PENGEMBANGAN PENGENALAN AKTIVITAS MANUSIA SECARA REAL TIME MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK DAN DEEP GATED RECURRENT UNIT

(SKRIPSI)

Oleh

Feren Ade Verilia

PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI
JURUSAN MATEMATIKAN DAN ILMU PENGETAHUAN ALAM
FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN
UNIVERSITAS LAMPUNG
2023

ABSTRAK

PENGEMBANGAN PENGENALAN AKTIVITAS MANUSIA SECARA REAL TIME MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK DAN DEEP GATED RECURRENT UNIT

Oleh

Feren Ade Verilia

Recurrent neural network (RNN) telah mencapai kesuksesan dalam memproses data sekuensial dan menjadi *state-of-the-art* dalam *speech recognition*, pengenalan sinyal digital, pemrosesan video, dan analisa data teks. Pada penelitian ini, diimplementasikan metode pengenalan aktivitas manusia dengan memproses data video menggunakan *convolutional neural network* (CNN) dan *deep gated recurrent unit* (GRU). Pertama, akan dilakukan pemilihan *frame* dengan cara memilih *frame* dengan urutan kelipatan enam. Hal ini dilakukan untuk mengurangi kompleksitas dan mengurangi fitur yang redundan. Fitur dari sebuah *frame* akan diekstrak menggunakan CNN dengan arsitektur MobileNetV2 yang sudah dilatih pada dataset ImageNet. Kemudian, fitur yang sudah diekstrak menggunakan CNN akan dimasukkan ke GRU dengan tujuan untuk menganalisa fitur spatiotemporal. Hasil penelitian ini dapat mencapai *F1 Score* sebesar 92.01% pada dataset YouTube 11 Actions. Metode ini dapat mencapai kecepatan sebesar 65.43 FPS.

Kata kunci: *Human Activity* Recognition, *Video Analysis*, *Deep Gated Recurrent Unit*, *Convolutional Neural Network*

ABSTRAK

DEVELOPMENT OF REAL TIME HUMAN ACTIVITY RECONGNITION BASED ON CONVOLUTIONAL NEURAL NETWORK AND DEEP GATED RECURRENT UNIT

Oleh

Feren Ade Verilia

Recurrent neural network (RNN) have achieved great success in processing sequential data and yielded the state-of-the-art results in speech recognition, digital signal processing, video processing, and text data analysis. In this thesis, proposed a human action recognition method by processing the video data using convolutional neural network (CNN) and deep gated recurrent unit (GRU) network. First, features are extracted from frame every multiple of six in the videos to helps reduce the redundancy and complexity. Next, the sequential information among frame features is learnt using deep GRU network, where multiple layers are stacked together to increase its depth. The result of this study achieved 92.01% F1 Score on YouTube 11 Actions dataset. This method achieve 65.43 FPS.

Keywords: Human Activity Recognition, Video Analysis, Deep Gated Recurrent Unit, Convolutional Neural Network

PENGEMBANGAN PENGENALAN AKTIVITAS MANUSIA SECARA REAL TIME MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK DAN DEEP GATED RECURRENT UNIT

Oleh

Feren Ade Verilia

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh
SARJANA PENDIDIKAN

pada

Program Studi Pendidikan Teknologi Informasi
Jurusan Pendidikan Matematika dan Ilmu Pengetahuan Alam

FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023

Judul Skripsi	:	**PENGEMBANGAN PENGENALAN AKTIVITAS MANUSIA SECARA REAL TIME MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK DAN DEEP GATED RECURRENT UNIT**
Nama Mahasiswa	:	Feren Ade Verilia
No. Pokok Mahasiswa	:	**1813025014**
Program Studi	:	**Pendidikan Teknologi Informasi**
Fakultas	:	**Keguruan dan Ilmu Pendidikan**

MENYETUJUI

1. Komisi Pembimbing

Dr. Rangga Firdaus, M.Kom.
NIP 19850915 200812 1001

Ing. Hery Dian Septama, S.T.
NIP 197410102008011015

2. Ketua Jurusan Pendidikan MIPA

Prof. Dr. Undang Rosidin, M.Pd.
NIP 19600301 198503 1 003

MENGESAHKAN

1. Tim Penguji

Ketua : **Dr. Rangga Firdaus, M.Kom.** .....

Sekretaris : **Ing. Hery Dian Septama, S.T.** .....

Penguji
Bukan Pembimbng : **Titin Yulianti, S.T., M.Eng.** .....

Dekan Fakultas Keguruan dan Ilmu Pendidikan

Prof. Dr. Sunyono, M.Si.
NIP 19651230 199111 1 001

Tanggal Lulus Ujian Skripsi : **12 Januari 2023**

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama	:	Feren Ade Verilia
NPM	:	1813025014
Fakultas/ Jurusan	:	Keguruan dan Ilmu Pendidikan/ Pendidikan PMIPA
Program Studi	:	Pendidikan Teknologi Informasi
Alamat	:	Perumas Bukit Kemiling Permai Blok J No.68, Kemiling, Kemiling Permai, Bandar Lampung, Lampung.

Menyatakan bahwa dalam skripsi saya yang berjudul "Pengembangan Pengenalan Aktivitas Manusia Secara Real Time Menggunakan Metode Convolutional Neural Network Dan Deep Gated Recurrent Unit" merupakan karya sendiri dan bukan hasil karya orang lain. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila dikemudian hari terbukti bahwa skripsi ini merupakan hasil salinan atau dibuat orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan Akademik yang berlaku.

Bandarlampung, 30 Januari 2023

Feren Ade Verilia
NPM 1813025014

RIWAYAT HIDUP

Penulis dilahirkan di Kecamatan Panjang, Kota Bandar Lampung, Lampung pada tanggal 18 Februari 1999. Penulis merupakan anak ketiga dari lima bersaudara, dari Bapak Ohmbrizal dan Ibu Warni.

Pendidikan awal yang penulis tempuh adalah pendidikan sekolah dasar di SD Negeri 3 Kemiling Permai yang diselesaikan pada tahun 2012, melanjutkan di SMP Negeri 28 Bandar Lampung pada tahun 2015, dan di SMA Negeri 14 Bandar Lampung pada tahun 2018.

Pada tahun 2018, penulis terdaftar sebagai mahasiswa Program Studi Pendidikan Teknologi Informasi Jurusan Pendidikan MIPA FKIP Unila melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis pernah menjabat sebagai Sekretaris Divisi Dana dan Usaha (Danus) di Forum Mahasiswa Pendidikan Teknologi Informasi (FORMATIF). Pada tahun 2021, penulis melaksanakan Praktik Industri di Badan Pengelolaan Keuangan dan Aset Daerah (BPKAD) Kabupaten Pesawaran.

MOTTO HIDUP

“ Dan hanya kepada Tuhanmulah hendaknya kamu berharap.”
(Q.S. Al-Insyirah: 8)

“ Barangsiapa mengerjakan kebajikan,baik laki-laki maupun perempuan dalam keadaan beriman, maka pasti akan kami berikan kepadanya kehidupan yang baik dan kami beri balasan dengan pahala yang lebih baik dari apa yang telah mereka kerjakan.”
(Q.S. An Nahl: 97)

“ Boleh jadi kamu membenci sesuatu padahal ia amat baik bagimu, dan boleh jadi pula kamu menyukai sesuatu padahal ia amat buruk bagimu, Allah mengetahui sedang kamu tidak mengetahui.”
(Q.S. Al-Baqarah: 216)

“ Life is a question and how we live it is our answer.”
(Gary Keller)

“ Percaya jalan Allah itu yang terbaik dan jangan lupa bersyukur.”
(Feren Ade Verilia)

PERSEMBAHAN

Assalamualaikum Warahmatullahi Wabarakatuh

Puji syukur kehadirat Allah Swt yang selalu memberikan limpahan nikmat dan rahmat-Nya dan semoga shalawat selalu tercurahkan kepada Nabi Muhammad SAW. Penulis mempersembahkan karya sederhana ini sebagai tanda bakti kasih tulus yang mendalam kepada:

1. Kedua orang tua tersayang penulis, Ibu Warni dan Bapak Ohmbrizal yang telah sepenuh hati membesarkan, mendidik, mendoakan, dan mendukung segala bentuk perjuangan penulis.
2. Kedua kakak dan kedua adik penulis, Uci Septian Adi Saputra, Vera Septi Dwi Cahyani, Jerry Maliki Dermawan dan Adam Putra, yang selalu memberikan dukungan dan mendoakan penulis.
3. Sahabat seperjuangan skripsi penulis, Dewi Ayu Lestari, Anisa Apriani, Dias Maharani Semedi, Lusi Dwi Wardhani, Trio Mahfuddin dan Okta Vika Putri Siregar.
4. Teman-teman angkatan 2018 Pendidikan Teknologi Informasi.
5. Keluarga besar FORMATIF FKIP Universitas Lampung.
6. Almamater tercinta, Universitas Lampung.

SANWACANA

Assalamualaikum Warahmatullahi Wabarakatuh

Alhamdulillah, puji syukur penulis ucapkan kehadirat Allah Swt, karena atas berkah dan rahmat-Nya, skripsi ini dapat diselesaikan.

Skripsi dengan judul "Pengembangan Pengenalan Aktivitas Manusia Secara Real Time Menggunakan Metode Convolutional Neural Network Dan Deep Gated Recurrent Unit" adalah salah satu syarat untuk memperoleh gelar Sarjana Pendidikan di Universitas Lampung.

Dalam kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Sunyono, M.Si. selaku dekan FKIP Universitas Lampung;
2. Bapak Prof. Dr. Undang Rosidin, M.Pd. selaku Ketua Jurusan Pendidikan Matematika dan Ilmu Pengetahuan Alam;
3. Bapak Dr. Doni Andra, S.Pd., M.Sc. selaku Ketua Program Studi Pendidikan Teknologi Informasi;
4. Bapak Dr. Rangga Firdaus, M.Kom. selaku Pembimbing I, atas kesediaannya untuk memberikan bimbingan, saran, arahan, dan masukan dalam penyelesaian skripsi ini serta terima kasih telah memberikan semangat sebagai Pembimbing Akademik penulis;
5. Bapak Ing. Hery Dian Septama, S.T. selaku Pembimbing II, atas kesediaannya memberikan bimbingan, saran, arahan, dan motivasi dalam proses penyelesaian skripsi ini;
6. Ibu Titin Yulianti, S.T., M.Eng. selaku Pembahas, atas saran dan perbaikan dalam proses penyelesaian skripsi ini;

7. Bapak Wayan Suana, S.Pd., M.Si. yang menjadi Dosen Pembimbing Akademik penulis.
8. Bapak dan Ibu Dosen Program Studi Pendidikan Teknologi Informasi yang telah memberikan ilmu selama berkuliah;
9. Bapak dan Ibu Staff Administrasi Program Studi Pendidikan Teknologi Informasi;
10. Bapak dan Ibu Staff Administrasi FKIP Universitas Lampung;
11. Sahabat penulis, Dewi Ayu Lestari, Anisa Apriani, Dias Maharani Semedi, Lusi Dwi Wardhani, Trio Mahfuddin, dan Ocha Orien Kharunisa.
12. Teman seperjuangan angkatan 2018 Program Studi Pendidikan Teknologi Informasi;
13. Diri sendiri

Penulis berharap semoga kebaikan yang telah diberikan kepada penulis mendapat pahala dari Allah Swt dan semoga skripsi ini dapat bermanfaat.

Bandarlampung, 30 Januari 2023
Penulis,

Feren Ade Verilia

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

v

I. PENDAHULUAN

1.1 Latar Belakang

Pengenalan aktivitas pada rangkaian video adalah masalah yang menantang pada visi komputer karena kemiripan dari konten visual (Nanda, Chauhan, K., & Bakshi, 2017), perubahan sudut pandang untuk aktivitas yang sama, gerakan kamera terhadap pelaku aktivitas, skala dan pose dari *actor*, dan perbedaan perubahan pencahayaan (Soomro, Zamir, & Shah, 2012). Aktivitas manusia sangat beragam, dari aktivitas sederhana melalui tangan dan kaki, hingga aktivitas yang kompleks dan terintegrasi antara tangan, kaki, dan tubuh. Contohnya, gerakan kaki untuk menendang bola adalah gerakan sederhana, sedangkan melompat untuk menyundul adalah gerakan kolektif antara tangan, kaki, kepala, dan tubuh (Herath, Harandi, & Porikli, 2017). Secara umum, aktivitas manusia adalah gerakan dari bagian tubuh dengan berinteraksi dengan objek pada lingkungan. Pada konteks video, aktivitas direpresentasikan menggunakan rangkaian *frame*, di mana manusia dapat memahami dengan mudah dengan menganalisa konten dari beberapa *frame* di suatu rangkaian *frame*. Pada skripsi ini, aktivitas manusia dikenali dengan cara yang mirip dengan pengamatan untuk aktivitas di dunia nyata. Skripsi ini menggunakan GRU untuk mempertimbangkan informasi dari *frame* sebelumnya untuk memahami aktivitas pada video.

Salah satu motivasi yang menarik untuk melakukan riset di bidang pengenalan aktivitas adalah domainnya yang luas dari pengaplikasiannya dalam video pengawasan (Nanda, Sa, Choudhury, Baksh, & Majhi, 2017), robotik, interaksi manusia-komputer (A. Aly, Alghamdi, Salim, & Gutub, 2013), analisa olahraga, permainan video untuk karakter pemain, dan manajemen video web (Ng, Hausknecht, Vijayanarasimhan, O., Monga, & G., 2015). Pengenalan aktivitas menggunakan analisa video membutuhkan komputasi yang besar karena video pendek dapat membutuhkan waktu yang lama karena *frame rate* yang

tinggi. Setiap *frame* memiliki peran penting dalam sebuah video, menyimpan informasi dari rangkaian *frame* yang panjang menjadikan sistem lebih baik. Peneliti telah memberikan banyak solusi seperti gerakan, fitur *space-time* (Schuldt, Laptev, & Caputo, 2004), dan *trajectory* (Murthy & Goecke, 2015). RNN adalah sebuah blok neuron yang terhubung dengan *input* unit, *output* unit, dan memiliki aktivasi pada waktu t, yang dapat memproses rangkaian data. GRU memproses satu elemen pada satu waktu sehingga dapat memodelkan *output* yang terdiri dari rangkaian elemen yang tidak *independent* (Lipton, Berkowitz, & Elkan, 2015). Arsitektur RNN dapat mencari pola tersembunyi di dalam data *time-space* seperti video, audio, dan teks. RNN memproses data dalam sebuah rangkaian pada waktu t mendapatkan *input* dari *hidden state* sebelumnya St-1 dan data baru xt. Data dikalikan dengan *weight* dan ditambah bias dan dimasukkan ke dalam fungsi aktivasi. Karena jumlah komputasi yang besar, semakin mendapatkan rangkaian data lebih banyak atau setelah beberapa *layer* akan semakin mengabaikan efek dari *input* awal yang menyebabkan *vanishing gradient problem*. Solusi dari masalah ini adalah LSTM. Arsitektur LSTM memiliki sel memori, gerbang *input*, gerbang *output*, dan gerbang *forget* yang dapat menjaga *state* dari waktu ke waktu dan gerbang *non linier* yang meregularisasi aliran informasi yang masuk dan keluar dari sel (Greff, Srivastava, Koutník, Steunebrin, & Schmidhuber, 2017). Para peneliti telah menyajikan beberapa jenis LSTM seperti *multi-layer* LSTM dan *bidirectional* LSTM untuk memproses data *sekuensial*. *Deep bidirectional* LSTM (DB-LSTM) juga digunakan dalam pengenalan aktivitas manusia (Ullah, Ahmad, Muhammad, Sajjad, & Baik, 2017). LSTM memiliki komputasi yang cukup besar, sehingga diperkenalkan metode GRU oleh Cho et al 2014 (Cho, et al., 2014). GRU memiliki kemiripan dengan LSTM yaitu menggunakan gerbang yang mengatur aliran informasi, tetapi GRU tidak memiliki sel memori sehingga GRU memiliki komputasi yang lebih efisien dibandingkan dengan LSTM (Chung, Gulcehr, Cho, & Bengio, 2014).

Pada skripsi ini, fitur dari video dianalisa untuk pengenalan aktivitas. Untuk mengurangi fitur *frame* yang redundan dan mengurangi waktu kompuatasi, fitur dari setiap *frame* kelipatan enam akan diekstrak menggunakan *MobileNet-V2* yang sudah di *training* pada *ImageNet* (Sandler, Howard, Zhu, Zhmoginov, & Chen,

2018). Arsitektur selanjutnya adalah GRU untuk mempelajari rangkaian informasi dari *frame* pada video. Untuk mempelajari fitur yang lebih kompleks, digunakan dua *layer* GRU.

Pada skripsi ini, diusulkan metode pengenalan aktivitas manusia dengan memproses data video menggunakan *Convolutional Neural Network* (CNN) dan *Deep Gated Recurrent Unit* (GRU). Dataset yang dipakai YouTube 11 Actions (UCF, 2011).

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam skripsi ini adalah :

1. Bagaimana proses pengenalan aktivitas manusia menggunakan CNN dan GRU?
2. Bagaimana performa CNN dan GRU pada pengenalan aktivitas?
3. Bagaimana kecepatan CNN dan GRU pada pengenalan aktivitas pada komputer tanpa GPU?

1.3 Batasan Permasalahan

Permasalahan yang dibahas pada skripsi ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Dataset yang dipakai YouTube 11 Action
2. Menggunakan bahasa Python 3

1.4 Tujuan Penelitian

Tujuan dari pembuatan skripsi ini, yaitu sebagai berikut:

1. Untuk membangun sebuah sistem pengenalan aktivitas manusia dengan menggunakan metode *Convolutional Neural Network* dan *Deep Gated Recurrent Unit* yang dapat mengenali aktivitas pada data video secara *real time.*

2. Untuk mencari performa terbaik dari model yang telah dibuat dari dataset Youtube 11 Action.
3. Untuk mengetahui rata-rata kecepatan yang telah dihasilkan dari performa terbaik dari model yang telah dibuat dari dataset Youtube 11 Action.

1.5 Manfaat Penelitian

Skripsi ini diharapkan dapat membantu menambah kemampuan yang ada pada pengenalan aktivitas manusia sehingga dapat diimplementasikan pada sistem-sistem yang membutuhkan pengenalan aktivitas manusia secara otomatis seperti video pengawasan.

1.6 Ruang Lingkup

Ruang lingkup penelitian pengembangan ini sebagai berikut:

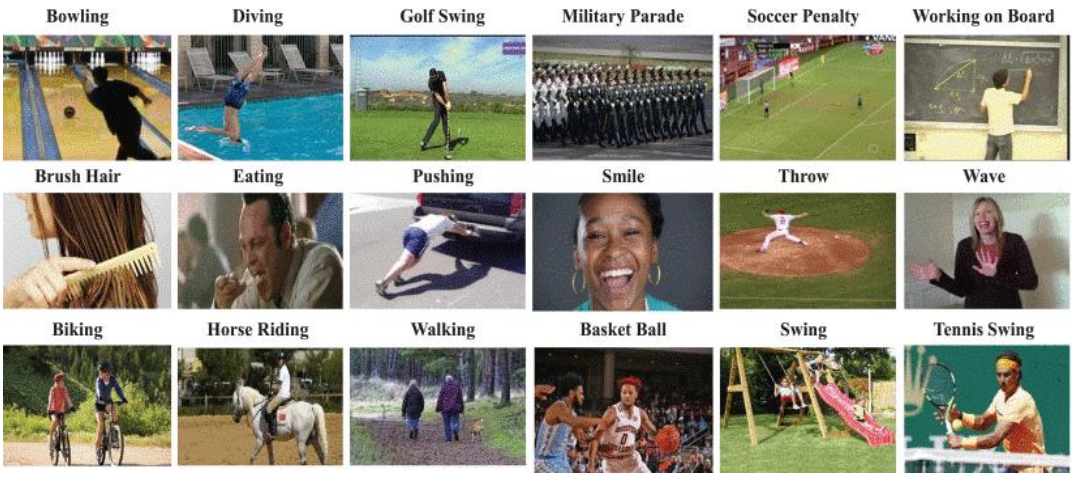
1. Penelitian pengembangan yang dimaksud ialah pengembangan pengenalan aktivitas manusia secara *real time* menggunakan metode CNN dan GRU.
2. Implementasi sistem menggunakan Python 3 sebagai bahasa pemrograman, dilengkapi dengan *library* antara lain Keras, TensorFlow, OpenCV, Numpy dan Matplotib dan perangkat keras menggunakan Google Colaboratory.
3. Tahap pengujian dan evaluasi dilakukan menggunakan 80% data video yang diambil dari dataset Youtube 11 Action yang disimpan pada Google Drive dan sisa data video 20% dijadikan data training. Evaluasi dilakukan dengan mengevaluasi model dari segi akurasi, *precision, recall,* serta *F1 score* untuk melihat performa model yang telah dibuat.

II. TINJAUAN PUSTAKA

Bab ini membahas mengenai teori-teori dasar yang digunakan dalam skripsi. Teori-teori tersebut adalah *Neural Network* dan beberapa teori lain yang mendukung pembuatan skripsi. Penjelasan ini bertujuan untuk memberikan gambaran umum dan diharapkan dapat mendukung sistem yang dibangun.

2.1 Pengenalan Aktivitas Manusia

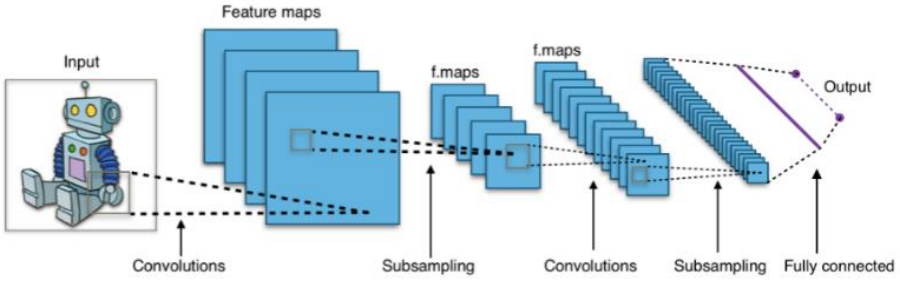
Pengenalan aktivitas manusia adalah salah satu permasalahan yang menantang di bidang kecerdasan buatan. Pengenalan aktivitas manusia berisi kecerdasan buatan yang dilatih untuk mengenali kelas kelas dari aktivitas manusia (Angelini, Fu, Long, Shao, & Naqvi, 2008). Aktivitas manusia berkisar dari aktivitas sederhana lengan atau kaki hingga aktivitas terintegrasi yang rumit antara tangan, kaki, dan tubuh. Misalnya, gerakan kaki untuk berjalan. Secara umum, tindakan manusia adalah gerakan bagian-bagian tubuh dengan berinteraksi dengan benda-benda di lingkungan. Dalam konteks video, suatu tindakan direpresentasikan menggunakan rangkaian *frame*, yang mudah dipahami manusia dengan menganalisis konten beberapa *frame* secara berurutan. Contoh aktivitas manusia dapat dilihat pada gambar 2.1.



Gambar 2.1 Ilustrasi aktivitas manusia (Ullah, Ahmad, Muhammad, Sajjad, & Baik, 2017)

2.2 Convolutional Neural Network (CNN)

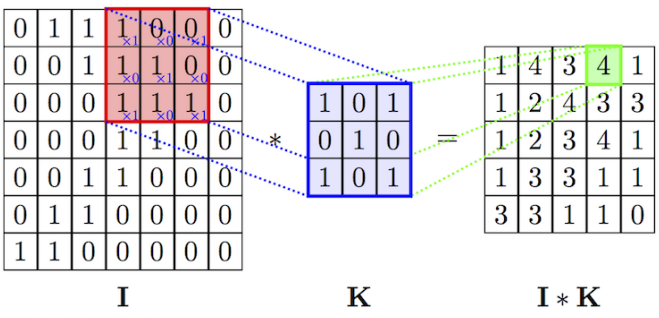
Di dalam ilmu *deep learning*, convolutional neural network (CNN, atau ConvNet) adalah kelas *deep neural network* yang paling sering digunakan untuk menganalisa gambar. CNN juga digunakan pada klasifikasi gambar, analisis gambar medis, dan *natural language processing* (Colloert & Weston, 2008). Contoh arsitektur CNN dapat dilihat pada gambar 2.2.



Gambar 2.2 Contoh arsitektur CNN (Aphex34, 2015)

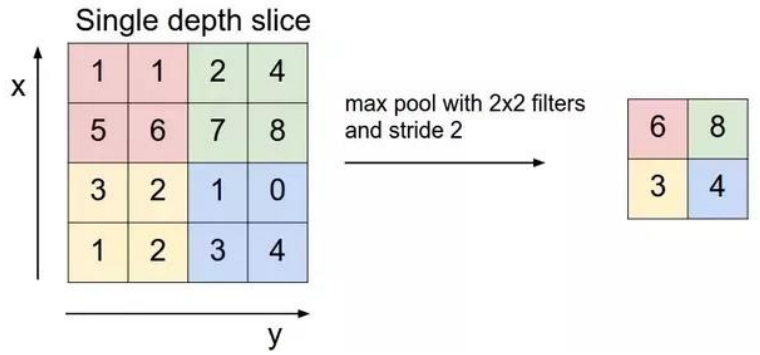
2.2.1 Convolution Layer

Convolution Layer melakukan operasi konvolusi pada *output* dari lapisan sebelumnya. Konvolusi adalah istilah matematis yang artinya mengaplikasikan sebuah fungsi pada *output* fungsi lain secara berulang. Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstrak fitur dari citra masukan (Fadillah, 2017). Ilustrasi cara kerja konvolusi bisa dilihat pada gambar 2.3, dimana I adalah citra, K adalah *filter* atau kernel yang digunakan, I * K adalah hasil operasi konvolusi.



Gambar 2.3 Ilustrasi cara kerja konvolusi (*Deep learning for complete beginners: convolutional neural networks with keras, 2017)*

2.2.2 Pooling Layer



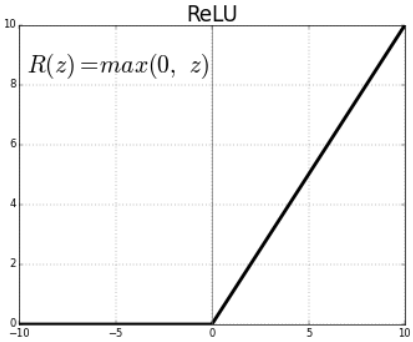
Gambar 2.4 Ilustrasi cara kerja Max Pooling *(An Intuitive Explanation of Convolutional Neural Networks,2016)*

Fungsi dari *Pooling Layer* adalah mereduksi ukuran dari data. Terdapat beberapa tipe *Pooling Layer* diantaranya yaitu *max, average*, *sum* dan lainnya. Metode *Pooling* dalam *Convolutional Neural Network* (CNN) yang biasa digunakan adalah *Max Pooling* & *Average Pooling*. *Max Pooling* membagi *output* dari *Convolution Layer* menjadi beberapa matriks kecil lalu mengambil nilai maksimal dari tiap matriks untuk menyusun matriks citra yang telah direduksi, sedangkan *Average Pooling* akan memilih nilai rata-ratanya. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun obyek citra mengalami translasi. Ilustrasi cara kerja *Max Pooling* bisa dilihat pada gambar 2.4.

2.2.3 Fully Connected Layer

Fully Connected Layer dalam penerapannya sama dengan *Multi Layer Perceptron* (MLP) yang bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara *linear*. *Feature map* dari *Convolution Layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu yang disebut *feature vector* sebelum dapat dimasukkan ke dalam sebuah *Fully Connected Layer*. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak *reversibel*, *Fully Connected Layer* diimplementasikan di akhir jaringan (Suartika, Wijaya, & Soelaiman, 2016).

2.2.4 ReLU Activation Function



Gambar 2.5 *ReLU Activation Function (Sena, 2017)*

Fungsi aktivasi berfungsi untuk menentukan apakah *neuron* tersebut harus aktif atau tidak berdasarkan nilai masukan. Salah satu contoh fungsi aktivasi adalah ReLU (*Rectified Linear Unit*) dimana fungsi ini melakukan *thresholding* dengan nilai nol terhadap nilai masukan, dimana seluruh nilai yang kurang dari nol akan dijadikan nol, seperti pada gambar 2.5.

2.2.5 Fungsi *Softmax*

Fungsi *softmax* biasa digunakan dalam klasifikasi banyak kelas. *Softmax* memberikan nilai *probabilitas* untuk setiap label kelas, dimana jumlah seluruh *probabilitas* adalah 1. Pada saat *softmax* digunakan buat model klasifikasi multi, hingga akan mengembalikan kesempatan dari tiap kelas sehingga kelas sasaran hindak memiliki *probabilitas* yang lebih besar dari kelas lain. Perhitungan *softmax* dengan memakai *eksponensial* dari nilai masukan yang diberikan dan jumlah dari nilai *eksponensial* dari segala nilai dalam masukkan(khaeriyah, 2019). *Softmax* pada dasarnya adalah *probabilitas eksponensial* yang dinormalisasi dari nilai masukan sejumlah kelas pada model klasifikasi seperti pada persamaan (2.5).

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \qquad (2.5)$$

Dimana S adalah softmax, y_i adalah nilai masukan, e^{y_i} adalah fungsi eksponensial standar yang diterapkan pada y_i dan $\sum_j e^{y_j}$ adalah normalisasi.

Operasi akan menghasilkan nilai *probabilitas*. Label dari data masukan akan ditentukan berdasarkan kelas dengan nilai *probabilitas* tertinggi.

2.2.6 Cross Entropy

Loss function merupakan fungsi yang menggambarkan kerugian yang dihasilkan oleh model. *Loss function* dikatakan baik, ketika menghasilkan *error* yang diharapkan paling rendah. Pada permasalahan klasifikasi banyak kelas, *cross entropy* adalah *loss function* yang biasa digunakan. *Cross entropy* akan menghitung *error* antara nilai prediksi *S* dengan nilai sebenarnya *T*, seperti pada persamaan (2.6). Selanjutnya, nilai *error* akhir diambil dari rata-rata hasil *cross entropy*, seperti pada persamaan (2.7).

$$D(S_i,T_i) = -\sum_j T_{ij} \log S_{ij} \qquad (2.6)$$

$$J(W,b) = \frac{1}{n}\sum_i D(S_i,T_i) \qquad (2.7)$$

Dimana D adalah *Cross Entropy*, S adalah nilai prediksi *softmax*, T adalah nilai sebenarnya, (W,b) adalah linear nodel, J adalah *mean square loss* dan $\frac{1}{n}$ adalah untuk menghitung rata-rata.

2.2.7 Stochastic Gradient Descent (SGD)

Ketika melatih sebuah model, dibutuhkan sebuah *loss function* yang dapat mengukur kualitas dari setiap bobot atau parameter tertentu. *Stochastic Gradient Descent* (SGD) adalah algoritma pengoptimalan. Tujuan pengoptimalan adalah untuk menemukan parameter yang dapat meminimalkan nilai *error* dari *loss function.* SGD adalah algoritma yang digunakan untuk memperbarui nilai bobot dan bias pada *neuron* di *neural network.* Pada dasarnya operasi yang dilakukan hanya mengurangi bobot awal dengan sebagian nilai dari nilai gradien yang sudah kita dapat. Nilai sebagian disini diwakili oleh parameter bernama *learning rate*, seperti yang terlihat pada Persamaan (2.8) dan (2.9).

$$w_{j+1} = w_j - \alpha \frac{\partial}{\partial w_j} J(W,b) \qquad (2.8)$$

$$b_{j+1} = w_j - \alpha \frac{\partial}{\partial b_j} J(W, b) \qquad (2.9)$$

Dimana w dan b adalah bobot atau parameter dari fungsi yang ingin dicari nilainya, j adalah 0,1,2,..., dan α adalah *learning rate*.

2.2.8 RMSProp

RMSProp (*Root Mean Square Propagation*) adalah metode pengoptimalan berbasis *adaptive learning rate* yang diusulkan oleh Geoffrey Hinton (Hinton, 2006). *RMSProp* memodifikasi Adagrad dengan mengganti akumulasi gradien menjadi rata-rata bergerak gradien yang diberi bobot secara kuadratik, seperti yang dapat dilihat pada Persamaan (2.10) dan (2.11).

$$s_j = \beta . s_{j-1} + (1 - \beta)(g_j)^2 \qquad (2.10)$$

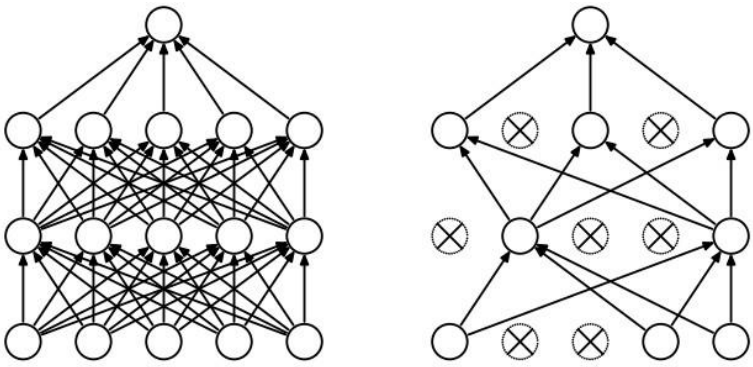
$$\theta_{j+1} = \theta_j - \frac{\alpha}{\sqrt{s_j + \varepsilon}} g_j \qquad (2.11)$$

Dimana s_j adalah rata-rata bergerak gradien kuadrat , β (beta) adalah parameter rata-rata bergerak, g_j adalah gradien dari fungsi biaya sehubungan dengan bobot, θ_j adalah parameter ,dan α adalah kecepatan pembelajaran (*learning rate*).

2.2.9 Dropout

Dropout merupakan proses mencegah terjadinya *overfitting* dan juga mempercepat proses *learning*. *Dropout* mengacu kepada menghilangkan *neuron* yang berupa *hidden* layer mapun *visible layer* di dalam jaringan (Budhiraja). Dengan menghilangkan suatu neuron, berarti menghilangkannya sementara dari jaringan yang ada. *Neuron* yang akan dihilangkan akan dipilih secara acak.

Pada gambar 2.6 (a) *neuron* tetap utuh pada *neural network* yang belum memakai *Dropout*, dan (b) *neural network* yang sebagian dari neuronnya tidak digunakan setelah diaplikasikan *Dropout*.



Gambar 2.6 *Ilustrasi neural network mengaplikasikan Dropout (Karpathy)*

2.2.10 Underfitting dan Overfitting

Underfitting dan *Overfitting* model merupakan perihal yang dapat terjalin kala membuat model CNN. Model yang *underfitting* dan *overfitting* tidak hendak melaksanakan prediksi dengan benar (NarasingaRao, Dr.M.R. & Prasad, 2018). Berikut uraian:

a. Underfitting

Terjadi kala model tidak dapat memandang logika dibelakang informasi, sampai tidak dapat melaksanakan prediksi dengan pas, baik buat dataset *training* ataupun dataset *testing*. Underfitting model hendak mempunyai *loss* besar serta akurasi rendah.

b. Overfitting

Terjadi sebab model yang terbuat sangat fokus pada *training* dataset tertentu, sampai tidak dapat melaksanakan prediksi dengan pas bila diberikan dataset *testing*. *Overfitting* umumnya hendak menangkap informasi *noise* yang sepatutnya diabaikan. *Overfitting* model hendak mempunyai *loss* rendah serta akurasi rendah.

2.2.11 MobileNet-V2

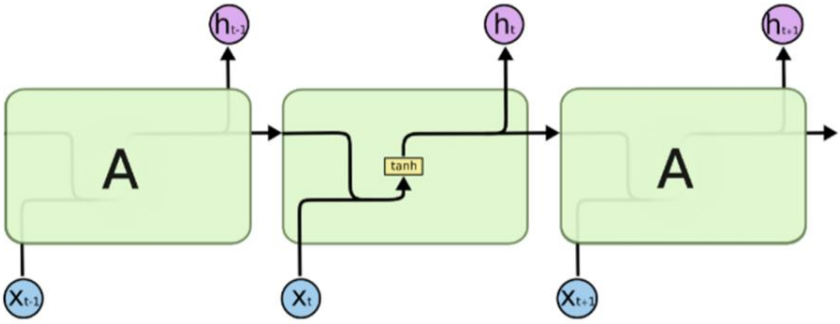
MobileNet adalah arsitektur CNN yang efisien untuk perangkat mobile dan aplikasi *embedded vision MobileNet* menggunakan *depth-wise separable convolution* untuk membangun *deep neural network* yang ringan. *MobileNets* juga efektif dalam banyak aplikasi seperti deteksi objek, klasifikasi gambar, atribut wajah, dan geo-lokalisasi skala besar (Howard, et al., 2017). *MobileNetV2* adalah perkembangan dari *MobileNet* yang berhasil meningkatkan performa model pada beberapa masalah. Alasan menggunakan *MobileNetV2* selain memang *score* akurasinya cukup tinggi, juga yang menjadi keunggulan utamanya ialah jumlah *training parameters* yang kecil dibandingkan dengan arsitektur CNN lainnya, sehingga kebutuhan akan komputasinya lebih ringan. Selain itu *model size MobileNetV2* juga kecil, hanya 14MB saja, namun tentunya dengan performansi yang baik. Sehingga kedepannya jika model tersebut akan di *deploy* kedalam sebuah *real app*, misal aplikasi android ataupun aplikasiberbasis *website* akan ringan dan berukuran kecil. Arsitektur *MobileNetV2* berbasis pada struktur i*nverted residual* di mana *input* dan *output* dari *residual block* adalah *layer bottleneck* yang tipis, berbeda dengan model *residual* tradisional di mana menggunakan *expanded representation* pada *input*. Seperti pendahulunya, *MobileNetV2* menggunakan *depth wise convolution* untuk mengambil fitur. Arsitektur *MobileNetV2* dapat dilihat pada gambar 2.7.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Gambar 2.7 Arsitektur *MobileNetV2*

2.3 Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) adalah salah satu metode pembelajaran mesin yang berfokus pada prediksi dari hasil pola. Secara umum, RNN mengingat masa lalu dan membuat keputusan yang dipengaruhi oleh apa yang telah dipelajari dari masa lalu (Venkatachalam). Selagi RNN belajar dengan cara yang sama saat pelatihan, di samping itu, RNN mengingat hal-hal yang dipelajari dari masukan sebelumnya saat menghasilkan keluaran. RNN dapat mengambil satu atau lebih vektor masukan dan menghasilkan satu atau lebih vektor keluaran. Keluaran tidak hanya dipengaruhi oleh bobot yang diterapkan pada masukan seperti *Neural Network* biasa, tetapi juga oleh vektor status "tersembunyi" yang mewakili konteks berdasarkan masukan atau keluaran sebelumnya. Jadi, masukan yang sama dapat menghasilkan keluaran yang berbeda tergantung pada masukan sebelumnya dalam suatu seri.



Gambar 2.8 Contoh arsitektur jaringan RNN

2.4 Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) adalah metode pengembangan dari RNN yang bekerja sangat baik dalam mengatasi masalah sekuensial (Cho, et al., 2014). GRU memiliki *update gate* dan *reset gate* untuk mengatasi *vanishing gradient problem*. GRU memiliki komputasi yang cukup ringan jika dibandingkan dengan LSTM. Formula dari GRU ke-j pada waktu t adalah:

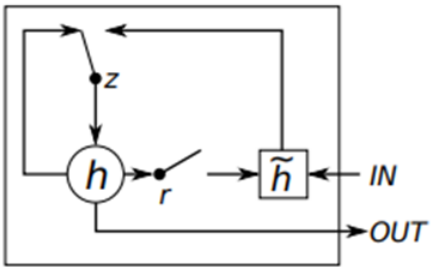
$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j \tilde{h}_t^j \quad (2.1)$$

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1})^j \quad (2.2)$$

$$\tilde{h}_t^j = tanh(Wx_t + U(r_t \odot h_{t-1}))^j \quad (2.3)$$

$$r_t^j = \sigma(W_r x_t + U_r h_{t-1})^j \quad (2.4)$$

W adalah bobot, x_t adalah input pada unit, σ adalah fungsi sigmoid, dan Θ adalah Hadamard product (Chung, Gulcehr, Cho, & Bengio, 2014). Ilustrasi arsitektur GRU dapat dilihat pada gambar 2.9.



Gambar 2.9 Ilustrasi GRU. r dan z adalah *reset gate* dan *update gate*, dan h dan h˜ adalah aktivasi dan kandidat aktivasi

2.5 Confusion Matrix dan *Classification Report*

Confusion Matrix adalah pengukuran kinerja untuk masalah klasifikasi pembelajaran mesin di mana keluaran bisa sebanyak dua atau lebih kelas (Narkhede). *Confusion Matrix* juga merupakan tabel N x N (di mana N adalah jumlah kelas/label/ kategori) yang berisi jumlah prediksi yang benar dan salah dari model klasifikasi, tujuannya yaitu membandingkan nilai aktual dengan nilai prediksi. Berikut adalah tabel dengan 4 kombinasi nilai prediksi dan aktual yang berbeda.

Tabel 2-1 Confusion matrix

		Nilai Aktual	
		Positif	Negatif
Nilai Prediksi	Positif	TP	FP
	Negatif	FN	TN

TP (*True Positive*) adalah ketika yang diprediksi positif dan kenyataannya benar. Contohnya, lelaki yang diprediksi bermain voli dan sesungguhnya benar bermain voli. *TN* (*True Negative*) adalah ketika yang diprediksi negatif dan kenyataannya benar. Contohnya,lelaki yang diprediksi tidak bermain voli dan sesungguhnya benar tidak bermain voli. *FP* (*False Positive*) adalah ketika yang diprediksi positif namun kenyataannya salah. Contohnya, lelaki yang diprediksi bermain voli namun sesungguhnya tidak bermain voli. *FN* (*False Negative*) adalah ketika yang diprediksi negatif namun kenyataannya salah. Contohnya, lelaki yang diprediksi tidak bermain voli namun sesungguhnya bermain voli.

Walaupun hasil yang ditampilkan *Confusion Matrix* telah perinci, tetapi sesungguhnya masih susah buat menguasai seberapa baik kinerja model dalam melaksanakan klasifikasi. Oleh sebab itu, informasi dari *Confusion Matrix* bisa digunakan buat menghitung *matrics* yang bisa mengukur kinerja model, yang setelah itu diucap selaku *Classification Report* (Laurenti, Giulio, 2020). Nilai-nilai dari *confusion matrix* dapat digunakan untuk mengukur tingkat validasi data. Berikut *metrics* digunakan antara lain:

Accuracy yaitu menggambarkan seberapa akurasi model dalam mengklasifikasi dengan benar.(1)

$$Precision = \frac{TP}{TP + FP}$$

Precision yaitu menggambarkan akurasi antara data yang diminta dengan hasil prediksi. (2)

$$Recall = \frac{TP}{TP + FN}$$

Recall yaitu menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. (3)

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

F1 Score yaitu menggambarkan perbandingan rata-rata *Precision* dan *Recal.l* (4)

2.6 Phyton

Python adalah bahasa pemrograman yang populer. Python sering dimanfaatkan dalam pengembangan web, perangkat lunak, penelitian, dan *system scripting*. Python dapat digunakan untuk menangani data besar dan melakukan operasi matematika yang kompleks. Python bekerja di berbagai *platform* seperti Windows, Mac, Linux, Raspberry Pi, dan lain-lain. Python dirancang untuk mudah dibaca, yaitu memiliki sintaks yang sederhana dan menggunakan bahasa inggris (PSF, 2005).

2.7 Library

Library merupakan sekumpulan program yang dapat digunakan pada program lain tanpa terikat satu dengan yang lainnya. Terdapat beberapa *library* yang digunakan dalam melakukan implementasi skripsi ini. *Library* yang digunakan antara lain:

2.7.1 Keras

Keras adalah *high-level neural networks* API, yang ditulis dalam bahasa pemrograman Python dan mampu berjalan di atas TensorFlow dan Theano. Keras dikembangkan dalam rangka memungkinkan eksperiman dilakukan dengan cepat. Keras dapat berjalan baik di CPU dan GPU. Keras berisi banyak implementasi *neural network* yang umum digunakan, fungsi aktivasi, *optimizer*, dan *tool* lain yang memudahkan dalam pengolahan citra dan data teks (Keras.S., 2020).

2.7.2 TensorFlow

TensorFlow adalah *library open source* untuk pembuatan program yang membutuhkan komputasi numerik berkinerja tinggi. TensorFlow dikembangakan oleh tim Google Brain. TensorFlow menyediakan fungsi-fungsi *machine learning* dan *deep learning*, dan dapat dijalankan dalam CPU atau GPU (TensorFlow.J., 2022).

2.7.3 OpenCV

OpenCV (*Open Source Computer Vision*) adalah *library* yang dimanfaatkan dalam pengolahan citra dinamis secara *real-time*.

OpenCV dapat digunakan dalam berbagai bahasa pemrograman seperti Python, C++, Java, atau MATLAB. OpenCV memiliki fitur seperti *Feature & Object Detection, Motion Analysis and Object Tracking, Image Filtering, Image Processing*, dan lain-lain (OpenCV, 2019).

2.7.4 Numpy

Numpy adalah *library Python* yang mendukung pengolahan data pada *array* dan matriks multidimensi yang besar. Numpy menyediakan kumpulan fungsi matematika, seperti aljabar linear, transformasi Fourier, pembuatan angka acak, dan lain-lain. *Numpy* bersifat *open source* sehingga banyak dimanfaatkan dalam pengolahan data penelitian (NumFOKUS, 2005).

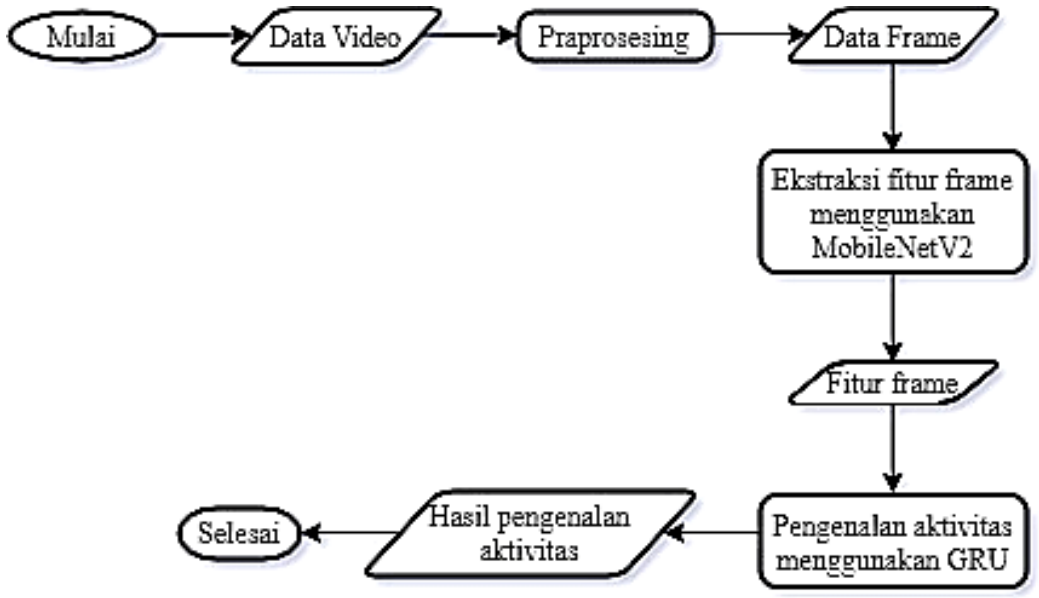
2.7.5 Matplotlib

Matplotlib adalah *library Python* yang mendukung pembuatan grafik dua dimensi dalam berbagai format dan dari berbagai jenis data. Matplotlib bersifat *open source* dan banyak digunakan untuk pengolahan data dalam penelitian. Matplotlib dapat membuat plot, histogram, spektrum daya, diagram batang, diagram kesalahan, plot pencar, dan lain-lain (Hunter.D.J., 2009)

III. METODE PENELITIAN

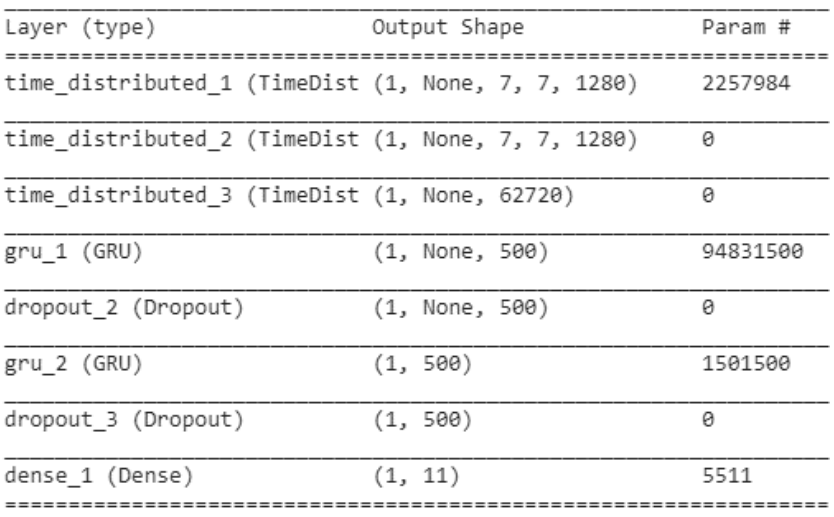
Bab ini menjelaskan tentang perancangan data dan sistem pengenalan aktivitas manusia *MobileNetV2* dan *Gated Recurrent Unit (GRU).* Bab ini juga akan menjelaskan gambaran umum sistem dalam bentuk diagram alir.

3.1 Desain Umum Sistem



Gambar 3.1 Diagram alir sistem yang dibangun

Sistem pengenalan aktivitas manusia yang dibangun memiliki proses utama di antaranya *praprosesing*, *ekstraksi fitur frame*, dan pengenalan aktivitas. Diagram alir dari sistem ditunjukkan pada Gambar 3.1. Arsitektur dapat dilihat pada gambar 3.2.



Layer (type)	Output Shape	Param #
time_distributed_1 (TimeDist	(1, None, 7, 7, 1280)	2257984
time_distributed_2 (TimeDist	(1, None, 7, 7, 1280)	0
time_distributed_3 (TimeDist	(1, None, 62720)	0
gru_1 (GRU)	(1, None, 500)	94831500
dropout_2 (Dropout)	(1, None, 500)	0
gru_2 (GRU)	(1, 500)	1501500
dropout_3 (Dropout)	(1, 500)	0
dense_1 (Dense)	(1, 11)	5511

Gambar 3.2 Arsitektur diagram alir sistem yang dibangun

3.2 Lingkungan Implementasi

Dalam mengimplementasikan aplikasi pengenalan ekspresi manusia diperlukan beberapa perangkat pendukung sebagai berikut.

3.2.1 Perangkat Keras

Implementasi skripsi ini menggunakan Google Colaboratory. Sistem operasi yang digunakan adalah Ubuntu 18.0.4. PC yang digunakan memiliki spesifikasi Intel Xeon dengan kecepatan 2.3 GHz single core, 2 thread per core, dan *Random Access Memory* (RAM) sebesar 13 GB, dan mempunyai *Graphics Processing Unit* (GPU) yaitu NVIDIA Tesla K80 sebesar 12 GB.

3.2.2 Perangkat Lunak

Perangkat lunak memiliki spesifikasi antara lain menggunakan bahasa pemrograman Python 3.6.9, dilengkapi dengan *library* antara lain OpenCV, Tensorflow-GPU, Keras-GPU, Numpy, dan Matplotlib.

3.3 Deskripsi Dataset

Dataset yang digunakan adalah YouTube11 Action. Data yang digunakan adalah data video yang berisi satu aktivitas tertentu. Data dibagi menjadi 11 kelas yang terdiri dari bermain bola basket, bersepeda, lompat indah, bermain golf, naik kuda, sepak bola, bermain ayunan, bermain tenis, melompat, bermain voli, dan berjalan bersama anjing. Video dalam satu subjek memiliki kemiripan fitur seperti aktor yang sama, *background* yang sama, dan sudut pandang yang sama. Dataset memiliki variasi yang besar pada pergerakan kamera, penampakan objek, pose, skala objek, sudut pandang, *background* yang berantakan, dan kondisi pencahayaan yang membuat dataset ini lebih menantang. Contoh dataset dapat dilihat pada gambar 3.3.

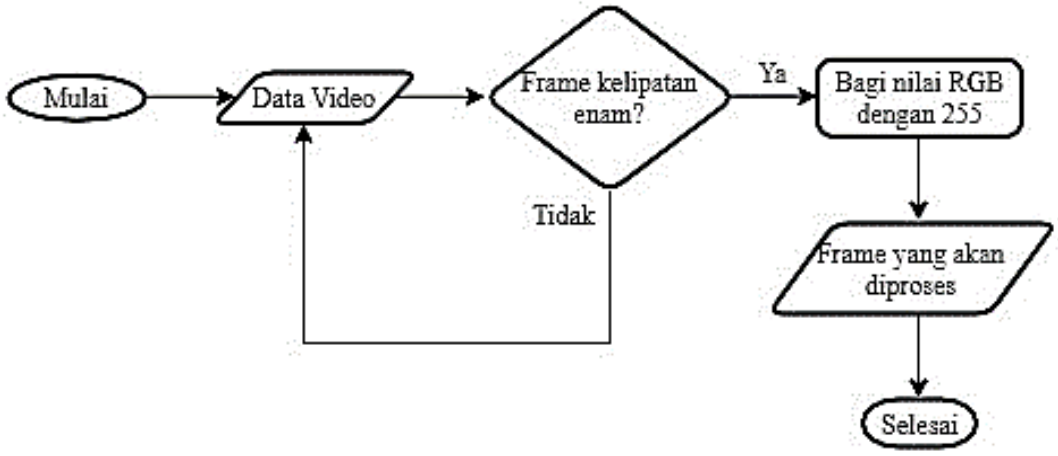
Gambar 3.3 Contoh dataset youtube 11 action (UCF, 2011)

Tabel 3-1 Spessifikasi dataset

No.	Kegiatan dalam Video	Jumlah Video	Ukuran Frame (pixel)
1.	Bermain bola basket	141	Tinggi gambar asli = 214, 240 px Lebar gambar asli = 320 px
2.	Bersepeda	145	Tinggi gambar asli = 239, 240 dan 262 px Lebar gambar asli = 320 px
3.	Lompat Indah	156	Tinggi gambar asli = 240, 256 px Lebar gambar asli = 320 px
4.	Bermain Golf	142	Tinggi gambar asli = 238, 240 px Lebar gambar asli = 320 px
5.	Naik kuda	198	Tinggi gambar asli = 214, 239, 240 dan 256 px Lebar gambar asli = 300, 320 px
6.	Sepak Bola	156	Tinggi gambar asli = 239, 240 px Lebar gambar asli = 320 px
7.	Bermain ayunan	137	Tinggi gambar asli = 240, 262 px Lebar gambar asli = 320 px
8.	Bermain Tenis	167	Tinggi gambar asli = 240, 262 px Lebar gambar asli = 320 px
9.	Melompat	119	Tinggi gambar asli =144, 240 dan 262 px Lebar gambar asli = 176, 320 px
10.	Bermain voli	116	Tinggi gambar asli = 239, 240 dan 256 px Lebar gambar asli = 300, 320 px
11.	Berjalan bersama anjing	123	Tinggi gambar asli = 182, 240 dan 256 px Lebar gambar asli = 320 px

3.4 Tahap Praproses

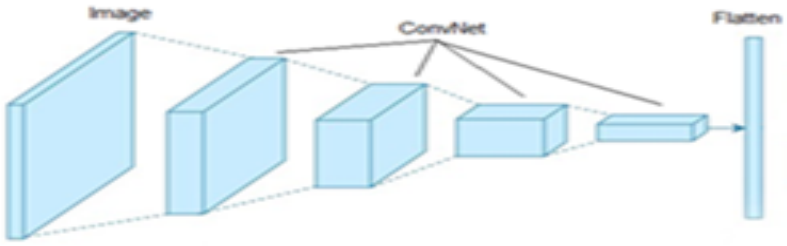
Tahap *preprosesing* dibagi menjadi tiga bagian yaitu memilih *frame*, *resize*, dan normalisasi. Sebuah video dengan *frame rate* yang tinggi memiliki banyak fitur yang redundan dan memiliki kompleksitas yang besar, maka dari itu tidak diproses seluruh *frame* pada video. *Frame* yang dipilih pada penelitian saya hanya *frame* yang kelipatan enam tujuannya untuk mengurangi komputasi dan mengurangi fitur-fitur yang redundan sehingga perubahan *frame* lebih signifikan hal ini didasari dari penelitian yang dilakukan oleh Ullah, Ahmad, Muhammad, Sajjad, dan Baik pada tahun 2018. Kemudian *frame* akan di-*resize* menjadi ukuran 224x224 pixel hal ini dilakukan untuk menyesuaikan dengan input pada arsitektur CNN yaitu *MobileNetV2*. Setelah itu dilakukan normalisasi yaitu proses membuat beberapa variabel memiliki rentang nilai yang sama dengan cara membagi nilai RGB dengan 255 agar setiap nilainya di antara 0 dan 1. Diagram alir tahap praproses dapat dilihat pada gambar 3.4.



Gambar 3.4 Diagram alir Praproses

3.5 Tahap Ekstraksi Fitur *Frame*

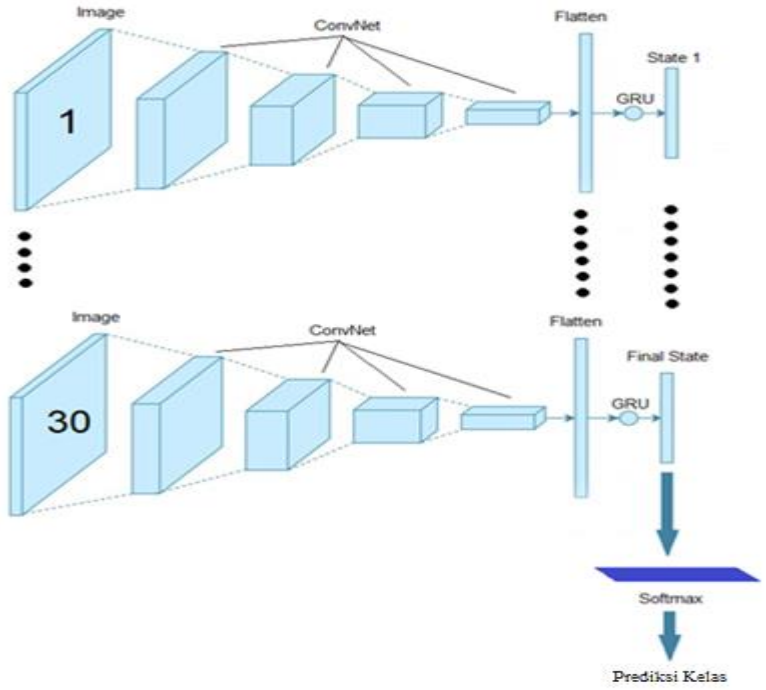
Pada tahap ini, akan dilakukan ekstraksi fitur menggunakan *MobileNetV2* untuk mengambil informasi yang berguna dari sebuah *frame* untuk pengenalan aktivitas. CNN adalah sumber fitur yang dominan dari representasi dan klasifikasi gambar. Melatih model *deep learning* untuk mendapatkan representasi gambar yang baik membutukan pelatihan pada ribuan bahkan jutaan gambar dan juga membutuhkan GPU dengan kekuatan tinggi untuk melatih CNN. Masalah ini dapat diselesaikan dengan menggunakan model yang sudah dilatih terlebih dahulu (Razavian, Azizpour, Sullivan, & Carlsson, 2014). *MobileNetV2* sudah dilatih pada dataset *ImageNet* agar dapat mengekstrak fitur dengan baik tanpa harus dilatih lebih lanjut pada dataset yang besar. Jumlah fitur yang dihasilkan dari *MobileNetV2* adalah konvolusi yang dijadikan *vector* sepanjang 62,720 elemen. Ilustrasi ekstraksi fitur dapat dilihat pada gambar 3.5.



Gambar 3.5 Ilustrasi ekstraksi fitur

3.6 Tahap Pengenalan Aktivitas

Pada tahap pengenalan aktivitas, fitur dari *frame* yang telah diekstrak oleh *MobileNetV2* akan dimasukkan ke GRU. Tahap ini bertujuan untuk menganalisa pola perubahan antar *frame*. Setiap fitur dari *frame* akan digunakan untuk *mengupdate state* dari GRU, kemudian *state* terakhir akan digunakan untuk menganalisa hasil akhir dari aktivitas pada video. Ilustrasi tahap ini dapat dilihat pada gambar 3.6.



Gambar 3.6 Ilustrasi pengenalan aktivitas

3.7 Pelatihan dan Pengujian

Proses pelatihan dilakukan untuk melatih atau membangun model dengan cara *mengupdate weight* setelah memproses *frame* terakhir dari suatu video. Dataset dibagi menjadi 3 bagian yaitu data *training*, data validasi, dan data tes dengan persamaan 60%, 20%, dan 20%. Tujuan memisahkan data menjadi data training dan data testing untuk menghindari terjadinya overfitting dimana suatu kondisi pelatihan yang hasil uji terhadap data yang dilatih sangat bagus tetapi diuji oleh data lain tidak digunakan dalam pelatihan sangat buruk. Sebelum dilakukan

update weight, *state* GRU diupdate setiap satu *frame* yang masuk hingga *frame* terakhir. *State* GRU terakhir yang akan digunakan untuk pengenalan akhir. *Output* dari pengenalan akhir digunakan untuk perhitungan *update weight*. Setiap satu epoch dilakukan validasi menggunakan data validasi untuk mengukur performa model saat pelatihan. Hasil terbaik dari suatu skenario validasi akan digunakan untuk test set untuk melihat keakuratanya atau performanya pada dataset dengan melakukan skenario uji coba. Percobaan dilakukan tanpa menggunakan GPU. skenario uji coba dibagi menjadi 4 yaitu:

1. Skenario Uji Coba pada pembekuan *weight MobileNetV2*
   Untuk uji coba hanya dilakukan pada pembekuan *weight MobileNetV2* dan tidak pembekuan pada *weight MobileNetV2* bertujuan untuk mengetahui apakah fitur yang diekstrak langsung dari *MobileNetV2* tanpa dilatih kembali menghasilkan performa lebih baik dibandingkan dengan *MobileNetV2* yang dilatih kembali dengan dataset yang ada.

2. Skenario Uji Coba pada *dropout*
   Pada skenario ini dilakukan uji coba pada *dropout* 0 dan 0,1 tujuannya untuk melihat apakah *dropout* dapat mengatasi *overfitting* pada model.

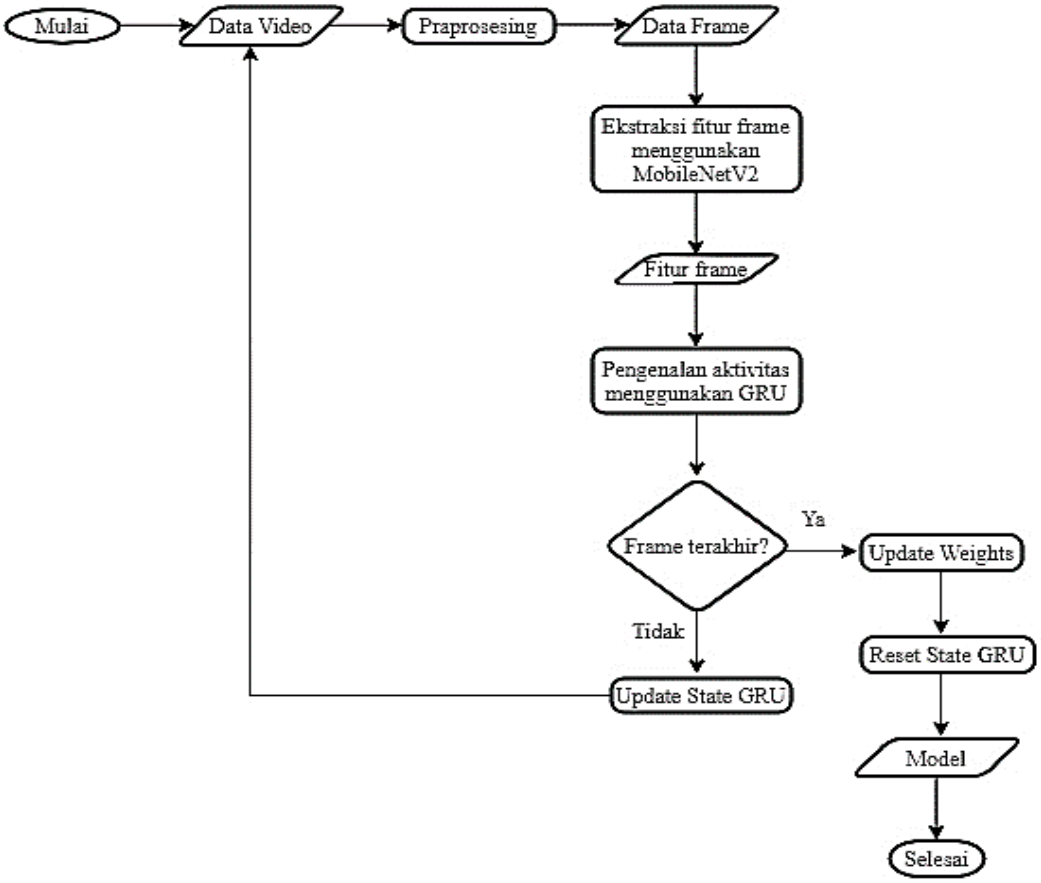
3. Skenario Uji Coba pada *learning rate*
   Pada skenario ini dilakukan uji coba pada *learning rate* 1.00E-5 dan 1.00E-6 tujuannya untuk mendapatkan hasil yang terbaik.

4. Skenario Uji Coba pada jumlah *hidden* unit GRU
   Pada skenario ini dilakukan uji coba pada 500 dan 250 *hidden* unit GRU untuk meningkatkan kecepatan model dengan cara mengurangi *hidden layer* pada GRU.

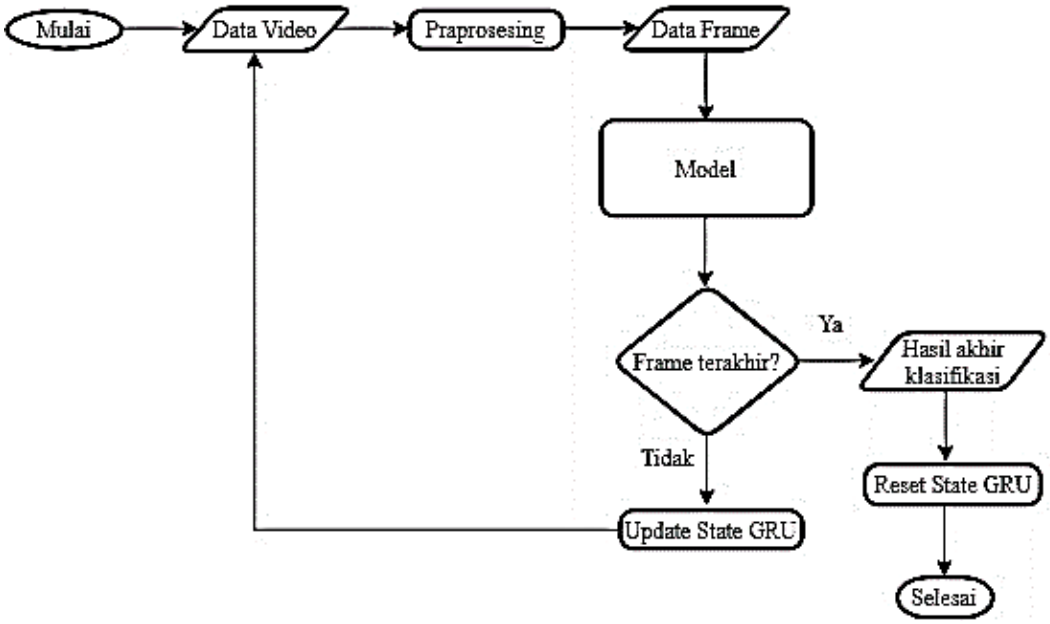
Tabel 3-2 Uji coba untuk pengujian

MobileNetV2 Weights	**Dropout**	**Learning Rate**	**Jumlah Hidden Unit GRU**	**Akurasi**	**F1Score**	**Presisi**	***Recall***	**FPS**
Dibekukan								
Tidak Dibekukan								

Pengujian dilakukan dengan cara mengukur performa model pada data *testing* dengan metrik akurasi, *precision*, *recall*, dan *f1 score* rumus-rumus ada di *confusion matrix* bagian jenis teknik validasi. Diagram alir proses pelatihan dapat dilihat pada gambar 3.7 dan untuk proses pengujian dapat dilihat pada gambar 3.8.



Gambar 3.7 Diagram alir pelatihan model.



Gambar 3.8 Diagram alir pengujian model.

V. KESIMPULAN DAN SARAN

Bab ini membahas tentang kesimpulan yang didasari oleh hasil uji coba yang telah dilakukan pada bab sebelumnya. Kesimpulan nantinya sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut di masa depan.

5.1 Kesimpulan

Dalam pengerjaan skripsi ini setelah melalui tahap perancangan sistem, implementasi metode, serta uji coba, diperoleh kesimpulan sebagai berikut:

1. Sistem yang memiliki performa paling baik adalah dengan membekukan *weight MobileNetV2*, *dropout* 0.1, *learning rate* 1e-5, dan 500 *hidden unit* GRU. Performa yang dihasilkan adalah *F1 score* 92.01% dan akurasi sebesar 92.3%.
2. Berdasarkan uji coba membekukan *weight MobileNetV2*, *dropout* 0.1, *learning rate* 1e-5, dan 500 *hidden unit* GRU, model yang dibangun menghasilkan rata-rata kecepatan 65 FPS (Frames Per Second).
3. Hasil pengujian dengan mengurangi jumlah *hidden unit* pada GRU dari 500 menjadi 250 dapat meningkatkan kecepatan sebesar 20.45 fps (frame per scond) tetapi menurunkan akurasi dan *F1 Score* pada model sebesar 1.16%.

5.2 Saran

Saran yang diberikan untuk pengembangan sistem pengenalan aktivitas manusia, yaitu:

1. Menambah variasi dataset berdasarkan kondisi sesungguhnya untuk menangani kondisi-kondisi pada realita.
2. Menambahkan deteksi manusia agar dapat mengenali aktivitas pada orang tertentu pada suatu video.
3. Melakukan eksplorasi arsitektur CNN selain *MobileNetV2*, seperti Residual Network 50, Inception-V3, Squeezenet dan GoogleNet.

DAFTAR PUSTAKA

A. Aly, S., Alghamdi, T. A., Salim, M., & Gutub, A. A. (2013). Data dissemination and collection algorithms for collaborative sensor devices using dynamic cluster heads. *Trends Appl. Sci. Res., 8*, 55-72.

Angelini, F., Fu, Z., Long, Y., Shao, L., & Naqvi, S. (2008). ActionXPose: A Novel 2D Multi-view Pose-based Algorithm for Real-time Human Action Recognition.

An Intuitive Explanation of Convolutional Neural Networks. (2016, August 11). (Ujjwalkarn) Retrieved November 29, 2022, from https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets

Aphex34. (2015, 12 16). *Convolutional Neural Network.* (Wikipedia) Retrieved 11 14, 2022, from https://en.wikipedia.org/wiki/File:Typical_cnn.png

Budhiraja, A. (n.d.). *Dropout in (Deep) Machine Learning*. Retrieved 11 11, 2022, from https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5

Cho, K., Merrienboer, B. v., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv*.

Chung, J., Gulcehr, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv*.

Colloert, R., & Weston, J. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. *Princeton*.

Deep learning for complete beginners: convolutional neural networks with keras. (2017, March 20). (Cambridgespark) Retrieved November 29, 2022, from https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html

Fadillah, S. (2017). *Penerapan Pengolahan Citra menggunakan Metode Deep Learning untuk Mendeteksi Kecacatan Permukaan Buah Manggis.* Skripsi, Universitas Muhammadiyah Yogyakarta, Teknik Informatika, Yogyakarta.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrin, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst., 28*, 2222-2232.

Herath, S., Harandi, M., & Porikli, F. (2017). Going deeper into action recognition: A survey. *Image Vis. Comput., 60*, 4-21.

Hinton, G. (2006). Neural Networks for Machine Learning.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv*.

Karpathy, A. (n.d.). *Convolutional Neural Networks for Visual Recognition*. (Stanford University) Retrieved November 30, 2022, from http://cs231n.github.io/

Keras,S. (2020). *The Python Deep Learning library*. (Keras) Retrieved November 30, 2022, from https://keras.io/

Khaeriyah, R. (2019). *Implementasi Metode Convolutional Neural Network Menggunakan Tensorflow Dalam Mendeteksi Sebuah Objek.* Yogyakarta: Skripsi UII.

Laurenti, Giulio. (2020). Confusion Matrix and Classification Report. [Online] Available at:https://medium.com/swlh/confusion-matrix-and-classification-report-88105288d48f

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv*.

Hunter,D.,J.(2008) (n.d.). (Matplotlib) Retrieved November 30, 2022, from https://matplotlib.org/index.html

Murthy, O. V., & Goecke, R. (2015). Ordered trajectories for human action recognition with large number of classes. *Image Vis. Comput., 42*, 22-34.

Nanda, A., Chauhan, D. S., K., S. P., & Bakshi, S. (2017). "Illumination and scale invariant relevant visual features with hypergraph-based learning for multi-shot person re-identification. *Multimedia Tools Appl*, 1-26.

Nanda, A., Sa, P. K., Choudhury, S. K., Baksh, S., & Majhi, B. (2017). A neuromorphic person re-identification framework for video surveillance. *IEEE Access, 5*, 6471-6482.

NarasingaRao, Dr.M.R., Prasad V Venkatesh. et al. (2018). Survey on Prevention

of Overfitting in Convolution Neural Networks Using ML Techniques. International Journal of Engineering & Technology,7 (2.32) (2018) 177-180.

Narkhede, S. (n.d.). *Understanding Confusion Matrix*. Retrieved May 29, 2022, from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Ng, J. Y.-H., Hausknecht, M., Vijayanarasimhan, S., O., V., Monga, R., & G., T. (2015). Beyond short snippets: Deep networks for video classification. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 4694-4702.

NumFOCUS. (2005.). (NumPy) Retrieved November 30, 2022, from http://www.numpy.org/

OpenCV. (2019.). Retrieved November 30, 2022, from https://opencv.org/

PSF.(2005) . (Python) Retrieved November 30, 2022, from https://www.python.org/about/

Razavian, A. S., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: An astounding baseline for recognition. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 806-813.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Schuldt, C., Laptev, I., & Caputo, B. (2004). Recognizing human actions: A local SVM approach. *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, 32-36.

Sena, S. (2017, October 28). *Pengenalan Deep Learning Neural Network*. Retrieved November 30, 2022, from https://medium.com/@samuelsena/pengenalan-deep-learning-8fbb7d8028ac

Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild.

Suartika, I. W., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi Citra Menggunakan Convolutional pada Caltech 101. *JURNAL TEKNIK ITS, 5*.

TensorFlow.J (2022.). (TensorFlow) Retrieved November 30, 2022, from https://www.tensorflow.org/

UCF. (2011, October 31). *www.crcv.ucf.edu*. (University of Central Florida) Retrieved June 24, 2021, from https://www.crcv.ucf.edu/data/UCF_YouTube_Action.php

Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M., & Baik, S. W. (2017). Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features. *IEEE Access, 6*, 1155 - 1166.

Venkatachalam, M. (n.d.). *Recurrent Neural Networks*. Retrieved June 13, 2022, from https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce