

TESIS

**KLASIFIKASI CITRA WAYANG PUNAKAWAN MENGGUNAKAN
METODE GAUSSIAN FILTER DAN ALGORITMA
CONVOLUTIONAL NEURAL NETWORK**



Disusun oleh:

Nama : Muhammad Resa Arif Yudianto
NIM : 19.51. 1182
Konsentrasi : Business Intelligence

**PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2021**

TESIS

**KLASIFIKASI CITRA WAYANG PUNAKAWAN MENGGUNAKAN
METODE GAUSSIAN FILTER DAN ALGORITMA
CONVOLUTIONAL NEURAL NETWORK**

**PUNAKAWAN PUPPET IMAGE CLASSIFICATION USING THE
GAUSSIAN FILTER METHOD AND CONVOLUTIONAL
NEURAL NETWORK ALGORITHM**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

Nama : Muhammad Resa Arif Yudianto
NIM : 19.51.1182
Konsentrasi : Business Intelligence

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA

2021

HALAMAN PENGESAHAN

KLASIFIKASI CITRA WAYANG PUNAKAWAN MENGGUNAKAN METODE GAUSSIAN FILTER DAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK

PUNAKAWAN PUPPET IMAGE CLASSIFICATION USING THE GAUSSIAN FILTER METHOD AND CONVOLUTIONAL NEURAL NETWORK ALGORITHM

Dipersiapkan dan Disusun oleh

Muhammad Resa Arif Yudianto

19.51.1182

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, 02 Februari 2021

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

Yogyakarta, 02 Februari 2021

Rektor

Prof. Dr. M. Suvanto, M.M.

NIK. 190302001

HALAMAN PERSETUJUAN

KLASIFIKASI CITRA WAYANG PUNAKAWAN MENGGUNAKAN METODE GAUSSIAN FILTER DAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK

PUNAKAWAN PUPPET IMAGE CLASSIFICATION USING THE GAUSSIAN FILTER METHOD AND CONVOLUTIONAL NEURAL NETWORK ALGORITHM

Dipersiapkan dan Disusun oleh

Muhammad Resa Arif Yudianto

19.51.1182

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Selasa, 02 Februari 2021

Pembimbing Utama

Dr. Kusriani, M.Kom

NIK. 190302106

Anggota Tim Penguji

Dr. Wing Wahyu Winarno, MAFIS,

Ak.
NIK. 555195

Pembimbing Pendamping

Alva Hendi Muhammad, S.T.,

M.Eng., Ph.D.
NIK. 190302493

Hanif Al Fatta, M.Kom

NIK. 190302096

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

Yogyakarta, 02 Februari 2021
Direktur Program Pascasarjana

Dr. Kusriani, M.Kom.

NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : **Muhammad Resa Arif Yudianto**
NIM : **19.51.1182**
Konsentrasi : **Business Intelligence**

Menyatakan bahwa Tesis dengan judul berikut:

**KLASIFIKASI CITRA WAYANG PUNAKAWAN MENGGUNAKAN
METODE GAUSSIAN FILTER DAN ALGORITMA CONVOLUTIONAL
NEURAL NETWORK**

Dosen Pembimbing Utama : **Dr. Kusriani, M.Kom**
Dosen Pembimbing Pendamping : **Hanif Al Fatta, M.Kom**

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, 02 Februari 2021
Yang Menyatakan,



Muhammad Resa Arif Yudianto

HALAMAN PERSEMBAHAN

Puji syukur atas kehadiran Allah SWT yang telah memberikan rahmat dan ridho-Nya, sehingga penulis dapat menyelesaikan karya tulis tesis ini. Pada kesempatan kali ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Kedua orang tua saya, Bapak Sugiyana dan Ibu Sri Murningsih yang tiada hentinya memberikan nasehat, tidak pernah lelah merawat, mendidik penulis dari kecil hingga sekarang, mengarahkan, memberikan dukungan dan perhatian, memberikan semangat dan pastinya doa yang selalu ada dalam setiap sujud, sehingga penulis dapat mencapai pada titik ini.
2. Bapak M. Suyanto, Prof, Dr., MM. selaku Rektor Universitas Amikom Yogyakarta.
3. Ibu Dr. Kusri, M.Kom selaku Direktur Program Pascasarjana Universitas Amikom Yogyakarta
4. Ibu Dr. Kusri, M.Kom, selaku dosen pembimbing 1, Bapak Hanif Al Fatta, M.Kom selaku dosen pembimbing 2, Bapak Mei P. Kurniawan, M.Kom., Bapak Dr. Suwanto Raharjo, S.Si., M.Kom., Bapak Emha Taufiq Luthfi, S.T., M.Kom., Bapak Dr. Wing Wahyu Winarno, MAFIS, Ak. dan Bapak Alva Hendi Muhammad, S.T., M.Eng., Ph.D. selaku dosen penguji yang telah memberikan masukan, arahan, dan motivasinya kepada penulis.
5. Bapak Arif Dwi Laksito, M.Kom, selaku dosen pembimbing ketika S1 dulu dan Ibu Acihmah Sidauruk, M.Kom yang selalu memberikan motivasi, semangat dan arahan sehingga penulis dapat menyelesaikan karya tulis tesis ini.
6. Teman-teman MTI angkatan 22 kelas A atas kebersamaan dan pengalamannya.
7. Teman-teman KARSA dan teman-teman dekat saya yang tidak bisa saya sebutkan satu persatu baik yang ada di Jogja maupun di luar Jogja yang telah memberikan motivasi dan semangat, sehingga penulis dapat menyelesaikan karya tulis tesis ini.

HALAMAN MOTTO

“Musuh yang paling berbahaya di atas dunia ini adalah penakut dan bimbang.
Teman yang paling setia, hanyalah keberanian dan keyakinan yang teguh.”

(Andrew Jackson)

“Banyak kegagalan dalam hidup ini dikarenakan orang-orang tidak menyadari
betapa dekatnya mereka dengan keberhasilan saat mereka menyerah.”

(Thomas Alva Edison)

“Waktu bagaikan pedang. Jika engkau tidak memanfaatkannya dengan baik
(untuk memotong), maka ia akan memanfaatkanmu (dipotong).

(HR. Muslim)

“What You Think is What You Get”

(Penulis)

“Orang Sukses Bukanlah Seberapa Banyak Orang Mengenalnya, akan tetapi
Seberapa Banyak Orang yang Bahagia Karena Mengenalnya”

(Penulis)

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT karena atas limpahan rahmat, hidayah serta inayah-NYA, penulis masih diberi kesempatan dan kemudahan untuk menyelesaikan tesis ini.

Tesis ini disusun dalam rangka memenuhi salah satu syarat kelulusan perguruan tinggi Program Studi Strata-2 Magister Teknik Informatika di Universitas AMIKOM Yogyakarta dan meraih gelar Magister Komputer (M.Kom). Selain itu tesis ini juga bertujuan untuk menambah pengetahuan tentang *computer vision* pengenalan tokoh wayang punakawan yang dibuat menggunakan algoritma *Convolutional Neural Network* dan metode *Gaussian Filter*.

Penulis juga mengucapkan terimakasih yang sebesar-besarnya kepada :

1. Bapak Prof. Dr. M. Suyanto, MM, selaku ketua Amikom Yogyakarta.
2. Ibu Dr. Kusriani, M.Kom dan Bapak Hanif Al Fatta, M.Kom selaku dosen pembimbing yang telah dengan sabar dalam memberikan masukan, saran, bantuan, dan bimbingan dalam menyelesaikan naskah skripsi ini.
3. Ibu Dr. Kusriani, M.Kom selaku Direktur Program Pascasarjana Universitas Amikom Yogyakarta.
4. Kedua Orang tua yang tak pernah lelah dalam memberikan dukungan 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.

Semoga tesis ini dapat menambah pengetahuan dan memberikan manfaat bagi para pembacanya maupun diri penulis sendiri serta dapat digunakan sebagai salah referensi untuk penelitian yang lain.

Yogyakarta, 14 Februari 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	7
1.3 Batasan Masalah	8
1.4 Tujuan Penelitian	9
1.5 Manfaat Penelitian	9
BAB II TINJAUAN PUSTAKA	11
2.1 Tinjauan Pustaka	11
2.2 Keaslian Penelitian	15

2.3	Landasan Teori.....	19
2.3.1	Klasifikasi.....	19
2.3.2	Pengolahan Citra Digital.....	20
2.3.3	Metode <i>Gaussian Filter</i>	21
2.3.4	<i>Convolutional Neural Network</i>	22
2.3.5	Konsep CNN.....	23
2.3.6	Arsitektur <i>Convolutional Neural Network</i>	24
2.3.7	<i>Confusion Matrix</i>	26
BAB III METODE PENELITIAN.....		28
3.1	Jenis, Sifat dan Pendekatan Penelitian.....	28
3.1.1	Metode Penelitian.....	28
3.2	Metode Pengumpulan Data.....	28
3.3	Metode Analisis Data.....	30
3.4	Alur Penelitian.....	31
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....		36
4.1	Pengumpulan Data.....	36
4.2	Analisa Data.....	42
4.2.1	Arsitektur CNN.....	42
4.2.2	<i>Fine Tuning</i>	46
4.2.3	Skenario Percobaan.....	50
4.2.4	<i>Preprocessing Data</i>	51

4.2.5	Proses Pembelajaran Data	57
4.3	Analisis Hasil Penelitian.....	63
4.3.1	Perbandingan Waktu Komputasi	66
4.3.2	Perbandingan <i>Training</i> dan <i>Testing Loss</i>	68
4.3.3	Perbandingan <i>Training</i> dan <i>Testing Accuracy</i>	70
4.3.4	Perbandingan Nilai Akurasi Ketiga Arsitektur.....	72
4.3.5	Pengujian Tambahan.....	75
BAB V	PENUTUP.....	78
5.1	Kesimpulan.....	78
5.2	Saran.....	78
DAFTAR	PUSTAKA.....	80

DAFTAR TABEL

Tabel 2.1. Matriks literatur review dan posisi penelitian.....	15
Tabel 2.2. Tabel <i>Confusion Matrix</i>	26
Tabel 4.1. Tabel contoh citra wayang punakawan	40
Tabel 4.2. Skenario percobaan	50
Tabel 4.3. Tabel <i>Confusion Matrix</i>	61
Tabel 4.4. Hasil percobaan.....	64
Tabel 4.5. Tabel pengujian tambahan.....	76



DAFTAR GAMBAR

Gambar 2.1. Arsitektur sederhana MLP	23
Gambar 2.2. Arsitektur <i>Convolutional Neural Network</i>	24
Gambar 2.3. <i>Convolutional Layer</i>	24
Gambar 2.4. Contoh perhitungan <i>Convolutional Layer</i>	25
Gambar 2.5. Proses pada <i>Pooling Layer</i>	25
Gambar 3.1. Alur pengumpulan data	29
Gambar 3.2. Alur penelitian	31
Gambar 4.1. Pelabelan dataset	40
Gambar 4.2. Arsitektur VGG-16	43
Gambar 4.3. Arsitektur ResNet50	44
Gambar 4.4. Arsitektur MobileNet	45
Gambar 4.5. Model VGG-16 sebelum <i>fine tuning</i>	46
Gambar 4.6. Model VGG-16 setelah <i>fine tuning</i>	47
Gambar 4.7. Model ResNet50 sebelum <i>fine tuning</i>	48
Gambar 4.8. Model ResNet50 setelah <i>fine tuning</i>	48
Gambar 4.9. Model MobileNet sebelum <i>fine tuning</i>	49
Gambar 4.10. Model MobileNet setelah <i>fine tuning</i>	50
Gambar 4.11. Hasil <i>preprocessing</i> skenario 1	53
Gambar 4.12. Hasil <i>preprocessing</i> skenario 2	54
Gambar 4.13. Hasil <i>preprocessing</i> skenario 3	55
Gambar 4.14. Hasil <i>preprocessing</i> skenario 4	56

Gambar 4.15. Proses konvolusi	58
Gambar 4.16. Pergerakan proses konvolusi	58
Gambar 4.17. Proses <i>Pooling Layer</i>	60
Gambar 4.18. Perbandingan waktu komputasi <i>epoch</i> 50.....	67
Gambar 4.19. Perbandingan waktu komputasi <i>epoch</i> 100.....	67
Gambar 4.20. <i>Training vs testing loss</i> ResNet50	68
Gambar 4.21. <i>Training vs testing loss</i> VGG-16.....	69
Gambar 4.22. <i>Training vs testing loss</i> MobileNet.....	69
Gambar 4.23. <i>Training vs testing accuracy</i> ResNet50.....	70
Gambar 4.24. <i>Training vs testing accuracy</i> VGG-16.....	71
Gambar 4.25. <i>Training vs testing accuracy</i> MobileNet.....	72
Gambar 4.26. Nilai akurasi <i>epoch</i> 50.....	73
Gambar 4.27. Nilai akurasi <i>epoch</i> 100.....	74

INTISARI

Memasuki industri 4.0 membuat berbagai hal dapat dikerjakan dengan lebih mudah, cepat dan otomatis. Salah satu bidang yang saat ini telah berkembang pesat adalah pengenalan objek secara otomatis oleh sistem atau sering disebut *computer vision*. Banyak algoritma *machine learning* maupun *deep learning* yang dapat digunakan dalam *computer vision*. Beberapa peneliti sebelumnya telah melakukan penelitian terhadap kinerja dari berbagai algoritma klasifikasi seperti SVM (*Support Vector Machine*) dan CNN (*Convolutional Neural Network*). Dari beberapa penelitian yang telah dilakukan, CNN memiliki kinerja yang lebih unggul dibandingkan SVM. Sehingga banyak penelitian tentang klasifikasi citra yang menggunakan algoritma ini. Pada penelitian-penelitian terdahulu, algoritma ini banyak diterapkan pada bidang kesehatan, pertanian dan ekonomi, akan tetapi penelitian terhadap bidang kebudayaan masih jarang dilakukan. Sedangkan saat ini, tingkat perkembangan budaya sudah semakin berkurang akibat pengaruh globalisasi, salah satu contohnya adalah budaya pewayangan yang saat ini hampir tidak ada.

Untuk mengantisipasi kepunahan budaya wayang tersebut, salah satu solusi yang dapat diterapkan adalah dengan menerapkan algoritma CNN dalam mengenali objek pewayangan secara otomatis. Terdapat banyak arsitektur CNN yang telah dikembangkan dan diikutsertakan ke dalam sebuah kompetisi untuk membandingkan kinerja dari setiap arsitektur. Beberapa contoh arsitektur yang sempat mendominasi kinerja/kemampuan mengenali objek yaitu ResNet50, VGG-16 dan MobileNet. Kinerja sebuah arsitektur tidak selalu sama, tergantung dari jenis data yang digunakan. Terdapat faktor lain yang dapat mempengaruhi kinerja arsitektur, salah satunya adalah proses *preprocessing*. Pada penelitian ini penulis membandingkan ketiga arsitektur tersebut dengan beberapa skenario percobaan dan penggunaan *gaussian filter* sebagai penghilang *noise* dan memperhalus citra. Dari hasil penelitian, didapatkan bahwa arsitektur dengan kinerja terbaik pada penelitian ini adalah VGG-16 yang dilakukan pada *epoch* 50 mendapatkan nilai akurasi sebesar 96%. Filter *gaussian* berpengaruh dalam meningkatkan akurasi ketika diterapkan pada citra RGB, akan tetapi akan mengalami penurunan akurasi ketika diterapkan pada citra *green channel*. Penggunaan CLAHE pada *preprocessing* data dapat meningkatkan akurasi klasifikasi pada model yang dihasilkan.

Kata Kunci: Klasifikasi, CNN, Wayang Kulit, Perbandingan Arsitektur

ABSTRACT

Entering industry 4.0 makes things easier, faster and more automatic. One area that is currently developing rapidly is automatic object recognition by systems or often called computer vision. Many machine learning and deep learning algorithms can be used in computer vision. Several researchers have previously conducted research on the performance of various classification algorithms such as SVM (Support Vector Machine) and CNN (Convolutional Neural Network). From several studies that have been conducted, CNN has a superior performance compared to SVM. So that many studies on image classification use this algorithm. In previous studies this algorithm has been widely applied in the fields of health, agriculture and economics, but research in the field of culture is still rarely done. Meanwhile, at this time the level of cultural development has decreased due to the influence of globalization, one example is the puppet culture which is currently almost non-existent.

To anticipate the extinction of the puppet culture, one solution that can be applied is to apply the CNN algorithm to recognize puppet objects automatically. There are many CNN architectures that have been developed and entered into a competition to compare the performance of each architecture. Some examples of architectures that had dominated the performance / ability to recognize objects, namely ResNet50, VGG-16 and MobileNet. The performance of an architecture is not always the same, depending on the type of data used. There are other factors that can affect architectural performance, one of which is the preprocessing process. In this study the authors compared the three architectures with several experimental scenarios and the use of a gaussian filter as noise removal and image refinement. From the research results, it was found that the architecture with the best performance in this study was VGG-16 which was carried out on epoch 50 and obtained an accuracy value of 96%. Gaussian filters have an effect in increasing accuracy when applied to RGB images, but will experience a decrease in accuracy when applied to green channel images. The use of CLAHE in data preprocessing can improve the classification accuracy of the resulting model.

Keywords: Classification, CNN, Puppet, Architectural Comparison

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Memasuki industri 4.0 yang serba digital ini membuat berbagai hal dapat dikerjakan dengan lebih mudah, cepat dan otomatis. Salah satu bidang yang saat ini telah berkembang pesat adalah pengenalan objek secara otomatis oleh sistem. Pengenalan objek merupakan sebuah teknik *computer vision* dalam mengidentifikasi objek yang ada di dalam gambar atau video. Penerapan *computer vision* merambah ke berbagai sektor untuk membantu manusia dalam mempermudah keperlunya. Beberapa contoh produk yang saat ini telah ada seperti mobil tesla yang otomatis dapat berjalan tanpa dikemudikan (*autopilot*), alat pendeteksi wajah untuk mengenali seseorang, alat pendeteksi sebuah penyakit dari gambar CT-scan atau citra Xray dan alat pendeteksi penyakit tanaman dari gambar/foto daun yang terserang hama.

Untuk dapat mengenali objek dalam suatu gambar, sebuah sistem perlu dilatih dengan data-data yang terkait dengan objek tersebut. Sistem akan belajar melalui data-data tersebut dengan menemukan sebuah ciri khusus dari suatu objek yang dapat dijadikan sebuah pola oleh sistem dalam mengenali objek yang sama pada kondisi yang berbeda. Proses pembelajaran sistem dengan data tersebut termasuk ke dalam bidang keilmuan *machine learning* atau *deep learning*. Terdapat banyak algoritma *machine learning* atau *deep learning* yang dapat digunakan dalam proses pembelajaran sistem terhadap objek citra. Seperti pada penelitian sebelumnya yang dilakukan Shin (2017), membandingkan metode klasifikasi citra

yaitu metode yang diusulkan berbasis SVM dengan metode CNN berbasis *deep learning*. Dalam penelitian tersebut digunakan objek citra berupa *dataset* penyakit polip yang didapatkan dari tiga sumber publik. Untuk meningkatkan validitas perbandingan terhadap kedua metode tersebut, dilakukan percobaan dengan berbagai skenario dan dataset yang berbeda-beda. Dari hasil percobaan yang telah dilakukan didapatkan akurasi terbaik adalah pada algoritma CNN dengan nilai akurasi sebesar 92.08%.

Berdasarkan acuan dari penelitian sebelumnya, tidak heran bahwa algoritma CNN banyak digunakan dalam proses klasifikasi citra. Beberapa penelitian sebelumnya, Alamsyah (2020) melakukan penelitian dalam mengklasifikasikan ekspresi citra wajah dengan algoritma CNN. Selain itu pada penelitian yang dilakukan Arrofiqoh (2018) juga menerapkan algoritma CNN pada klasifikasi tanaman. Pada bidang lain, seperti bidang kesehatan algoritma ini juga banyak diterapkan dalam mengembangkan sebuah sistem pendeteksian otomatis seperti yang dilakukan oleh Rokhana (2019), yaitu melakukan pembelajaran sistem dengan algoritma CNN untuk mendeteksi patah tulang femur pada citra ultrasonik B-Mode.

Dari beberapa penelitian sebelumnya, penggunaan CNN banyak diterapkan pada klasifikasi objek di bidang ekonomi, pertanian dan kesehatan. Sedangkan untuk bidang yang lain seperti bidang kebudayaan, penerapannya masih jarang dilakukan penelitian dalam mendeteksi objek-objek pada bidang kebudayaan. Sedangkan saat ini, tingkat kelestarian budaya sudah semakin berkurang akibat pengaruh globalisasi. Salah satu bentuk kebudayaan yang dimiliki Indonesia yang

saat ini mulai menurun adalah wayang. Menurut para ahli, wayang dibedakan menjadi 2 jenis yaitu wayang orang yang diperankan langsung oleh beberapa orang dan jenis yang kedua adalah wayang yang berwujud boneka yang digerakkan oleh seseorang yang disebut dalang. Menurut Pasha (2011), Salah satu jenis wayang boneka adalah wayang kulit. Wayang kulit terbuat dari boneka kayu dan kulit sebagai pakaiannya.

D.R.Indah (2019), salah satu contoh tokoh pewayangan yang banyak dikenal adalah punakawan yang terdiri dari Semar dan ketiga anaknya yaitu Gareng, Petruk dan Bagong. Tokoh-tokoh ini memiliki karakter yang ceria, lucu dan khas dengan pesan moral yang selalu dibawakan dalam setiap pertunjukan wayang. Akan tetapi kebudayaan wayang yang dahulunya berkembang dengan jumlah yang banyak, saat ini tinggal berjumlah 25 yang masih ada di masyarakat. Hal tersebut terjadi karena lemah dan berkurangnya ketiga faktor yang mempengaruhi eksistensi dari wayang ini. Ketiga faktor tersebut adalah faktor pelaku yang berkarya, kelompok masyarakat yang mencintai budaya wayang dan dukungan dari pemerintah Suseno (1991). Oleh karena itu kebudayaan ini lama kelamaan akan mengalami kemusnahan dan tidak sedikit dari generasi berikutnya yang tidak akan mengetahui tokoh-tokoh pewayangan.

Untuk mengantisipasi terjadinya kepunahan budaya wayang tersebut, salah satu solusinya adalah dengan pengenalan objek dengan sistem. Penelitian serupa dengan objek wayang yang dilakukan oleh Sandy (2019), mendeteksi dan mengklasifikasikan wayang menurut bentuknya menggunakan algoritma KNN (*K-Nearest Neighbor*) sebagai algoritma klasifikasi dan GLCM (*Gray Level Co-*

occurrence Matrix) yang bertugas dalam mengekstrak fitur. Jumlah data yang digunakan sebanyak 100 citra yang dibagi menjadi 5 kelas dengan rasio perbandingan 60% sebagai data latih dan 40% sebagai data uji. Citra yang digunakan memiliki ukuran 480x460 piksel. Data citra diperoleh melalui proses pengambilan gambar menggunakan kamera. Hasil penelitian menunjukkan tingkat akurasi yang didapatkan sebesar 77,5%.

Pada penelitian sebelumnya algoritma yang digunakan dalam melakukan pengenalan objek wayang adalah algoritma KNN dengan nilai akurasi sebesar 77,5%. Penelitian-penelitian sebelumnya tentang klasifikasi citra, algoritma CNN kebanyakan masih diterapkan pada bidang kesehatan dan pertanian. Oleh karena itu untuk menguji tingkat kinerja dan kemampuan algoritma CNN terhadap berbagai jenis dataset, dapat diterapkan pada pengenalan objek wayang ini. Pada penelitian ini digunakan algoritma CNN, karena pada penelitian sebelumnya oleh Shin (2017), CNN memiliki hasil yang lebih baik dari algoritma *machine learning*. Selain itu, karena banyaknya arsitektur algoritma CNN ini maka dapat dilakukan percobaan terhadap beberapa arsitektur untuk mendapatkan nilai akurasi yang lebih baik.

Penelitian sebelumnya Setiawan (2017), membandingkan beberapa arsitektur CNN yang diujikan pada dataset fundus. Dataset yang digunakan adalah data Fundus retina yang terdiri dari dua kelas yaitu normal dan neovaskularisasi. Jumlah data sebanyak 128 citra yang dibagi menjadi 16 bagian yang sama, sehingga menghasilkan 2048 patches. Percobaan dilakukan dengan dua skenario yaitu proses *learning* tanpa optimasi dan proses *learning* dengan optimasi. Dari hasil percobaan yang telah dilakukan, didapatkan hasil arsitektur terbaik pada skenario tanpa

optimasi adalah arsitektur VGG19 dengan nilai akurasi sebesar 89,3%, sensitivitas 90,4% dan spesifisitas 88,2%. Sedangkan pada skenario penambahan optimasi dengan *gradient descent*, arsitektur terbaik adalah VGG16 dengan nilai akurasi 92,31%, nilai sensitivitas 90,7% dan spesifisitas sebesar 94%.

Penelitian terdahulu oleh Cheng (2017) ingin membandingkan algoritma CNN dengan parameter tambahan ERACNN dibandingkan dengan algoritma RBF-SVM, Linear-SVM dan KNN. Dalam penelitian tersebut objek yang digunakan adalah dataset emosional seseorang dengan sinyal EEG (*electroencephalogram*). Jumlah data latih yang digunakan sebanyak 40 partisipan/orang dengan data uji sebanyak 20 partisipan yang diambil secara acak. Dari hasil percobaan didapatkan bahwa algoritma CNN mengungguli dari ketiga algoritma yang lain dengan tingkat akurasi sebesar 83,45% pada 2 kategori emosional (optimis dan pesimis) dan 68,8% untuk 3 kategori emosional (optimis, kalem dan pesimis). Minimnya dataset sangat mempengaruhi hasil akurasi yang didapatkan.

Putra (2020) menerapkan algoritma CNN dalam mengklasifikasikan penyakit retinopati yang dapat menyebabkan kebutaan terhadap mata. Arsitektur CNN yang digunakan adalah GoogLeNet, ResNet18, ResNet50, dan ResNet101. Terdapat 5 tahapan yang dijalankan pada penelitian ini yaitu *preprocessing data*, peningkatan kualitas citra, ekstraksi fitur, pengurangan fitur dan klasifikasi. Dataset citra sebesar 224 x 224 piksel yang terbagi menjadi 75% sebagai data latih dan 25% sebagai data uji. Untuk meningkatkan kualitas citra, digunakan 3 jenis metode yang masing-masing akan diujikan, yaitu metode CLAHE (*Contrast Limited Adaptive Histogram Equalization*), *Homomorphic filter*, dan *Morphological Contrast*

Enhancement. Dari hasil percobaan yang telah dilakukan didapatkan hasil akurasi tertinggi yaitu pada arsitektur ResNet50 untuk klasifikasi dataset dua kelas dengan nilai 86.76%.

Penelitian yang dilakukan Tammina (2019) yaitu mengaplikasikan transfer learning dengan arsitektur VGG16 pada objek dataset hewan dengan jumlah 2 kelas yaitu anjing dan kucing yang berjumlah 25000 data. Untuk meningkatkan jumlah data dilakukan proses augmentasi dengan 8 parameter augmentasi. Skenario percobaan yang diujikan yaitu membandingkan rancangan CNN yang dibuat sendiri dengan arsitektur VGG16. Pada arsitektur VGG16 dilakukan penyesuaian lapisan *fully connected layer*, disesuaikan dengan jumlah kelas yang digunakan yaitu 2 kelas. Setelah dilakukan percobaan terhadap semua skenario, didapatkan hasil nilai akurasi pada CNN buatan sendiri sebesar 72.40%, lalu dilakukan penyesuaian parameter (*fine tuning*) akurasi meningkat menjadi 79.20%. Sedangkan untuk arsitektur VGG16 memperoleh akurasi sebesar 95.40%.

Penelitian sebelumnya Hendriyana (2020) melakukan penelitian tentang klasifikasi jenis kayu di Indonesia. Dalam penelitiannya, arsitektur yang digunakan adalah mobileNet. Jumlah kelas yang digunakan sebanyak 1000 data yang terbagi ke dalam 10 kelas jenis kayu. Data dikumpulkan secara mandiri oleh peneliti melalui pengambilan dengan kamera. Proses *preprocessing* pada penelitian ini adalah mengubah ukuran citra menjadi 150 x 150 piksel kemudian dilakukan proses augmentasi dengan menggeser sudut citra sebesar 30% untuk menambah variasi dataset. Rasio pembagian dataset pada penelitian ini adalah 90% sebagai data latih dan 10% sebagai data uji. Citra yang diujikan berupa citra RGB yaitu dengan

parameter masukan adalah $150 \times 150 \times 3$. Dari hasil percobaan yang telah dilakukan, didapatkan nilai akurasi sebesar 93.3%.

Khilmawan (2018) melakukan penelitian yaitu membandingkan metode yang digunakan dalam perbaikan citra yaitu antara *median filter* dan *gaussian filter* terhadap citra tulang. Dalam penelitiannya, dipilih 2 metode terbaik untuk menghilangkan *noise*. Berdasarkan hasil percobaan didapatkan hasil bahwa metode *gaussian filter* dan *median filter* memiliki kinerja yang sama baiknya dengan selisih 0,002 saja dan terbukti dapat memperbaiki kualitas dari citra.

Dari beberapa uraian di atas, penulis tertarik ingin melakukan penelitian menggunakan algoritma CNN dan metode *Gaussian Filter* pada citra wayang punakawan sebagai objeknya. Pemilihan algoritma CNN, karena memiliki kinerja yang terbaik dibandingkan dengan algoritma klasifikasi yang lain berdasarkan hasil penelitian-penelitian sebelumnya serta masih jaranganya penelitian CNN terhadap dataset pada bidang kebudayaan. Untuk mendapatkan akurasi yang lebih tinggi dari penelitian sebelumnya, penulis akan melakukan percobaan dengan membandingkan beberapa arsitektur CNN dengan beberapa skenario yang terdiri dari kombinasi perlakuan citra (*preprocessing*) dan penggunaan metode *Gaussian Filter*.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah pada penelitian ini adalah :

1. Bagaimana rancangan algoritma CNN yang terbaik dalam klasifikasi citra wayang punakawan untuk mendapatkan nilai akurasi lebih tinggi dari 77,5% pada penelitian sebelumnya terhadap objek yang sama?
2. Apakah perlakuan citra wayang punakawan dengan metode *Gaussian Filter* dapat meningkatkan nilai akurasi yang dihasilkan?

1.3 Batasan Masalah

Batasan masalah dalam suatu penelitian sangat diperlukan, sehingga penelitian dapat lebih terarah dan tidak menyimpang jauh. Adapun batasan masalah dalam penelitian ini adalah :

1. Algoritma klasifikasi citra yang digunakan dalam penelitian ini adalah CNN.
2. Jenis wayang yang digunakan untuk data penelitian adalah wayang kulit.
3. *Objek* citra wayang yang digunakan dalam penelitian ini adalah wayang punakawan.
4. Data *objek* citra wayang untuk proses *training* dan *testing* berasal dari internet (*google image*).
5. Dilakukan *preprocessing* data dengan melakukan proses augmentasi citra untuk meningkatkan performa model klasifikasi proses *training*.
6. Dilakukan proses segmentasi citra dengan metode *thresholding* untuk menangani citra wayang dengan *background* berwarna.
7. Ekstraksi fitur citra berdasarkan ciri-ciri bentuk dari *objek* citra wayang dengan algoritma CNN.

8. Proses filterisasi citra untuk menghilangkan *noise* menggunakan metode *Gaussian Filter*.
9. Skenario percobaan menggunakan 3 arsitektur CNN yang berbeda-beda dan akan dikombinasikan dengan penggunaan metode *Gaussian Filter*.
10. Proses evaluasi model klasifikasi menggunakan metode *confusion matrix* dan pengukuran kecepatan klasifikasi citra menggunakan *time execution process* pada python.

1.4 Tujuan Penelitian

Tujuan dari penelitian yang penulis lakukan adalah sebagai berikut :

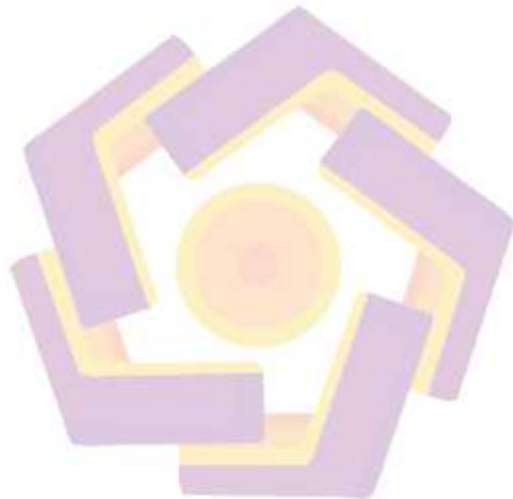
1. Mencari rancangan arsitektur *Convolutional Neural Network* terbaik untuk mendapatkan tingkat akurasi yang tertinggi pada klasifikasi citra wayang punakawan.
2. Mengetahui apakah perlakuan citra sebelum dilakukan proses klasifikasi dengan metode *Gaussian Filter* memiliki pengaruh terhadap tingkat akurasi yang dihasilkan.

1.5 Manfaat Penelitian

Manfaat yang didapatkan dari penelitian ini adalah :

1. Rancangan arsitektur yang diterapkan dapat dijalankan dan dapat mengklasifikasikan citra wayang dengan akurasi yang baik sesuai dengan batasan yang telah ditentukan.

2. Memberikan kontribusi penelitian terhadap arsitektur *Convolutional Neural Network* yang cocok digunakan dan menghasilkan tingkat akurasi yang tinggi.
3. Memberikan kontribusi penelitian terhadap pengaruh filterisasi citra menggunakan metode *Gaussian Filter* terhadap nilai akurasi yang dihasilkan.



BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Dalam tinjauan pustaka ini peneliti memiliki beberapa penelitian sebelumnya yang relevan dengan metode yang diusulkan. Penelitian sebelumnya dijadikan rujukan dan referensi dalam penelitian ini.

Penelitian yang dilakukan oleh Andry Chowanda dan Rhio Sutoyo (2019) dengan judul *Deep Learning for Visual Indonesian Place Classification with Convolutional Neural Networks*. Dalam penelitiannya, bertujuan untuk membandingkan akurasi klasifikasi dari ketiga model arsitektur algoritma CNN yaitu VGG-16, VGG-19 dan GoogleNet pada objek citra tempat-tempat di Indonesia. Jumlah dataset utama yang didapatkan oleh peneliti adalah sebanyak 2040 citra, Kemudian dilakukan proses augmentasi, sehingga dataset menjadi 16828 yang terbagi ke dalam 6 kelas. Ukuran piksel yang digunakan dalam penelitian ini adalah 224x224 piksel. Proses evaluasi dari setiap model arsitektur CNN menggunakan parameter atau variabel tingkat akurasi saja. Arsitektur terbaik didapatkan oleh model VGG-19 dengan nilai akurasi sebesar 92%.

Penelitian lain yang dilakukan oleh Saya Fujino, Taichi Hatanaka, Naoki Mori dan Keinosuke Matsumoto (2019) yang berjudul *Evolutionary Deep Learning Based on Deep Convolutional Neural Network for Anime Storyboard Recognition*. Fokus dalam penelitian ini adalah membandingkan akurasi pembelajaran terhadap rasio pembagian dataset dengan pendekatan *evolusiner deep learning*

menggunakan algoritma CNN. Objek citra untuk diklasifikasi adalah tulisan tangan *story board* anime yang terdiri dari 12 episode dengan jumlah gambar setiap episodanya berkisar antara 860 gambar dan terbagi menjadi 2 kelas. Proses pembuatan model arsitektur menggunakan *tool* atau bahasa pemrograman java. Dalam penelitiannya, dilakukan 2 skenario percobaan yaitu skenario 1 (E1) membagi data training 1-5 episode sebagai data *training* dan sisanya 6-12 sebagai data *testing*. Skenario kedua (E2) yaitu membagi dataset untuk data *training* dan *testing* dilakukan secara acak dengan rasio 5:7. Jumlah *epoch* yang digunakan juga termasuk ke dalam variabel pembeding. Dari hasil penelitian didapatkan bahwa skenario E2 mendapatkan nilai akurasi terbaik yaitu 94% dengan jumlah *epoch* adalah 500 dan 94% untuk 2000 *epoch*.

Dalam penelitian yang dilakukan oleh Tianmei Guo, Jiwen Dong, Henjian Li dan Yunxing Gao (2017) dengan judul *Simple Convolutional Neural Network on Image Classification*. Penelitian ini menganalisis perbedaan metode dari tingkat pembelajaran dan optimasi algoritma untuk mencari tahu pengaruh parameter. Citra yang digunakan berupa gambar tulisan tangan angka 0 sampai 9 yang berjumlah 60000 dengan ukuran 28x28 piksel. Arsitektur yang diusulkan pada penelitian ini terdiri dari 3 *convolutional layer* dan 2 *activation layer* menggunakan *relu* dan *dropout*. Evaluasi hanya berdasarkan tingkat pembelajaran dan *error rate* saja. Dari hasil penelitiannya, didapatkan parameter yang berpengaruh terhadap tingkat pembelajaran adalah *multistep* yaitu semakin banyak perulangan pada proses *training*, nilai tingkat pembelajaran semakin baik. Meskipun tingkat *error* dari arsitektur yang diusulkan ini sebesar 0,66% lebih besar dibandingkan dengan

arsitektur CNN yang sudah banyak digunakan, arsitektur dalam penelitian ini memiliki proses komputasi yang cukup ringan dan cepat.

Dalam penelitiannya Rokhana (2019) yang berjudul *Convolutional Neural Network* untuk Pendeteksian patah Tulang Femur pada Citra Ultrasonik B-Mode, berusaha menguji pengaruh lapisan CONV dan fungsi aktivasinya terhadap nilai akurasi klasifikasi citra patah tulang. Selain membandingkan jumlah dan kombinasi lapisan CONV, penelitian ini juga membandingkan lapisan CNN terbaik dengan algoritma lain seperti SVM dan KNN. Dataset yang digunakan berjumlah 9600 data yang dibagi menjadi 4 kelas dengan ukuran 128x128 piksel. Rasio pembagian data *training* dan *testing* sama yaitu 5:5. Variabel evaluasi algoritma yang digunakan adalah tingkat akurasi, sensitivitas dan *specificity* nya terhadap jumlah *batch*, *epoch*, penggunaan *dropout* dan waktu pelatihan. Dari hasil percobaan perbandingan terhadap CNN1, CNN2, CNN3, CNN4, CNN5, SVM dan KNN didapatkan lapisan atau arsitektur terbaik adalah CNN4 dengan 4 CONV *layer*, 2 aktivasi *layer* dan 1 *fully connected layer* dengan nilai akurasi 95,3%.

Penelitian berikutnya oleh Febian Fitra Maulana dan Naim Rochmawati (2019) dengan judul Klasifikasi Citra Buah Menggunakan *Convolutional Neural Network*. Fokus pada penelitian ini adalah untuk menguji tingkat akurasi pengklasifikasian buah dengan arsitektur CNN dengan perpaduan 3 *layer* konvolusi dan 2 *hidden layer*. Dataset buah didapatkan dari situs *kaggle* dengan nama dataset *Fruit-360*. Dari keseluruhan dataset hanya diambil 13% dari total kelas yang ada secara manual dipilih oleh peneliti yaitu total 3720 data. Dari 3720 data dibagi menjadi 90% sebagai data *training* dan 10% sebagai data *testing* dengan ukuran

citra 100 x 100 piksel. Penilaian terhadap model arsitektur pada penelitian ini hanya didasarkan pada tingkat akurasinya saja tanpa menguji tingkat presisi, dan *recall*. Tingkat akurasi yang didapatkan adalah sebesar 97,97%.

Dalam penelitian yang dilakukan oleh Muhammad Zufar dan Budi Setiyono (2016) dengan judul *Convolutional Neural Network* untuk Pengenalan Wajah secara *Real Time*. Jumlah data yang digunakan adalah 126 gambar yang diperoleh dari pengambilan menggunakan kamera webcam dan dibedakan menjadi 2 jenis yaitu dataset gambar wajah *indoor* (pencahayaannya minim) dan gambar wajah *outdoor* (pencahayaannya terang). Sebelum dilakukan proses klasifikasi, dilakukan pengolahan pra proses terhadap citra yaitu dengan mengubah menjadi *grayscale* dan mengubah ukuran citra menjadi 46 x 46 piksel dengan metode *Extended Local Binary Pattern*. Arsitektur CNN yang diterapkan pada penelitian ini menggunakan kedalaman 7 *layer*. Dari percobaan yang telah dilakukan, didapatkan nilai akurasi sebesar 87%. Dalam penelitian ini, didapatkan juga hasil bahwa penggunaan metode *Extended Local Binary Pattern* dapat mengatasi pengaruh intensitas cahaya pada gambar.

2.2 Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian

Klasifikasi Citra Wayang Punakawan Menggunakan Metode *Gaussian Filter* Dan Algoritma *Convolutional Neural Network*

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	<i>Deep Learning for Visual Indonesian Place Classification with Convolutional Neural Networks</i>	Andry Chowanda dkk, 4th International Conference on Computer Science and Computational Intelligence, 2019	Membandingkan tingkat akurasi klasifikasi ketiga arsitektur antara VGG-16, VGG-19 dan GoogleNet dengan algoritma <i>Convolutional Neural Network</i> .	Dengan rasio data latih sebesar 91%, data uji sebesar 9% dan ukuran citra 224 x 224 piksel didapatkan arsitektur CNN terbaik adalah model VGG-19 dengan 15 <i>convolutional layer</i> , 5 <i>pooling phases</i> dan 4 <i>fully connected</i> dan <i>classifier layer</i> (flatten, dense dan softmax classifier) memiliki tingkat akurasi sebesar 92%.	Beberapa tempat di Indonesia memiliki ciri/fitur yang hampir sama dan karena minimnya dataset primer sebelum dilakukan pemutaran dimensi citra untuk menambah dataset, sehingga menimbulkan kesalahan dalam pengklasifikasian. Selain itu tidak dilakukannya pra proses citra sebelum dilakukan klasifikasi	Perbedaan antara penelitian sebelumnya dengan penelitian yang akan dilakukan, diantaranya pada penelitian yang akan dilakukan akan melalui pra proses salah satunya dengan metode <i>gaussian filter</i> untuk menghilangkan noise dan meningkatkan blur pada citra.
2	<i>Evolutionary deep learning based on deep convolutional neural network for anime storyboard recognition</i>	Saya Fujino dkk, Elsevier, 2019	Membandingkan tingkat akurasi terhadap perbedaan rasio dataset yang digunakan untuk proses learning dan	Pendekatan dengan cara membagi data latih dan data uji secara acak dari total dataset dengan rasio 5:7 memiliki tingkat akurasi yang	Tidak adanya tahap pra proses terhadap citra, metode ini membutuhkan <i>resource</i> yang besar karena proses komputasi yang berat	Perbedaan antara penelitian sebelumnya dengan penelitian yang akan dilakukan diantaranya, <i>tool</i> yang digunakan pada penelitian ini menggunakan

Tabel 2.1. Matriks literatur review dan posisi penelitian

Klasifikasi Citra Wayang Punakawan Menggunakan Metode *Gaussian Filter* Dan Algoritma *Convolutional Neural Network*

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			testing dengan pendekatan baru <i>evolutioner Deep Learning</i> dengan algoritma <i>Convolutional Neural Network</i> .	lebih tinggi dengan nilai akurasi 93% untuk 500 epoch dan 94% epoch. Jumlah epoch yang digunakan untuk melakukan pelatihan tidak begitu berpengaruh pada penelitian ini.	dengan proses pelatihan dilakukan sebanyak 3 kali untuk memastikan dan mendapatkan nilai akurasi yang baik.	Java akan tetapi pada penelitian yang akan dilakukan menggunakan python dengan bantuan <i>library tensorflow</i> dan <i>keras</i> .
3	<i>Simple Convolutional Neural Network on Image Classification</i>	Tianmei Guo dkk, <i>IEEE 2nd International Conference on Big Data Analysis</i> , 2017	Untuk menganalisis perbedaan metode dari tingkat pembelajaran dan perbedaan optimasi algoritma untuk menyelesaikan parameter yang berpengaruh dalam klasifikasi gambar.	Parameter terbaik dan yang paling berpengaruh terhadap tingkat pembelajaran adalah <i>multistep</i> , semakin banyak dilakukan perulangan dalam proses pelatihan akan meningkatkan nilai tingkat pembelajaran. Arsitektur pada penelitian ini dengan 3 layer konvolusi dan 2 layer aktivasi dengan <i>relu</i> dan <i>dropout</i> menghasilkan tingkat	Tidak dilakukan perbandingan hasil tingkat eror dari dataset yang melalui pra proses dengan dataset yang langsung dilakukan klasifikasi. Perbandingan untuk mengetahui pengaruh parameter hanya terhadap tingkat eror saja yang dirasa kurang mewakili.	Perbedaan antara penelitian sebelumnya dengan penelitian yang akan dilakukan diantaranya, penelitian ini perbandingan hanya terhadap tingkat eror sedangkan pada penelitian yang akan dilakukan akan membandingkan arsitektur CNN berdasarkan dari tingkat akurasi, presisi dan <i>recall</i> dan pengaruh tahap pra proses dengan <i>gaussian filter</i> .

Tabel 2.1. Matriks literatur review dan posisi penelitian

Klasifikasi Citra Wayang Punakawan Menggunakan Metode *Gaussian Filter* Dan Algoritma *Convolutional Neural Network*

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				error (<i>error rate</i>) sebesar 0,66%.		
4	<i>Convolutional Neural Network</i> untuk Pendeteksian patah Tulang Femur pada Citra Ultrasonik B-Mode	Rika Rokhana dkk, JNETETI, 2019	Untuk menguji pengaruh lapisan CONV dan fungsi aktivasinya terhadap nilai akurasi klasifikasi citra patah tulang.	Konfigurasi/arsitektur terbaik adalah CNN4 dengan 4 CONV layer, 2 aktivasi layer dan 1 <i>fully connected layer</i> memperoleh nilai akurasi sebesar 95,3%.	Tidak adanya skenario percobaan dengan mengubah rasio jumlah dataset untuk data latih dan data uji yaitu dengan rasio 5:5 yang digunakan untuk membandingkan pengaruh jumlah lapisan arsitektur CNN.	Perbedaan antara penelitian sebelumnya dengan penelitian yang akan dilakukan diantaranya, penelitian ini variabel evaluasi yang digunakan adalah akurasi, sensitivitas dan <i>specificity</i> sedangkan pada penelitian yang akan dilakukan variabel evaluasi yang akan digunakan adalah akurasi, presisi dan <i>recall</i> .
5	Klasifikasi Citra Buah Menggunakan <i>Convolutional Neural Network</i> .	Febian Fitra Maulana dkk, <i>Journal of Informatics and Computer Science</i> , 2019	Untuk menguji tingkat akurasi pengklasifikasian buah dengan arsitektur CNN dengan perpaduan 3 <i>Convolutional Layer</i> dan 2 <i>Hidden Layer</i>	Arsitektur CNN dengan perpaduan 3 <i>convolutional layer</i> dan 2 <i>hidden layer</i> mampu mendapatkan tingkat akurasi sebesar 97,97% dengan perbandingan rasio data latih 90% dan data uji 10%	Pemilihan kelas buah hanya 13% dari total dataset yang digunakan dan dipilih secara manual dan tidak secara acak, sehingga kurang mewakili tingkat akurasi untuk dataset tersebut	Perbedaan antara penelitian sebelumnya dengan penelitian yang akan dilakukan diantaranya, penelitian ini hanya menggunakan variabel akurasi saja untuk menguji arsitektur CNN. Sedangkan pada penelitian yang akan dikerjakan menggunakan

Tabel 2.1. Matriks literatur review dan posisi penelitian

Klasifikasi Citra Wayang Punakawan Menggunakan Metode *Gaussian Filter* Dan Algoritma *Convolutional Neural Network*

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
						variabel akurasi, presisi dan <i>recall</i> untuk menilai arsitektur CNN.
6	<i>Convolutional Neural Network</i> untuk Pengenalan Wajah secara <i>Real Time</i>	Muhammad Zufar dkk, JURNAL SAINS DAN SENI ITS, 2016	Untuk mengidentifikasi pengenalan wajah secara <i>realtime</i>	Tingkat akurasi yang didapatkan dari arsitektur dengan kedalaman 7 layer sebesar 87%, selain itu penggunaan metode <i>Extended Local Binary Pattern</i> dapat mengatasi pengaruh intensitas cahaya pada gambar	Gambar primer dan masih dibagi menjadi 2 jenis yaitu dataset wajah <i>indoor</i> (pencahayaan minim) dan gambar wajah <i>outdoor</i> (pencahayaan terang), mengingat arsitektur CNN memerlukan dataset yang banyak dan juga karena pendeteksian wajah secara <i>real time</i> sangat bergantung dengan jumlah data untuk setiap kelas/wajahnya	Perbedaan antara penelitian sebelumnya dengan penelitian yang akan dilakukan diantaranya, pada penelitian ini digunakan metode <i>Extended Local Binary Pattern</i> karena terdapat 2 jenis kondisi dataset. Sedangkan pada penelitian yang akan dilakukan menggunakan <i>gaussian filter</i> untuk menghilangkan <i>noise</i>

2.3 Landasan Teori

2.3.1 Klasifikasi

Dalam bukunya Santoso (2007) yang dikutip dari Agus Mulyanto (2009), klasifikasi merupakan langkah atau cara dalam upaya membentuk suatu model atau fungsi yang digunakan dalam menjelaskan atau membedakan konsep kelas data. Dengan melakukan proses ini suatu objek dapat dikenali dan dikelompokkan berdasarkan kelasnya dengan cara memperkirakan berdasarkan hasil dari model yang telah dibentuk. Klasifikasi dokumen adalah bagian proses yang penting dalam bidang sistem informasi, khususnya untuk pengetahuan bisnis.

Penelitian oleh Han & Kamber (2001) menjelaskan bahwa dalam proses klasifikasi terbagi menjadi dua tahapan yaitu pelatihan (*learning*) dan pengujian (*testing*). Kedua tahapan ini saling berurutan dalam proses klasifikasi. Pada tahap *learning* (pelatihan) ini merupakan tahap di mana dilakukan proses pembelajaran terhadap suatu data yang telah diketahui kelasnya atau sering disebut data latih. Tahapan ini dimaksudkan agar komputer dapat belajar mengenal beberapa objek (data latih) berdasarkan kelasnya sehingga dihasilkan satu model klasifikasi. Kemudian tahapan kedua adalah proses *testing* (pengujian), tahap ini berfungsi untuk melakukan evaluasi tingkat kinerja dari model hasil dari tahap *learning* dengan data baru yang disebut data uji. Keluaran dari tahap ini berupa tingkat keakuratan suatu model dalam memprediksi data yang belum diketahui kelasnya (data uji).

Jika nilai akurasi dari tahap *testing* ini tinggi, maka dapat model hasil pembelajaran pada tahap *learning* layak untuk digunakan dalam memprediksi data-

data baru yang belum diketahui kelasnya. Proses klasifikasi sangat erat hubungannya dengan teknik atau algoritma yang dapat belajar dan mengelompokkan data ke dalam kelas-kelasnya. Beberapa algoritma yang dapat digunakan dalam proses klasifikasi diantaranya adalah *Naive Bayes Classifier*, *Decision Tree*, *Rule Based Classifier* dan Jaringan Saraf Tiruan atau lebih dikenal *Neural Network*. Masing-masing algoritma memiliki cara kerja yang berbeda-beda dalam proses klasifikasi data, selain itu setiap algoritma juga memiliki kelebihan dan kekurangan masing-masing dalam melakukan klasifikasi pada banyak kasus Han & Kamber (2001).

2.3.2 Pengolahan Citra Digital

Dalam bukunya Basuki, Palandi & Fatchurrohman (2005), pengambilan objek gambar dengan mesin dalam bentuk pendekatan berdasarkan contoh dan jumlah merupakan representasi/gambaran dari citra digital. Besarnya kotak-kotak yang disusun dalam baris dan kolom atau piksel adalah yang dimaksud dengan sampling. Sedangkan jumlah atau kuantisasi menyatakan besar kecilnya tingkat kecerahan dari citra sesuai dengan jumlah bit biner mesin yang digunakan untuk mendapatkan citra tersebut.

Proses pengolahan data teks berbeda dengan pengolahan citra. Dalam pengolahan citra ini dilakukan dengan beberapa teknik untuk memanipulasi citra agar dapat diketahui perbedaan atau ciri khas yang membedakan antara citra yang satu dengan citra yang lain. Pengolahan citra adalah proses memanipulasi citra dengan mesin komputer dengan maksud agar kualitas citra tersebut menjadi lebih

baik Munir (2004). Menurut Iriyanto dan Zaini (2014) ada satu hal yang penting dalam konsep pengolahan citra yaitu mengekstrak informasi dari citra tersebut. Sebagai contoh seperti pengenalan berdasarkan bentuknya. Salah satu metode pengolahan citra yaitu *edge detection*, yaitu pemrosesan citra terhadap bidang *feature detection* dan *feature extraction*. *Edge* merupakan batas-batas dalam sebuah objek yang digunakan untuk mendeteksi dan mengidentifikasi tepi dalam citra digital.

2.3.3 Metode Gaussian Filter

Menurut ahmad (2005) *Gaussian Filter* adalah salah satu metode yang berfungsi untuk melakukan filterisasi citra sebelum dilakukan klasifikasi yang merupakan salah satu filter linier dengan nilai pembobotan untuk setiap anggotanya dan dipilih berdasarkan bentuk dari fungsi *Gaussian*. Metode ini dipilih karena dapat memfilter citra dengan memperhalusnya berdasarkan pertimbangan bahwa filter ini mempunyai pusat kernel. Filter ini sangat efektif untuk menghilangkan noise yang bersifat sebaran normal. Noise sebaran normal ini banyak ditemukan pada sebaran citra hasil proses digitasi menggunakan kamera karena merupakan fenomena alamiah akibat sifat pantulan cahaya dan kepekaan sensor cahaya pada kamera itu sendiri.

Untuk menghitung atau menentukan nilai-nilai setiap elemen dalam filter penghalus *Gaussian* yang akan dibentuk dapat dihitung melalui persamaan (2.1) :

$$\frac{h(x,y)}{c} = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.1)$$

σ = Lebar dari fungsi *Gaussian*

c = Konstanta normalisasi

$g(x,y)$ = Citra hasil konvolusi

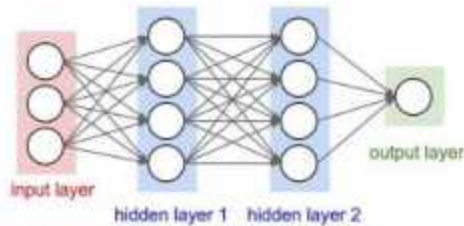
Fungsi dari kernel filter *Gaussian* ini membuat nilai-nilai tiap piksel akan saling berdekatan satu sama lain dengan piksel sebelahnya dan akan menghasilkan efek penghalusan.

2.3.4 *Convolutional Neural Network*

Convolutional Neural Network (CNN) adalah algoritma klasifikasi hasil dari pengembangan *Multilayer Perceptron* (MLP) yang dibangun untuk memproses data dua dimensi. Karena kedalaman jaringan yang tinggi dan banyak diterapkan pada data gambar/citra sehingga CNN termasuk *Deep Neural Network*. Algoritma ini pertama kali dikembangkan oleh Kunihiko Fukushima yaitu peneliti asal Jepang dan diberi nama NeoCognitron Fukushima (1980). Penelitian yang dilakukan Kunihiko kemudian disempurnakan oleh Yann LeCun peneliti asal AT&T Bell Laboratories di Holmdel, New Jersey USA. Pada penelitiannya LeCun (1990), model CNN dengan nama LeNet berhasil diaplikasikan pada penelitiannya mengenai pengenalan angka dan tulisan tangan. Kemudian pada tahun 2012, Alex Krizhevsky menerapkan algoritma ini pada suatu ajang perlombaan dan berhasil menjuarai kompetisi tersebut. Dengan kemenangannya tersebut menjadi ajang pembuktian bahwa metode *Deep Learning* khususnya dengan algoritma CNN terbukti berhasil mengungguli metode/algoritma yang lainnya seperti SVM pada kasus klasifikasi citra.

2.3.5 Konsep CNN

Dikutip dari penelitian Putra (2016), konsep dasar cara kerja algoritma CNN mirip dengan MLP tetapi dalam CNN setiap neuron direpresentasikan dalam bentuk 2 dimensi.



Gambar 2.1. Arsitektur sederhana MLP

Pada Gambar 2.1 terlihat bahwa memiliki layer i (kotak merah dan biru) dengan masing-masing layer juga berisi neuron. MLP menerima masukan data satu dimensi dan mempropagasikan data tersebut pada jaringan hingga menghasilkan keluaran. Setiap hubungan antar neuron pada dua layer yang bersebelahan memiliki parameter bobot satu dimensi yang menentukan kualitas mode. Disetiap data input pada layer dilakukan operasi linear dengan nilai bobot yang ada, kemudian hasil komputasi akan ditransformasi menggunakan operasi non linear yang disebut sebagai fungsi aktivasi. Pada CNN, data yang dipropagasikan pada jaringan adalah data dua dimensi, sehingga operasi linear dan parameter bobot pada CNN berbeda. Pada CNN operasi linear menggunakan operasi konvolusi, sedangkan bobot tidak lagi satu dimensi saja, namun berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi.

2.36 Arsitektur *Convolutional Neural Network*

Secara garis besar bagian-bagian yang terdapat pada arsitektur algoritma *Convolutional Neural Network* dapat dilihat pada Gambar 2.2 :



Gambar 2.2. Arsitektur *Convolutional Neural Network*

2.3.6.1 *Convolutional Layer*

Convolutional layer adalah sebuah proses dimana citra dimanipulasi dengan menggunakan *external mask* atau *subwindows* untuk menghasilkan citra yang baru. Proses ini adalah mereduksi dimensi dari citra dengan operasi *convolutional* untuk mengekstrak fitur-fitur penting dari citra seperti deteksi ciri/*edge*, warna, orientasi gradien, dll melalui proses *encoding*. Secara matematis, menurut Madenda (2015) konvolusi adalah jumlah total dari hasil kali antara setiap elemen yang bersesuaian (memiliki posisi koordinat yang sama) dalam dua matriks atau dua vektor, seperti yang ditunjukkan Gambar 2.3.

$$\begin{array}{|c|c|c|c|c|} \hline 3 & 9 & 1 & 2 & 7 & 4 \\ \hline 1 & 3 & 8 & 9 & 3 & 1 \\ \hline 5 & 7 & 2 & 5 & 1 & 3 \\ \hline 0 & 1 & 3 & 1 & 7 & 8 \\ \hline 4 & 2 & 1 & 6 & 2 & 8 \\ \hline 2 & 0 & 3 & 2 & 3 & 9 \\ \hline \end{array}
 \quad
 \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array}
 \quad
 \begin{array}{|c|c|c|} \hline 5 & -4 & 0 & 0 \\ \hline -10 & -2 & -2 & 3 \\ \hline 0 & -2 & -4 & -7 \\ \hline -3 & -2 & 3 & -6 \\ \hline \end{array}$$

$6 \times 6 \qquad 3 \times 3 \qquad 4 \times 4$

Gambar 2.3. *Convolutional Layer*

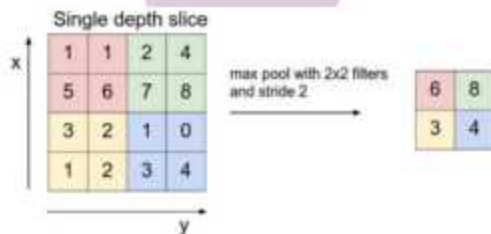
Secara matematis perhitungan dalam layer konvolusi (*convolutional layer*) dapat dicontohkan seperti pada Gambar 2.4.

$$\begin{array}{|c|c|c|} \hline 2 & 1 & 3 \\ \hline 4 & 6 & 5 \\ \hline 8 & 9 & 7 \\ \hline \end{array} \cdot \begin{array}{|c|c|c|} \hline 10 & 20 & 10 \\ \hline 20 & 10 & 20 \\ \hline 10 & 20 & 10 \\ \hline \end{array} = C = 2 \times 10 + 1 \times 20 + 3 \times 10 + \\ 4 \times 20 + 6 \times 10 + 5 \times 20 + \\ 8 \times 10 + 9 \times 20 + 7 \times 10 \\ = 640$$

Gambar 2.4. Contoh perhitungan *Convolutional Layer*

2.3.6.2 Pooling Layer

Pada bagian *pooling layer* ini terjadi proses mereduksi ukuran sebuah data citra. Dalam pengolahan citra, *pooling* juga bertujuan untuk meningkatkan invariansi posisi dari fitur serta mempercepat komputasi dan mengontrol terjadinya *overfitting*. *Layer* ini juga bertugas dalam mengurangi dimensi. Dalam prosesnya, membutuhkan kemampuan komputasi yang tinggi dalam memproses data. *Layer* ini sangat berguna untuk mengekstrak fitur-fitur penting dengan berbagai variasi posisi citra dan rotasi sehingga proses pelatihan atau *training* nantinya akan efektif seperti Gambar 2.5.



Gambar 2.5. Proses pada *Pooling Layer*

2.3.6.3 Fully Connected Layer

Setiap neuron pada *convolution layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah *fully connected layer*. *Feature map* yang dihasilkan dari proses *feature extraction* dengan *convolutional* dan *pooling layer* masih berbentuk *multidimensional array*, sehingga harus dilakukan *flatten* atau *reshape feature map* menjadi sebuah *vector* agar bisa digunakan sebagai input dari *fully connected layer*. *Vector* hasil proses *flatten* akan diumpungkan dan diproses dengan *feed forward neural network* dan *backpropagation* untuk setiap proses pelatihannya dengan serangkaian jumlah epoch. Hasil keluaran dari proses ini dapat membedakan antara fitur yang berpengaruh dan mendominasi dengan fitur tingkat rendah pada citra dan mengklasifikasikannya menggunakan teknik klasifikasi *softmax*.

2.3.7 Confusion Matrix

Dalam bukunya Sokolova & Lapalme (2009) *Confusion matrix* adalah salah satu metode yang digunakan untuk mengevaluasi kinerja algoritma klasifikasi. Tabel 2.2 merupakan gambaran sederhana untuk mempermudah pemahaman tentang istilah *confusion matrix* dalam keluaran klasifikasi.

Tabel 2.2. Tabel *Confusion Matrix*

		Kelas Prediksi	
		Positif	Negatif
Kelas Sesungguhnya	Positif	TP	FN
	Negatif	FP	TN

Nilai *True Negative* (TN) adalah data yang di klasifikasi dengan tepat sebagai keluaran negatif atau salah. *True Positive* (TP) adalah data yang diklasifikasi dengan tepat sebagai keluaran positif atau benar. *False Positive* (FP) adalah data yang diklasifikasi dengan kurang tepat apabila keluaran berupa positif atau benar. *False Negative* (FN) adalah data yang diklasifikasi dengan kurang tepat.

$$\text{Presisi} = \frac{\sum_i^n \frac{TP_i}{TP_i + FP_i}}{n} \quad (2.2)$$

Persamaan (2.2) merupakan perhitungan rata-rata nilai *precision* yaitu dari data hasil klasifikasi seberapa banyak data yang benar antara nilai sebenarnya dengan prediksi yang diberikan oleh sistem.

$$\text{Recall} = \frac{\sum_i^n \frac{TP_i}{TP_i + FN_i}}{n} \quad (2.3)$$

Persamaan (2.3) merupakan perhitungan rata-rata nilai *recall* yaitu dari seluruh data benar seberapa banyak data yang keluar dalam hasil klasifikasi. Evaluasi *recall* digunakan apabila lebih memilih nilai *False Positive* daripada *False Negative* Ghoneim (2019).

$$\text{Akurasi} = \frac{\sum_i^n \frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i}}{n} \quad (2.4)$$

Persamaan (2.4) merupakan perhitungan rata-rata nilai akurasi untuk menunjukkan tingkat efektifitas per kelas dari sebuah klasifikasi Sokolova & Lapalme (2009).

BAB III

METODE PENELITIAN

3.1 Jenis, Sifat dan Pendekatan Penelitian

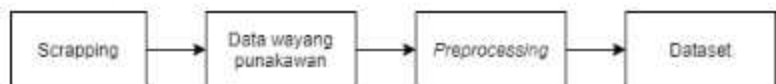
3.1.1 Metode Penelitian

Jenis penelitian yang digunakan adalah penelitian eksperimental, yaitu menerapkan serangkaian tindakan/skenario percobaan untuk membuktikan suatu konsep yang diusulkan. Sedangkan sifat penelitian ini bersifat kausal, yaitu memperoleh informasi yang berhubungan dengan hubungan sebab akibat dengan pendekatan penelitian kuantitatif yaitu menjelaskan bagaimana sesuatu dibangun dan bekerja yang berupa fakta dan bersifat objektif menggunakan skala numerik. Dalam penelitian ini melakukan percobaan dengan melakukan pembelajaran terhadap klasifikasi citra wayang kulit punakawan dengan penerapan arsitektur *Convolutional Neural Network* dan metode *Gaussian Filter* dengan beberapa skenario percobaan kemudian akan dievaluasi hasil pembelajaran tersebut untuk menilai keakuratan dalam mengklasifikasikan citra wayang kulit punakawan dari setiap skenario untuk diketahui fakta-fakta dari hasil penelitian.

3.2 Metode Pengumpulan Data

Data yang dibutuhkan untuk penelitian ini berupa gambar/citra wayang kulit punakawan. Metode pengumpulan data yang penulis lakukan untuk mendapatkan citra wayang yaitu dengan cara *scrapping* dari google menggunakan *library* python *scrappy*. Hasil *scrapping* akan dipisahkan ke dalam 4 folder yang berbeda yaitu

berdasarkan nama tokoh pewayangannya yang terdiri dari Gareng, Semar, Petruk dan Bagong. Data tersebut nanti akan diseleksi berdasarkan kelayakan citra untuk dijadikan sebagai dataset.



Gambar 3.1. Alur pengumpulan data

Secara garis besar alur pengumpulan data pada penelitian ini ditunjukkan pada Gambar 3.1. Pada tahapan pertama yaitu akan dilakukan proses pengunduhan data berupa gambar/citra wayang kulit punakawan dari sumber internet yaitu *google image* dengan cara *scrapping*. Tahap berikutnya adalah *preprocessing* yaitu terdapat beberapa tahapan dan perlakuan terhadap citra. Perlakuan pertama yaitu proses pelabelan dan penyortiran citra, yaitu citra dibedakan menjadi 4 folder dengan nama folder sesuai nama dari kelas/tokoh pewayangannya. Pada proses penyortiran citra yang sama akan diambil salah satu, karena tidak menutup kemungkinan akan terjadi duplikasi citra yang berasal dari sumber yang berbeda.

Kemudian tahap atau perlakuan berikutnya adalah proses augmentasi pada citra dengan tujuan untuk menambah jumlah dataset yang akan digunakan sehingga diharapkan performa model klasifikasi akan meningkat. Beberapa proses augmentasi yaitu membalikan gambar secara horizontal, melakukan zoom-in, melakukan rotasi gambar, dan melakukan *mirroring* citra. Dari beberapa tahap di atas, citra wayang kulit telah menjadi dataset yang siap digunakan untuk proses berikutnya yaitu pengolahan citra menggunakan metode *Gaussian Filter* dan algoritma *Convolutional Neural Network (CNN)*.

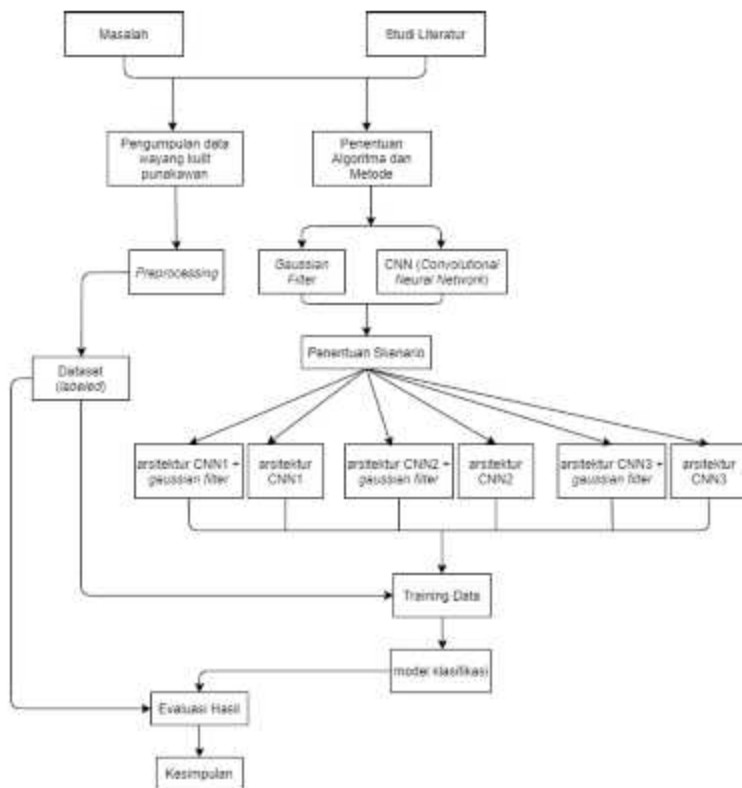
3.3 Metode Analisis Data

Metode analisis yang digunakan dalam penelitian ini adalah analisis kuantitatif menggunakan algoritma *Convolutional Neural Network*. Data yang sudah terkumpul menjadi dataset citra wayang punakawan selanjutnya 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 3 skenario utama terhadap penggunaan kombinasi dari arsitektur CNN. Dari ketiga skenario utama tersebut masing-masing akan dilakukan percobaan terhadap citra yang melalui proses filterisasi dengan metode *Gaussian Filter* dan tidak melalui proses filterisasi. Dari proses *training* akan terbentuk 6 model klasifikasi yang masing-masing akan diuji tingkat akurasi, presisi dan *recall* dan *F-score* untuk mencari arsitektur yang paling baik dan mengetahui pengaruh dari penggunaan metode *Gaussian Filter*. Dalam melakukan percobaan ini dilakukan dengan bahasa pemrograman python dengan bantuan *libray* Keras dan Tensorflow dalam membangun arsitektur CNN.

3.4 Alur Penelitian

Berikut adalah bagan alur dari penelitian:



Gambar 3.2. Alur penelitian

Alur penelitian secara sistematis tahap demi tahap dapat dilihat pada Gambar 3.2. dan lebih jelasnya akan dijelaskan sebagai berikut:

1. Identifikasi masalah

Dalam proses identifikasi masalah dengan mencari tahu permasalahan yang ada pada objek penelitian dengan membaca beberapa artikel terkait.

2. Studi literatur

Dalam studi literatur yang dilakukan adalah dengan mencari dan membaca jurnal serta buku yang relevan dengan permasalahan yang akan diangkat sebagai bahan rujukan dalam memilih metode atau algoritma yang sesuai dan menentukan objek penelitian.

3. Pengumpulan data wayang kulit punakawan serta menentukan algoritma dan metode

Setelah membaca beberapa literatur yang relevan, tahap berikutnya adalah proses pengumpulan data berupa wayang kulit dengan tokoh punakawan dari sumber internet dengan metode *scrapping*. Selain itu setelah membaca penelitian-penelitian terdahulu yang relevan terkait *image classification*, penulis menentukan untuk menggunakan algoritma *Convolutional Neural Network* dan metode *Gaussian Filter* dalam penelitian ini.

4. *Preprocessing*

Dalam proses ini dilakukan pengolahan terhadap kumpulan citra wayang kulit punakawan sebelum dilakukan klasifikasi. Beberapa *preprocessing* seperti labeling data, menyamakan ukuran piksel semua

citra, proses segmentasi dan proses augmentasi pada citra untuk menambah jumlah dataset citra.

5. Dataset

Hasil dari pra proses akan menghasilkan satu dataset wayang kulit punakawan yang terdiri dari 4 kelas (Semar, Gareng, Petruk dan Bagong). Dataset ini akan digunakan untuk proses *training* dan *testing*.

6. Penentuan algoritma dan metode

Dalam penelitian ini penulis menggunakan algoritma *Convolutional Neural Network* (CNN) untuk proses klasifikasi wayang kulit punakawan. Kemudian dalam penelitian ini menggunakan metode *Gaussian Filter* dalam proses filterisasi untuk menghilangkan noise yang ada pada citra.

7. Penentuan skenario

Pada penelitian ini digunakan tiga arsitektur CNN yang telah teruji kemampuannya dalam imagenet yaitu VGG16, ResNet50 dan MobileNet.

8. Training data

Pada tahap ini, akan dilakukan proses pelatihan menggunakan data *training* dari dataset wayang kulit punakawan. Proses pelatihan atau *training* ini dilakukan terhadap keenam skenario pada gambar 7. di atas sehingga akan dihasilkan 6 model klasifikasi berdasarkan total jumlah skenario yang ada. Skenario percobaan yang dilakukan adalah kombinasi antar arsitektur CNN dengan metode *Gaussian Filter* dan

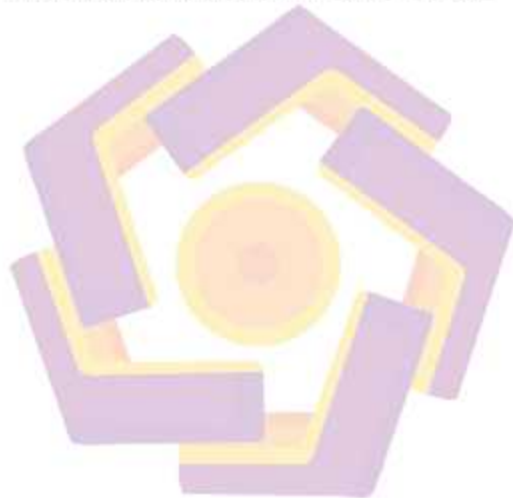
arsitektur CNN saja tanpa melalui tahap pembersihan noise dengan metode *Gaussian Filter*. Proses pelatihan dengan algoritma *Convolutional Neural Network* akan melalui proses *features extraction* yang akan ditangani oleh 2 lapisan dalam arsitektur CNN ini yaitu lapisan *convolutional layer* dan *pooling layer*. Pada penelitian ini proses ekstraksi fitur dari citra berdasarkan deteksi ciri/*edge* dari citra wayang punakawan. Setelah dilakukan pengurangan dimensi dan didapatkan fitur-fitur yang dominan pada citra, berikutnya akan dilakukan proses pelatihan atau *training* yang terjadi pada lapisan *fully connected layer* hingga terbentuk model klasifikasi yang siap untuk dievaluasi melalui proses *testing*. Komponen *fully connected layer* ini berperan dalam mengklasifikasikan citra ke dalam 4 kelas berdasarkan tingkat probabilitas yang paling mendekati.

9. Evaluasi hasil

Pada tahap ini dilakukan proses pengujian terhadap model klasifikasi yang telah dibuat dari sejumlah skenario percobaan. Dalam tahapan ini akan didapatkan nilai-nilai *confusion matrix* dari setiap skenario untuk menilai keakuratan, presisi, dan *recall* dari model klasifikasi. Kecepatan waktu proses mulai dari proses *preprocessing* masing-masing skenario hingga proses pembelajaran model akan diukur pada setiap skenario percobaan menggunakan *library* dari python yaitu *time execution process*.

10. Kesimpulan

Setelah didapatkan beberapa hasil fakta dengan beberapa skenario percobaan, tahap selanjutnya adalah membuat kesimpulan dengan menyajikan hasil dari percobaan yang telah dilakukan dengan beberapa fakta terkait arsitektur CNN, perlakuan terhadap dataset dan pengaruh penggunaan metode *Gaussian Filter* terhadap tingkat akurasi klasifikasi dan arsitektur mana yang memiliki kinerja terbaik.



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 dataset, penentuan arsitektur dan skenario penelitian, perlakuan awal terhadap dataset sebelum dilakukan proses pembelajaran dengan arsitektur CNN, proses pembuatan model dan evaluasi hasil percobaan.

4.1 Pengumpulan Data

Data yang digunakan pada penelitian ini adalah wayang punakawan yang terdiri dari wayang Gareng, Semar, Petruk dan Bagong. Data wayang punakawan tersebut didapatkan dari sumber google image. Tahapan pengumpulan data pada penelitian ini dilakukan secara melalui proses *scrapping image* dengan *library* selenium dari python. List kode untuk melakukan scrapping data dapat dilihat pada baris kode di bawah ini.

```
from selenium import webdriver
import os
import time
import requests
import io
from PIL import Image
import hashlib

def fetch_image_urls(query: str, max_links_to_fetch: int, wd: webdriver,
sleep_between_interactions: int = 1):
    def scroll_to_end(wd):
        wd.execute_script("window.scrollTo(0, document.body.scrollHeight);")
        time.sleep(sleep_between_interaction)
```

```

search_url =
"https://www.google.com/search?safe=off&site=&tbm=isch&source=hp&q={q}&oq
={q}&gs_l=img"
wd.get(search_url.format(q=query))
image_urls = set()
image_count = 0
results_start = 0
while image_count < max_links_to_fetch:
    scroll_to_end(wd)
    thumbnail_results = wd.find_elements_by_css_selector("img.Q4LuWd")
    number_results = len(thumbnail_results)
    print(f"Found: {number_results} search results. Extracting links from
{results_start}:{number_results}")
    for img in thumbnail_results[results_start:number_results]:
        try:
            img.click()
            time.sleep(sleep_between_interactions)
        except Exception:
            continue

    actual_images = wd.find_elements_by_css_selector('img.n3VNCb')
    for actual_image in actual_images:
        if actual_image.get_attribute('src') and 'http' in
actual_image.get_attribute('src'):
            image_urls.add(actual_image.get_attribute('src'))

    image_count = len(image_urls)

    if len(image_urls) >= max_links_to_fetch:
        print(f"Found: {len(image_urls)} image links, done!")
        break
    else:
        print("Found:", len(image_urls),
            "image links, looking for more ...")
        time.sleep(30)
        return
    load_more_button = wd.find_element_by_css_selector(".mye4qd")
    if load_more_button:
        wd.execute_script("document.querySelector('.mye4qd').click();")

    results_start = len(thumbnail_results)

return image_urls

```

Kode di atas merupakan inisiasi *package* dengan cara melakukan import beberapa *library* yang digunakan. Setelah itu dibuat sebuah fungsi dengan awalan `def` seperti pada baris kode di atas. Fungsi tersebut digunakan untuk mengaktifkan `webdriver` dari `selenium` dan membuka browser `google chrome` dengan set target url adalah pencarian pada `google` yang akan mengirimkan parameter nilai `keyword` yang dimasukkan untuk diletakkan pada mesin pencarian `google`.

```
def persist_image(folder_path: str, url: str):
    try:
        image_content = requests.get(url).content
    except Exception as e:
        print(f"ERROR - Could not download {url} - {e}")
    try:
        image_file = io.BytesIO(image_content)
        image = Image.open(image_file).convert('RGB')
        file_path = os.path.join(folder_path, hashlib.sha1(
            image_content).hexdigest()[:10] + '.jpg')
        with open(file_path, 'wb') as f:
            image.save(f, "JPEG", quality=85)
        print(f"SUCCESS - saved {url} - as {file_path}")
    except Exception as e:
        print(f"ERROR - Could not save {url} - {e}")

def search_and_download(search_term: str, driver_path: str, number_images: int,
    target_path='./images'):
    target_folder = os.path.join(
        target_path, '_'.join(search_term.lower().split(' ')))
    if not os.path.exists(target_folder):
        os.makedirs(target_folder)
    with webdriver.Chrome(executable_path=driver_path) as wd:
        res = fetch_image_urls(search_term, number_images,
            wd=wd, sleep_between_interactions=0.5)
    for elem in res:
        persist_image(target_folder, elem)
```

Kemudian setelah berhasil dibuat sebuah fungsi untuk menginisiasi dan membuka search bar pada google chrome dengan memasukkan keyword yang dimasukkan, tahap selanjutnya adalah pembuatan fungsi dengan nama `persist_image` seperti baris kode di atas. Fungsi tersebut digunakan untuk melakukan validasi image yang didapatkan dan sekaligus untuk proses pengunduhan/menyimpan image. Kemudian pada baris kode selanjutnya dibuat sebuah fungsi dengan nama `search_and_download`. Fungsi ini akan merupakan fungsi utama dalam proses scrapping ini, karena pada fungsi ini akan mengeksekusi 2 fungsi yang telah didefinisikan.

```
DRIVER_PATH = 'C:\webdrivers\chromedriver.exe'
search_term = input("Masukkan kata kunci : ")
jumlah = input("Masukkan jumlah gambar : ")
while not jumlah:
    jumlah = input("Masukkan jumlah gambar : ")
jml = int(jumlah)

search_and_download(search_term=search_term,
                    driver_path=DRIVER_PATH, number_images=jml)
```

Tahap selanjutnya pada pembuatan skrip *scrapping image* ini adalah pemanggilan fungsi utama yang telah dibuat tadi untuk dijalankan dengan parameter yang dimasukkan oleh penulis. Pada baris kode di atas jika penulis tidak memasukkan jumlah wayang yang akan disrape maka sistem akan tetap meminta penulis untuk memasukkan jumlah wayang yang akan disrape. Penulis cukup memasukkan inputan keyword untuk wayang yang ingin dicari dan jumlahnya, kemudian program akan berjalan dengan membuka browser chrome dan mengetikkan keyword masukan dari penulis tadi, dan sistem akan membuka satu-satu gambar wayang tersebut dan mengunduhnya dan disimpan pada tujuan folder

yang telah ditentukan. Data gambar wayang yang diperoleh terdapat beberapa duplikasi, sehingga diperlukan proses pemilihan secara manual gambar yang sama dilakukan penghapusan.



Gambar 4.1. Pelabelan dataset

Pada Gambar 4.1 ditunjukkan proses pelabelan dataset citra wayang. Proses pelabelan kelas cukup dengan memberikan nama wayang pada folder tempat menyimpan citra wayang. Dataset disimpan di dalam *google drive*, karena proses pengolahan citra menggunakan *google colab*. Setiap kelas terdiri dari 100 dataset citra wayang yang berbeda. Berikut ditunjukkan sampel untuk masing-masing kelas citra wayang punakawan pada Tabel 4.1.

Tabel 4.1. Tabel contoh citra wayang punakawan

Kelas	Citra Wayang
Bagong	

Tabel 4.1. Tabel contoh citra wayang punakawan

Gareng	
Semar	
Petruk	

Citra wayang yang didapatkan seperti ditunjukkan pada Tabel 4.1 memiliki latar belakang yang beragam, sehingga diperlukan proses *preprocessing* untuk mengolah dataset tersebut sebelum dilakukan proses pelatihan model. Total citra wayang yang digunakan penulis dalam penelitian ini berjumlah 400 data yang masing-masing terbagi sama rata untuk setiap kelasnya yaitu 100 citra yang berbeda untuk setiap kelas. Citra memiliki variasi latar belakang dan posisi yang bermacam-macam, hal tersebut untuk meningkatkan proses pembelajaran sistem dalam mengenali citra.

4.2 Analisa Data

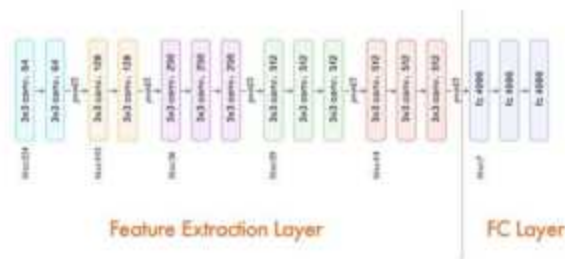
Setelah data citra wayang berhasil dikumpulkan dan telah diberikan label sesuai dengan nama kelas yang disesuaikan dengan nama tokoh wayang, tahap selanjutnya adalah proses pengolahan data citra wayang menjadi dataset sehingga dapat digunakan sebagai masukan dalam proses pembentukan model klasifikasi. Proses pembentukan model klasifikasi ini akan dilakukan terhadap tiga arsitektur CNN yang dikombinasikan dengan penggunaan *Gaussian Filter* dan perlakuan *preprocessing* data untuk mendapatkan akurasi yang terbaik dari skenario percobaan yang dilakukan.

4.2.1 Arsitektur CNN

Penelitian ini digunakan percobaan terhadap 3 arsitektur terbaik yang telah teruji tingkat kinerja dari masing-masing arsitektur pada sebuah kompetisi ImageNet. Ketiga arsitektur tersebut adalah VGG-16, ResNet50 dan MobileNet. Masing-masing arsitektur tersebut akan dikombinasikan dengan beberapa kondisi dan penggunaan filter gaussian untuk melatih dataset wayang dan dihasilkan akurasi terbaik dari semua skenario yang diujikan.

4.2.1.1 Arsitektur VGG-16

Arsitektur VGG-16 telah memenangkan kompetisi ILSVR (*Imagenet*) pada tahun 2014. Seperti namanya, arsitektur ini terdiri dari 16 blok layer yang disebut sebagai feature extraction layer, yang terdiri dari layer konvolusi dan layer pooling yang bertugas mengekstrak fitur dan mengecilkan dimensi dari citra.

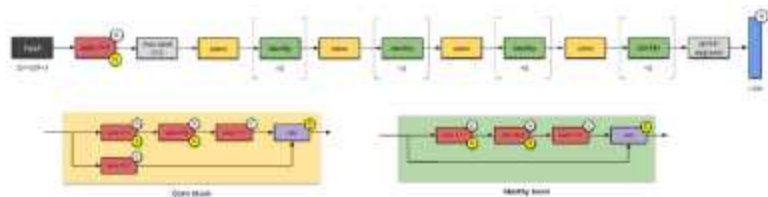


Gambar 4.2. Arsitektur VGG-16

Gambar di atas seperti ditunjukkan pada Gambar 4.2 merupakan struktur dari arsitektur VGG-16. Dalam arsitektur VGG-16 ini digunakan ukuran filter yang konsisten sama yaitu 3×3 dengan jumlah *stride* adalah 1 untuk *Convolutional Layer*. Pada *Pooling Layer* digunakan jenis *max pooling* dengan parameter *padding* semuanya adalah "same" dengan jumlah filter 2×2 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, karena pada *Imagenet* terdiri dari 1000 kelas. Fungsi dari *Fully Connected Layer* ini adalah untuk menentukan kelas pada citra berdasarkan tingkat probabilitas dari setiap kelas yang paling mendekati. Total parameter arsitektur VGG-16 ini adalah sebesar 138,357,544.

4.2.1.2 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.

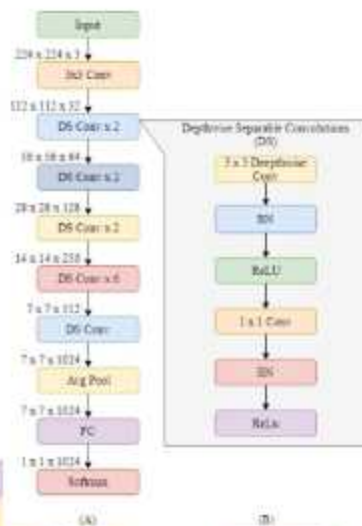


Gambar 4.3. Arsitektur ResNet50

Pada Gambar 4.3 ditunjukkan gambaran tentang struktur arsitektur ResNet50. Arsitektur ini terkenal dengan kedalaman jaringannya, yaitu terdiri lebih dari 177 lapisan. Arsitektur ini pertama kali yang memperkenalkan konsep *skip connection*. Salah satu keunggulan dari arsitektur ini adalah penggunaan konsep *skip connection*, yaitu menumpuk lapisan konvolusi 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.

4.2.1.3 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 4.4. Arsitektur MobileNet

Arsitektur MobileNet ini menggunakan konsep konvolusi yang dapat dipisahkan secara mendalam kecuali pada lapisan pertama seperti ditunjukkan pada Gambar 4.4. Lapisan konvolusi pertama merupakan lapisan konvolusi 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. Oleh karena itu arsitektur ini dikenal karena ringan dengan proses komputasi model yang lebih cepat, sehingga dapat digunakan pada aplikasi mobile. 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 *layer* konvolusi dengan ketebalan filter yang sesuai

dengan ketebalan dari input image. MobileNet membagi konvolusi menjadi *depthwise convolution* dan *pointwise convolution*.

4.2.2 Fine Tuning

Sebelum dilakukan proses pembelajaran menggunakan ketiga arsitektur, tahap sebelumnya yang perlu dilakukan adalah proses *fine tuning*. Proses ini merupakan tahap pengambilan bobot dari arsitektur yang telah dilatih sebelumnya menggunakan dataset Imagenet. Selain itu proses fine tuning juga digunakan untuk menyesuaikan jumlah kelas pada layer konvolusi terhadap objek/dataset yang akan dilatih. Karena ketiga arsitektur tersebut menggunakan 1000 kelas pada Imagenet, sehingga diperlukan penyesuaian jumlah kelas terhadap jumlah objek/dataset yang akan digunakan. Pada penelitian ini tahapan *fine tuning* yang diterapkan pada ketiga arsitektur adalah pemangkasan layer *fully connected* dan penambahan layer dropout dan *classification layer* dengan jumlah kelas adalah 4 kelas.

block5_conv1 (Conv2D)	(None, 14, 14, 512)	2159040
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2159040
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2159040
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
Flatten (Flatten)	(None, 25088)	0
FC1 (Dense)	(None, 4096)	102744384
FC2 (Dense)	(None, 4096)	16781312
prod1_logits (Dense)	(None, 1000)	4007000

total params: 138,357,544		
trainable params: 138,357,544		
non-trainable params: 0		

lan(model.layers)

23

Gambar 4.5. Model VGG-16 sebelum *fine tuning*

Pada Gambar 4.5 ditunjukkan keluaran model history dari arsitektur VGG-16 sebelum dilakukan proses *fine tuning*. Pada arsitektur ini terdapat 5 blok *feature extraction layer* yang terdiri dari *convolution layer* dan *pooling layer*. Jumlah kelas yang ada pada layer dense adalah 1000 kelas sesuai dengan jumlah kelas pada ImageNet. Jumlah parameter pada model/arsitektur ini mula-mulanya adalah berjumlah 138.357.544 parameter dengan jumlah layer keseluruhan adalah 23 layer.

conv_1 (Conv2D)	(None, 32, 32, 3)	216000
conv_2 (Conv2D)	(None, 32, 32, 3)	216000
conv_3 (Conv2D)	(None, 64, 64, 3)	432000
conv_4 (Conv2D)	(None, 64, 64, 3)	432000
conv_5 (Conv2D)	(None, 128, 128, 3)	0
flatten (Flatten)	(None, 21600)	0
fc1 (Dense)	(None, 4096)	20176000
fc2 (Dense)	(None, 4096)	1079136
dropout (Dropout)	(None, 4096)	0
dense (Dense)	(None, 4)	10000
Total params: 134,276,932		
Trainable params: 134,276,932		
Non-trainable params: 0		
1 (Dropout Layer)		
24		

Gambar 4.6. Model VGG-16 setelah *fine tuning*

Pada Gambar 4.6 diperlihatkan model history dari arsitektur VGG-16 setelah dilakukan proses *fine tuning*. Dari gambar di atas terlihat pada layer dense yang semula memiliki nilai 1000 sebagai jumlahnya, sekarang setelah dilakukan proses *fine tuning* berubah menjadi 4 sesuai dengan jumlah kelas pada tokoh wayang punakawan. Selain itu jumlah parameter saat ini menjadi 134.276.932. Pada tahap ini penulis juga menambahkan 1 layer sebelum ke dense yaitu layer *dropout* dengan nilai 0.5, hal ini bertujuan untuk mencegah

Pada Gambar 4.8 ditunjukkan model history arsitektur ResNet50 setelah dilakukan proses *fine tuning*. Pada proses ini penulis memangkas layer FC yang awal mulanya memiliki jumlah kelas 1000 dan menggantinya dengan layer FC dengan jumlah dense layer adalah 4 kelas. Pada layer di atasnya penulis menambahkan layer dropout juga untuk mencegah terjadinya *overfitting*, sehingga jumlah keseluruhan layer saat ini adalah 178 layer. Beberapa parameter juga telah dibekukan/dinonaktifkan akibat proses *fine tuning* ini sehingga total parameter saat ini yang digunakan berjumlah 23.595.908 parameter dengan trainable params sebanyak 23.542.788 dan non trainable params sebanyak 53.120.

Proses *fine tuning* juga dilakukan pada arsitektur MobileNet. Pada Gambar 4.9 ditunjukkan potongan model history dari arsitektur MobileNet dengan jumlah 93 layer. Jumlah kelas pada layer klasifikasi berjumlah 1000 kelas dengan total parameter yang digunakan berjumlah 4.253.864 yang terbagi menjadi dua yaitu *trainable params* dengan jumlah 4.231.976 parameter dan *non trainable params* sebanyak 21.888.

```

conv_13_mha (Conv2D) (None, 7, 7, 1632) 0
conv_14 (Conv2D) (None, 7, 7, 1632) 184376
conv_15_m (keras.layers.Conv2D) (None, 7, 7, 1632) 488
conv_16_mha (Conv2D) (None, 7, 7, 1632) 0
global_average_pooling_2 (None, 1632) 0
maxpool_2 (MaxPool) (None, 1, 1, 1632) 0
dropout (Dropout) (None, 1, 1, 1632) 0
conv_17 (Conv2D) (None, 1, 1, 1000) 182288
maxpool_3 (MaxPool) (None, 1000) 0
global_pool (Activation) (None, 1000) 0
-----
Total params: 4,253,864
Trainable params: 4,231,976
Non trainable params: 21,888
-----
}
model.layers
}

```

Gambar 4.9. Model MobileNet sebelum *fine tuning*

Hasil dari proses *fine tuning* terhadap arsitektur MobileNet dapat dilihat pada Gambar 4.10.

```

conv_pw_13 (Conv2D) (None, 7, 7, 1024) 1044576
conv_pw_13_bn (BatchNormaliz) (None, 7, 7, 1024) 4096
conv_pw_13_relu (ReLU) (None, 7, 7, 1024) 0
global_average_pooling2d_3 (None, 1024) 0
reshape_1 (Reshape) (None, 1, 1, 1024) 0
dropout (Dropout) (None, 1, 1, 1024) 0
conv_pw_14 (Conv2D) (None, 1, 1, 1000) 1021000
reshape_2 (Reshape) (None, 1000) 0
dropout_1 (Dropout) (None, 1000) 0
dense_1 (Dense) (None, 4) 4004
-----
Total params: 4,257,888
Trainable params: 4,235,980
Non-trainable params: 21,888

len(model.layers)
54

```

Gambar 4.10. Model MobileNet setelah *fine tuning*

Setelah dilakukan proses *fine tuning* jumlah parameter yang digunakan berubah menjadi 4.257.868 dengan *total trainable params* sebanyak 4.235.980 dan *non trainable params* sebanyak 21.888 params.

4.2.3 Skenario Percobaan

Untuk menghasilkan hasil yang diharapkan, diperlukan beberapa skenario percobaan yang akan diterapkan untuk ketiga arsitektur. Dalam penelitian ini penulis menentukan 4 skenario utama yang dapat dilihat pada Tabel 4.2 berikut.

Tabel 4.2. Skenario percobaan

No	Skenario	Keterangan
1	S1	Arsitektur + Gaussian Filter + Thresholding + CLAHE

Tabel 4.2. Skenario percobaan

2	S2	Arsitektur + Gaussian Filter + Thresholding
3	S3	Arsitektur + Thresholding + CLAHE
4	S4	Arsitektur + Thresholding

Empat skenario yang ditunjukkan pada Tabel 4.2 akan diterapkan terhadap ketiga arsitektur yang telah penulis sebutkan sebelumnya. Setiap skenario percobaan di atas akan diujikan dalam pembelajaran data menggunakan *epoch* 50 dan 100 dengan rasio pembagian data latih sebesar 80% dan 20% untuk data uji. Dalam penelitian ini penulis ingin mengetahui pengaruh penggunaan *gaussian filter* terhadap nilai akurasi dari model selain itu dengan penggunaan CLAHE apakah mampu meningkatkan akurasi klasifikasi dari model atau tidak.

4.2.4 *Preprocessing Data*

Sebelum dataset digunakan sebagai umpan masukan dalam tahap pembelajaran, diperlukan satu tahapan yaitu *preprocessing data*. Tujuan dari tahap ini adalah untuk mengolah dataset dengan beberapa perlakuan termasuk beberapa skenario di atas, sehingga dataset citra wayang punakawan telah siap untuk diproses ke dalam arsitektur CNN. Pada penelitian ini preprosesing data yang wajib diterapkan terhadap keempat skenario di atas adalah proses augmentasi data, yang berfungsi untuk meningkatkan jumlah dataset yang digunakan pada proses training.

```

for i in dirs: #loop all directory
    count = 0
    for pic in glob.glob(PATH+i+'/*'):
        im = cv2.imread(pic)
        im = cv2.resize(im,(224,224))
        #get green channel
        green_image = im.copy()
        green_image[:, :, 0] = 0
        green_image[:, :, 2] = 0

```

Selain proses augmentasi, pada penelitian ini *preprocessing* yang wajib dilakukan dan diberlakukan sama terhadap semua skenario percobaan adalah menseragamkan ukuran citra menjadi ukuran 224 x 224 piksel. Selain itu channel warna dari citra RGB yang digunakan pada penelitian ini menggunakan channel warna hijau atau *green channel* yang dapat dilihat pada potongan kode di atas.

4.2.4.1 Preprocessing Skenario 1

Setelah citra wayang dilakukan konversi ukuran citra wayang menjadi ukuran 224 x 224 piksel dan perubahan citra ke dalam *green channel*, tahap selanjutnya adalah *preprocessing* citra berdasarkan dari skenario masing-masing.

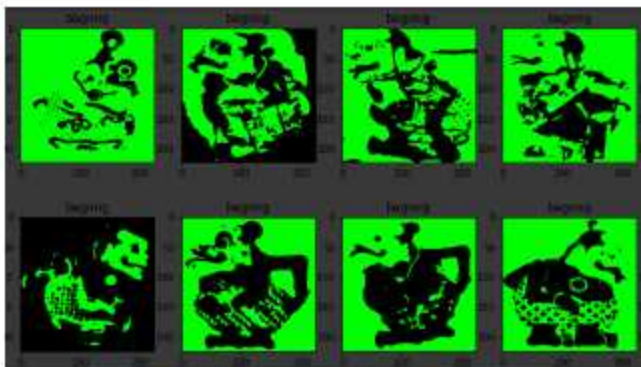
```

lab_im = cv2.cvtColor(green_image , cv2.COLOR_BGR2LAB)
l,a,b =cv2.split(lab_im)
clahe = cv2.createCLAHE(clipLimit=3.0, tileGridSize=(8,8))
clahe_im = clahe.apply(l)
updated_lab_im2 = cv2.merge((clahe_im,a,b))
CLAHE_im = cv2.cvtColor(updated_lab_im2, cv2.COLOR_LAB2BGR)
gaus = cv2.GaussianBlur(CLAHE_im,(5,5),cv2.BORDER_DEFAULT)
ret,thresh = cv2.threshold(gaus,127,255,cv2.THRESH_BINARY)
im = np.array(thresh)

```

Pada skenario satu ini dilakukan proses penambahan CLAHE, lalu citra diproses dengan filter *Gaussian* dan dilakukan proses *thresholding* seperti pada

potongan kode di atas. Proses *preprocessing* pada penelitian ini dilakukan dengan menggunakan *library* open cv yang dijalankan pada *google colab*. Hasil pengolahan pada skenario ini akan menghasilkan citra seperti yang ditunjukkan pada Gambar 4.11.



Gambar 4.11. Hasil *preprocessing* skenario 1

4.2.4.2 Preprocessing Skenario 2

Preprocessing yang dilakukan pada skenario 2 ini hampir sama dengan skenario 1, hanya berbeda pada penambahan proses CLAHE seperti pada potongan kode di bawah ini.

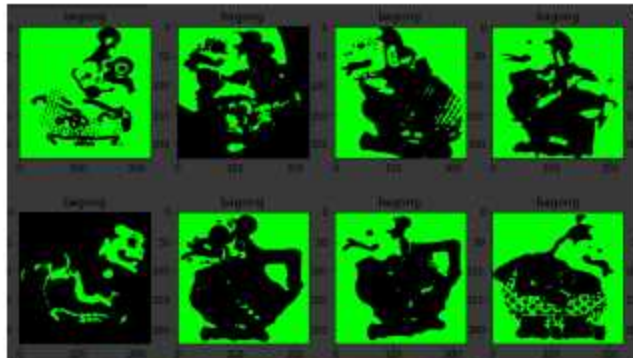
```

gaus = cv2.GaussianBlur(green_image,(5,5),cv2.BORDER_DEFAULT)
ret,thresh = cv2.threshold(gaus,127,255,cv2.THRESH_BINARY)
im = np.array(thresh)

```

Pada skenario 2 citra wayang tidak dilewatkan proses CLAHE, akan tetapi dari *green channel* langsung diolah melalui filter *gaussian* dan *thresholding*. Pada semua skenario akan dilewatkan pada proses segmentasi gambar yaitu pada proses

thresholding. Hasil pengolahan dataset pada skenario 2 ini dapat dilihat pada Gambar 4.12.



Gambar 4.12. Hasil *preprocessing* skenario 2

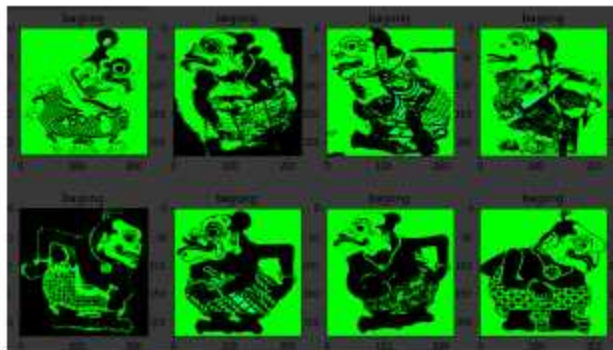
4.2.4.3 Preprocessing Skenario 3

Pada proses pengolahan data (*preprocessing*) terhadap skenario 3 ini tidak dilewatkan proses filterisasi menggunakan gaussian filter.

```
lab_im = cv2.cvtColor(green_image , cv2.COLOR_BGR2LAB)
l,a,b =cv2.split(lab_im)
clahe = cv2.createCLAHE(clipLimit=3.0, tileGridSize=(8,8))
clahe_im = clahe.apply(l)
updated_lab_im2 = cv2.merge((clahe_im,a,b))
CLAHE_im = cv2.cvtColor(updated_lab_im2, cv2.COLOR_LAB2BGR)
ret,thresh = cv2.threshold(CLAHE_im,127,255,cv2.THRESH_BINARY)
im = np.array(thresh)
```

Potongan kode di atas merupakan proses pengolahan dataset pada skenario 3. Citra inputan sudah dalam kondisi *green channel* yang akan dilewatkan pada proses CLAHE dan kemudian akan disegmentasi dengan metode *thresholding*.

Berikut adalah hasil pengolahan citra wayang punakawan pada skenario 3 dapat dilihat pada Gambar 4.13.



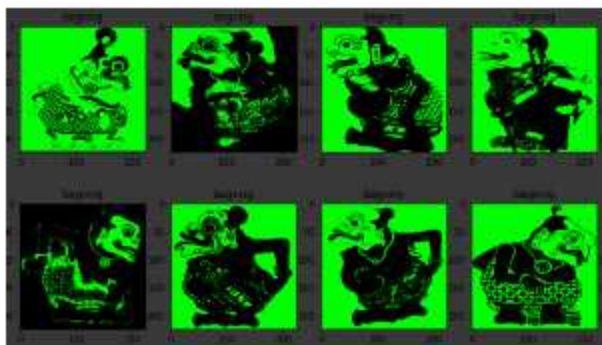
Gambar 4.13. Hasil *preprocessing* skenario 3

4.2.4.4 *Preprocessing* Skenario 4

Proses *preprocessing* yang dilakukan pada skenario 4 ini hampir sama dengan skenario 3 dengan perbedaan pada skenario ini tidak menggunakan CLAHE. Berikut adalah potongan kode *preprocessing* skenario 4.

```
ret,thresh = cv2.threshold(green_image,127,255,cv2.THRESH_BINARY)
im = np.array(thresh)
```

Pada potongan kode di atas, masukan citra pada tahap segmentasi berupa citra green channel tanpa melalui proses penambahan CLAHE maupun filter gaussian. Berikut adalah hasil *preprocessing* pada skenario 4 dapat dilihat pada Gambar 4.14.



Gambar 4.14. Hasil *preprocessing* skenario 4

Pada penelitian ini dilakukan penggandaan data gambar pada data latih melalui proses augmentasi. Berikut adalah potongan kode proses augmentasi citra.

```
datagen = ImageDataGenerator(
    rescale=1. / 255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True)
```

Potongan kode di atas merupakan skrip untuk melakukan augmentasi citra pada data latih. Pada penelitian ini menggunakan bantuan *package* ImageDataGenerator dari *library* Keras. Proses augmentasi dilakukan pada 4 metode, yaitu citra asli akan dilakukan skala ulang dengan besar skala adalah nilai dari parameter *rescale* dikalikan dengan ukuran citra. Selanjutnya citra dilakukan metode *shear_range*, yaitu melakukan pergeseran sudut citra asli sejauh nilai yang diberikan. Proses augmentasi yang ketiga yaitu proses *zoom* pada citra asli sebesar nilai yang diberikan pada parameter *zoom_range*. Metode augmentasi yang terakhir digunakan adalah *horizontal_flip* yaitu membalik citra secara horizontal. Dari proses augmentasi ini akan didapatkan tambahan data latih sebanyak 4 kali dari

total data latih, yaitu untuk masing-masing kelas sebanyak $80 \times 4 = 320$ data. Penentuan penggunaan parameter augmentasi harus bijaksana dan tepat untuk masing-masing jenis dataset, tidak berarti jika semua parameter augmentasi digunakan akan menghasilkan nilai akurasi yang tinggi dengan jumlah dataset yang semakin banyak.

4.2.5 Proses Pembelajaran Data

Setelah dataset dilakukan proses *preprocessing* dataset citra telah siap digunakan untuk proses pembelajaran dalam membentuk model klasifikasi menggunakan ketiga arsitektur di atas. Proses pembelajaran atau *training* pada penelitian ini dilakukan melalui beberapa tahapan, diantaranya yaitu tahap ekstraksi fitur dan klasifikasi. Proses ekstraksi fitur pada CNN dilakukan pada layer konvolusi dan *pooling layer*. Sedangkan proses klasifikasi ditangani oleh *fully connected layer*. Berikut akan dijelaskan proses perhitungan pada *convolutional layer*, *pooling layer* dan proses klasifikasi pada *fully-connected layer* menggunakan *library keras*.

4.2.5.1 Proses Konvolusi

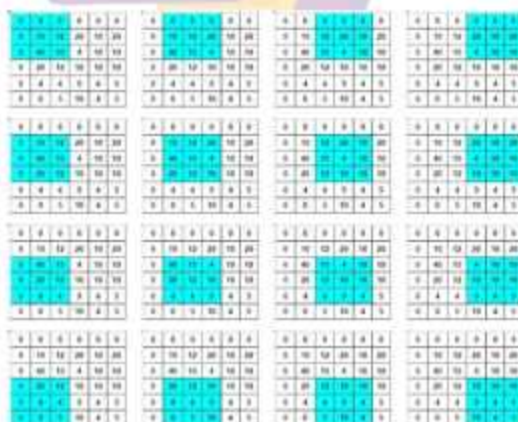
Proses konvolusi yang terjadi pada *convolutional layer* berfungsi dalam mengekstrak fitur-fitur penting yang ada di dalam citra. Proses ini menggunakan operasi perkalian matriks antara matriks citra dengan matriks kernel/filter. Gambar dapat diolah oleh mesin/sistem dengan cara mengubahnya menjadi kumpulan 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.

0	0	0	0	0	0	×	1	0	-1	=	-28	16	8	6
0	10	12	20	10	20		2	0	2		-54	72	14	-12
0	40	15	4	10	10		1	0	1		43	51	5	2
0	20	12	10	10	10						-25	-10	3	10
0	4	4	9	4	5									
0	0	5	10	4	5									

Gambar 4.15. Proses konvolusi

Pada Gambar 4.15 ditunjukkan contoh/sampel perhitungan operasi konvolusi. Pada sampel ini penulis menggunakan 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 konvolusi angka-angka tersebut berkisar antara 0 – 255.



Gambar 4.16. Pergerakan proses konvolusi

Pada Gambar 4.16 diperlihatkan pergerakan pada proses konvolusi terhadap matriks 6 x 6 di atas dengan stride yang digunakan adalah 1. Setiap blok matriks pada citra tersebut akan dikalikan dengan matriks kernel yang berukuran sama yaitu 3 x 3. Kernel/filter yang digunakan penulis sebagai 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.

$$\begin{aligned}
 P1 &= (0*1) + (0*0) + (0*(-1)) + (0*2) + (10*0) + (12*(-2)) + (0*1) + (40*0) + (15*(-1)) = -39 \\
 P2 &= (0*1) + (0*0) + (0*(-1)) + (10*2) + (12*0) + (20*(-2)) + (40*1) + (15*0) + (4*(-1)) = 16 \\
 P3 &= (0*1) + (0*0) + (0*(-1)) + (12*2) + (20*0) + (10*(-2)) + (15*1) + (4*0) + (10*(-1)) = 9
 \end{aligned}$$

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

4.2.5.2 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		
-54	72	14	-12	=	72
-43	51	9	-2		14
-25	-10	3	13		51
					13

Gambar 4.17. Proses *Pooling Layer*

Pada Gambar 4.17 ditunjukkan proses pooling terhadap citra hasil proses konvolusi sebelumnya tadi. Proses pooling yang ada pada gambar di atas menggunakan jumlah filter adalah 2×2 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 4.17 yaitu dengan ukuran 2×2 .

4.2.5.3 Proses Klasifikasi

Proses klasifikasi citra ditangani oleh *fully connected layer*. Citra hasil pengolahan pada tahap sebelumnya yaitu pada proses ekstraksi fitur masih berbentuk matriks multidimensi. Sedangkan pada layer klasifikasi ini tidak dapat menerima inputan berupa matriks multidimensi yang berbentuk array, sehingga perlu dilakukan proses *flatten*. Fungsi *flatten* ini akan 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 (karena kelas lebih dari 2) yang akan bertugas dalam

menentukan kelas dari setiap citra yang diinputkan berdasarkan tingkat probabilitas yang paling mendekati dengan kelas citra.

4.2.5.4 Proses Pengujian Model

Penelitian ini menggunakan *Confusion Matrix* sebagai teknik/metode dalam menguji kinerja dari model yang telah dibuat. Dalam metode *Confusion Matrix* ini dapat digunakan dalam mencari nilai akurasi, presisi dan recall dari model klasifikasi yang telah dibuat. Berikut penulis akan menunjukkan contoh melakukan perhitungan pada *Confusion Matrix*. Untuk dapat menghitung nilai akurasi, presisi dan recall diperlukan sebuah tabel *Confusion Matrix* untuk memudahkan dalam perhitungannya. Pada kasus kali ini penulis akan menggunakan jumlah kelas yang sesuai dengan kelas pada penelitian ini, yaitu 4 kelas. Tabel *Confusion Matrix* dapat dilihat pada Tabel 4.3.

Tabel 4.3. Tabel *Confusion Matrix*

		Prediksi			
		Bagong	Gareng	Petruk	Semar
Aktual	Bagong	24	0	2	0
	Gareng	2	12	0	4
	Petruk	1	0	17	0
	Semar	0	2	1	15

Dalam menentukan nilai akurasi, presisi dan recall diperlukan 3 istilah yaitu TP (*True Positive*) yaitu kelas aktual yang diprediksi dengan benar ditunjukkan pada Tabel 4.3 pada kolom yang berwarna hijau. Kemudian terdapat istilah FP (*False Positive*), sebagai contoh sampel FP untuk kelas Bagong ditunjukkan pada

kolom berwarna biru dan FN (*False Negatif*) untuk kelas Bagong ditunjukkan pada kolom berwarna kuning.

1. Akurasi

Untuk menentukan nilai akurasi model melalui tabel *Confusion Matrix* dibutuhkan total nilai TP yang dibagi dengan jumlah data latih. Seperti pada Tabel 4.3 jumlah nilai TP adalah $24 + 12 + 17 + 15 = 68$ dan jumlah data latih yang digunakan adalah 80. Sehingga nilai akurasi nya adalah $68/80 = 0.85$ atau 85%.

2. Presisi

Rumus yang digunakan dalam menentukan nilai presisi adalah $TP/(TP + FP)$. Dalam menentukan nilai presisi pada *multi class* sedikit berbeda dengan *binary class*. Pada kelas banyak diperlukan perhitungan presisi terhadap masing-masing kelas, kemudian hasil dari perhitungan tersebut dijumlahkan dan dibagi dengan jumlah kelas. Mengacu pada Tabel 4.3 adalah nilai presisinya :

a. Nilai presisi kelas Bagong

$$P(\text{bagong}) = 24/(24+2+1+0) \Rightarrow 24/27 \Rightarrow 0.88$$

b. Nilai presisi kelas Gareng

$$P(\text{gareng}) = 12/(12+0+2+0) \Rightarrow 12/14 \Rightarrow 0.86$$

c. Nilai presisi kelas Petruk

$$P(\text{petruk}) = 17/(17+2+0+1) \Rightarrow 17/20 \Rightarrow 0.85$$

d. Nilai presisi kelas Semar

$$P(\text{semar}) = 15/(15+0+4+0) \Rightarrow 15/19 \Rightarrow 0.79$$

Sehingga nilai presisinya adalah $(0.88 + 0.86 + 0.85 + 0.79) / 4 \Rightarrow 0.845$.

3. Recall

Rumus yang digunakan dalam menentukan nilai recall adalah $TP/(TP+FN)$. Pada penentuan recall ini dilakukan sama dengan penentuan presisi, yaitu diperlukan perhitungan recall terhadap masing-masing kelas dan dilakukan proses penjumlahan dan pembagian dengan jumlah kelasnya.

a. Nilai recall kelas Bagong

$$R(\text{bagong}) = 24/(24+0+2+0) \Rightarrow 24/26 \Rightarrow 0.92$$

b. Nilai recall kelas Gareng

$$R(\text{gareng}) = 12/(12+2+0+4) \Rightarrow 12/18 \Rightarrow 0.66$$

c. Nilai recall kelas Petruk

$$R(\text{petruk}) = 17/(17+1+0+0) \Rightarrow 17/18 \Rightarrow 0.94$$

d. Nilai recall kelas Semar

$$R(\text{semar}) = 15/(15+0+2+1) \Rightarrow 0.83$$

Sehingga nilai recall yang diperoleh adalah $(0.92+0.66+0.94+0.83)/4 \Rightarrow 0.8375$.

4.3 Analisis Hasil Penelitian

Setelah dilakukan pengolahan data (*preprocessing*) dan proses pembelajaran model terhadap masing-masing skenario percobaan, diperoleh beberapa model klasifikasi sebanyak skenario yang digunakan. Model yang

dihasilkan merupakan representasi pengetahuan terhadap pembelajaran citra wayang punakawan. Algoritma CNN berhasil mengidentifikasi jenis wayang dari proses pembelajaran data wayang berdasarkan ciri masing-masing kelas. Proses identifikasi citra dilakukan pada layer konvolusi, di mana pada layer ini dilakukan ekstraksi fitur/ciri dari citra yang kemudian dilakukan pengecilan dimensi pada proses pooling layer agar ciri pada citra semakin terlihat perbedaannya.

Setelah didapatkan fitur pada citra, tahap selanjutnya akan dilakukan proses pengenalan dan pengidentifikasian pada layer fully connected layer yang akan menentukan kelas dari citra wayang punakawan. Model tersebut kemudian dilakukan pengujian menggunakan data uji untuk mengetahui tingkat akurasi dan kinerja dari masing-masing model. Berikut adalah hasil pengujian terhadap model klasifikasi dapat dilihat pada Tabel 4.4.

Tabel 4.4. Hasil percobaan

Epoch 50							
ResNet50							
Skenario	Time (s)	Training Loss	Testing Loss	Training Acc	Testing Acc	Precision	Recall
S1	348.49	0.0103	2.1994	0.9979	0.4042	0.3200	0.3833
S2	330.40	0.0086	2.0538	0.9990	0.4042	0.3400	0.4133
S3	360.29	0.0047	1.3540	0.9990	0.4875	0.5333	0.4800
S4	364.40	0.0110	1.1563	0.9979	0.4917	0.5867	0.4967
VGG-16							
Skenario	Time (s)	Training Loss	Testing Loss	Training Acc	Testing Acc	Precision	Recall
S1	521.32	0.0041	0.2638	0.9990	0.9167	0.9200	0.9167
S2	479.53	0.0041	0.6366	0.9990	0.8708	0.8767	0.8700

Tabel 4.4. Hasil Percobaan

S3	492.48	0.0074	0.0828	0.9979	0.9667	0.9667	0.9633
S4	470.80	0.0005	0.3605	0.9990	0.9042	0.9067	0.9033
MobileNet							
Skenario	Time (s)	Training Loss	Testing Loss	Training Acc	Testing Acc	Precision	Recall
S1	468.95	0.0178	1.0483	0.9917	0.8375	0.8533	0.8367
S2	419.65	0.0072	1.1279	0.9979	0.8042	0.8133	0.8067
S3	302.21	0.0091	0.3515	0.9969	0.9250	0.9233	0.9233
S4	335.32	0.0181	1.0292	0.9958	0.8833	0.8867	0.8833
Epoch 100							
ResNet50							
Skenario	Time (s)	Training Loss	Testing Loss	Training Acc	Testing Acc	Precision	Recall
S1	602.51	0.0032	0.6038	0.9990	0.7958	0.8000	0.7967
S2	527.01	0.0046	1.0051	0.9990	0.7083	0.7433	0.7067
S3	622.99	0.0037	0.3990	0.9990	0.8875	0.8900	0.8867
S4	507.11	0.0059	0.6013	0.9979	0.8458	0.8500	0.8433
VGG-16							
Skenario	Time (s)	Training Loss	Testing Loss	Training Acc	Testing Acc	Precision	Recall
S1	649.26	0.0085	0.2562	0.9969	0.9292	0.9367	0.9233
S2	516.83	0.0263	0.5780	0.9917	0.8583	0.8667	0.8500
S3	647.81	0.0002	0.1441	0.9990	0.9625	0.9533	0.9500
S4	639.31	0.0003	0.3304	0.9990	0.9208	0.9167	0.9167
MobileNet							
Skenario	Time (s)	Training Loss	Testing Loss	Training Acc	Testing Acc	Precision	Recall
S1	514.17	0.0098	0.7893	0.9979	0.8333	0.8333	0.8300
S2	475.77	0.0124	1.2851	0.9979	0.8375	0.8433	0.8400

Tabel 4.4. Hasil Percobaan

S3	514.89	0.0115	0.3321	0.9969	0.9250	0.9267	0.9233
S4	513.00	0.0094	0.9967	0.9969	0.8917	0.8933	0.8933

Pada Tabel 4.4 ditunjukkan hasil percobaan semua skenario percobaan terhadap ketiga arsitektur. Dari hasil percobaan yang telah dilakukan, diperoleh nilai akurasi tertinggi dari ketiga arsitektur di atas, baik yang menggunakan epoch 50 maupun 100 adalah arsitektur VGG-16. Dari ketiga arsitektur, VGG-16 yang paling lama proses komputasinya. Percobaan arsitektur ResNet50 pada epoch 50 tidak menunjukkan hasil akurasi yang baik dan baru meningkat setelah dilakukan pelatihan menggunakan epoch 100. Pada arsitektur ResNet50 mengalami proses overfitting pada percobaan dengan epoch 50. Nilai akurasi pada arsitektur MobileNet terlihat stabil pada epoch 50 maupun 100 dan tidak menunjukkan perbedaan yang signifikan, akan tetapi arsitektur ini memiliki waktu komputasi yang paling kecil dan ringan dibandingkan kedua arsitektur yang lain. Untuk melihat perbedaan dari ketiga arsitektur dengan lebih jelas penulis membuat beberapa perbandingan dalam bentuk grafik.

4.3.1 Perbandingan Waktu Komputasi

Dalam penelitian ini dilakukan pencatatan terhadap waktu komputasi yang dibutuhkan dalam proses pembelajaran model untuk setiap skenario percobaan pada ketiga arsitektur. Berikut adalah grafik perbandingan waktu komputasi pada ketiga arsitektur terhadap semua skenario percobaan yang dilakukan pada *epoch* 50 dan 100.



Gambar 4.18. Perbandingan waktu komputasi *epoch* 50

Pada Gambar 4.18 di atas ditunjukkan grafik perbandingan waktu proses dalam percobaan dengan *epoch* 50. Grafik di atas merepresentasikan total waktu komputasi mulai dari proses *preprocessing* hingga proses pembelajaran model. Dari grafik tersebut terlihat bahwa arsitektur VGG-16 memiliki waktu komputasi yang tertinggi.



Gambar 4.19. Perbandingan waktu komputasi *epoch* 100

Pada Gambar 4.19 diperlihatkan waktu proses arsitektur VGG-16 masih tertinggi dibandingkan kedua arsitektur yang lain, akan tetapi pada *epoch* 100 ini

arsitektur ResNet50 hampir mendekati waktu proses dari arsitektur VGG-16, sedangkan arsitektur MobileNet memiliki waktu proses yang paling kecil.

4.3.2 Perbandingan *Training* dan *Testing Loss*

Dalam penelitian ini dilakukan perbandingan nilai loss/error pada saat training dan testing data. Berikut terdapat tiga grafik untuk masing-masing arsitektur.



Gambar 4.20. *Training vs testing loss* ResNet50

Pada Gambar 4.20 diperlihatkan hasil perbandingan antara proses training dan testing pada keempat skenario untuk arsitektur ResNet50. Dari grafik tersebut terlihat bahwa perbedaan antara nilai loss pada proses training dan testing sangat signifikan, terlebih lagi untuk percobaan pada epoch 50. Hal tersebut mengakibatkan terjadinya *overfitting*.



Gambar 4.21. *Training vs testing loss VGG-16*

Nilai loss pada grafik yang ditunjukkan pada Gambar 4.21 masih tinggi pada percobaan *epoch* 50. Skenario 2 memiliki nilai loss tertinggi dibandingkan ketiga arsitektur yang lain. Pada percobaan menggunakan *epoch* 100, nilai error dapat ditekan dengan baik.

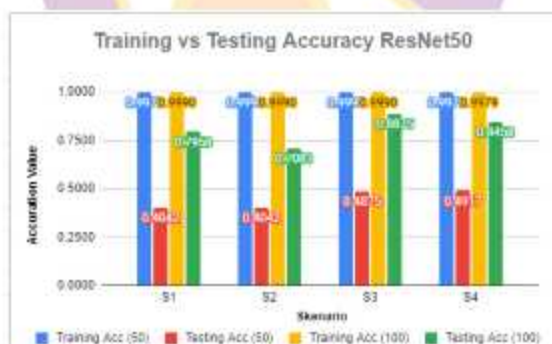


Gambar 4.22. *Training vs testing loss MobileNet*

Pada Gambar 4.22 diperlihatkan nilai loss pada proses pembelajaran model terhadap keempat skenario pada arsitektur MobileNet. Pada grafik di atas sama seperti arsitektur lainnya, nilai loss pada proses pengujian masih tinggi dan skenario 2 memiliki nilai loss tertinggi, sedangkan nilai loss pada skenario 3 memiliki nilai terkecil dan cenderung stabil baik untuk *epoch* 50 maupun 100.

4.3.3 Perbandingan *Training* dan *Testing Accuracy*

Parameter akurasi menjadi perbandingan utama dalam penelitian ini. Nilai akurasi dapat menggambarkan bagus tidaknya suatu model klasifikasi yang nantinya akan digunakan untuk memprediksi data/objek baru.



Gambar 4.23. *Training vs testing accuracy* ResNet50

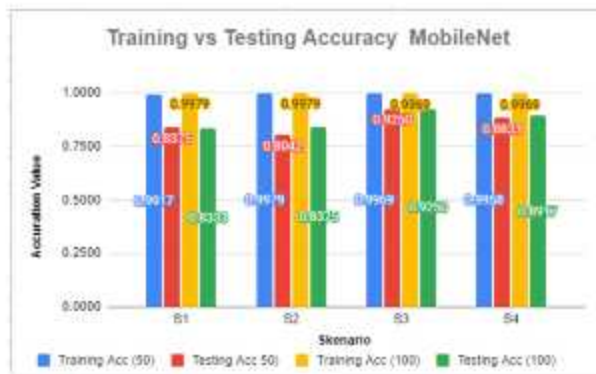
Pada Gambar 4.23 diperlihatkan nilai akurasi dari hasil percobaan terhadap keempat skenario pada arsitektur ResNet50. Pada grafik tersebut terlihat bahwa nilai akurasi pada proses training cenderung sama. Akan tetapi untuk nilai akurasi

pada proses pengujian di *epoch* 50 sangat rendah dan baru meningkat pada penggunaan *epoch* 100 dengan nilai tertinggi sebesar 0.8875 pada skenario 3.



Gambar 4.24. *Training vs testing accuracy VGG-16*

Nilai akurasi dari hasil percobaan menggunakan arsitektur VGG-16 dapat dilihat pada Gambar 4.24 di atas. Pada grafik tersebut nilai akurasi pada proses testing tidak begitu jauh perbedaannya dengan nilai akurasi testing, seperti pada arsitektur ResNet50 tadi yang memperlihatkan perbedaan yang sangat signifikan. Nilai akurasi testing pada *epoch* 50 dan 100 tidak mengalami kenaikan yang signifikan atau dapat dikatakan *epoch* 50 pada arsitektur VGG-16 sudah cukup. Akurasi tertinggi didapatkan pada skenario 3.

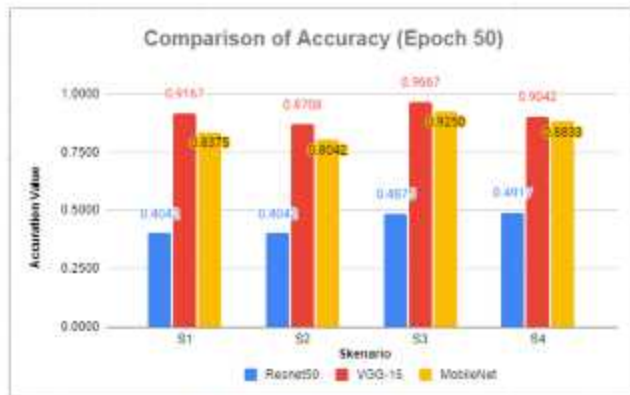


Gambar 4.25. *Training vs testing accuracy MobileNet*

Pada Gambar 4.25 ditunjukkan nilai akurasi pada setiap skenario pada arsitektur MobileNet. Sama seperti arsitektur lain, nilai akurasi proses training pada setiap skenario memiliki nilai yang cenderung stabil. Sedangkan untuk nilai akurasi pada proses pengujian didapatkan bahwa skenario 3 memperoleh nilai akurasi tertinggi baik pada *epoch* 50 maupun 100.

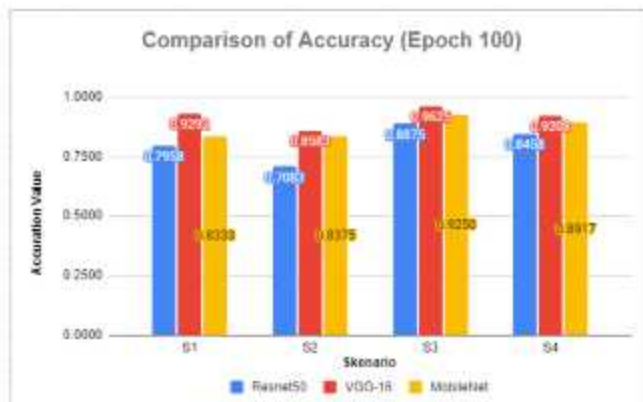
4.3.4 Perbandingan Nilai Akurasi Ketiga Arsitektur

Untuk lebih mempermudah dalam melihat perbandingan antara arsitektur ResNet50, VGG-16 dan MobileNet pada semua skenario, penulis membuat grafik perbandingan ketiga arsitektur tersebut ke dalam satu grafik.



Gambar 4.26. Nilai akurasi *epoch* 50

Pada Gambar 4.26 ditunjukkan gambar grafik perbandingan nilai akurasi dari ketiga arsitektur terhadap semua skenario percobaan pada *epoch* 50. Dari grafik tersebut arsitektur ResNet50 memiliki nilai akurasi terendah di semua skenario percobaan. Akurasi pada arsitektur VGG-16 mengungguli dari kedua arsitektur di semua skenario. Skenario percobaan dengan penambahan CLAHE yaitu skenario 1 dan 3 pada semua arsitektur memiliki nilai akurasi lebih tinggi dibandingkan skenario percobaan yang tanpa menggunakan CLAHE. Skenario percobaan yang terdapat proses filter gaussian memiliki nilai akurasi lebih rendah dibandingkan skenario tanpa filter gaussian.



Gambar 4.27. Nilai akurasi *epoch* 100

Pada Gambar 4.27 ditunjukkan gambar grafik perbandingan nilai akurasi dari ketiga arsitektur terhadap semua percobaan dalam *epoch* 100. Pada percobaan ini, nilai akurasi dari arsitektur ResNet50 meningkat secara drastis pada semua skenario percobaan. Sedangkan untuk arsitektur VGG-16 dan MobileNet perbedaan nilai akurasi dari setiap skenario pada *epoch* 50 dan 100 tidak meningkat secara signifikan yaitu hanya bertambah 2% saja. Bahkan ada beberapa skenario yang tidak mengalami peningkatan nilai akurasi dan justru mengalami penurunan. Arsitektur VGG-16 masih menjadi arsitektur dengan nilai akurasi tertinggi pada semua skenario. Nilai akurasi skenario yang menggunakan CLAHE selalu lebih unggul, dan skenario yang menerapkan filter *gaussian* cenderung lebih rendah dibandingkan arsitektur yang tidak menggunakan filter *gaussian*.

Dari beberapa pembahasan terhadap hasil percobaan di atas relevan dengan penelitian sebelumnya, seperti penelitian yang dilakukan Setiawan (2017) membandingkan 9 arsitektur CNN termasuk di dalamnya adalah VGG16 dan

ResNet50. Dalam penelitiannya tersebut objek yang digunakan adalah dataset fundus. Arsitektur dengan akurasi tertinggi adalah arsitektur VGG16 dengan nilai akurasi 92.31% setelah dilakukan optimasi dengan gradient descent. Pada penelitian ini, dari ketiga arsitektur yang diujikan arsitektur VGG16 memiliki akurasi tertinggi dengan nilai akurasi sebesar 96.25% dengan jumlah epoch sebanyak 50 epoch.

4.3.5 Pengujian Tambahan

Skenario pengujian tambahan pada penelitian ini merupakan masukan dari salah satu dosen penguji untuk mengetahui apakah filter *Gaussian* tidak dapat meningkatkan akurasi pada objek lain selain wayang punakawan. Selain itu, apakah dengan pengujian tambahan terhadap perlakuan citra yang berbeda dari skenario yang telah ditentukan pada penelitian ini juga tidak berpengaruh dalam meningkatkan akurasi klasifikasi. Oleh karena itu, penulis melakukan pengujian tambahan terhadap objek citra lain dan menggunakan skenario tambahan untuk mencari temuan lain dari penelitian ini.

Skenario yang diujikan pada penelitian ini menunjukkan bahwa *Gaussian Filter* kurang efektif dalam meningkatkan nilai akurasi, karena skenario-skenario yang menggunakan filter ini justru mengalami penurunan nilai akurasi. Oleh karena itu peneliti mencoba mengujikan arsitektur terbaik pada penelitian ini yaitu VGG16 dengan dataset lain dan perlakuan dataset yang berbeda. Hasil dari pengujian yang telah peneliti lakukan dapat dilihat pada Tabel 4.5.

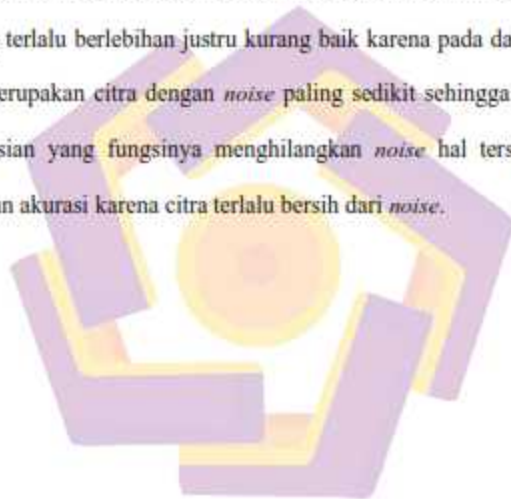
Tabel 4.5. Tabel pengujian tambahan

No	Dataset	Skenario	Nilai Akurasi	Waktu
1	Citra X-Ray (320 data => 2 kelas)	VGG16 + green channel + CLAHE + gaussian filter + thresholding	96.88%	221.48 s
2		VGG16 + green channel + CLAHE ++ thresholding	98.44%	223.23 s
3		VGG16 + CLAHE + gaussian filter + thresholding	98.44%	225.42 s
4		VGG16 + CLAHE + thresholding	96.88%	227.47 s
5	Wayang punakawan (400 data => 4 kelas)	VGG16 + green channel + CLAHE + gaussian filter + thresholding	92.92%	275.38 s
6		VGG16 + green channel + CLAHE ++ thresholding	96.25%	266.88 s
7		VGG16 + CLAHE + gaussian filter + thresholding	98.75%	279.29 s
8		VGG16 + CLAHE + thresholding	97.50%	269.17 s

Seperti yang terlihat pada Tabel 4.5 peneliti melakukan pengujian tambahan terhadap dataset citra X-Ray dengan dua kelas dan dataset wayang dengan perlakuan tambahan yaitu penggunaan warna citra asli yaitu RGB tanpa transformasi citra menjadi *green channel*. Dari hasil percobaan tersebut penggunaan filter gaussian efektif dan berpengaruh terhadap nilai akurasi yang dihasilkan pada skenario penggunaan channel RGB asli dari citra. Sebaliknya pada skenario dengan penggunaan *green channel* akurasi menurun dengan penggunaan

filter gaussian. Seperti pada penelitian sebelumnya oleh Soomro et al. (2018), Informasi pada *red* dan *blue channel* memiliki tingkat noise yang tinggi, sedangkan *green channel* memiliki noise yang paling sedikit.

Sedangkan fungsi dari filter gaussian seperti pada penelitian yang dilakukan Khilmawan (2018) mengatakan bahwa 2 metode filterisasi terbaik dalam menghilangkan noise adalah median dan gaussian filter yang dapat meningkatkan kualitas citra. Dari dua penelitian sebelumnya dapat dikatakan bahwa penghilangan noise yang terlalu berlebihan justru kurang baik karena pada dasarnya citra *green channel* merupakan citra dengan *noise* paling sedikit sehingga ketika diterapkan filter gaussian yang fungsinya menghilangkan *noise* hal tersebut justru dapat menurunkan akurasi karena citra terlalu bersih dari *noise*.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan analisa hasil pada percobaan yang telah dilakukan terhadap ketiga arsitektur dengan 4 skenario percobaan dapat disimpulkan bahwa :

1. Rancangan arsitektur terbaik pada penelitian ini terhadap klasifikasi wayang punakawan adalah arsitektur VGG-16 yang dikombinasikan dengan penambahan CLAHE dan proses segmentasi menggunakan teknik *thresholding*. Nilai akurasi yang didapatkan adalah sebesar 0.9667 atau 96% dengan jumlah *epoch* 50.
2. Penggunaa *gaussian filter* berpengaruh dalam meningkatkan akurasi ketika diterapkan pada citra RGB akan tetapi akan mengalami penurunan akurasi ketika diterapkan pada citra *green channel*.
3. Penggunaan CLAHE dapat meningkatkan nilai akurasi klasifikasi, nilai akurasi pada skenario percobaan yang menggunakan filter *gaussian* meningkat secara signifikan dari 0.8583 setelah ditambahkan CLAHE akurasi meningkat menjadi 0.9292.

5.2 Saran

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

1. 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.
2. Sebaiknya melakukan eksplorasi proses fine tuning pada setiap arsitektur untuk mendapatkan hasil yang lebih maksimal.
3. Sebaiknya menambah variasi skenario percobaan seperti penggunaan variasi ukuran citra dan rasio pembagian data latih dan data uji.
4. 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

- Ahmad, Usman. 2005. *Pengolahan Citra Digital dan Teknik Pemrogramannya*. Edisi Pertama. Yogyakarta: Graha Ilmu.
- Basuki, A., Palandi, J., & Fatchurrohman. 2005. *Pengolahan Citra Digital Menggunakan Visual Basic*. Yogyakarta: Graha Ilmu
- Madenda, S. (2015). *Pengolahan Citra & Video Digital*. Jakarta: Erlangga
- Munir, R., 2004, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Jakarta : Informatika
- Pasha, L. (2011). *Buku Pintar Wayang*. Yogyakarta: Bentang Pustaka
- Santoso, B. 2007. *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu
- Suseno, F. (1991). *Wayang dan Panggilan Manusia*. Jakarta: Gramedia Pustaka Utama
- S.Y. Iriyanto dan T.M. Zaini, *Pengolahan Citra Digital*, Lampung: Anugrah Utama Raharja, 2014

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

- Cheng, C., Wei, X. and Jian, Z. (2017) 'Emotion recognition algorithm based on convolution neural network', *Proceedings of the 2017 12th International Conference on Intelligent Systems and Knowledge Engineering, ISKE 2017, 2018-January*, pp. 1–5. doi: 10.1109/ISKE.2017.8258786
- Chowanda, A. and Sutoyo, R. (2019) 'Deep learning for visual Indonesian place classification with convolutional neural networks', *Procedia Computer Science. Elsevier B.V.*, 157, pp. 436–443. doi: 10.1016/j.procs.2019.08.236
- D. R. Indah. (2019). "The symbolic meaning of ' Punakawan Javanese Wayang ' (a value imaging study in character education at the character education course in STKIP Bina Insan Mandiri Surabaya)", *SELL Journal*, e-ISSN : 2580-8400,pp. 99–106.
- Eka Putra, W. S. (2016) 'Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101', *Jurnal Teknik ITS*, 5(1). doi: 10.12962/j23373539.v5i1.15696.

- Fujino, S. et al. (2019) 'Evolutionary deep learning based on deep convolutional neural network for anime storyboard recognition', *Neurocomputing*. Elsevier B.V., 338, pp. 393–398. doi: 10.1016/j.neucom.2018.05.124
- Guo, T., Dong, J., Li, H. and Gao, Y. (2017) 'Simple Convolutional Neural Network on Image Classification', 2nd International Conference on Big Data Analysis. IEEE, pp. 721-724. doi: 978-1-5090-3619-6
- Hendriyana and Maulana, Y. H. (2020) 'Identifikasi Jenis Kayu menggunakan Convolutional Neural Network', *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 4(1), pp. 70–76
- Hubel, D. and Wiesel, T. (1968). Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology (London)*, 195, 215–243
- Islam, K. T., Raj, R. G. and Al-Murad, A. (2018) 'Performance of SVM, CNN, and ANN with BoW, HOG, and Image Pixels in Face Recognition', 2nd International Conference on Electrical and Electronic Engineering, ICEEE 2017. IEEE, (December), pp. 1–4. doi: 10.1109/ICEEE.2017.8412925
- Khilmawan, M. R. and Riadi, A. A. (2018) 'Implementasi Pengurangan Noise Pada Citra Tulang Menggunakan Metode Median Filter Dan Gaussian Filter', *JIPi (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 3(2), pp. 116–121. doi: 10.29100/jipi.v3i2.865
- K. Fukushima, "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biological Cybernetics*, 1980
- Maulana, F. F. and Rochmawati, N. (2019) 'Klasifikasi Citra Buah Menggunakan Convolutional Neural Network', *Journal of Informatics and Computer Science*, 01, pp. 104–108
- Putra, R. E., Tjandrasa, H. and Suciati, N. (2020) 'Severity classification of non-proliferative diabetic retinopathy using convolutional support vector machine', *International Journal of Intelligent Engineering and Systems*, 13(4), pp. 156–170. doi: 10.22266/IJIES2020.0831.14
- Rokhana, R. et al. (2019) 'Convolutional Neural Network untuk Pendeteksian Patah Tulang Femur pada Citra Ultrasonik B-Mode', *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, 8(1), p. 59. doi: 10.22146/jnteti.v8i1.491
- Sandy, B., Siahaan, J.K., Permana, P. dan Muthahir. (2019) 'Klasifikasi Citra Wayang Dengan Menggunakan Metode K-NN & GLCM', *Prosiding Seminar Nasional Teknologi Informatika*, ISBN: 978-602-50006-1-4, pp. 71–77

- Setiawan, W. (2020) 'Perbandingan Arsitektur Convolutional Neural Network Untuk Klasifikasi Fundus', *Jurnal Simantec*, 7(2), pp. 48–53. doi: 10.21107/simantec.v7i2.6551
- Shin, Y. and Balasingham, I. (2017) 'Comparison of hand-craft feature based SVM and CNN based deep learning framework for automatic polyp classification', *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 3277–3280. doi: 10.1109/EMBC.2017.8037556
- Sokolova, M., & Lapalme, G. (2009). Information Processing and Management. A systematic analysis of performance measures for classification tasks, 427-437
- Tamina, S. (2019) 'Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images', *International Journal of Scientific and Research Publications (IJSRP)*, 9(10), p. p9420. doi: 10.29322/ijstrp.9.10.2019.p9420
- Y. LeCun, "Handwritten Digit Recognition with a BackPropagation Network," 1990
- Zufar, M. and Setiyono, B. (2016) 'Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-time', *Jurnal Sains dan Seni ITS*, 5(2), p. 128862

PUSTAKA ELEKTRONIK

- Ghoneim, S. (2019, April 2). Accuracy, Recall, Precision, F-Score & Specificity, which to optimize on? Retrieved April 2020, from Towards Data Science: <https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124>