengan *accuracy* 99,41% dengan waktu komputasi dibawah 0,03 detik per gambar sehingga baik digunakan untuk kamera pemantau dengan *frame rate* 30 fps.

Kata kunci: Mask, Klasifikasi, *Convolutional Neural Network*, *Support Vector Machine*

ABSTRACT

One alternative to preventing the transmission of COVID-19, on April 6, 2020 the World Health Organization (WHO) issued a recommendation to use masks in the community but disobedience by some people to health protocols as an effort to prevent the transmission of COVID-19 has become a concern for the general public so that innovation in the form of a system that can carry out supervision. machine learning can be used as a monitoring system by building a model that can classify the facial image of mask users and not use masks. In this case, the Convolutional Neural Network (CNN) will compare performance and computation time per image with a combination of Convolutional Neural Network (CNN) and Support Vector Machine (SVM) using 1700 images divided into 850 face images using masks and 850 face images that use masks. don't use a mask. To get the best model, this study uses three proven architectures, namely VGG-16, ResNet50 and MobileNet. The research concludes that CNN using ResNet50 gets the highest performance value with an accuracy of 99.41% with a computation time below 0.03 seconds per image so it is good for monitoring cameras with a frame rate of 30 fps. Abstract merupakan intisari dalam Bahasa Inggris, ditulis dengan huruf miring.

Keyword: Mask, Classification, Convolutional Neural Network, Support Vector Machine

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Coronavirus Disease-2019 (COVID-19) adalah koronavirus jenis baru yang ditemukan pada Desember 2019 di Wuhan Cina. COVID-19 menjadi pandemi di berbagai negara seperti Amerika Serikat (US), Inggris, Italia, Thailand, Singapore, Malaysia dan berbagai negara lain termasuk Indonesia. Menurut Badan Nasional Penanggulangan Bencana (BNPB), perkembangan COVID-19 di Indonesia sejak Maret sampai 14 Mei 2020 tercatat 16.006 kasus positif dan 1.043 dinyatakan meninggal dunia. Menurut *World Health Organization* (WHO) jumlah kasus positif COVID-19, Indonesia masuk dalam peringkat 35 dari 215 negara dan wilayah/teritorial.

Cepatnya penyebaran virus hingga banyaknya korban jiwa, memberikan dampak negatif bagi kehidupan bermasyarakat menjadi masalah yang sangat serius bagi setiap negara sehingga membutuhkan penanganan yang serius pula untuk mencegah semakin meluasnya penularan COVID-19. Salah satu alternatif pencegahan penularan COVID-19, pada 6 April 2020 *World Health Organization* (WHO) mengeluarkan anjuran penggunaan masker di tengah masyarakat, selama perawatan di rumah dan ditempat pelayanan kesehatan yang diperuntukkan bagi anggota masyarakat, tenaga profesional kesehatan masyarakat dan pencegahan dan pengendalian infeksi (PPI). Berdasarkan rekomendasi anjuran tersebut, pemerintah

Republik Indonesia mewajibkan warga untuk menggunakan masker saat beraktivitas diluar rumah.

Ketidak patuhan sebagai masyarakat terhadap protokol kesehatan sebagai upaya pencegahan penularan COVID-19, menjadi keresahan bagi masyarakat secara umum. Seperti penelitian yang dilakukan oleh (Sari and Atiqoh 2020) untuk mengetahui hubungan antara pengetahuan masyarakat dengan kepatuhan penggunaan masker sebagai upaya pencegahan COVID-19 memaparkan 25,81% dari 62 responden tidak patuh menggunakan masker. Penggunaan masker sebagai alat perlindungan diri (APD) terhadap penularan COVID-19 jika disamakan dengan penggunaan APD pada industri untuk mengurangi resiko kerja, ketidak patuhan pekerja juga menjadi masalah yang serius. Menurut (Astiningsih, Kurniawan, and Suroto 2018) Kurangnya inspeksi dan pengawasan yang mencapai 72% mengakibatkan 54% pekerja tidak menggunakan APD sehingga penelitian tersebut menunjukkan adanya hubungan antara penerapan program inspeksi dan pengawasan dengan kepatuhan penggunaan APD ($p=0,024$). Kurangnya kesadaran untuk menggunakan APD, mengharuskan pengawasan yang lebih ketat bagi para pekerja. Akan tetapi menurut (Shrestha et al. 2015) metode pengawasan tradisional terkadang gagal karena tidak bisa mengawasi para pekerja setiap jam dan setiap hari. Sehingga memerlukan inovasi berupa sistem pengawasan yang *real time* dan secara otomatis dapat mendeteksi para pekerja yang tidak menggunakan APD.

Untuk membangun sebuah sistem pengawasan diperlukan metode/algorithm *machine learning* atau *deep learning* yang dapat mengklasifikasi

citra dengan kondisi yang berbeda-beda seperti penelitian yang dilakukan oleh (Naufal 2021) yang menganalisis perbandingan dari kinerja algoritma *Support Vector Machine* (SVM), *K-Nearest Neighbors* (KNN) dan *Convolutional Neural Network* (CNN) untuk melakukan klasifikasi citra dengan kelas *cloudy*, *rain*, *shine* dan *sunrise*. Penelitian ini menyimpulkan algoritma CNN lebih unggul dengan nilai akurasi yang mencapai 0,942 dibandingkan SVM dengan nilai akurasi 0,860 dan KNN dengan nilai akurasi 0,766. Kinerja *Convolutional Neural Network* (CNN) juga telah diuji oleh (Faisal, Butarbutar, and Sirait 2019) dengan membandingkan algoritma tersebut dengan algoritma *Support Vector Machine* (SVM) untuk mengidentifikasi penyakit pada tomat berdasarkan citra daun. Penelitian tersebut menyimpulkan bahwa CNN adalah algoritma *classifier* yang lebih unggul dengan nilai akurasi 97,5% dibandingkan dengan SVM dengan nilai akurasi 95%.

Berbeda dengan penelitian yang dilakukan oleh (Faisal, Butarbutar, and Sirait 2019) yang membandingkan CNN dan SVM, penelitian yang dilakukan oleh (Chen et al. 2018) justru mengkombinasikan algoritma CNN dengan SVM untuk selanjutnya dibandingkan dengan algoritma CNN sendiri untuk melakukan klasifikasi gestur tangan dari sinyal *Surface Electromyography* (sEMG). Dari hasil penelitian ini, penggabungan CNN-SVM memperoleh akurasi lebih tinggi yaitu 68,2% dari CNN dengan akurasi 65,4%.

Penelitian yang dilakukan oleh (Chaiyasarn et al. 2018) juga menggunakan algoritma CNN untuk mengekstraksi fitur dari gambar RGB dan menggunakan algoritma SVM untuk proses klasifikasi yang bertujuan untuk mendeteksi retakan pada struktur batu di situs bersejarah. Penelitian ini juga menggunakan dua

skenario pengujian dengan membandingkan akurasi dari proses yang hanya menggunakan algoritma CNN dengan CNN-SVM. Penelitian ini menyimpulkan bahwa gabungan CNN-SVM juga lebih baik dengan akurasi dari validasi dataset 85,94% dan testing dataset 74,9% daripada CNN dengan akurasi validasi dataset 82,94% dan testing dataset 67,5%. Selanjutnya penelitian yang dilakukan oleh (Wang et al. 2018) yang bertujuan untuk mengklasifikasi gambar mikroskop kanker payudara menggunakan algoritma CNN dan SVM. Dengan menggunakan skenario yang sama, penelitian ini justru memperoleh hasil yang berbeda dengan penelitian (Chen et al. 2018) dan (Chafiyasari et al. 2018). Penelitian ini menyimpulkan hasil klasifikasi gambar menggunakan algoritma CNN lebih tinggi dengan nilai akurasi 92,5% dari hasil validasi dataset dan data testing 91,7% daripada CNN+SVM dengan nilai akurasi 82,25% dari hasil validasi dataset dan 80,6% untuk data testing.

Selain mempertimbangkan kinerja model berdasarkan akurasi, waktu proses juga perlu dipertimbangkan karena menurut (Thirumala et al. 2019) pemantauan secara otomatis hanya dapat dilakukan jika sistem dapat bekerja dengan cepat. Menurut (Lotfi, Ahmad, and Saeed 2016) *frame rate* yang umum digunakan pada kamera pemantau adalah 30 *frame per secon*. Untuk mengurangi resiko kehilangan informasi pada setiap *frame*, perlu mempertimbangkan waktu komputasi dari model klasifikasi yang kurang dari atau sama dengan 0,03 detik untuk satu citra.

Berdasarkan uraian di atas, dimana algoritma CNN (*Convolutional Neural Network*) memiliki akurasi yang lebih tinggi dibanding beberapa algoritma klasifikasi lain serta terdapat perbedaan kinerja dari perbandingan algoritma CNN

(*Convolutional Neural Network*) dengan kombinasi *Convolutional Neural Network* (CNN) dan *Support Vector Machine* (SVM) maka penelitian ini akan membandingkan kembali algoritma CNN (*Convolutional Neural Network*) dengan kombinasi *Convolutional Neural Network* (CNN) dan *Support Vector Machine* (SVM) untuk melakukan klasifikasi citra wajah yang menggunakan masker dan tidak menggunakan masker dengan mengukur kinerja berdasarkan *accuracy*, *precision*, *recall* dan waktu proses.

1.2. Rumusan Masalah

Berdasarkan pemaparan latar belakang di atas, dapat diperoleh rumusan masalah sebagai berikut :

- a. Berapa nilai *accuracy*, *precision* dan *recall* dari algoritma *Convolutional Neural Network* (CNN) untuk mengklasifikasi citra wajah yang menggunakan masker dan tidak menggunakan masker ?
- b. Berapa nilai *accuracy*, *precision* dan *recall* dari algoritma *Convolutional Neural Network* (CNN) yang dikombinasikan dengan *Support Vector Machine* (SVM) untuk mengklasifikasi citra wajah yang menggunakan masker dan tidak menggunakan masker ?
- c. Apakah waktu komputasi setiap model kurang dari atau sama dengan 0,03 detik dalam melakukan klasifikasi ?
- d. Jika berbeda, Mengapa terdapat perbedaan kinerja dari algoritma *Convolutional Neural Network* (CNN) dengan kombinasi algoritma *Convolutional Neural Network* (CNN) dan *Support Vector Machine* (SVM) ?

1.3. Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

- a. Algoritma yang digunakan untuk klasifikasi citra adalah *Convolutional Neural Network (CNN)* dan *Support Vector Machine (SVM)*.
- b. Penelitian ini akan mengklasifikasikan citra wajah yang menggunakan masker dan tidak menggunakan masker.
- c. Jenis masker yang digunakan untuk data penelitian adalah masker medis atau masker kain yang menutupi mulut dan hidung.
- d. Citra untuk proses *training* dan *testing* diperoleh dari *kaggle* dan *google image*.
- e. Penelitian ini akan melakukan klasifikasi citra menggunakan algoritma *Convolutional Neural Network (CNN)* dan mengkombinasikan algoritma *Convolutional Neural Network (CNN)* dan *Support Vector Machine (SVM)*.
- f. Proses evaluasi model klasifikasi menggunakan metode *confusion matrix* dan pengukuran kecepatan klasifikasi citra menggunakan *time execution process* pada python.

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah tujuan dari penelitian ini adalah:

- a. Mengetahui nilai *accuracy*, *precision* dan *recall* dari algoritma *Convolutional Neural Network (CNN)* untuk mengklasifikasi citra wajah yang menggunakan masker dan tidak menggunakan masker.

- b. Mengetahui nilai *accuracy*, *precision* dan *recall* dari algoritma *Convolutional Neural Network* (CNN) yang dikombinasikan dengan *Support Vector Machine* (SVM) untuk mengklasifikasi citra wajah yang menggunakan masker dan tidak menggunakan masker.
- c. Mengetahui waktu komputasi dari algoritma *Convolutional Neural Network* (CNN) dan kombinasi algoritma *Convolutional Neural Network* (CNN) dengan *Support Vector Machine* (SVM) dalam melakukan klasifikasi per gambar
- d. Mengetahui penyebab adanya perbedaan kinerja dari algoritma *Convolutional Neural Network* (CNN) dan kombinasi algoritma *Convolutional Neural Network* (CNN) dengan *Support Vector Machine* (SVM)

1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah :

- a. Bagi peneliti, mengetahui tingkat akurasi tertinggi antara model klasifikasi *Convolutional Neural Network* (CNN) dan kombinasi *Convolutional Neural Network* dengan *Support Vector Machine* (SVM).
- b. Memberikan pengetahuan baru mengenai model yang dapat melakukan klasifikasi citra wajah yang menggunakan masker dan tidak menggunakan masker
- c. Rancangan model yang dibangun dapat diterapkan pada sistem pemantau penggunaan masker sehingga memberikan kontribusi terhadap pencegahan penyebaran COVID-19.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian yang dilakukan oleh Weili Fang, Lieyun Ding, Botao Zhong, Peter E.D. Love dan Hanbin Luo (2018) dengan judul *Automated detection of workers and heavy equipment on construction sites: A convolutional neural network approach*. Penelitian ini bertujuan untuk melakukan pengawasan dengan mendeteksi secara otomatis pekerja dan alat berat di lokasi konstruksi menggunakan algoritma *Convolutional Neural Network* (CNN). Penelitian ini mampu mendeteksi keberadaan pekerja dan *excavator* dengan akurasi mencapai 91% dan 95%.

Penelitian yang dilakukan oleh Weili Fang, Botao Zhong, Neng Zhao, Peter E.D. Love, Hanbin Luo, Jiayue Xue dan Shuangjie Xu (2019) dengan judul *A deep learning-based approach for mitigating falls from height with computer vision: Convolutional neural network*. Penelitian ini juga melakukan pengawasan dengan mendeteksi pekerja yang melakukan perilaku tidak aman seperti melintasi *structural supports* (beton atau baja) pada konstruksi bangunan menggunakan algoritma *Convolutional Neural Network* (CNN). Penelitian tersebut menyimpulkan tingkat penarikan dan presisi untuk deteksi yang tumpang tindih mencapai 75% dan 90%.

Penelitian yang dilakukan oleh Hongfeng Chen, Runze Tong, Minjie Chen, Yinfeng Fang dan Honghai Liu (2018) dengan judul *A Hybrid CNN-SVM Classifier*

for Hand Gesture Recognition With Surface EMG Signals. Penelitian ini melakukan dua skenario pengujian dimana skenario pertama hanya menggunakan algoritma CNN dan skenario kedua menggunakan algoritma CNN untuk ekstraksi fitur dan SVM untuk melakukan klasifikasi gestur tangan dari sinyal Surface Electromyography (sEMG). Dari hasil penelitian ini, penggabungan CNN-SVM memperoleh akurasi lebih tinggi yaitu 68,2% dari skenario pertama dengan akurasi 65,4%.

Penelitian yang dilakukan oleh Krisada Chaiyasarna, Wasif Khanb, Luqman Alib, Mayank Sharmab, Daniel Brackenburyc dan Matthew DeJong (2018) dengan judul *Crack Detection in Masonry Structures using Convolutional Neural Networks and Support Vector Machines*. Penelitian ini bertujuan untuk mendeteksi retakan pada struktur batu di situs bersejarah menggunakan algoritma CNN untuk mengekstraksi fitur dari gambar RGB dan menggunakan algoritma SVM untuk proses klasifikasi. Penelitian ini juga menggunakan dua skenario pengujian dengan membandingkan akurasi dari proses yang hanya menggunakan algoritma CNN dengan CNN-SVM. Penelitian ini menyimpulkan gabungan CNN-SVM lebih baik dengan akurasi dari validasi dataset 85,94% dan testing dataset 74,9% daripada CNN dengan akurasi validasi dataset 82,94% dan testing dataset 67,5%.

Penelitian yang dilakukan oleh Yaqi Wang, Lingling Sun, Kaiqiang Ma dan Jiannan Fang (2018) dengan judul *Breast Cancer Microscope Image Classification Based on CNN with Image Deformation*. Penelitian ini bertujuan untuk mengklasifikasi gambar mikroskop kanker payudara menggunakan algoritma CNN dan SVM. Dengan menggunakan skenario yang sama, penelitian ini

memperoleh hasil yang berbeda dengan dua peneliti sebelumnya (Chen et al. 2018) dan (Chaiyasarn et al. 2018) penelitian ini menyimpulkan hasil klasifikasi gambar menggunakan algoritma CNN lebih tinggi dengan nilai akurasi 92,5% dari hasil validasi dataset dan data testing 91,7% daripada CNN+SVM dengan nilai akurasi 82,25% dari hasil validasi dataset dan 80,6% untuk data testing.

Penelitian yang dilakukan oleh Bhima Caraka, Bakhtiar Aldino Ardi Sumbodo dan Ika Candradewi (2017) dengan judul Klasifikasi Sel Darah Putih Menggunakan Metode *Support Vector Machine (SVM)* Berbasis Pengolahan Citra Digital. Penelitian ini bertujuan untuk mengklasifikasikan jenis dari sel darah putih dari gambar *mikroskop* menggunakan algoritma SVM dan menggunakan metode *Histogram Oriented Gradient* untuk ekstraksi fitur. Penelitian ini menyimpulkan bahwa klasifikasi jenis sel darah putih menggunakan algoritma SVM dapat mencapai akurasi dengan nilai rata-rata 72,26%.

2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian Analisis Komparasi *Convolutional Neural Network (CNN)* Dengan *Support Vector Machine (SVM)* Untuk Klasifikasi Citra Pengguna Masker

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	<i>Automated detection of workers and heavy equipment on construction sites: A convolutional neural network approach</i>	Weili Fang, Lieyun Ding, Botao Zhong, Peter E.D. Love, Hanbin Luo, Elsevier, 2018.	Penelitian ini bertujuan untuk melakukan pengawasan dengan mendeteksi secara otomatis pekerja dan alat berat di lokasi konstruksi menggunakan algoritma Convolutional Neural Network (CNN)	Penelitian ini mampu mendeteksi keberadaan pekerja dan excavator dengan akurasi mencapai 91% dan 95%.	Penelitian ini tidak menjelaskan <i>pre-processing</i> dataset sebelum dilakukan proses <i>training</i> atau <i>testing</i>	Perbandingan penelitian ini dengan penelitian yang akan dilakukan diantaranya, penelitian ini mendeteksi objek dan mengklasifikasikan jenis objek yang dideteksi (manusia dan alat berat) sedangkan penelitian yang akan dilakukan adalah hanya mendeteksi manusia dan mengklasifikasikan objek apakah menggunakan masker atau tidak

Tabel 2.1. Matriks literatur review dan posisi penelitian Analisis Komparasi *Convolutional Neural Network (CNN)* Dengan *Support Vector Machine (SVM)* Untuk Klasifikasi Citra Pengguna Masker (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
2	<i>A deep learning-based approach for mitigating falls from height with computer vision: Convolutional neural network</i>	Weili Fang, Botao Zhong, Neng Zhao, Peter E.D. Love, Hanbin Luo, Jiayue Xue, Shuangjie Xu, Elsevier, 2018	Penelitian ini bertujuan untuk melakukan pengawasan dengan mendeteksi pekerja yang melakukan perilaku tidak aman seperti melintasi <i>structural supports</i> (beton atau baja) pada konstruksi bangunan menggunakan algoritma Convolutional Neural Network (CNN)	Penelitian tersebut menyimpulkan tingkat penarikan dan presisi untuk deteksi yang tumpang tindih mencapai 75% dan 90%.	<ul style="list-style-type: none"> E. Salah satu kelemahan dari proses deteksi pekerja yang diakui oleh penulis adalah tidak dapat mendeteksi pekerja yang terhalang oleh objek lain (tidak menutupi seluruh badan) disekitar area kerja F. Penelitian ini juga tidak menjelaskan Preprocessing dataset 	Salah satu perbedaan penelitian ini dengan penelitian yang akan dilakukan diantaranya, penelitian ini mendeteksi manusia dan perilakunya sedangkan penelitian yang akan dilakukan adalah mendeteksi manusia dan atribut yang dikenakannya

Tabel 2.1. Matriks literatur review dan posisi penelitian Analisis Komparasi *Convolutional Neural Network (CNN)* Dengan *Support Vector Machine (SVM)* Untuk Klasifikasi Citra Pengguna Masker (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
3	<i>A Hybrid CNN-SVM Classifier for Hand Gesture Recognition With Surface EMG Signals</i>	Hongfeng Chen, Runze Tong, Minjie Chen, Yinfeng Fang, Honghai Liu, IEEE, 2018	Penelitian bertujuan untuk melakukan klasifikasi gestur tangan dari sinyal Surface Electromyography (sEMG) dengan menggunakan dua skenario pengujian dimana skenario pertama hanya menggunakan algoritma CNN dan skenario kedua menggunakan algoritma CNN untuk ekstraksi fitur dan SVM untuk klasifikasi	Dari hasil penelitian ini, penggabungan CNN-SVM memperoleh akurasi lebih tinggi yaitu 68,2% dibandingkan dengan hanya menggunakan CNN yaitu 65,4%.	<ul style="list-style-type: none"> E. Penelitian ini tidak menampilkan hasil dari proses pengambilan sinyal <i>Surface Electromyography (sEMG)</i> F. Penelitian ini menggunakan dataset yang sangat sedikit. Dimana dataset diperoleh dari 8 orang dan terdiri dari 10 gestur tangan G. Penelitian ini juga tidak menjelaskan Preprocessing dataset 	Salah satu perbedaan penelitian ini dengan penelitian yang akan dilakukan diantaranya, penelitian ini melakukan proses klasifikasi berdasarkan citra dari sinyal <i>Surface Electromyography (sEMG)</i> sedangkan penelitian yang akan dilakukan adalah mendeteksi manusia dan melakukan klasifikasi berdasarkan atribut yang dikenakannya

Tabel 2.1. Matriks literatur review dan posisi penelitian Analisis Komparasi *Convolutional Neural Network (CNN)* Dengan *Support Vector Machine (SVM)* Untuk Klasifikasi Citra Pengguna Masker (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
4	<i>Crack Detection in Masonry Structures using Convolutional Neural Networks and Support Vector Machines</i>	Krisada Chaiyasarna, Wasif Khanb, Luqman Alib, Mayank Sharmab, Daniel Brackenburyc, Matthew DeJong, ISARC Proceedings, 2018	Penelitian ini bertujuan untuk mendeteksi retakan pada struktur batu di situs bersejarah menggunakan algoritma CNN untuk mengekstraksi fitur dari gambar RGB dan menggunakan algoritma SVM untuk proses klasifikasi. Penelitian ini juga menggunakan dua skenario pengujian dengan membandingkan akurasi dari proses yang hanya menggunakan algoritma CNN dengan CNN-SVM	Penelitian ini menyimpulkan gabungan CNN-SVM lebih baik dengan akurasi dari validasi dataset 85,94% dan testing dataset 74,9% daripada CNN dengan akurasi validasi dataset 82,94% dan testing dataset 67,5%.	Kelemahan yang diakui penulis adalah kurangnya dataset mengakibatkan sulitnya sistem mengidentifikasi retakan pada susunan batu bata dibandingkan retakan pada permukaan beton.	Walau metode dan skenario pengujian pada penelitian ini dan penelitaian yang akan dilakukan sama, tetapi objek penelitaian serta proses pengumpulan data yang akan dilakukan pada penelitian selanjutya berbeda

Tabel 2.1. Matriks literatur review dan posisi penelitian Analisis Komparasi *Convolutional Neural Network (CNN)* Dengan *Support Vector Machine (SVM)* Untuk Klasifikasi Citra Pengguna Masker (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
5	<i>Breast Cancer Microscope Image Classification Based on CNN with Image Deformation</i>	Yaqi Wang, Lingling Sun, Kaiqiang Ma, Jiannan Fang, Springer International Publishing, 2018	Penelitian ini bertujuan untuk mengklasifikasi gambar mikroskop kanker payudara menggunakan algoritma CNN dan SVM. Penelitian ini juga menggunakan dua skenario pengujian dengan membandingkan akurasi dari proses yang hanya menggunakan algoritma CNN dengan CNN-SVM	Penelitian ini menyimpulkan hasil klasifikasi gambar menggunakan algoritma CNN lebih tinggi dengan nilai akurasi 92,5% dari hasil validasi dataset dan data testing 91,7% daripada CNN+SVM dengan nilai akurasi 82,25% dari hasil validasi dataset dan 80,6% untuk data testing		Penelitian ini juga menggunakan metode dan skenario pengujian yang sama dengan penelitian yang akan dilakukan, tetapi objek penelitian serta proses pengumpulan data yang akan dilakukan pada penelitian selanjutnya berbeda

Tabel 2.1. Matriks literatur review dan posisi penelitian Analisis Komparasi *Convolutional Neural Network (CNN)* Dengan *Support Vector Machine (SVM)* Untuk Klasifikasi Citra Pengguna Masker (Lanjutan)

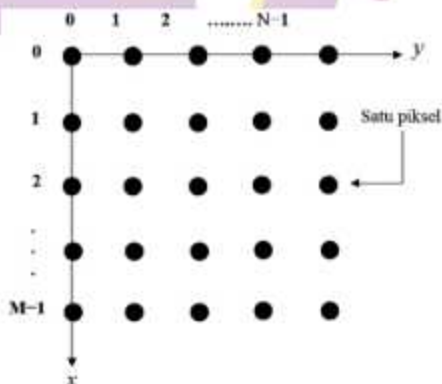
No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
6	Klasifikasi Sel Darah Putih Menggunakan Metode Support Vector Machine (SVM) Berbasis Pengolahan Citra Digital	Bhima Caraka, Bakhtiar Alldino Ardi Sumbodo, Ika Candradewi, IJEIS, 2017	Penelitian ini bertujuan untuk mengklasifikasikan jenis dari sel darah putih dari gambar <i>mikroskop</i> menggunakan algoritma SVM dan menggunakan metode <i>Histogram Oriented Gradient</i> untuk ekstraksi fitur	Penelitian ini menyimpulkan bahwa klasifikasi jenis sel darah putih menggunakan algoritma SVM dapat mencapai akurasi dengan nilai rata-rata 72,26%	Kelemahan yang diakui penulis adalah perlunya penambah variasi data <i>training</i> yang lebih banyak untuk sel – sel darah putih sehingga proses deteksi bisa lebih akurat	Perbandingan penelitian ini dengan penelitian yang akan dilakukan diantaranya, penelitian ini menggunakan algoritma SVM untuk melakukan klasifikasi dan menggunakan metode <i>Histogram Oriented Gradient</i> untuk ekstraksi fitur sedangkan penelitian yang akan dilakukan menggunakan dua skenario pengujian dimana skenario pertama hanya menggunakan algoritma CNN dan skenario kedua menggunakan algoritma CNN untuk ekstraksi fitur dan SVM untuk klasifikasi

2.3. Landasan Teori

2.3.1. Pengolahan Citra Digital

Pengolahan citra digital adalah ilmu yang mempelajari tentang algoritma transformasi citra (Asmara 2018). Citra adalah kumpulan piksel dalam ruang warna yang berbeda yang menjadi representasi visual dari objek kehidupan nyata dalam bentuk dua dimensi (Singh 2019) dan menurut (Kusumanto and Tompunu 2011) citra digital secara matematis merupakan fungsi kontinu dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer, citra digital harus melalui proses digitalisasi sehingga dapat dipresentasikan secara numerik dengan nilai-nilai diskrit.

Menurut (Andono, Sutojo, and Muljono 2017) citra digital merupakan matriks M baris dan N kolom yang terbentuk dari bilangan real. Perpotongan antara baris dan kolom dari matriks tersebut, disebut piksel atau elemen terkecil dari sebuah citra seperti pada gambar 1.



Gambar 2.1. Sistem kordinat yang mewakili citra

Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x,y) adalah $f(x,y)$ sehingga sebuah citra digital dapat ditulis dalam bentuk matriks berikut:

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (1)$$

Citra $f(x,y)$ dalam fungsi matematis dapat ditulis $(0 \leq x \leq M-1)$, $(0 \leq y \leq N-1)$ dan $(0 \leq f(x,y) \leq G-1)$

Dimana :
 M = Jumlah piksel baris pada array citra
 N = Jumlah piksel kolom pada array citra
 G = Nilai skala keabuan (*graylevel*)

Besarnya nilai M , N dan G pada fungsi di atas umumnya merupakan perangkatan dari dua sehingga dapat ditulis sebagai berikut.

$$M = 2^m ; N = 2^n ; G = 2^k \quad (2)$$

Nilai m , n dan k adalah bilangan bulat positif. Interval $(0,G)$ disebut skala keabuan (*grayscale*). Besar G tergantung pada proses digitalisasinya. Biasanya keabuan 0 (nol) menyatakan intensitas hitam dan 1 (satu) menyatakan intensitas putih. Untuk citra 8 bit, nilai G sama dengan $2^8 = 256$ warna (derajat keabuan).

2.3.2. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah salah satu algoritma *deep learning* yang digunakan untuk kasus-kasus penggunaan *computer vision* seperti mengklasifikasikan gambar atau video dan mendeteksi objek di dalam gambar atau

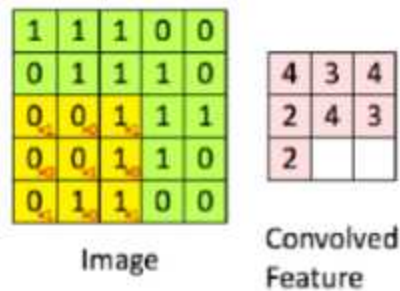
bahkan wilayah dalam gambar (Moolayil 2019). Dan menurut (Zufar and Setiyono 2016) CNN adalah variasi dari *Multilayer Perceptron* (MLP) yang terinspirasi dari jaringan syaraf pada manusia. CNN merupakan suatu layer yang memiliki susunan neuron 3D (lebar, tinggi dan kedalaman). Lebar dan tinggi merupakan ukuran layer sedangkan kedalaman mengacu pada jumlah layer. Secara umum jenis layer pada CNN dibedakan menjadi dua yaitu:

2.3.2.1. Layer Ekstraksi Fitur Gambar

Layer ini berada pada awal arsitektur tersusun atas beberapa layer dan setiap layer tersusun atas neuron yang terkoneksi pada daerah lokal (local region) *layer* sebelumnya. *Layer* ini menerima input gambar secara langsung dan memprosesnya hingga menghasilkan keluaran berupa vektor untuk diolah pada layer berikutnya. Lapisan ekstraksi fitur ini terdiri dari :

a. Convolutional Layer

Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada *output* fungsi lain secara berulang. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah kernel (kotak kuning) pada citra disemua offset yang memungkinkan seperti yang ditunjukkan pada Gambar 2. Kotak hijau secara keseluruhan adalah citra yang akan dikonvolusi. Kernel bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar disebelah kanannya.



Gambar 2.2. Operasi konvolusi

Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra input bahkan menurut (Liu 2018) konvolusi adalah cara yang efisien untuk ekstraksi fitur. Konvolusi akan menghasilkan transformasi linear dari data input sesuai informasi spasial pada data. Bobot pada layer tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN.

b. *Rectified Linear Unit* (ReLU)

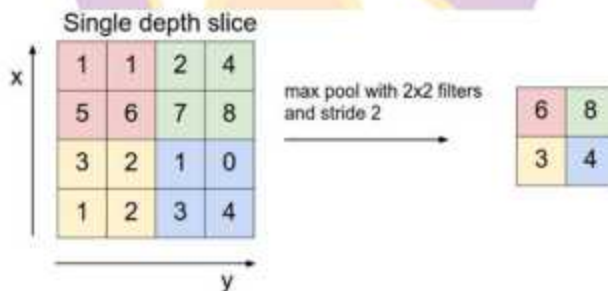
ReLU bertujuan untuk menjaga hasil citra proses konvolusi berada pada domain definit positif. ReLU merupakan salah satu fungsi aktivasi populer dalam deep neural network. Pada fungsi aktivasi dapat mengubah jumlah angka pembobotan dari input yang masuk ke dalam neuron buatan. Fungsi ini harus bersifat non-linear untuk mengkodekan pola yang kompleks dari data. Aktivasi memiliki dua jenis yaitu *Sigmoid* dan *Tanh*. Persamaan yang biasa digunakan dalam fungsi ini adalah :

$$f(x) = \max(x, 0) \quad (3)$$

Dimana : x = nilai pada *feature map*

c. *Subsampling Layer* atau *Pooling Layer*

Proses ini bertujuan untuk mereduksi ukuran sebuah data citra. Dalam pengolahan citra, *subsampling* juga bertujuan untuk meningkatkan invariansi posisi dari fitur. Dalam sebagian besar CNN, metode *subsampling* yang digunakan adalah *max pooling*. *Max pooling* membagi *output* dari *convolution layer* menjadi beberapa *grid* kecil lalu mengambil nilai maksimal dari setiap *grid* untuk menyusun matriks citra yang telah direduksi seperti yang ditunjukkan pada Gambar 3. *Grid* yang berwarna merah, hijau, kuning dan biru merupakan kelompok *grid* yang akan dipilih nilai maksimumnya. Sehingga hasil dari proses tersebut dapat dilihat pada kumpulan *grid* disebelah kanannya. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun objek citra mengalami translasi (pergeseran).



Gambar 2.3. Operasi *max pooling*

2.3.2.2. *Layer* klasifikasi

Layer ini menerima input dari hasil keluaran *layer* ekstraksi fitur gambar berupa vektor kemudian ditransformasikan seperti *Multi Neural Networks* dengan tambahan beberapa *hidden layer*. Hasil keluaran berupa skoring kelas untuk klasifikasi. Lapisan klasifikasi terdiri dari :

a. *Flatten*

Proses ini bertujuan untuk membentuk ulang fitur (*reshape feature map*) menjadi sebuah vector agar bisa kita gunakan sebagai *input* dari *fully-connected layer*.

b. *Fully-connected*

Lapisan *Fully-connected* akan menghitung skor kelas. Seperti Jaringan Saraf biasa dan seperti namanya, setiap neuron dalam lapisan ini akan terhubung ke semua angka dalam volume.

c. *Softmax*

Fungsi *Softmax* menghitung probabilitas dari setiap kelas target atas semua kelas target yang memungkinkan dan akan membantu untuk menentukan kelas target untuk input yang diberikan. Keuntungan utama menggunakan *Softmax* adalah rentang probabilitas output dengan nilai 0 hingga 1, dan jumlah semua probabilitas akan sama dengan satu. Jika fungsi *softmax* digunakan untuk model multi-klasifikasi, dia akan mengembalikan peluang dari masing-masing kelas dan kelas target akan memiliki probabilitas tinggi. *Softmax* menggunakan eksponensial (*e-power*) dari nilai input yang diberikan dan jumlah nilai eksponensial dari semua nilai dalam input. Maka rasio eksponensial dari nilai input dan jumlah nilai eksponensial adalah output dari fungsi *softmax*.

2.3.3. Support Vector Machine (SVM)

SVM adalah algoritma yang dapat digunakan untuk klasifikasi pola (*pattern*) yang dikembangkan oleh V. Vapnic bersama timnya (Fadlisyah 2007) dan menurut (Islam, Raj, and Al-Murad 2017) SVM adalah model keputusan terawasi yang terhubung dengan beberapa algoritma pembelajaran lainnya untuk melakukan tugas *machine learning* dan teori-teori statistik. SVM banyak digunakan untuk menganalisis data, terutama di bidang pengolahan gambar, SVM berfungsi sebagai model classifier atau sebagai model regresi untuk memberikan informasi yang bermakna dengan menganalisis data input.

Tujuan SVM adalah untuk menemukan *hyperplane* optimal $f(w,x) = w \cdot x + b$ untuk memisahkan dua kelas dalam dataset dengan fitur $x \in R^n$. SVM mempelajari parameter dengan memecahkan masalah optimisasi dengan menggunakan persamaan berikut

$$\min \frac{1}{2} w^T w + C \sum_{i=1}^p \max(0, 1 - y'_i (w^T x_i + b)) \quad (4)$$

Dimana :

- $w^T w$ = Norma Manhattan (juga dikenal sebagai norma L1)
- C = Parameter penalti (dapat berupa nilai arbitrer atau nilai yang dipilih menggunakan penyetelan parameter-hiper)
- y' = Label aktual
- $w^T x_i + b$ = Fungsi prediktor

2.3.4. Pengelompokan Data

Agar dapat berfungsi dengan baik, *Convolutional Neural Networks* memerlukan *dataset* yang besar yang berisi berbagai macam kondisi dan variasi (Garcia and Barbedo 2018). Menurut (Putra 2020) pengelompokan data sangat diperlukan untuk proses *training* dan *testing* dalam membangun sebuah model Pembelajaran Mesin atau *Deep Learning*. *Training* merupakan proses konstruksi model yang membutuhkan *training set* (data training) yang merupakan himpunan data yang digunakan untuk melatih atau membangun model dan *validation set* (data validasi) yang merupakan himpunan data untuk mengoptimisasi saat melatih model. *Testing* merupakan proses menguji kinerja model pembelajaran yang membutuhkan *testing set* (data uji) yang merupakan himpunan data untuk menguji model setelah proses latihan.

Training, validation dan *testing data* harus memiliki karakteristik yang sama dan sebaiknya didistribusikan dengan jumlah yang seimbang 50% kasus positif dan 50% kasus negatif. Rasio pembagian *dataset* (*training: validation: testing*) pada umumnya dibagi menjadi (80%: 10%: 10%) atau (90%: 5%: 5%) dan apabila *dataset* berukuran kecil, dapat dibagi menjadi *training* dan *testing* dengan rasio (90%: 10%), (80%: 20%), (70%: 30%) atau (50%: 50%).

2.3.5. Confusion Matrix

Confusion matrix merupakan sebuah tabel yang terdiri dari atas banyaknya baris data uji yang diprediksi benar dan tidak benar oleh model klasifikasi, tabel ini

diperlukan untuk menentukan kinerja suatu model klasifikasi (Wijayanto 2015).

Berikut merupakan tabel dari sebuah *confusion matrix* :

Tabel 2.2. *Confusion matrix*

		Kelas Prediksi	
		<i>Positive</i>	<i>Negative</i>
Hasil Prediksi	<i>Positive</i>	<i>TP (True Positive)</i>	<i>FP (False Positive)</i>
	<i>Negative</i>	<i>FN (False Negative)</i>	<i>TN (True Negative)</i>

True Positive (TP) merupakan data yang diprediksi dengan tepat sebagai keluaran positive atau benar, *True Negative* (TN) merupakan data yang diprediksi tepat sebagai keluaran negatif atau salah, *False Positive* (FP) merupakan data prediksi dengan kurang tepat apabila keluaran berupa positif atau benar dan *False Negative* (FN) merupakan data yang diprediksi kurang tepat.

Untuk mengukur kinerja model sebuah *matrix* dari *confusion matrix*, ada beberapa model kinerja yang biasanya digunakan yaitu, *accuracy*, *precision* dan *specificity*.

a. *Accuracy*

Accuracy akan menghitung seberapa tepat dan akurat sebuah model atau arsitektur mengklasifikasi dengan benar.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

Persamaan (5) merupakan rasio prediksi benar (positif dan negatif) dari keseluruhan data. Dapat dikatakan bahwa akurasi adalah tingkat kedekatan nilai prediksi dengan nilai sebenarnya.

b. *Precision*

Precision merupakan gambaran dari tingkat keakuratan data permintaan dengan hasil prediksi yang diberikan model.

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

Persamaan (6) merupakan rasio *Precision* benar positif dibandingkan dengan keseluruhan hasil yang telah diprediksi positif.

c. *Specificity*

Specificity merupakan gambaran perhtingan kebenaran memprediksi negatif dibandingkan keseluruhan data negatif

$$Specificity = \frac{TN}{TN+FP} \quad (7)$$



BAB III

METODE PENELITIAN

3.1. Jenis, Sifat, dan Pendekatan Penelitian

Pendekatan yang dilakukan pada penelitian ini bersifat kuantitatif dengan jenis penelitian eksperimental, dimana penelitian ini melakukan skenario pengujian dengan membandingkan antara algoritma *convolutional neural network* (CNN) dan kombinasi algoritma *convolutional neural network* (CNN) dan *Support Vector Machine* (SVM) dalam mendeteksi penggunaan masker dari citra yang diperoleh dari kamera *Closed Circuit Television* (CCTV). Skenario percobaan kemudian akan dievaluasi hasil untuk menilai keakuratan dalam mendeteksi dan mengklasifikasikan citra dari setiap skenario untuk diketahui fakta-fakta dari hasil penelitian.

3.2. Metode Pengumpulan Data

Alur pengumpulan data dapat dilihat pada gambar 4. Pada tahap pertama, dilakukan proses pengumpulan dataset yang diperoleh dari media internet. Dari hasil pengumpulan data, diperoleh gambar seseorang yang menggunakan masker berbagai jenis dan tanpa menggunakan masker. Sebagian besar gambar yang diperoleh menampilkan banyak objek sehingga memerlukan *preprocessing* data dengan melakukan *cropping* pada bagian kepala untuk membuang objek yang tidak diperlukan. Gambar selanjutnya akan dipisahkan ke dalam 2 folder di mana folder

pertama menyimpan gambar pengguna masker dan folder kedua menyimpan gambar bukan pengguna masker untuk selanjutnya dijadikan sebagai dataset.



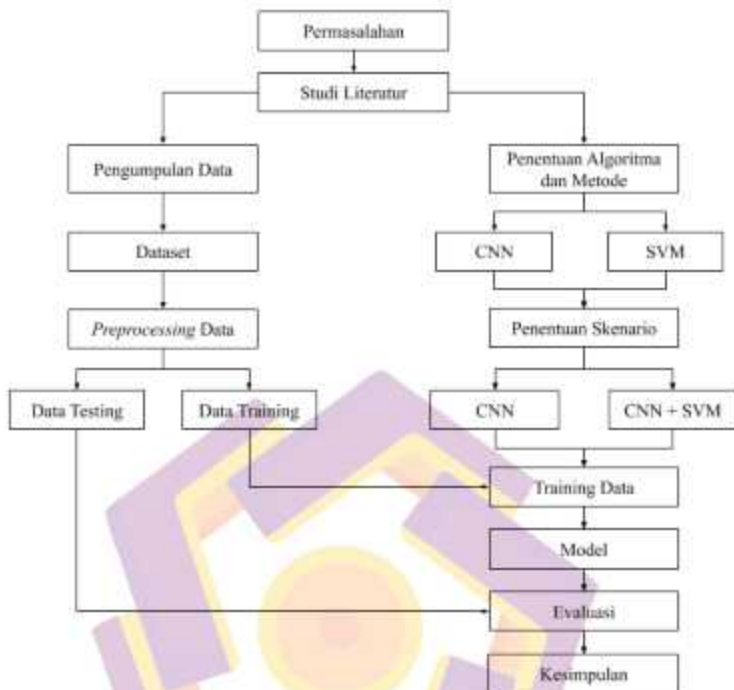
Gambar 3.1. Alur pengumpulan data

3.3. Metode Analisis Data

Metode analisis yang digunakan dalam penelitian ini adalah analisis kuantitatif menggunakan algoritma *Convolutional Neural Network* (CNN) dan *Support Vector Machine* (SVM). Citra yang diperoleh dari proses pengumpulan data akan dilakukan preprocessing sebelum dilakukan ekstraksi fitur dan klasifikasi citra, seperti mengubah citra menjadi grayscale, menyeragamkan ukuran piksel semua citra dan melakukan segmentasi pada citra dengan teknik thresholding untuk memisahkan background dengan foreground sebagai objek citra. Proses segmentasi citra ini dilakukan untuk mengekstraksi background berwarna agar tidak dideteksi sebagai objek/fitur. Percobaan pada penelitian ini melalui 2 skenario utama yaitu menggunakan *Convolutional Neural Network* (CNN) dan kombinasi dari algoritma *Convolutional Neural Network* (CNN) dan *Support Vector Machine* (SVM). Dalam melakukan percobaan ini dilakukan dengan bahasa pemrograman python dengan bantuan libray Keras dan Tensorflow dalam membangun arsitektur CNN dan SVM

3.4. Alur Penelitian

Alur penelitian dapat dilihat pada gambar 5 berikut ini.



Gambar 3.2. Alur penelitian

Alur penelitian secara sistematis dapat dilihat pada Gambar 5. dan akan dijelaskan sebagai berikut:

a. Permasalahan

Tahap ini merupakan tahap awal untuk memulai sebuah penelitian dengan melihat permasalahan yang berada di tengah masyarakat.

b. Studi Literatur

Setelah menentukan permasalahan, proses selanjutnya adalah mencari informasi mengenai hal yang berhubungan dengan masalah tersebut melalui studi literatur. Studi literatur dilakukan dengan membaca jurnal penelitian dan buku yang dianggap relevan dengan permasalahan yang akan diangkat. Proses ini juga

sebagai bahan rujukan penelitian untuk memilih metode atau algoritma yang dianggap sesuai dengan permasalahan.

c. Penentuan algoritma dan metode

Dari studi literatur, penulis telah menentukan algoritma dan metode yang digunakan untuk pendeteksian pengguna masker menggunakan algoritma *Convolutional Neural Network (CNN)* dan *Support Vector Machine (SVM)* pada penelitian ini.

d. Skenario

Setelah menentukan algoritma dan metode, diperlukan skenario untuk penelitian yang akan dilakukan. Dari proses ini ditentukan dua skenario yang akan dilakukan. Skenario pertama, proses ekstraksi dan klasifikasi dilakukan dengan menggunakan algoritma *Convolutional Neural Network (CNN)* dan skenario kedua, proses ekstraksi dilakukan dengan menggunakan *Convolutional Neural Network (CNN)* selanjutnya hasil ekstraksi di klasifikasi menggunakan *Support Vector Machine (SVM)*.

e. Pengumpulan dataset

Machine Learning membutuhkan data untuk memperoleh kecerdasan, sehingga proses pengumpulan data sangat penting pada penelitian ini. Data diperoleh dari media internet yang berupa gambar yang telah melalui proses pelabelan. Akan tetapi, gambar tersebut masih menampilkan objek lain sehingga dilakukan *preprocessing* terlebih dahulu.

f. Dataset

Data yang telah dikumpulkan dari tahap pengumpulan dataset akan diolah dengan cara melabelkan sesuai dengan informasi citra. Label yang akan digunakan pada penelitian ini adalah pengguna masker dan tidak menggunakan masker. Data selanjutnya dikelompokkan ke dalam folder sesuai dengan kelasnya.

g. Training Data dan Model

Setelah menentukan skenario yang ada yaitu CNN dan CNN + SVM, kita mulai melakukan percobaan menggunakan dataset yang telah disediakan sebelumnya. Training data ini juga bertujuan untuk mencari yang mana dari kedua skenario ini yang mempunyai akurasi atau hasil yang terbaik dan nantinya arsitektur tersebut menjadi sebuah model untuk direkomendasikan untuk melakukan evaluasi atau pengujian data.

h. Evaluasi

Model yang telah terbentuk dari proses training, akan dievaluasi dengan menggunakan data testing untuk mendapatkan nilai akurasi menggunakan *Confusion Matrix*.

i. Kesimpulan

Setelah melakukan proses evaluasi dari skenario yang telah ditentukan sebelumnya, proses ini akan menyajikan hasil dari penelitian. Hasil penelitian berupa data fakta yang dihasilkan oleh akurasi dari *confusion matrix* terkait dengan Arsitektur dari algoritma CNN dan SVM

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

Pada bab ini akan dijelaskan tentang analisa dan implementasi dari rancangan tahapan alur penelitian. Terdapat beberapa tahapan proses mulai dari pengumpulan data, penentuan arsitektur, skenario penelitian, *preprocessing dataset* proses pembuatan model dan evaluasi hasil percobaan dari arsitektur CNN dan CNN + SVM.

4.1. Membangun *Dataset*

4.1.1. Pengumpulan Data

Dataset yang digunakan pada penelitian ini adalah gambar orang yang menggunakan masker dan tidak menggunakan masker yang diperoleh dari kaggle yang merupakan situs dan platform untuk perlombaan membuat model terbaik untuk menganalisa dan memprediksi suatu dataset melalui link (<https://www.kaggle.com/wobotintelligence/face-mask-detection-dataset>). Selain kaggle, *google image* juga digunakan sebagai sumber untuk mendapatkan gambar wajah yang menggunakan masker dengan motif wajah dan dilakukan pengunduhan secara manual. Dataset yang diperoleh dilakukan proses *cropping* untuk mendapatkan gambar wajah yang menggunakan masker dan tidak menggunakan masker seperti pada gambar 4.1.



Gambar 4.1. *Cropping* gambar

Proses *cropping* dilakukan terhadap 1267 gambar dari kaggle sehingga menjadi 1650 gambar yang terbagi menjadi 800 gambar wajah yang menggunakan masker dan 850 gambar wajah yang tidak menggunakan masker dan 27 gambar dari *google image* sehingga menjadi 50 gambar wajah yang menggunakan masker dengan motif wajah dan selanjutnya dilakukan proses pelabelan. Total *dataset* yang digunakan adalah 1700 gambar yang terbagi menjadi 850 gambar wajah yang menggunakan masker dan 850 gambar wajah yang tidak menggunakan masker

My Drive > dataset -

Folders

mask

nonmask

Gambar 4.2. Proses pelabelan dataset

Dataset yang telah melalui proses *cropping* dikumpulkan kedalam folder berdasarkan kelas gambar untuk selanjutnya dilakukan proses pelabelan dengan memberikan nama pada folder berdasarkan kelasnya seperti pada gambar 4.2. Dataset disimpan di dalam *google drive*, karena proses pengolahan citra menggunakan *google colab*. Berikut ditunjukkan sampel untuk masing-masing kelas citra yang dapat dilihat pada gambar 4.3.



Gambar 4.3. Sampel dataset *mask* dan *nonmask*.

Dataset yang terkumpul memiliki ukuran yang berbeda dikarenakan posisi wajah yang beragam pada gambar aslinya, sehingga diperlukan proses *resize* untuk menyamakan ukuran dataset tersebut sebelum dilakukan proses pelatihan model.

4.1.2. *Preprocessing Data*

Setelah data terkumpul kedalam folder berdasarkan kelas masing-masing, selanjutnya dilakukan *preprocessing* terhadap data tersebut. Pada tahap ini, proses

diawali dengan melakukan *load* data dari *google drive* ke dalam *google colab* untuk selanjutnya dilakukan *resize* dengan tujuan untuk menyamakan ukuran data menjadi 224 x 224 piksel. Setelah semua data memiliki ukuran yang seragam, selanjutnya data dan label data disimpan kedalam variabel dalam bentuk array. Proses ini dilakukan dengan perintah sebagai berikut :

```
import numpy as np
import os, re, glob
from google.colab import drive
drive.mount('/content/drive')

PATH = '/content/drive/My Drive/dataset/'
dirs = os.listdir(PATH)

im_size = 224
label = 0
im_arr = []
lb_arr = []
X = []
y = []
for i in dirs:
    count = 0
    for pic in glob.glob(PATH+i+'*'):
        im = cv2.imread(pic)
        im = cv2.resize(im, (im_size, im_size))
        im = np.array(im)
        count = count + 1
        X.append(im)
        y.append(label)
        if(count <= 10):
            im_arr.append((str(i):im))
    print("Jumlah "+str(i)+" : "+str(count))
    label = label + 1
    lb_arr.append(i)
X = np.array(X)
y = np.array(y):
```

Hasil akhir dari *preprocessing* adalah matriks dalam bentuk array multidimensi yang tersimpak dalam variabel dimana *X* menyimpan data citra dan *y*

label dari citra dan jika data citra yang telah melalui proses ini ditampilkan kembali akan terlihat seperti pada gambar 4.4.



Gambar 4.4. Hasil *Preprocessing*

Gambar 4.4 menunjukkan bahwa *dataset* telah memiliki ukuran yang seragam dan memiliki label yang sesuai yaitu *mask* dan *nonmask* yang ditampilkan dalam format gambar *red green blue* (RGB).

4.1.3. Pengelompokan Data

Setelah melakukan *preprocessing* terhadap *dataset* selanjutnya dilakukan proses pengelompokan data dengan membagi *dataset* menjadi tiga bagian yaitu data latih: data validasi: data uji dengan rasio 70%:10%:20%. *Train_test_split* yang disediakan oleh *Sklearn* merupakan *library Python* yang digunakan untuk membagi data secara acak dengan perintah sebagai berikut :

```
from sklearn.model_selection import train_test_split

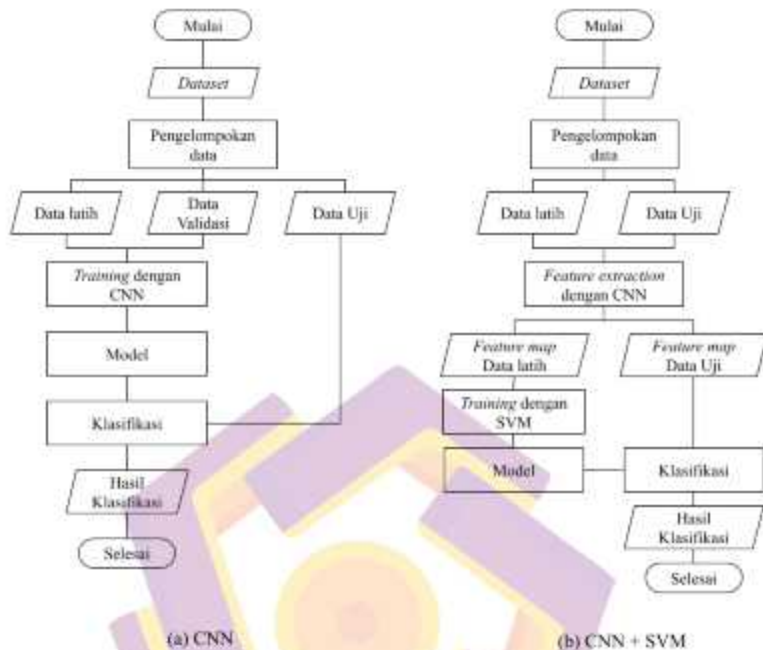
train_ratio = 0.70
validation_ratio = 0.10
test_ratio = 0.20

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=1 - train_ratio, random_state=1)

X_val, X_test, y_val, y_test = train_test_split(X_test,
                                                y_test, test_size=test_ratio/(test_ratio +
                                                validation_ratio), random_state=1)
```

4.2. Analisis Data

Setelah dataset dilakukan proses *preprocessing* dataset citra telah siap digunakan untuk proses pembelajaran dalam membentuk model klasifikasi menggunakan *Convolutional Neural Network* (CNN) dan dilakukan pula percobaan dengan mengkombinasikan *Convolutional Neural Network* (CNN) dengan *Support Vector Machine* (SVM). Proses analisis data yang dilakukan pada penelitian ini dapat dilihat pada gambar 4.5.



Gambar 4.5. Alur Prose Analisis Data

Berdasarkan gambar 4.5 terlihat perbedaan proses antara algoritma CNN dengan kombinasi algoritma CNN dan SVM. Algoritma CNN menggunakan 3 kelompok data yaitu data latih dan data validasi untuk proses traing dan data uji untuk proses klasifikasi sedangkan proses untuk kombinasi algoritma CNN dan SVM hanya menggunakan 2 kelompok data yaitu data latih dan data uji. CNN digunakan untuk melakukan proses *feature extraction* yang akan menghasilkan *feature map* untuk digunakan sebagai masukan pada proses *training* dan *testing* dari algoritma SVM.

4.2.1. Convolutional Neural Network (CNN)

Untuk mendapatkan hasil klasifikasi dengan akurasi yang baik, percobaan ini menggunakan 3 arsitektur yang telah teruji tingkat kinerja dari masing-masing arsitektur pada sebuah kompetisi ImageNet. Ketiga arsitektur tersebut adalah VGG-16, ResNet50 dan MobileNet. Berikut perintah untuk mengunduh arsitektur.

```
# Download arsitektur VGG16
import tensorflow as tf
vgg_model = tf.keras.applications.vgg16.VGG16()

# Download arsitektur ResNet
from keras.applications.resnet50 import ResNet50
resnet_model=ResNet50()

# Download arsitektur MobileNet
from keras.applications import MobileNet
mobile = MobileNet()
```

Untuk menjalankan arsitektur CNN pada pemrograman *python*, percobaan dimulai dengan meng-compile model kemudian menggunakan fungsi *fit()* untuk melakukan pelatihan menggunakan data latih dan menggunakan fungsi *evaluate()* untuk melakukan validasi hasil dari pelatihan menggunakan data validasi. Selain itu, fungsi *time()* juga digunakan untuk menghitung waktu pelatihan dan evaluasi. Proses ini dilakukan dengan menggunakan perintah sebagai berikut :


```

vgg_model.compile(optimizer=SGD(lr=0.001,
                                momentum=0.9,
                                nesterov=False),
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])

start_time = time.time()
history = vgg_model.fit(X_train,
                        y_train,
                        validation_data=(X_val, y_val),
                        epochs=50,
                        batch_size=16)
print("--- %s seconds ---" % (time.time() - start_time))

```

Untuk dapat melakukan pembelajaran hingga klasifikasi, CNN akan melakukan beberapa proses terhadap citra seperti *convolution*, *Pooling* dan *flatten*.

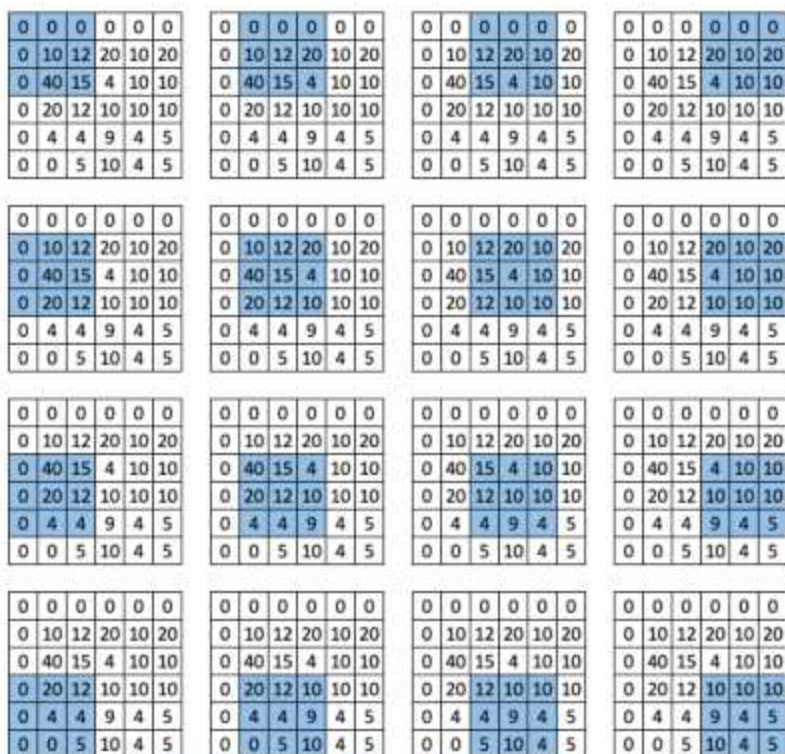
4.2.1.1. Proses *Convolution*

Proses *convolution* yang terjadi pada *convolutional layer* berfungsi mengekstrak fitur-fitur penting yang ada di dalam citra. Proses ini menggunakan operasi perkalian matriks antara matriks citra dengan matriks kernel/filter. Gambar dapat diolah oleh mesin/sistem dengan cara mengubahnya menjadi kumpulan matriks sehingga dapat diolah. Untuk mempermudah dalam memahami proses konvolusi ini, penulis menggunakan sampel matriks dari sebuah citra dengan ukuran 6 x 6 piksel. Penulis tidak menggunakan citra asli pada penelitian ini, karena ukuran citra sesungguhnya yang digunakan pada penelitian ini memiliki ukuran 224 x 224 piksel.

$$\begin{array}{|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 10 & 12 & 20 & 10 & 20 \\ \hline 0 & 40 & 15 & 4 & 10 & 10 \\ \hline 0 & 20 & 12 & 10 & 10 & 10 \\ \hline 0 & 4 & 4 & 9 & 4 & 5 \\ \hline 0 & 0 & 5 & 10 & 4 & 5 \\ \hline \end{array}
 \times
 \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array}
 =
 \begin{array}{|c|c|c|c|} \hline -39 & 16 & 9 & -6 \\ \hline -54 & 72 & 14 & -12 \\ \hline -43 & 51 & 9 & -2 \\ \hline -25 & -10 & 3 & 13 \\ \hline \end{array}$$

Gambar 4.6. Proses *convolution*

Pada Gambar 4.6 ditunjukkan contoh perhitungan operasi *convolution*. Pada contoh ini digunakan ukuran citra dalam matriks 6 x 6 piksel dengan kernel 3 x 3 dan stride 1 yang berarti pergerakan akan bergeser 1 matriks ke kanan dan satu matriks ke bawah ketika sudah sampai ujung paling kanan. Nilai-nilai yang terdapat di setiap kolom matriks merepresentasikan dari fitur citra. Dalam proses matriks *convolution* angka-angka tersebut berkisar antara 0 – 255.



Gambar 4.7. Pergerakan proses *convolution*

Pada Gambar 4.7 diperlihatkan pergerakan *kernel* 3×3 pada proses *convolution* terhadap matriks 6×6 dengan nilai *stride* 1. Setiap blok matriks pada citra tersebut akan dikalikan dengan matriks *kernel*. *Kernel* yang digunakan contoh di atas menggunakan jenis *kernel* vertikal. Dalam praktiknya nilai-nilai bobot dalam kernel tersebut akan selalu berubah karena proses update bobot pada saat dilakukan proses *training*. Berikut adalah 3 sampel perhitungan konvolusi pada tiga pergerakan pertama

$$P1 = (0 \times 1) + (0 \times 0) + (0 \times (-1)) + (0 \times 2) + (10 \times 0) + (12 \times (-2)) \\ + (0 \times 1) + (40 \times 0) + (15 \times (-1))$$

$$P1 = -39$$

$$P2 = (0 \times 1) + (0 \times 0) + (0 \times (-1)) + (10 \times 2) + (12 \times 0) + (20 \times (-2)) \\ + (40 \times 1) + (15 \times 0) + (4 \times (-1))$$

$$P2 = 16$$

$$P3 = (0 \times 1) + (0 \times 0) + (0 \times (-1)) + (12 \times 2) + (20 \times 0) + (10 \times (-2)) \\ + (15 \times 1) + (4 \times 0) + (10 \times (-1))$$

$$P3 = 9$$

Perhitungan di atas merupakan contoh operasi perkalian antara blok pada matriks citra dengan matriks *kernel* sehingga menghasilkan matriks baru dengan ukuran 4 x 4. Setelah dilakukan proses *convolution*, tahap selanjutnya adalah proses *pooling*.

4.2.1.2. Proses *Pooling*

Tahap pada proses ini berfungsi untuk mengecilkan dimensi dari citra. Proses *pooling* ini terdiri dari 2 jenis yaitu *max pooling* dan *average pooling*. Pada penelitian kali ini penulis akan mencotohkan proses *pooling* menggunakan jenis *max pooling* yaitu akan memilih nilai yang tertinggi di dalam satu blok.

-39	16	9	-6	=	72	14
-54	72	14	-12		51	13
-43	51	9	-2			
-25	-10	3	13			

Gambar 4.8. Proses *Pooling*

Pada Gambar 4.8 ditunjukkan proses pooling terhadap citra hasil proses konvolusi sebelumnya tadi. Proses pooling yang ada pada gambar di atas menggunakan jumlah filter adalah 2 x 2 dengan jumlah stride adalah 2, yang berarti pergeseran terjadi sebanyak 2 matriks. Blok pertama ditandai dengan warna kuning dan karena penulis menggunakan teknik max pooling, sehingga dari keempat nilai pada blok tersebut akan dipilih nilai yang terbesar, dalam kasus kali ini nilai terbesar adalah 72 untuk blok pertama, sehingga didapatkan matriks baru hasil proses pooling seperti ditunjukkan pada Gambar 4.7 yaitu dengan ukuran 2 x 2.

4.2.1.3. *Fully Connected*

Proses klasifikasi citra ditangani oleh *fully connected layer*. Citra hasil pengolahan proses *feature extraction* masih berbentuk matriks multidimensi sedangkan pada layer klasifikasi, tidak dapat menerima inputan berupa matriks multidimensi yang berbentuk array, sehingga perlu dilakukan proses *flatten*.

-39	16	9	-6
-54	72	14	-12
-43	51	9	-2
-25	-10	3	13

=

-39
16
9
-6
-54
72
14
-12
-43
51
9
-2
-25
-10
3
13

Gambar 4.9. Proses *Flatten*

Proses *flatten* seperti yang terlihat pada gambar 4.9 bertujuan untuk mengubah masukan yang berupa array menjadi sebuah vektor yang akan dijadikan sebagai inputan dalam *fully connected layer*. Pada tahap ini terdapat 1 layer yaitu *dense layer* dengan fungsi aktivasi yaitu *softmax* yang akan bertugas dalam menentukan kelas dari setiap citra yang diinputkan berdasarkan tingkat probabilitas yang paling mendekati dengan kelas citra.

4.2.1.4. Arsitektur VGG-16

Arsitektur VGG-16 terdiri dari 16 blok layer yang disebut sebagai *feature extraction layer*, yang terdiri dari *layer convolution* dan *layer pooling* yang bertugas mengekstrak fitur dan mengecilkan dimensi dari citra.

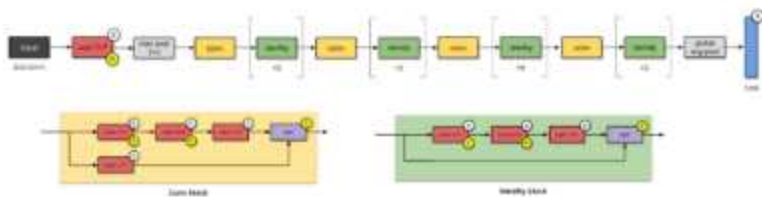


Gambar 4.10. Arsitektur VGG-16

Gambar 4.10 merupakan struktur dari arsitektur VGG-16. Arsitektur ini digunakan ukuran *filter* atau *karnel* yang konsisten sama yaitu 3 x 3 dengan jumlah *stride* adalah 1 untuk *Convolutional Layer*. Pada *Pooling Layer* digunakan jenis *max pooling* dengan parameter *padding* semuanya adalah "same" dengan jumlah filter 2 x 2 dan *stride* yang digunakan adalah 2. Lapisan terakhir yaitu lapisan *Fully Connected (FC) Layer* yang terdiri dari 2 FC dengan aktivasi *softmax*. Arsitektur ini menggunakan jumlah parameter *output* sebesar 1000 sehingga perlu dilakukan *fine tuning* untuk mengubah parameter *output* menjadi 2. Fungsi dari *Fully Connected Layer* ini adalah untuk menentukan kelas pada citra berdasarkan tingkat probabilitas dari setiap kelas yang paling mendekati.

4.2.1.5. Arsitektur ResNet50

Resnet atau kepanjangan dari *Residual Network* merupakan jaringan saraf klasik yang banyak digunakan pada dunia komputer vision. Arsitektur ini telah memenangkan kompetisi Imagenet pada tahun 2015 lalu.

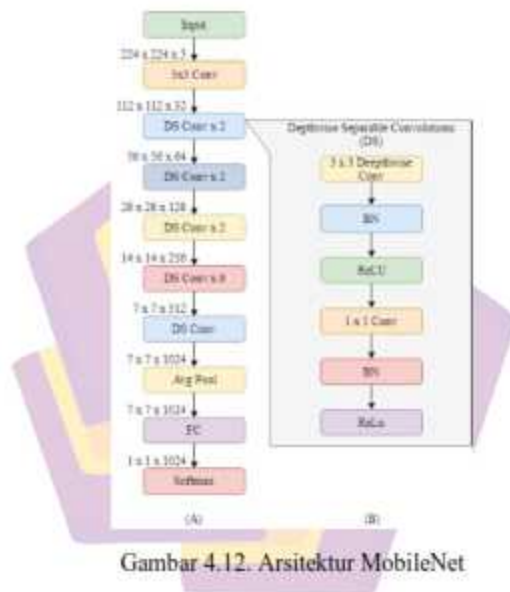


Gambar 4.11. Arsitektur ResNet50

Pada gambar 4.11 ditunjukkan arsitektur ResNet50. Arsitektur ini terkenal dengan kedalaman jaringannya, yaitu terdiri lebih dari 177 lapisan. Arsitektur ini yang pertama kali memperkenalkan konsep *skip connection*. Salah satu keunggulan dari arsitektur ini adalah penggunaan konsep *skip connection*, yaitu menumpuk lapisan *convolution* dan melewatinya dari inputan asli ke lapisan selanjutnya. Penumpukan jaringan yang dalam tidak berpengaruh signifikan dalam meningkatkan akurasi, hal tersebut justru akan menurunkan kinerja proses *learning*. Oleh karena itu pada arsitektur ResNet50 ini diterapkan konsep *skip connection* untuk mengantisipasi terjadinya penurunan kinerja proses pembelajaran model. Arsitektur ini juga menggunakan jumlah parameter *output* sebesar 1000 sehingga perlu dilakukan *fine tuning* untuk mengubah parameter *output* menjadi 2.

4.2.1.6. Arsitektur MobileNet

Arsitektur MobileNet ini merupakan arsitektur CNN yang paling ringan dibandingkan dengan arsitektur CNN yang lain, karena dalam proses pembelajaran model tidak membutuhkan resource yang banyak dan proses komputasi yang ringan.



Gambar 4.12. Arsitektur MobileNet

Gambar 4.12 memperlihatkan arsitektur MobileNet yang menggunakan konsep *convolution* yang dapat dipisahkan secara mendalam kecuali pada lapisan pertama. Lapisan *convolution* pertama merupakan lapisan *convolution* yang penuh. Semua lapisan diikuti dengan normalisasi batch dan operasi non linear yaitu aktivasi ReLU. Tujuan dari konsep ini adalah untuk mengurangi ukuran dan proses komputasi model. Total keseluruhan *layer* yang ada pada arsitektur ini berjumlah 93 *layer* yang terdiri dari *feature extraction* dan *classification layer*. Perbedaan

mendasar antara arsitektur MobileNet dengan arsitektur CNN pada umumnya adalah penggunaan lapisan atau *convolution layer* dengan ketebalan filter yang sesuai dengan ketebalan dari input image. MobileNet membagi konvolusi menjadi *depthwise convolution* dan *pointwise convolution*.

4.2.1.7. Fine Tuning

Arsitektur yang digunakan pada peengujian ini adalah VGG16, ResNet50 dan MobileNet. Ketiga arsitektur yang dilakukan menggunakan 1000 kelas sehingga memerlukan proses *fine tuning* untuk menyesuaikan jumlah kelas pada layer konvolusi terhadap dataset yang akan diproses.

block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 1000)	4097000

Gambar 4.13. Layer VGG16 Sebelum *Fine Tuning*

Dibagian akhir pada *layer* arsitektur VGG16, ResNet50 dan MobileNet memiliki jumlah kelas gambar yang tidak sesuai dengan jumlah kelas pada pengujian yang dilakukan. Dicontohkan pada arsitektur VGG16 yang terlihat pada gambar 4.13 menunjukkan bahwa arsitektur ini menggunakan 1000 kelas, sehingga diperlukan penyesuaian jumlah kelas. Untuk dapat melakukan klasifikasi menggunakan *softmax* digunakan perintah sebagai berikut :

```
x = mobile.layers[-2].output
output = Dense(units=2, activation='softmax')(x)
vgg_model = keras.Model(inputs=mobile.input, outputs=output)
```

Untuk melakukan *feature extraction* untuk sebagai masukan algoritma SVM, dilakukan dengan menghilangkan beberapa *layer* dibagian akhir setiap arsitektur hingga menyisakan proses *flatten* atau *reshape* sebagai *layer* akhir. Sebagai contoh pada skenario 4, perintah untuk menghilangkan tiga *layer* akhir pada arsitektur ini adalah sebagai berikut :

```
x = mobile.layers[-4].output
vgg_model = keras.Model(inputs=mobile.input, outputs=x)
```

Gambar 4.13 bagian (a) diperlihatkan *layer* akhir dari arsitektur VGG-16 setelah dilakukan proses *fine tuning*. *Dense* yang semula memiliki nilai 1000 sebagai jumlah kelasnya, sekarang setelah dilakukan proses *fine tuning* berubah menjadi 2 sesuai dengan jumlah kelas pada gambar yang akan di proses pada pengujian ini yaitu *mask* dan *nonmask*.

block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
dense_1 (Dense)	(None, 2)	8194

(a)

block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0

(b)

Gambar 4.14. Layer VGG16 Setelah *Fine Tuning*

Gambar 4.14 bagian (b) diperlihatkan layer akhir dari arsitektur VGG-16 setelah dilakukan proses *fine tuning*. Tiga *layer Dense* terlihat pada gambar 4.5, sekarang setelah dilakukan proses *fine tuning* tiga *layer* tersebut telah hilang dan arsitektur dari VGG16 di akhiri dengan *layer flatten*. Proses *fine tuning* tidak hanya dilakukan pada arsitektur VGG16, dua arsitektur lain yaitu ResNet50 dan MobileNet juga dilakukan proses *fine tuning* seperti yang telah dicontohkan pada arsitektur VGG16.

4.2.2. Support Vector Machine (SVM)

Pada penelitian ini, digunakan algoritma SVM untuk melakukan klasifikasi citra yang telah di *extraction* menggunakan arsitektur dari CNN. Untuk

mendapatkan *feature map* dari citra, digunakan fungsi *predict()* dari model CNN dan digunakan fungsi *time()* untuk menghitung waktu *extraction*. Sebagai contoh perintah untuk melakukan *extraction* dengan menggunakan arsitektur VGG-16 terlihat seperti pada kode berikut :

```
start_time = time.time()
train_extractor = vgg_model.predict(X_train)
test_extractor = vgg_model.predict(X_test)
print("--- %s seconds ---" % (time.time() - start_time))
```

Setelah data *dataset* di *extraction*, selanjutnya dilakukan proses pelatihan. Pelatihan dimulai dengan memberikan parameter $C = 100$ dan *kernel* = "rbf" kemudian menggunakan fungsi *fit()* untuk melakukan pelatihan menggunakan data latih dan fungsi *time()* digunakan untuk menghitung waktu pelatihan. Proses ini dilakukan dengan menggunakan perintah sebagai berikut :

```
start_time = time.time()
SVM_model = svm.SVC(C=100, kernel='rbf')
SVM_model.fit(train_extractor, y_train)
print("--- %s seconds ---" % (time.time() - start_time))
```

Model yang telah terbentuk dari hasil pelatihan disimpan ke dalam variabel *SVM_model* untuk selanjutnya digunakan pada proses pengujian. Fungsi *predict()* digunakan untuk melakukan proses pengujian dengan menggunakan data uji dan fungsi *time()* juga digunakan menghitung waktu pengujian seperti yang terlihat pada kode berikut:

```
start_time = time.time()
predictions = SVM_model.predict(test_extractor)
print("--- %s seconds ---" % (time.time() - start_time))
```

Hasil pengujian akan ditampilkan dan dievaluasi menggunakan *confusion matrix* untuk mendapatkan nilai *accuracy*, *precision* dan *recall*.

4.2.3. Skenario Percobaan

Untuk mendapatkan model klasifikasi yang diharapkan, skenario percobaan yang akan dibuat dengan membandingkan metode CNN dengan metode CNN yang dikombinasikan dengan SVM yang menggunakan tiga arsitektur dari CNN. Dalam penelitian ini akan melakukan 6 skenario percobaan yang dapat dilihat pada Tabel 4.1.

Tabel 4.1. Skenario percobaan

No	Skenario	Keterangan
1	S1	CNN (Arsitektur VGG-16)
2	S2	CNN (Arsitektur ResNet50)
3	S3	CNN (Arsitektur MobileNet)
4	S4	CNN (Arsitektur VGG-16) + SVM
5	S5	CNN (Arsitektur ResNet50) + SVM
6	S6	CNN (Arsitektur MobileNet) + SVM

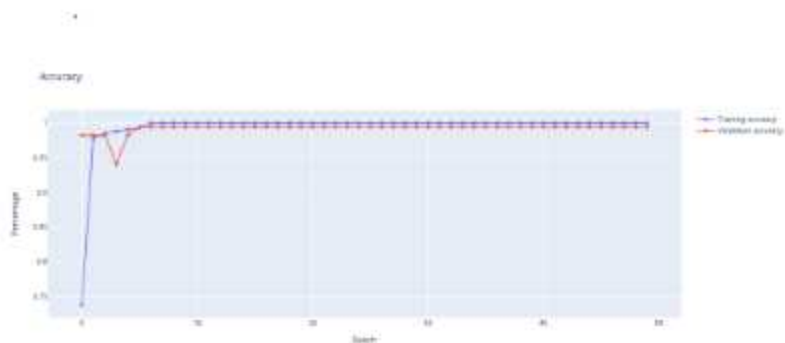
4.2.4. Hasil Training dan Testing Data

Setelah *dataset* dan arsitektur telah siap untuk digunakan, tahap selanjutnya pada pengujian ini adalah melakukan *training* dengan menggunakan data latih dan data validasi sebagai masukan untuk menghasilkan semua model sesuai dengan skenario pengujian. Selanjutnya, model yang telah dibangun akan diuji menggunakan data uji.

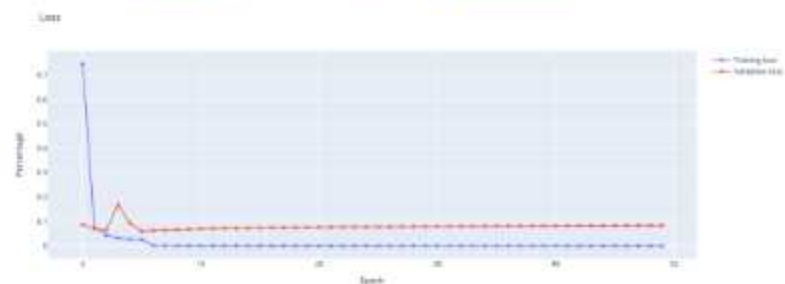
4.2.3.1. Skenario Satu

Skenario pertama pada pengujian ini adalah melakukan klasifikasi citra menggunakan CNN dengan memanfaatkan arsitektur VGG16. Hasil dari proses

training dan *validation* pada pengujian ini jika ditampilkan kedalam grafik maka akan terlihat seperti pada gambar 4.15 dan gambar 4.16.



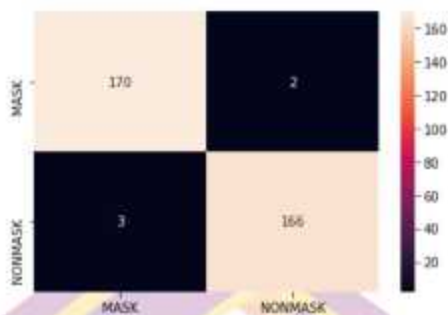
Gambar 4.15. Grafik *training and validation accuracy* skenario I



Gambar 4.16. Garafik *training and validation loss* skenario I

Gambar 4.15 memperlihatkan proses *training* dan *validation* dari *epoch* pertama sampai 50 mengalami peningkatan dimana *training accuracy* mencapai 1,00 dan *validation accuracy* mencapai 0.9706. Pada gambar 4.16 memperlihatkan nilai *loss* dari *epoch* pertama sampai dengan 50 yang semakin menurun dimana *training loss* mencapai 0.000036396 dan *validation loss* mencapai 0.0843 dengan

waktu *training* dan *validatin* 892.364872694015 detik. Model yang yetah terbentuk dari proses *training* akan diuji kembali menggunakan data uji dan dievaluasi menggunakan *confusion matrix*.



Gambar 4.17. Garafik *confusion matrix* pengujian skenario 1

Fungsi *predict()* digunakan untuk melakukan prediksi atau pengujian dengan menngunakan 341 data uji dimana data tersebut tidak terdapat pada data latih dan data validasi. Gambar 4.17 memperlihatkan hasil pengujian dimana dari 341 data yang terbagi dalam kelas *mask* 172 data dan *nonmask* 169 data diketahui 170 data kelas *mask* di prediksi benar, 2 data di prediksi salah dan 166 data kelas *nonmask* di prediksi benar, 3 data di prediksi salah. Maka dari hasil dari proses ini dapat dihitung nilai *accuracy*, *precision* dan *recall*

$$accuracy = \frac{170 + 166}{341} = 0,9853$$

$$precision(mask) = \frac{170}{170 + 3} = 0,9827$$

$$precision(nonmask) = \frac{166}{166 + 2} = 0,9881$$

$$\textit{precision} = \frac{0,9827 + 0,9881}{2} = 0,9854$$

$$\textit{Recall}(\textit{mask}) = \frac{170}{170 + 2} = 0,9884$$

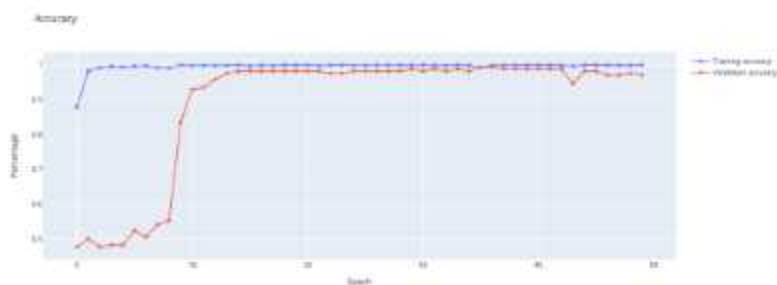
$$\textit{Recall}(\textit{nonmask}) = \frac{166}{166 + 3} = 0,9822$$

$$\textit{Recall} = \frac{0,9884 + 0,9822}{2} = 0,9853$$

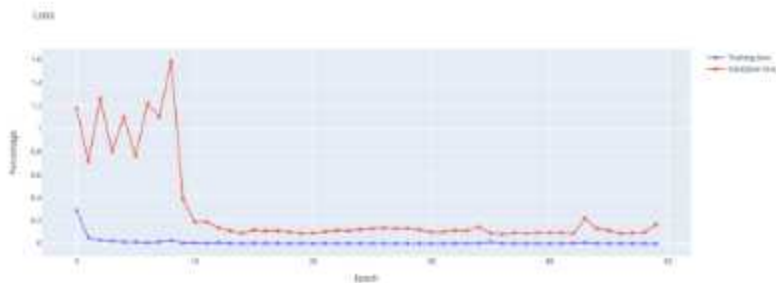
Dari perhitungan diatas, pengujian terhadap model yang telah dibangun diketahui menghasilkan nilai *accuracy* 0,9853, nilai *precision* 0,9854 dan nilai *recall* 0,9853. Fungsi *time()* juga digunakan pada prose pengujian dan menghasilkan waktu 1.54427170753479 detik.

4.2.3.2. Skenario Dua

Skenario kedua pada pengujian ini adalah melakukan klasifikasi citra menggunakan CNN dengan memanfaatkan arsitektur ResNet50. Hasil dari proses *training* dan *validation* pada pengujian ini jika ditampilkan kedalam grafik maka akan terlihat seperti pada gambar 4.18 dan gambar 4.19.

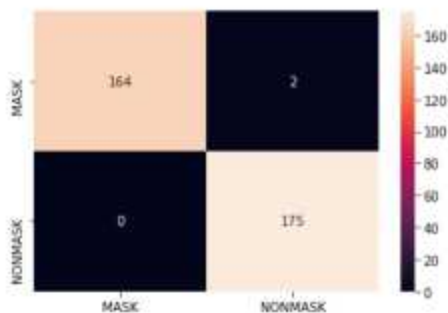


Gambar 4.18. Grafik *traing and validation accuracy* skenario 2



Gambar 4.19. Grafik *training and validation loss* skenario 2

Gambar 4.18 memperlihatkan proses *training* dan *validation* dari *epoch* pertama sampai 50 mengalami peningkatan dimana *training accuracy* mencapai 1,00 dan *validation accuracy* mencapai 0,9706. Pada gambar 4.19 memperlihatkan nilai *loss* dari *epoch* pertama sampai dengan 50 yang semakin menurun dimana *training loss* mencapai 0,00021927 dan *validation loss* mencapai 0,1666 dengan waktu *training* dan *validatin* 703,546879053115 detik. Model yang telah terbentuk dari proses *training* akan diuji kembali menggunakan data uji dan dievaluasi menggunakan *confusion matrix*.



Gambar 4.20. Garafik *confusion matrix* pengujian skenario 2

Fungsi *predict()* digunakan untuk melakukan prediksi atau pengujian dengan menggunakan 341 data uji dimana data tersebut tidak terdapat pada data latih dan data validasi. Gambar 4.20 memperlihatkan hasil pengujian dimana dari 341 data yang terbagi dalam kelas *mask* 166 data dan *nonmask* 175 data diketahui 164 data kelas *mask* di prediksi benar, 2 data di prediksi salah dan 175 data kelas *nonmask* di prediksi benar, tidak ada data di prediksi salah. Maka dari hasil dari proses ini dapat dihitung nilai *accuracy*, *precision* dan *recall*

$$\mathbf{accuracy = \frac{164 + 175}{341} = 0,9941}$$

$$precision(mask) = \frac{164}{164 + 0} = 1$$

$$precision(nonmask) = \frac{175}{175 + 2} = 0,9887$$

$$\mathbf{precision = \frac{1 + 0,9887}{2} = 0,9944}$$

$$Recall(mask) = \frac{164}{164 + 2} = 0,9880$$

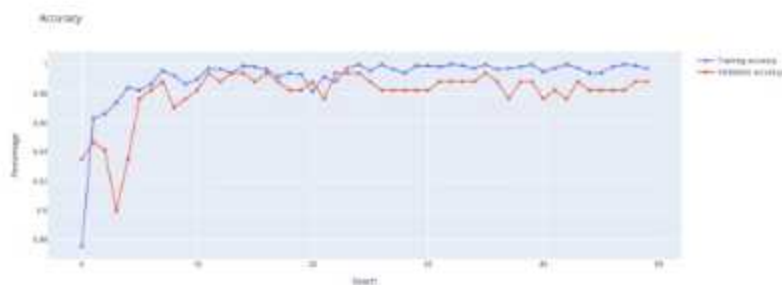
$$Recall(nonmask) = \frac{175}{175 + 0} = 1$$

$$\mathbf{Recall = \frac{0,9880 + 1}{2} = 0,9940}$$

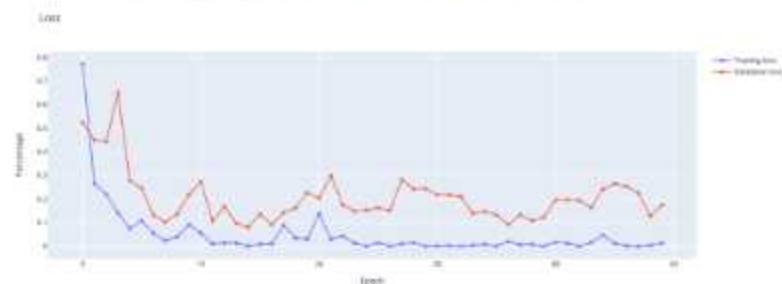
Dari perhitungan diatas, pengujian terhadap model yang telah dibangun diketahui menghasilkan nilai *accuracy* 0,9941, nilai *precision* 0,9944 dan nilai *recall* 0,9940. Fungsi *time()* juga digunakan pada prose pengujian dan menghasilkan waktu 1.79551267623901 detik.

4.2.3.3. Skenario Tiga

Skenario ketiga pada pengujian ini adalah melakukan klasifikasi citra menggunakan CNN dengan memanfaatkan arsitektur MobileNet. Hasil dari proses *training* dan *validation* pada pengujian ini jika ditampilkan kedalam grafik maka akan terlihat seperti pada gambar 4.21 dan gambar 4.22



Gambar 4.21. Grafik *training and validation accuracy* skenario 3



Gambar 4.22. Grafik *training and validation loss* skenario 3

Gambar 4.21 memperlihatkan proses *training* dan *validation* dari *epoch* pertama sampai 50 mengalami peningkatan dimana *training accuracy* mencapai 0.999 dan *validation accuracy* mencapai 0.9882. Pada gambar 4.22

memperlihatkan nilai *loss* dari *epoch* pertama sampai dengan 50 yang semakin menurun dimana *training loss* mencapai 0.0058 dan *validation loss* mencapai 0.1745 dengan waktu *training* dan *validatin* 411.085926055908 detik. Model yang yethah terbentuk dari proses *training* akan diuji kembali menggunakan data uji dan dievaluasi menggunakan *confusion matrix*.



Gambar 4.23. Garafik *confusion matrix* pengujian skenario 2

Fungsi *predict()* digunakan untuk melakukan prediksi atau pengujian dengan menggunakan 341 data uji dimana data tersebut tidak terdapat pada data latih dan data validasi. Gambar 4.23 memperlihatkan hasil pengujian dimana dari 341 data yang terbagi dalam kelas *mask* 177 data dan *nonmask* 164 data diketahui 170 data kelas *mask* di prediksi benar, 7 data di prediksi salah dan 164 data kelas *nonmask* di prediksi benar, tidak ada data di prediksi salah. Maka dari hasil dari proses ini dapat dihitung nilai *accuracy*, *precision* dan *recall*

$$\text{accuracy} = \frac{170 + 164}{341} = 0,9795$$

$$\text{precision}(\text{mask}) = \frac{170}{170 + 0} = 1$$

$$\text{precision}(\text{nonmask}) = \frac{164}{164 + 7} = 0,9591$$

$$\text{precision} = \frac{1 + 0,9591}{2} = 0,9795$$

$$\text{Recall}(\text{mask}) = \frac{170}{170 + 7} = 0,9605$$

$$\text{Recall}(\text{nonmask}) = \frac{164}{164 + 0} = 1$$

$$\text{Recall} = \frac{0,9795 + 1}{2} = 0,9802$$

Dari perhitungan diatas, pengujian terhadap model yang telah dibangun diketahui menghasilkan nilai *accuracy* 0,9795, nilai *precision* 0,9795 dan nilai *recall* 0,9802. Fungsi *time()* juga digunakan pada prose pengujian dan menghasilkan waktu 0.773337602615356 detik.

4.2.3.4. Skenario Empat

Skenario kelima pada pengujian ini adalah melakukan klasifikasi citra menggunakan SVM dengan memanfaatkan CNN yang menggunakan arsitektur VGG16 untuk menghasilkan *feature map*. Proses *extraction* pada skenario ini dilakukan dalam waktu 48.3147387504577 detik, pelatihan dilakukan dalam waktu 27.221982717514 detik dan pengujian dilakukan dalam waktu 6.94587159156799 detik. Hasil pengujian akan ditampilkan dan dievaluasi menggunakan *confusion matrix*.



Gambar 4.24. Garafik *confusion matrix* pengujian skenario 4

Gambar 4.24 memperlihatkan hasil pengujian dimana dari 340 data yang terbagi dalam kelas *mask* 163 data dan *nonmask* 177 data diketahui 152 data kelas *mask* di prediksi benar, 11 data di prediksi salah dan 171 data kelas *nonmask* di prediksi benar, 6 data di prediksi salah. Maka dari hasil dari proses ini dapat dihitung nilai *accuracy*, *precision* dan *recall*

$$\text{accuracy} = \frac{152 + 171}{340} = 0,9500$$

$$\text{precision}(\text{mask}) = \frac{152}{152 + 6} = 0,9620$$

$$\text{precision}(\text{nonmask}) = \frac{171}{171 + 11} = 0,9396$$

$$\text{precision} = \frac{0,9500 + 0,9396}{2} = 0,9508$$

$$\text{Recall}(\text{mask}) = \frac{152}{152 + 11} = 0,9325$$

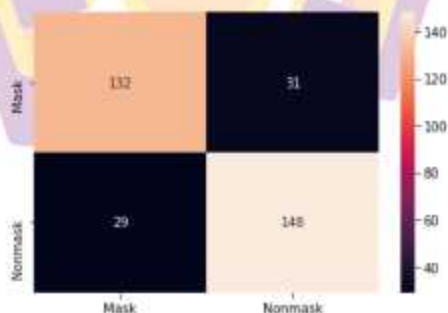
$$\text{Recall}(\text{nonmask}) = \frac{171}{171 + 6} = 0,9661$$

$$\text{Recall} = \frac{0,9325 + 0,9661}{2} = 0,9493$$

Dari perhitungan diatas, pengujian terhadap model yang telah dibangun diketahui menghasilkan nilai *accuracy* 0,9500, nilai *precision* 0,9508 dan nilai *recall* 0,9493.

4.2.3.5. Skenario Lima

Skenario kelima pada pengujian ini adalah melakukan klasifikasi citra menggunakan SVM dengan memanfaatkan CNN yang menggunakan arsitektur ResNet50 untuk menghasilkan *feature map*. Proses *extraction* pada skenario ini dilakukan dalam waktu 44.1188886165618 detik, pelatihan dilakukan dalam waktu 157.420652389526 detik dan pengujian dilakukan dalam waktu 36.4525887966156 detik. Hasil pengujian akan ditampilkan dan dievaluasi menggunakan *confusion matrix*.



Gambar 4.25. Garafik *confusion matrix* pengujian skenario 5

Gambar 4.25 memperlihatkan hasil pengujian dimana dari 340 data yang terbagi dalam kelas *mask* 163 data dan *nonmask* 177 data diketahui 132 data kelas *mask* di prediksi benar, 31 data di prediksi salah dan 148 data kelas *nonmask* di

prediksi benar, 29 data di prediksi salah. Maka dari hasil dari proses ini dapat dihitung nilai *accuracy*, *precision* dan *recall*

$$\text{accuracy} = \frac{132 + 168}{340} = 0,8235$$

$$\text{precision}(\text{mask}) = \frac{132}{132 + 29} = 0,8199$$

$$\text{precision}(\text{nonmask}) = \frac{168}{168 + 31} = 0,8268$$

$$\text{precision} = \frac{0,8199 + 0,8268}{2} = 0,8233$$

$$\text{Recall}(\text{mask}) = \frac{132}{132 + 31} = 0,8098$$

$$\text{Recall}(\text{nonmask}) = \frac{168}{168 + 29} = 0,8362$$

$$\text{Recall} = \frac{0,8098 + 0,8362}{2} = 0,8230$$

Dari perhitungan diatas, pengujian terhadap model yang telah dibangun diketahui menghasilkan nilai *accuracy* 0,8235, nilai *precision* 0,8233 dan nilai *recall* 0,8230.

4.2.3.6. Skenario Enam

Skenario keenam pada pengujian ini adalah melakukan klasifikasi citra menggunakan SVM dengan memanfaatkan CNN yang menggunakan arsitektur MobileNet untuk menghasilkan *feature map*. Proses *extraction* pada skenario ini dilakukan dalam waktu 34.2554574012756 detik, pelatihan dilakukan dalam waktu 94.4188511371612 detik dan pengujian dilakukan dalam waktu

15.7483372688293 detik. Hasil pengujian akan ditampilkan dan dievaluasi menggunakan *confusion matrix*.



Gambar 4.26. Garafik *confusion matrix* pengujian skenario 6

Gambar 4.26 memperlihatkan hasil pengujian dimana dari 340 data yang terbagi dalam kelas *mask* 165 data dan *nonmask* 175 data diketahui 161 data kelas *mask* di prediksi benar, 4 data di prediksi salah dan 170 data kelas *nonmask* di prediksi benar, 5 data di prediksi salah. Maka dari hasil dari proses ini dapat dihitung nilai *accuracy*, *precision* dan *recall*

$$\text{accuracy} = \frac{161 + 170}{340} = 0,9735$$

$$\text{precision}(\text{mask}) = \frac{161}{161 + 5} = 0,9699$$

$$\text{precision}(\text{nonmask}) = \frac{170}{170 + 4} = 0,9770$$

$$\text{precision} = \frac{0,9735 + 0,9770}{2} = 0,9734$$

$$\text{Recall}(\text{mask}) = \frac{161}{161 + 4} = 0,9758$$

$$Recall(nonmask) = \frac{170}{170 + 5} = 0,9714$$

$$Recall = \frac{0,9758 + 0,9714}{2} = 0,9736$$

Dari perhitungan diatas, pengujian terhadap model yang telah dibangun diketahui menghasilkan nilai *accuracy* 0,9735, nilai *precision* 0,9734 dan nilai *recall* 0,9736.

4.3. Analisis Hasil Penelitian

Setelah dilakukan pengolahan data dan proses pembelajaran model terhadap masing-masing skenario percobaan dengan menggunakan data latih, diperoleh beberapa model klasifikasi sebanyak skenario yang digunakan. Model tersebut kemudian dilakukan pengujian menggunakan data uji untuk mengetahui tingkat akurasi dan kinerja dari masing-masing model. Berikut adalah hasil pelatihan dan pengujian terhadap model klasifikasi dapat dilihat pada Tabel 4.2 berikut ini

Tabel 4.2. Hasil Percobaan

Skenario	Algoritma	Arsitektur	Training		Validation		Tasting		
			Acc	Loss	Acc	Loss	Acc	Prec	Rec
S1	CNN	VGG-16	1	3.64E-05	0.9941	0.0843	0.9853	0.9854	0.9853
S2	CNN	ResNet50	1	2.19E-04	0.9706	0.1666	0.9941	0.9944	0.9940

Tabel 4.2. Hasil Percobaan (Lanjutan)

Skenario	Algoritma	Arsitektur	Training		Validation		Tasting		
			Acc	Loss	Acc	Loss	Acc	Prec	Rec
S3	CNN	MobileNet	0.999	0.0058	0.9882	0.1745	0.9795	0.9795	0.9802
S4	CNN + SVM	VGG-16	-	-	-	-	0.95	0.9508	0.9493
S5	CNN + SVM	ResNet50	-	-	-	-	0.8235	0.8233	0.8230
S6	CNN + SVM	MobileNet	-	-	-	-	0.9735	0.9734	0.9736

4.3.1. Perbandingan Hasil *Training* dan *Validation*

Dalam percobaan ini dilakukan perbandingan nilai *accuracy* dan *loss* pada saat *training* dan *validation* pada data. Perbandingan ini hanya dilakukan untuk S1, S2 dan S3 karena skenario lain yang menggunakan SVM, *accuracy* dan *loss* dari proses *training* tidak dapat diukur. berikut grafik untuk masing-masing skenario tersebut.

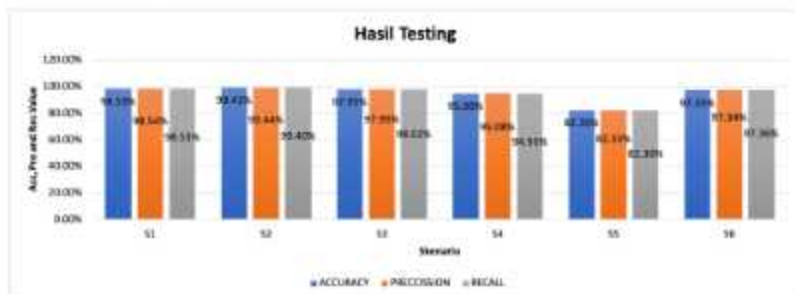


Gambar 4.27. Hasil *training and validation*

Pada Gambar 4.27 diperlihatkan perbandingan persentase *accuracy* dan *loss* antara proses *training* dan *validation* pada ketiga skenario. Dari grafik tersebut terlihat adanya persamaan nilai *training accuracy* pada S1 dan S2 sedangkan S3 mendapatkan nilai yang lebih rendah. Perbandingan antara nilai *trainig accuracy* dan *validation accuracy* terlihat tidak jauh berbeda.

4.3.2. Perbandingan Hasil *Testing*

Dalam percobaan ini dilakukan perbandingan hasil *training* berdasarkan nilai *accuracy*, *preccission* dan *recall* dari keenam skenario. Berikut grafik untuk masing-masing skenario tersebut.



Gambar 4.28. Hasil *testing*

Pada Gambar 4.28 memperlihatkan nilai persentasi dari *accuracy*, *preccission* dan *recall* antara keenam skenario. Dari grafik tersebut terlihat S2 mendapatkan nilai *accuracy*, *precision* dan *recall* tertinggi sedangkan S5 mendapatkan nilai *accuracy*, *precision* dan *recall* terendah. Jika dibandingkan dengan S5, hasil *testing* S1, S3 dan S6 mendapatkan nilai yang jauh lebih tinggi bahkan hampir mendekati nilai dari S2 sebagai skenario dengan nilai persentasi tertinggi.

4.3.3. Perbandingan Waktu Proses

Untuk mengetahui apakah model dapat digunakan dengan baik pada kamera pemantau dengan *frame rate* 30 fps, dilakukan perhitungan waktu komputasi untuk *preprocessing* dan proses *testing* pada data uji. Karena data uji diproses secara bersamaan, peneliti hanya mendapatkan waktu komputasi untuk keseluruhan data uji. Detail waktu komputasi dapat dilihat pada tabel 4.3

Tabel 4.3 Waktu komputasi data uji

No	Skenario	<i>Preprocessing</i>	<i>Testing</i>	Jumlah Data Uji	Rata-rata (s)
		Second (s)			
1	S1	3.262	1.544	340	0.014
2	S2	3.235	1.796	340	0.015
3	S3	3.234	0.773	340	0.012
4	S4	3.269	6.946	341	0.030
5	S5	3.249	36.453	341	0.116
6	S6	3.249	15.748	341	0.056

Nilai rata-rata dari akumulasi *preprocessing* dan *tasting* digunakan untuk mendapatkan waktu komputasi per gambar sehingga dapat diketahui apakah model

yang dibangun dapat melakukan klasifikasi dengan waktu komputasi $\leq 0,03$ detik. Berikut grafik perbandingan waktu komputasi untuk masing-masing skenario tersebut.



Gambar 4.29. Waktu Komputasi

Pada Gambar 4.29 memperlihatkan waktu proses dari keenam skenario. Dari grafik tersebut terlihat S3 merupakan skenario dengan waktu tercepat sedangkan S4 merupakan skenario dengan waktu telat. Waktu komputasi S5 dan S4 $> 0,03$ detik sehingga dapat diketahui bahwa CNN dengan tiga arsitektur yang digunakan dan CNN + SVM dengan arsitektur VGG-16 dapat digunakan secara maksimal pada sistem pemantauan dengan *frame rate* 30 fps.

4.3.4. Kelemahan Sistem

Sistem yang melalui proses *training* akan diuji kembali melalui proses *testing* dengan data baru yang tidak terdapat pada data yang digunakan pada proses *training*

untuk mengetahui kinerja dari sistem yang telah dibangun. Hasil *testing* terhadap enam skenario dengan jumlah data uji sebanyak 340 untuk CNN dan 341 untuk CNN + SVM menunjukkan sistem yang dibangun tidak dapat mengklasifikasi keseluruhan data dengan benar. Gambar 4.30 dibawah memperlihatkan sebagian gambar yang tidak dapat diklasifikasi oleh sistem dengan benar.

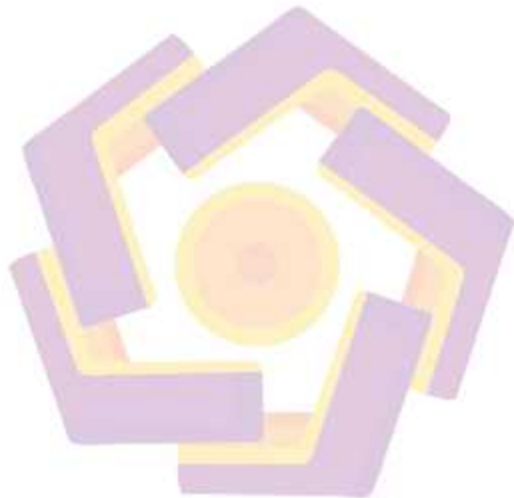


Gambar 4.30 Gandra dengan hasil klasifikasi yang salah

Gambar 4.30 menampilkan contoh gambar yang tidak dapat diklasifikasi dengan benar sehingga dapat disimpulkan bahwa sistem tidak dapat bekerja dengan baik pada kondisi :

- a. Gambar wajah yang menggunakan masker dengan motif tidak dapat diklasifikasi dengan benar meskipun telah dilakukan penambahan 50 data gambar dengan motif wajah
- b. Gambar wajah dengan posisi menyamping tidak dapat diklasifikasi dengan benar baik yang menggunakan masker maupun tidak menggunakan masker

- c. Gambar wajah dengan janggut dan kumis lebat terkadang diklasifikasi sebagai *mask*
- d. Sistem tidak mengklasifikasi dengan benar gambar dengan kualitas rendah baik kelas *mask* maupun *nonmask*



BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan analisa hasil pada percobaan yang telah dilakukan terhadap ketiga arsitektur dengan enam skenario percobaan dapat disimpulkan bahwa :

- a. Dari hasil percobaan 3 arsitektur, algoritma *Convolutional Neural Network* (CNN) dengan arsitektur ResNet50 mendapatkan nilai kinerja tertinggi dengan *accuracy* 99,41%, *precision* 99,44% dan *recall* 99,40% jika dibandingkan dengan arsitektur VGG-16 dengan nilai kinerja *accuracy* 98,53%, *precision* 98,54% dan *recall* 98,53% dan arsitektur MobileNet dengan nilai kinerja *accuracy* 97,95%, *precision* 97,95% dan *recall* 98,02%.
- b. Dari hasil percobaan 3 arsitektur dari algoritma *Convolutional Neural Network* (CNN) yang dikombinasi dengan *Support Vector Machine* (SVM), arsitektur MobileNet mendapatkan nilai kinerja tertinggi dengan *accuracy* 97,35%, *precision* 97,34% dan *recall* 97,36% jika dibandingkan dengan arsitektur VGG-16 dengan nilai kinerja *accuracy* 95,00%, *precision* 95,08% dan *recall* 94,93% dan arsitektur ResNet50 dengan nilai kinerja *accuracy* 82,35%, *precision* 82,33% dan *recall* 82,30%.
- c. Algoritma *Convolutional Neural Network* (CNN) dengan tiga arsitektur dalam skenario pengujian dapat melakukan klasifikasi per gambar dalam waktu kurang dari 0,03 detik dimana waktu komputasi arsitektur VGG-16 0,014 detik, ResNet50 0,116 detik dan MobileNet 0,012 detik. Algoritma kombinasi

Convolutional Neural Network (CNN) dengan *Support Vector Machine (SVM)* untuk arsitektu VGG-16 dapat melakukan klasifikasi dalam waktu sama dengan 0,03 detik sedangkan dua arsitektur lainnya lebih dari 0,03 detik dimana waktu komputasi arsitektur ResNet50 0,015 detik dan MobileNet 0,056 detik.

- d. Algoritma *Convolutional Neural Network (CNN)* dalam melakukan klasifikasi dengan kelas *mask* dan *nonmask* memperoleh nilai kinerja yang lebih baik dari *Support Vector Machine (SVM)* karena arsitektur yang digunakan pada penelitian ini telah dilatih dengan 1000 kelas termasuk di dalamnya kelas *mask* ditambah dengan kemampuan CNN yang dirancang khusus untuk melakukan klasifikasi citra.

5.2. Saran

Berikut ini adalah beberapa saran yang dapat dijadikan pedoman untuk melakukan pengembangan penelitian ini, diantaranya adalah :

- a. Sebaiknya dilakukan penambahan dataset agar hasil perbandingan lebih optimal, karena dengan penggunaan dataset yang banyak dapat dilakukan proses validasi selain proses pengujian untuk mendapatkan hasil yang lebih baik.
- b. Sebaiknya melakukan eksplorasi proses fine tuning pada setiap arsitektur untuk mendapatkan hasil yang lebih maksimal.
- c. Sebaiknya menambah variasi skenario percobaan seperti penggunaan variasi ukuran citra dan rasio pembagian data latih dan data uji.
- d. Sebaiknya proses *training* dilakukan menggunakan library dari python yang dijalankan secara lokal pada komputer. Karena jika menggunakan google colab

waktu komputasi kurang tepat, karena hal tersebut sangat berpengaruh terhadap kecepatan internet pada setiap percobaan yang dilakukan.



DAFTAR PUSTAKA

PUSTAKA BUKU

Andono, Pulung Nurtantio, T. Sutojo, and Muljono. 2017. *Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.

Asmara, Rosa Andrie. 2018. *Pengolahan Citra Digital*. Malang: Polinema Press.

Fadlisyah. 2007. *Computer Vision Dan Pengolahan Citra*. Yogyakarta: Penerbit Andi.

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Astiningsih, Hayu, Bina Kurniawan, and Suroto. 2018. "Hubungan Penerapan Program K3 Terhadap Kepatuhan Penggunaan Apd Pada Pekerja Konstruksi Di Pembangunan Gedung Parkir Bandara Ahmad Yani Semarang." *Jurnal Kesehatan Masyarakat (e-Journal)* 6(4): 300–308.

Chaiyasarn, Krisada et al. 2018. "Crack Detection in Masonry Structures Using Convolutional Neural Networks and Support Vector Machines." In *International Symposium on Automation and Robotics in Construction (ISARC 2018)*, Berlin: ISARC Proceedings, 118–25.

Chen, Hongfeng et al. 2018. "A Hybrid Cnn-Svm Classifier For Hand Gesture Recognition With Surface Emg Signals." In *2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, Chengdu: IEEE, 619–24.

Faisal, Said, Theresia F M Butarbutar, and Pahala Sirait. 2019. "Implementasi CNN Dan SVM Untuk Identifikasi Penyakit Tomat via Daun." 20(2): 117–34.

Garcia, Jayme, and Arnal Barbedo. 2018. "Impact of Dataset Size and Variety on the Effectiveness of Deep Learning and Transfer Learning for Plant Disease Classification." *Computers and Electronics in Agriculture* 153(July): 46–53. <https://doi.org/10.1016/j.compag.2018.08.013>.

Islam, Kh Tohidul, Ram Gopal Raj, and Abdullah Al-Murad. 2017. "Performance of SVM , CNN , and ANN with BoW , HOG , and Image Pixels in Face Recognition." *2017 2nd International Conference on Electrical & Electronic Engineering (ICEEE)* (December): 1–4.

- Kusumanto, R D, and Alan Novi Tompunu. 2011. "Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB." *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2011* 2011(Semantik).
- Liu, Yu Han. 2018. "Feature Extraction and Image Recognition with Convolutional Neural Networks Feature Extraction and Image Convolutional Neural Networks Recognition With." *First International Conference on Advanced Algorithms and Control Engineering*.
- Lotfi, Mehdi, Seyed Ahmad, and Motamedi Saeed. 2016. "Time-Based Feedback-Control Framework for Real-Time Video Surveillance Systems with Utilization Control." *Journal of Real-Time Image Processing*.
- Moolayil, Jojo. 2019. *Learn Keras for Deep Neural Networks*. Canada: Apress.
- Naufal, Muhammad Farid. 2021. "COMPARATIVE ANALYSIS OF IMAGE CLASSIFICATION ALGORITHM FOR WEATHER DATASET." *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)* 8(2): 311–17.
- Sari, Devi Pramita, and Nabila Sholihah Atiqoh. 2020. "Hubungan Antara Pengetahuan Masyarakat Dengan Kepatuhan Penggunaan Masker Sebagai Upaya Pencegahan Penyakit Covid-19 Di Ngronggah." *Jurnal Ilmiah Rekam Medis dan Informatika Kesehatan* 10(1).
- Shrestha, Kishor, Pramen P Shrestha, Dinesh Bajracharya, and Evangelos A Yfantis. 2015. "Hard-Hat Detection for Construction Safety Visualization." *Journal of Construction Engineering* 2015.
- Singh, Himanshu. 2019. *Practical Machine Learning and Image Processing For Facial Recognition, Object Detection, and Pattern Recognition Using Python*. Uttar Pradesh: Apress.
- Thirumala, Karthik, Sushmita Pal, Trapti Jain, and Amod C Umarikar. 2019. "A Classification Method for Multiple Power Quality Disturbances Using EWT Based Adaptive Filtering and Multiclass SVM." *Neurocomputing*: 1–10. <https://doi.org/10.1016/j.neucom.2019.01.038>.
- Wang, Yaqi, Lingling Sun, Kaiqiang Ma, and Jiannan Fang. 2018. "Breast Cancer Microscope Image Classification Based on CNN with Image Deformation." In *15th International Conference Image Analysis and Recognition ICIAR 2018*, Póvoa de Varzim: Springer International Publishing, 845–52.

http://dx.doi.org/10.1007/978-3-319-93000-8_96.

Wijayanto, Hanang. 2015. "Klasifikasi Batik Menggunakan Metode K-Nearest Neighbour Berdasarkan Gray Level Co-Occurrence Matrices (GLCM)." *Jurusan Teknik Informatika FIK UDINUS* (5).

Zufar, Muhammad, and Budi Setiyono. 2016. "Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real - Time." *Jurnal Sains dan Seni ITS* 5(2): 72-77.

PUSTAKA ELEKTRONIK

Putra, Jan Wira Gotama. 2020. *Pengenalan Konsep Pembelajaran Mesin Dan Deep Learning*. 1.4. Tokyo. <https://wiragotama.github.io/resources/ebook/intro-to-ml-secured.pdf>.

