

SKRIPSI

DETEKSI MANIPULASI CITRA DENGAN *ERROR LEVEL ANALYSIS* DAN *CONVOLUTIONAL NEURAL NETWORK*

Disusun dan diajukan oleh:

**RIZKY ALFIANSYAH
D42116007**



**PROGRAM STUDI SARJANA TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2023**

LEMBAR PENGESAHAN SKRIPSI
DETEKSI MANIPULASI CITRA DENGAN ERROR LEVEL ANALYSIS
DAN CONVOLUTIONAL NEURAL NETWORK

Disusun dan diajukan oleh

RIZKY ALFIANSYAH

D42116007

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin pada tanggal 12 Juli 2023 dan dinyatakan telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,



Dr. Ir. Ingrid Nurtanio, MT
Nip. 196108131988112001



Prof. Dr. Ir. Indrabayu, S.T., M.T.,
M.Bus.Sys., IPM., ASEAN.Eng.
Nip. 197507162002121004

Ketua Program Studi,



Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM., ASEAN.Eng.
Nip. 197507162002121004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Rizky Alfiansyah
NIM : D42116007
Program Studi : Teknik Informatika
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Deteksi Manipulasi Citra Dengan *Error Level Analysis* Dan *Convolutional Neural Network*

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 18 Mei 2023

Yang Menyatakan



Rizky Alfiansyah

ABSTRAK

RIZKY ALFIANSYAH. *Deteksi Manipulasi Citra Dengan Error Level Analysis Dan Convolutional Neural Network* (dibimbing oleh Dr. Ir. Ingrid Nurtanio, M.T. dan Prof. Dr. Indrabayu, S.T., M.T., M. Bus.Sys., IPM, ASEAN.Eng)

Pemalsuan citra digital telah menjadi salah satu taktik penyebar hoax yang merupakan ajang penyebar provokasi, menimbulkan kerusuhan serta kebencian. Untuk mengidentifikasi suatu citra asli atau palsu, sangat sulit dilihat dengan mata telanjang, diperlukan teknik-teknik khusus dan ketelitian tertentu agar dapat mengetahui dengan pasti suatu citra merupakan citra asli atau sudah mengalami modifikasi. Bagi orang awam, hal ini mungkin sulit untuk dilakukan. Mesin dapat melakukannya dengan mudah untuk sejumlah besar data dengan bantuan teknik kecerdasan buatan yang digunakan untuk mengembangkan sistem yang akan membantu dalam mengenali apakah citra tersebut asli atau palsu. Teknologi ini membutuhkan banyak data citra, dan setiap citra memiliki banyak pixel-pixel penyusunnya. Metode *Convolutional Neural Network* (CNN) adalah salah satu algoritma dari *Deep Learning* yang merupakan pengembangan dari *Multi Layer Perceptron* (MLP) yang dirancang untuk mengolah data citra. CNN bisa digunakan untuk mendeteksi dan mengenali object pada sebuah image. Oleh karena itu pada penelitian ini membahas tentang implementasi metode metode *Error Level Analysis* untuk preprocessing dan klasifikasi menggunakan metode *Convolutional Neural Network*. yang dilakukan pada sekumpulan data untuk mengetahui model yang paling cocok untuk pengembangan sistem pengklasifikasi citra asli dan palsu. Dataset yang telah disiapkan dibagi menjadi 3 folder sesuai dengan label *Real*, *Splice*, dan *Copy Move* yang diberi pada masing-masing data yaitu sebanyak 7442 data *Real*, 1799 data *Splice*, dan 3224 Data *Copy Move* dengan 50 data *testing* untuk masing-masing label. Parameter yang digunakan yakni Input size 128 x 128 dengan 3 channel, epoch 200, batch size 32, dan *learning rate* 0,0001. Hasil pengujian deteksi citra, *system* mampu mendeteksi citra dengan total akurasi rata-rata mencapai 95% *Real*, 100% *Splice*, 98% *Copy Move*.

Kata Kunci: Deteksi Manipulasi Citra, *Error Level Analysis*, *Convolutional Neural Network*

ABSTRACT

RIZKY ALFIANSYAH. *Image Forgery Detection with Error Level Analysis and Convolutional Neural Network* (supervised by Dr. Ir. Ingrid Nurtanio, M.T. and Prof. Dr. Indrabayu, S.T., M.T., M. Bus.Sys., IPM, ASEAN.Eng)

Digital image forgery has become one of the tactics used to spread hoaxes, incite riots, and generate hatred. Identifying whether an image is genuine or fake is challenging to determine with the naked eye. Special techniques and precision are required to ascertain whether an image is original or has been modified. This may be difficult for ordinary individuals, but machines can easily accomplish this task for large amounts of data with the assistance of artificial intelligence techniques used to develop systems that aid in recognizing whether an image is genuine or fake. This technology relies on a significant amount of image data, with each image composed of numerous pixels. The Convolutional Neural Network (CNN) method is one algorithm from Deep Learning, which is an extension of the Multi-Layer Perceptron (MLP) designed to process image data. CNN can be utilized to detect and recognize objects in an image. Therefore, this research focuses on implementing the Error Level Analysis method for preprocessing and classification using the Convolutional Neural Network method. This is performed on a dataset to determine the most suitable model for developing a system that classifies genuine and fake images. The prepared dataset is divided into three folders based on the labels: Real, Splice, and Copy Move, consisting of 7442 Real data, 1799 Splice data, and 3224 Copy Move data, with 50 testing data for each label. The parameters used are an input size of 128 x 128 with 3 channels, 200 epochs, a batch size of 32, and a learning rate of 0.0001. The results of the image detection testing indicate that the system is capable of detecting images with an average accuracy of 95% for Real images, 100% for Splice images, and 100% for Copy Move images.

Keywords: *Image Forgery Detection, Error Level Analysis, Convolutional Neural Network*

DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI.....	i
PERNYATAAN KEASLIAN.....	ii
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR	vii
DAFTAR SINGKATAN DAN ARTI SIMBOL	ix
DAFTAR LAMPIRAN.....	x
KATA PENGANTAR	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian/Perancangan.....	2
1.4 Manfaat Penelitian/Perancangan.....	3
1.5 Ruang Lingkup/Asumsi perancangan.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Citra Digital	4
2.2 Pemalsuan Citra Digital	4
2.3 <i>Error Level Analysis (ELA)</i>	5
2.4 <i>Data Augmentation</i>	7
2.5 <i>Deep Learning</i>	10
2.6 Convolutional Neural Network (CNN).....	12
BAB III METODE PENELITIAN/PERANCANGAN	21

3.1	Waktu dan Lokasi Penelitian	21
3.2	Benda Uji dan Alat	21
3.3	Prosedur Penelitian	22
3.4	Teknik Pengambilan Data.....	23
3.5	Perancangan Sistem	23
BAB IV HASIL DAN PEMBAHASAN		37
4.1.	Hasil Penelitian.....	37
4.2.	Pembahasan	45
BAB V KESIMPULAN DAN SARAN.....		48
5.1	Kesimpulan	48
5.2	Saran	48
DAFTAR PUSTAKA		49
LAMPIRAN.....		51

DAFTAR GAMBAR

Gambar 1 Algoritma perkiraan kualitas JPEG.....	5
Gambar 2 Ilustrasi nilai kualitas algoritma ELA	6
Gambar 3 Contoh data augmentasi pada citra.....	9
Gambar 4 <i>Deep Neural Network</i> (Dadang, 2018).....	11
Gambar 5 Ilustrasi arsitektur Convolutional Neural Network (CNN).....	12
Gambar 6 Ilustrasi proses <i>convolutional layer</i> (Khan et al., 2018)	14
Gambar 7 Ilustrasi <i>weights</i> dan <i>bias</i> pada <i>layer</i> (Malik, 2019)	15
Gambar 8 Grafik Fungsi Aktivasi <i>ReLU</i> (Khan et al., 2018).....	16
Gambar 9 ilustrasi fungsi aktivasi <i>softmax</i>	17
Gambar 10 Operasi max pooling (Khan et al., 2018)	18
Gambar 11 Confusion Matrix menampilkan total positive dan negative tuple	20
Gambar 12 Tahapan Penelitian	22
Gambar 13 <i>Flowchart</i> tahap <i>training</i> dan tahap <i>testing</i>	23
Gambar 14 Contoh Data <i>Training</i>	24
Gambar 15 Hasil ELA dari gambar 14.	25
Gambar 16 <i>Flowchart</i> konversi citra ke ELA.....	26
Gambar 17 Identifikasi nilai RGB dari beberapa pixel.....	26
Gambar 18 Ilustrasi nilai kualitas JPG dengan kernel 8 x 8	27
Gambar 19 Potongan <i>Source Code</i> proses ELA.	28
Gambar 20 <i>Flowchart</i> proses <i>augmentasi</i> data.....	28
Gambar 21 Arsitektur pembangunan model CNN.....	29
Gambar 22 <i>Flowchart</i> proses <i>training</i>	30
Gambar 23 Parameter yang digunakan untuk <i>training</i>	32
Gambar 24 Layer Arsitektur CNN yang digunakan	33
Gambar 25 Proses <i>training</i> data	33
Gambar 26 <i>Source Code Testing</i> Data	34
Gambar 27 Kurva Akurasi dan <i>Kurva Loss Training Accuracy</i> tanpa metode ELA	38
Gambar 28 <i>Confusion matrix</i> hasil kinerja model tanpa menggunakan metode ELA	38
Gambar 29 Kurva Akurasi dan <i>Kurva Loss Training Accuracy</i> dengan metode ELA	39
Gambar 30 <i>Confusion Matrix</i> kinerja model dengan metode <i>ELA</i>	39

DAFTAR TABEL

Tabel 1 Tabel konversi RGB ke YCbCr	27
Tabel 2 Confusion matrix model evaluation.....	35
Tabel 3 Tabel hasil pengujian skenario.....	37
Tabel 4 Uji Deteksi Citra Asli.....	41
Tabel 5 Uji Deteksi Citra <i>Splice</i>	42
Tabel 6 Uji Citra <i>Copy Move</i>	44
Tabel 7 Kesalahan Deteksi.....	45

DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
ELA	<i>Error Level Analysis</i>
CNN	<i>Convolutional Neural Network</i>
MLP	<i>Multi Layer Perceptron</i>
CGI	<i>Computer Generated Image</i>
RGB	“ <i>Red, Green Blue</i> ”, pallete warna yang terdiri dari tiga warna primer yakni Merah, Hijau, dan Biru.
YCrCb	ruangan warna yang dipakai sebagai bagian dari <i>Color image pipeline</i> dalam sistem video dan fotografi digital.
ReLu	<i>Rectifier Linear Unit</i>
σ	Definisi <i>fungsi softmax</i>
CE	<i>Categorical Cross Entropy</i>
TP	“ <i>True Positive</i> ”, Jumlah data dengan nilai sebenarnya positif dan nilai prediksi positif
FP	“ <i>False Positive</i> ”, Jumlah data dengan nilai sebenarnya negatif dan nilai prediksi positif
FN	“ <i>False Negative</i> ”, Jumlah data dengan nilai sebenarnya positif dan nilai prediksi negative
TN	“ <i>True Negative</i> ”, Jumlah data dengan nilai sebenarnya negatif dan nilai prediksi negatif

DAFTAR LAMPIRAN

Lampiran 1 Source Code	51
Lampiran 2 Dataset	51

KATA PENGANTAR

Assalamualaikum Warahmatullaahi Wabarakatuh

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa, yang telah membagikan rahmat serta karunia- Nya, sehingga penulis bisa menyelesaikan laporan tugas akhir dengan judul “Deteksi Manipulasi Citra Dengan *Error Level Analysis* Dan *Convolutional Neural Network*”. Tak lupa pula Shalawat dan salam senantiasa tercurah kepada Rasulullah SAW yang mengantarkan manusia dari zaman kegelapan ke zaman yang terang benderang ini. Laporan tugas akhir ini ialah salah satu syarat untuk mendapatkan gelar Sarjana Strata Satu (S1) pada Departemen Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa penulisan ini tidak dapat terselesaikan tanpa dukungan dari berbagai pihak baik moril maupun materil. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu dalam penyusunan skripsi ini terutama kepada:

1. Kedua orang tua, Bapak Sukiman dan Ibu Sitti Wajida yang telah memberikan dukungan baik moril maupun materil serta senantiasa memberikan doa yang tiada henti-hentinya kepada penulis.
2. Ibu Dr. Ir. Ingrid Nurtanio, M.T., selaku dosen pembimbing I dan Bapak Prof. Dr. Indrabayu, S.T., M.T., M. Bus.Sys., IPM, ASEAN.Eng selaku dosen pembimbing II, yang selalu menyediakan waktu, tenaga dan pikirannya yang luar biasa untuk mengarahkan penulis dalam penyusunan Tugas Akhir ini.
3. Segenap Dosen dan Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin, yang telah banyak membantu penulis selama masa perkuliahan.
4. AIMP Family, yang telah memberikan begitu banyak bantuan selama penelitian, pengambilan data, dan diskusi terkait penyusunan skripsi.
5. Saudara-saudari IGNITER 16, yang selalu menyemangati dan membantu penyelesaian skripsi ini serta mengisi hari-hari menjadi sangat menyenangkan.
6. Semua pihak atas dukungan dan bantuannya yang tidak dapat penulis tuliskan satu persatu.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna dikarenakan terbatasnya pengalaman dan pengetahuan yang dimiliki penulis. Oleh karena itu, penulis mengharapkan segala bentuk saran serta masukan bahkan kritik yang membangun dari berbagai pihak. Semoga skripsi ini dapat memberikan manfaat untuk mendorong penelitian penelitian selanjutnya.

Gowa, 18 Mei 2023

Rizky Alfiansyah
D42116007

BAB I PENDAHULUAN

1.1 Latar Belakang

Menurut *EU High Level Expert Group* (2018), berita palsu didefinisikan sebagai disinformasi, yaitu segala bentuk informasi yang tidak akurat, salah, ataupun menyesatkan yang dipresentasikan, dipromosikan, atau didesain. Di balik berita-berita palsu, terdapat beberapa alasan mengenai publikasi tersebut. Salah satunya adalah untuk mendapatkan keuntungan secara ekonomi, entah itu melalui peningkatan jumlah klik berita maupun membuat berita yang tidak seharusnya untuk menguntungkan salah satu pihak (Özgöbek and Gulla, 2018).

Pemalsuan dan perusakan informasi tumbuh secara eksponensial, karena semakin banyaknya platform media sosial. Informasi yang dipalsukan disebarluaskan melalui berbagai platform dan dalam berbagai bentuk. Platform media sosial merupakan lokasi yang sangat ideal untuk melakukan tindakan pemalsuan dan perusakan tersebut karena mengingat informasinya bisa menjangkau sejumlah besar audiens (Parikh et al., 2019).

Saat ini penyebaran informasi atau berita melalui media *online* tidak hanya dilakukan oleh situs berita yang sudah dikenal oleh masyarakat, namun oleh siapa saja pengguna internet dapat berperan dalam penyebaran suatu informasi. Sayangnya banyak informasi atau berita yang disebarluaskan secara individu atau berkelompok lebih banyak yang tidak dapat dipertanggungjawabkan kebenarannya atau teindikasi *hoax*. *Hoax* merupakan informasi atau berita yang berisi hal-hal yang belum pasti atau yang benar-benar bukan merupakan fakta yang terjadi.

Kini informasi atau berita yang dianggap benar tidak lagi mudah ditemukan. Survey Mastel (2017) mengungkapkan bahwa dari 1.146 responden, 44,3% diantaranya menerima berita *hoax* setiap hari dan 17,2% menerima lebih dari satu kali dalam sehari. Bahkan media arus utama yang diandalkan sebagai media yang dapat dipercaya terkadang ikut terkontaminasi penyebaran *hoax*. Media arus utama juga menjadi saluran penyebaran informasi/berita *hoax*, masing-masing sebesar 1,20% (radio), 5% (media cetak) dan 8,70% (televisi). Tidak saja oleh media arus utama, kini *hoax* sangat banyak beredar di masyarakat melalui media *online*. Hasil penelitian yang dilakukan oleh Mastel (2017) menyebutkan bahwa saluran yang banyak digunakan dalam penyebaran *hoax* adalah situs web, sebesar 34,90%, aplikasi chatting (Whatsapp, Line, Telegram) sebesar 62,80%, dan melalui media sosial (Facebook, Twitter, Instagram, dan Path) yang merupakan media terbanyak digunakan yaitu mencapai 92,40%. Sementara itu, data yang dipaparkan oleh Kementerian

Komunikasi dan Informatika menyebut ada sebanyak 800 ribu situs di Indonesia yang terindikasi sebagai penyebar hoax dan ujaran kebencian (Juditha, 2018).

Pemalsuan citra digital telah menjadi salah satu taktik penyebar *hoax* yang merupakan ajang penyebar provokasi, menimbulkan kerusuhan serta kebencian. Untuk mengidentifikasi suatu citra asli atau palsu, sangat sulit dilihat dengan mata telanjang, diperlukan teknik-teknik khusus dan ketelitian tertentu agar dapat mengetahui dengan pasti suatu citra merupakan citra asli atau sudah mengalami modifikasi. Bagi orang awam, hal ini mungkin sulit untuk dilakukan. Mesin dapat melakukannya dengan mudah untuk sejumlah besar data dengan bantuan teknik kecerdasan buatan yang digunakan untuk mengembangkan sistem yang akan membantu dalam mengenali apakah citra tersebut asli atau palsu. Teknologi ini membutuhkan banyak data citra, dan setiap citra memiliki banyak pixel-pixel penyusunnya. *Convolutional Neural Network* (CNN) adalah salah satu algoritma dari *Deep Learning* yang merupakan pengembangan dari *Multi Layer Perceptron* (MLP) yang dirancang untuk mengolah data citra. CNN bisa digunakan untuk mendeteksi dan mengenali *object* pada sebuah *image*. Oleh karena itu pada penelitian ini membahas tentang implementasi metode *deep learning* yang dilakukan pada sekumpulan data untuk mengetahui model yang paling cocok untuk pengembangan sistem pengklasifikasi citra asli dan palsu.

1.2 Rumusan Masalah

- a. Bagaimana sistem dapat mendeteksi pemalsuan citra?
- b. Bagaimana kinerja dari metode *Error Level Analysis* untuk mendeteksi adanya indikasi pemalsuan (*splice* atau *copy-move*) dalam citra?
- c. Bagaimana hasil testing model dari system klasifikasi gambar asli dan palsu (*splice* atau *copy-move*) menggunakan *Convolutional Neural Network*?

1.3 Tujuan Penelitian/Perancangan

- a. Membuat system yang dapat mendeteksi citra yang telah dimanipulasi.
- b. Mengetahui hasil kinerja dari *Error Level Analysis* dalam mendeteksi adanya indikasi pemalsuan *splice* atau *copy-move* dalam citra.
- c. Mengetahui hasil testing model dari klasifikasi citra asli dan palsu (*splice* atau *copy-move*) menggunakan *Convolutional Neural Network*.

1.4 Manfaat Penelitian/Perancangan

Dapat memberikan masyarakat pertimbangan yang lebih baik dalam menentukan apakah citra tersebut asli atau palsu (*splice* atau *copy-move*), yang berdampak positif dalam memastikan integritas visual dan keamanan informasi.

1.5 Ruang Lingkup/Asumsi perancangan

Terdapat beberapa batasan yang berlaku pada penambangan sistem ini, yaitu:

- a. Data mentah harus berupa citra dengan *lossy compression* (contohnya .jpg).
- b. Citra bukan merupakan *computer generated image* (CGI).

BAB II TINJAUAN PUSTAKA

2.1 Citra Digital

Citra secara umum adalah merupakan suatu gambar, foto ataupun berbagai tampilan dua dimensi yang menggambarkan suatu visualisasi objek. Citra dapat diwujudkan dalam bentuk tercetak ataupun digital. Citra digital adalah larik angka angka secara dua dimensional. Citra digital tersimpan dalam suatu bentuk larik (array) angka digital yang merupakan hasil kuantifikasi dari tingkat kecerahan masing-masing piksel penyusun citra tersebut. Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (continue) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, scanner dan lain sebagainya sehingga bayangan objek yang disebut citra tersebut terekam (Prabowo and Abdullah, 2018).

Dalam komputer, setiap piksel diwakili oleh dua buah bilangan bulat (integer) untuk menunjukkan lokasinya dalam bidang citra dan sebuah nilai dalam bilangan bulat (integer) untuk menunjukkan cahaya atau keadaan terang-gelap piksel tersebut. Suatu piksel dapat ditunjukkan lokasinya dengan menggunakan koordinat (0,0) yang digunakan untuk posisi kiri atas dalam bidang citra. Tingkat pencahayaan suatu piksel dapat ditunjukkan dengan bilangan bulat yang besarnya 8 bit, dengan lebar selang nilai 0-255, di mana 0 untuk warna hitam, 255 untuk warna putih. Citra digital yang diperoleh dari kamera digital akan menghasilkan intensitas pantulan yang menggambarkan terang dan gelap pada penampilan piksel-piksel penyusunannya selain itu juga akan memberikan informasi warna (Pradhitya, 2015).

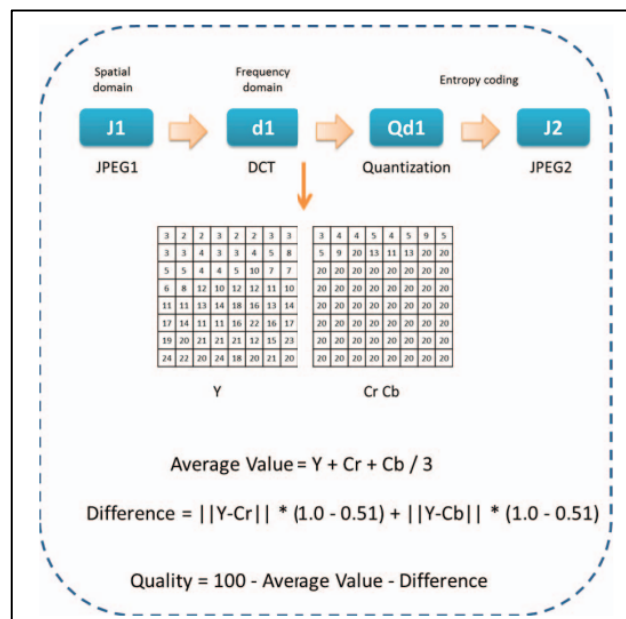
2.2 Pemalsuan Citra Digital

Pemalsuan citra sering tak dapat dikenali secara kasat mata karena citra hasil modifikasi sulit ditentukan keasliannya. Kasus modifikasi citra yang terjadi misalnya adalah kasus beredarnya foto-foto yang melibatkan pejabat, artis atau publik figur yang memuat konten-konten yang sensitif sehingga menimbulkan kehebohan di masyarakat. Kemunculan citra-citra digital yang dimodifikasi akan menimbulkan permasalahan-permasalahan dalam kehidupan sosial seperti penyebaran informasi-informasi yang tidak benar, sehingga sangat mudah terjadi kesalah pahaman. JPEG “Joint Photographic Experts

Group” merupakan salah satu format citra yang paling sering digunakan, dimana JPEG adalah sebuah format citra yang memiliki format lossy (dimana metode untuk mengkompresi data dengan hasil perbedaan data yang tidak terlalu jauh dari data sebelum kompresi), tetapi banyaknya error yang tersimpan tidak sama dengan citra sebelum proses kompresi (Darmawan et al., 2019).

2.3 Error Level Analysis (ELA)

ELA adalah algoritma kompresi JPEG untuk deteksi forensik gambar. koefisien frekuensi dari gambar dikuantisasi oleh tabel kuantisasi dan diikuti oleh proses *entropi encoded*. Krawetz bersama timnya (Hacker Factor Solution) mengadopsi proses kuantisasi untuk mengembangkan algoritma untuk mendekati kualitas JPEG. Gambar 1 menyajikan algoritma pendekatan kualitas JPEG. Melalui persamaan tersebut, perhitungan kualitas JPEG dapat diringkas dengan menghitung selisih antara nilai rata-rata dari tabel kuantisasi Y (*luminance*) dan CrCb (*chrominance*).



Gambar 1 Algoritma perkiraan kualitas JPEG

(Warifetal., 2015)

Menggunakan ELA, gambar selanjutnya dibagi menjadi blok 8 x 8 dan dikompresi ulang secara independen pada tingkat kesalahan yang diketahui seperti 95%. Setiap kotak harus memberikan tingkat kualitas yang kira-kira sama jika gambar tidak dimodifikasi

sama sekali. Tingkat kesalahan adalah hilangnya informasi selama gambar disimpan dalam format JPEG. Ilustrasi nilai kualitas algoritma ELA ditunjukkan pada Gambar 2. Pada gambar tersebut, perbedaan tingkat kesalahan pada blok-blok tertentu dapat menentukan area yang dimodifikasi.

Selain itu, tingkat kesalahan akan meningkat pada operasi penyimpanan ulang. Operasi resave selanjutnya dapat mengurangi potensi tingkat kesalahan dan ditunjukkan melalui hasil ELA yang lebih gelap. Setelah sejumlah operasi penyimpanan ulang, kisi persegi dapat mencapai tingkat kesalahan minimumnya. Oleh karena itu frekuensi dan detail dapat hilang oleh setiap operasi penyimpanan ulang.

81	81	81	81	81	81	81	81
81	81	81	81	81	81	81	81
90	90	90	81	81	81	81	81
90	90	90	81	81	81	81	81
90	90	90	81	81	81	81	81
90	90	90	81	81	81	81	81
81	81	81	81	81	81	81	81
81	81	81	81	81	81	81	81

Gambar 2 Ilustrasi nilai kualitas algoritma ELA

(Warif et al., 2015)

Namun, jumlah kesalahan dalam ELA terbatas pada blok 8 x 8. Kesalahan blok telah mencapai minimum lokalnya jika tidak ada perubahan pada tingkat kualitas. Piksel tidak pada minimum lokalnya jika ada sejumlah besar biaya yang kemudian dapat mengidentifikasi perubahan. Misalnya, menyimpan 90% lagi akan menghasilkan gambar yang hampir sama untuk penghematan satu kali sebesar 81%. Selain itu, area biru atau merah yang dirancang oleh ELA mencerminkan perubahan yang dilakukan oleh perangkat lunak komersial seperti photoshop dan GIMP. Kesalahan yang dihasilkan oleh teknik ELA dapat membantu mengidentifikasi manipulasi pada gambar JPEG. Bagaimanapun, teknik ELA dapat diringkas sebagai proses evaluasi tingkat kualitas kisi-kisi yang dikuadratkan di dalam gambar.

Berdasarkan karakteristik ELA, teknik ini diyakini dapat berguna untuk citra forensik area (Warifetal., 2015).

2.4 Data Augmentation

Augmentasi data merupakan proses memodifikasi atau memanipulasi suatu citra sehingga citra asli dalam bentuk standar akan diubah bentuk dan posisinya. Augmentasi data bertujuan agar mesin bisa belajar dan mengenali berbagai macam citra yang berbeda-beda, sekaligus dapat dimanfaatkan untuk memperbanyak data. Pada kebanyakan kasus, penggunaan augmentasi data berhasil meningkatkan performa dari model. Peningkatan tersebut terjadi karena mesin dapat mengenali dengan lebih baik objek dari berbagai bentuk serta pola yang bervariasi. Dapat dilihat juga pada gambar di bawah ini, satu citra saja mampu menghasilkan banyak gambaran baru dengan posisi yang beragam. Contoh penggunaan augmentasi data dapat ditemukan pada gambar di bawah ini (Zuhri et al., 2022).

Augmentasi citra dalam bentuk distorsi data dapat ditemukan dalam *LeNet-5*. Ini merupakan salah satu pengaplikasian awal dari CNN pada klasifikasi angka tulisan tangan. Augmentasi data juga telah diteliti dalam aplikasi *oversampling*. *Oversampling* adalah teknik yang digunakan untuk merekayasa ulang sampel distribusi kelas yang tidak seimbang agar model tidak terlalu condong dalam memberi label pada instansi sebagai tipe kelas mayoritas. Pendekatan sederhana adalah *Random Oversampling* (ROS) yang menduplikasi gambar secara acak dari kelas minoritas hingga mencapai rasio kelas yang diinginkan. Bukti awal yang menunjukkan efektivitas Augmentasi Data berasal dari transformasi sederhana seperti membalik horizontal, augmentasi ruang warna, dan pemotongan acak (Shorten and Khoshgoftaar, 2019).

Beberapa macam transformasi geometri yang sering dilakukan adalah:

1. *Flipping*

Membalik sumbu horizontal jauh lebih umum daripada membalik sumbu vertikal. Augmentasi ini adalah salah satu yang paling mudah diimplementasikan.

2. *Color Space*

Data citra digital umumnya diwakili dalam bentuk tensor dengan dimensi (tinggi \times lebar \times saluran warna). Salah satu strategi yang praktis untuk melakukan augmentasi adalah dengan mengubah data dalam ruang saluran warna. Terdapat beberapa augmentasi warna sederhana, seperti mengisolasi saluran warna tunggal seperti R, G, atau B. Dengan dengan cepat, citra dapat diubah menjadi representasi dalam satu saluran warna dengan mengisolasi matriks tersebut dan menambahkan dua matriks nol dari saluran warna lainnya. Selain itu, nilai-nilai RGB dapat dengan mudah dimanipulasi melalui operasi matriks sederhana untuk meningkatkan atau mengurangi kecerahan gambar. Terdapat juga augmentasi warna yang lebih canggih, seperti

mengubah histogram warna yang menggambarkan citra. Dengan mengubah nilai intensitas dalam histogram tersebut, pencahayaan citra dapat diubah, serupa dengan yang digunakan dalam aplikasi pengeditan foto.

3. *Cropping*

Cropping citra dapat digunakan sebagai langkah pemrosesan yang praktis untuk data citra dengan dimensi tinggi dan lebar yang bervariasi. Salah satu metode adalah dengan memotong bagian tengah setiap citra untuk mendapatkan patch yang diinginkan. Selain itu, pemotongan acak juga dapat digunakan untuk menciptakan efek yang mirip dengan translasi. Perbedaan antara pemotongan acak dan translasi adalah bahwa pemotongan akan mengurangi ukuran input citra, misalnya dari (256, 256) menjadi (224, 224), sedangkan translasi mempertahankan dimensi spasial citra. Namun, tergantung pada ambang batas pengurangan yang digunakan dalam pemotongan, transformasi ini mungkin tidak menjaga keberlanjutan label.

4. *Rotation*

Augmentasi rotasi dilakukan dengan memutar citra ke kanan atau kiri pada sumbu antara 1° hingga 359° . Keamanan dari augmentasi rotasi sangat ditentukan oleh parameter derajat rotasi. Rotasi yang sedikit, seperti antara 1 hingga 20 atau -1 hingga -20 , mungkin berguna pada tugas pengenalan angka seperti MNIST. Namun, ketika derajat rotasi semakin meningkat, label dari data tidak lagi dipertahankan setelah transformasi dilakukan.

5. *Translation*

Menggeser citra ke kiri, kanan, atas, atau bawah dapat menjadi transformasi yang sangat berguna untuk menghindari bias posisi pada data. Sebagai contoh, jika semua citra dalam dataset berada di pusat, yang umum terjadi pada dataset pengenalan wajah, maka model juga perlu diuji pada citra yang berada di pusat secara sempurna. Saat citra asli digeser ke arah tertentu, ruang yang tersisa dapat diisi dengan nilai konstan seperti 0 atau 255, atau dapat diisi dengan noise acak atau Gaussian. Penambahan ini mempertahankan dimensi spasial dari citra setelah augmentasi dilakukan.

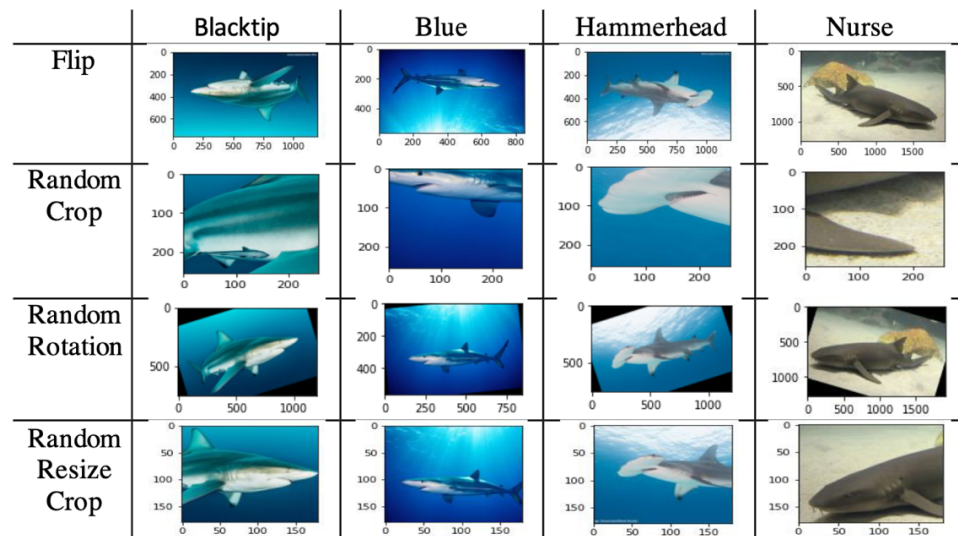
6. *Color Space Transformation*

Data citra dikodekan ke dalam tiga matriks yang saling ditumpuk, masing-masing memiliki ukuran tinggi \times lebar. Matriks-matriks ini mewakili nilai piksel untuk setiap nilai warna RGB secara individu. Bias pencahayaan merupakan salah satu tantangan umum dalam masalah pengenalan citra. Oleh karena itu, efektivitas transformasi ruang warna, yang juga dikenal sebagai transformasi fotometrik, relatif mudah dipahami. Salah satu cara cepat untuk memperbaiki citra yang terlalu terang atau terlalu gelap

adalah dengan meningkatkan atau menurunkan nilai piksel dengan konstanta. Manipulasi ruang warna lainnya adalah dengan memisahkan matriks warna RGB secara individual. Transformasi lain melibatkan batasan nilai piksel pada rentang nilai minimum atau maksimum tertentu. Representasi intrinsik warna dalam citra digital menyediakan banyak strategi augmentasi yang dapat digunakan.

Transformasi ruang warna juga dapat dilakukan melalui aplikasi pengeditan citra. Nilai piksel dalam setiap saluran warna RGB digabungkan untuk membentuk histogram warna. Histogram ini dapat dimanipulasi untuk menerapkan filter yang mengubah karakteristik ruang warna citra. Terdapat kebebasan kreativitas yang besar dalam augmentasi ruang warna. Mengubah distribusi warna pada citra dapat menjadi solusi yang baik untuk mengatasi tantangan pencahayaan pada data pengujian.

Dataset citra dapat disederhanakan dengan mengonversi matriks RGB menjadi citra *grayscale* tunggal. Ini akan menghasilkan citra yang lebih kecil dengan dimensi tinggi \times lebar \times 1, yang memungkinkan perhitungan yang lebih cepat. Namun, telah terbukti bahwa ini dapat mengurangi akurasi kinerja citra.



Gambar 3 Contoh data augmentasi pada citra

Transformasi geometri merupakan solusi yang baik untuk mengatasi bias posisi yang ada dalam data pelatihan. Terdapat banyak sumber potensial bias yang dapat memisahkan distribusi data pelatihan dari data pengujian. Jika terdapat bias posisi, seperti pada dataset pengenalan wajah di mana setiap wajah berada di pusat frame secara sempurna, transformasi geometri adalah solusi yang tepat. Selain kemampuan mereka yang kuat dalam mengatasi bias posisi, transformasi geometri juga berguna karena mudah

diimplementasikan. Terdapat banyak perpustakaan pengolahan gambar yang menyediakan operasi seperti flip horizontal dan rotasi yang mudah digunakan. Beberapa kelemahan dari transformasi geometri termasuk penggunaan memori tambahan, biaya komputasi transformasi, dan waktu pelatihan tambahan. Beberapa transformasi geometri seperti translasi atau pemotongan acak harus diamati secara manual untuk memastikan bahwa mereka tidak mengubah label citra (Shorten and Khoshgoftaar, 2019).

2.5 Deep Learning

Belakangan ini *Deep Learning* menjadi sorotan dalam pengembangan *Machine Learning*. Alasannya karena *Deep Learning* telah mencapai hasil yang luar biasa dalam visi komputer. *Deep Learning* merupakan cabang dari *Machine Learning* yang terinspirasi dari kortek manusia dengan menerapkan jaringan syaraf buatan yang memiliki banyak hidden layer. *Convolutional Neural Network (CNN)* merupakan salah satu metode dalam *Deep Learning* yang dibuat untuk menutupi kelemahan dari metode sebelumnya. Terdapat beberapa kelemahan dalam metode sebelumnya, tetapi dengan model ini sejumlah parameter bebas dapat dikurangi dan deformasi gambar input seperti translasi, rotasi dan skala dapat ditangani (Santoso and Ariyanto, 2018).

Algoritma pada *Deep Learning* memiliki fitur yang unik yaitu sebuah fitur yang mampu mengekstraksi secara otomatis. Hal ini berarti algoritma yang dimilikinya secara otomatis dapat menangkap fitur yang relevan sebagai keperluan dalam pemecahan suatu masalah. Algoritma semacam ini sangat penting dalam sebuah kecerdasan buatan karena mampu mengurangi beban pemrograman dalam memilih fitur yang eksplisit. Algoritma ini dapat digunakan untuk memecahkan permasalahan yang perlu pengawasan (*supervised*), tanpa pengawasan (*unsupervised*), dan semi terawasi (*semi supervised*). Teknik dan algoritma *deep learning* telah digunakan dalam beberapa bidang seperti klasifikasi gambar, klasifikasi video, deteksi objek, pengenalan pola, *text-to-speech*, *natural language processing*, robotika, klasifikasi teks, dan lain sebagainya.

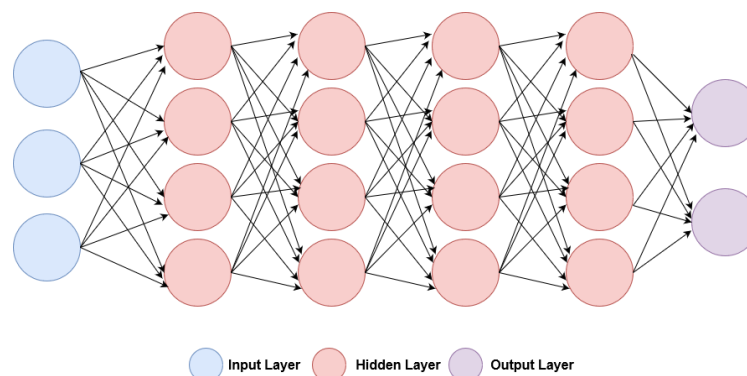
Jaringan saraf yang dimiliki oleh *Deep Learning* terbentuk dari hirarki sederhana dengan beberapa lapisan hingga tingkat tinggi atau banyak lapisan (*multi layer*). Berdasarkan hal itulah *Deep Learning* dapat digunakan untuk memecahkan masalah kompleks yang lebih rumit dan terdiri dari sejumlah besar lapisan transformasi *non-linier*. *Deep Learning* bekerja berdasarkan jaringan dan prosedural optimal yang digunakan pada arsitektur. Setiap *output* dari lapisan per lapisan yang tersembunyi dapat dipantau dengan menggunakan grafik khusus yang dirancang untuk setiap output neuron. Kombinasi dan rekombinasi dari setiap neuron yang saling terhubung dari semua unit lapisan tersembunyi

dilakukan menggunakan gabungan dari fungsi aktivasi. Prosedur-prosedur tersebut dikenal sebagai Transformasi *Non-Linier* yang digunakan untuk prosedur optimal untuk menghasilkan bobot optimal pada setiap unit lapisan guna mendapatkan nilai target yang dibutuhkan.

Dalam proses perancangan, apabila jumlah saraf yang ditambahkan sangat banyak, hal tersebut tidak akan pernah cocok untuk menyelesaikan setiap masalah. Persoalan terpenting dalam *Deep Learning* adalah jaringan sarafnya dilatih dengan cara penurunan gradien secara sederhana. Pada saat kita menambahkan lapisan jaringan yang semakin banyak, maka sebaliknya penurunan dari gradien semakin berkurang sehingga dapat mempengaruhi nilai outpunya. Jaringan Saraf Tiruan (*Artificial Neural Network*) adalah jaringan saraf yang biasanya menggunakan jaringan seperti umpan maju (*feed forward*) atau *recurrent network* yang hanya memiliki 1 atau 2 lapisan tersembunyi. Tetapi, jika lapisan jaringan sarafnya lebih dari 2 layer atau bahkan mencapai ratusan lapisan itulah yang disebut sebagai *Deep Learning*. Pada Jaringan Saraf Tiruan, arsitektur jaringan yang dimilikinya kurang kompleks dan membutuhkan lebih banyak informasi tentang data input sehingga dapat menentukan algoritma mana yang dapat digunakan.

Dalam Jaringan Saraf Tiruan terdiri dari beberapa algoritma yaitu Model Hebb, *Perceptron*, *Adaline*, *Forward Propagation*, dll. Sedangkan pada algoritma jaringan saraf *Deep Learning* tidak memerlukan informasi apapun terhadap data yang akan dipelajarinya, dan algoritmanya dapat secara mandiri melakukan tuning (penyetelan) dan pemilihan model yang paling optimal (Dadang, 2018).

Lapisan layer pada jaringan *Deep Learning Neural Network* dapat dilihat pada gambar 4 berikut:



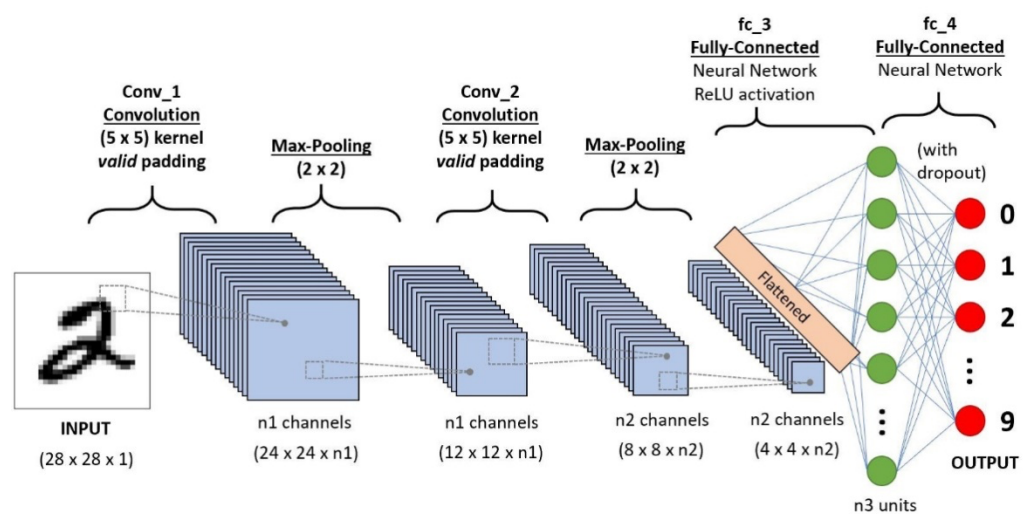
Gambar 4 Lapisan layer pada *Deep Learning Neural Network* (Dadang, 2018).

2.6 Convolutional Neural Network (CNN)

CNN adalah tipe network yang berbasis *feed forward*, yang alur informasinya hanyalah satu arah, yaitu dari masukan ke keluaran. Walaupun ada beberapa jenis arsitektur CNN, pada umumnya, CNN memiliki beberapa *convolutional layer* dan *pooling layer*. Kemudian, diikuti oleh satu atau lebih *fully connected layer*. Pada klasifikasi citra, masukan pada CNN adalah dalam bentuk citra, sehingga setiap pixel-nya dapat diolah.

Secara singkat, *convolutional layer* digunakan sebagai pengestraksi fitur yang mempelajari representasi fitur tersebut dari citra yang menjadi masukan pada CNN. Sedangkan, *pooling layer* bertugas untuk mengurangi resolusi spasial dari peta-peta fitur. Umumnya, sebelum *fully connected layer*, terdapat tumpukan beberapa *convolutional* dan *pooling layer* yang berfungsi untuk mengekstrak representasi fitur yang lebih abstrak. Setelahnya, *fully connected layer* akan menginterpretasi fitur-fitur tersebut dan melakukan fungsi-fungsi yang membutuhkan *high-level reasoning*. Klasifikasi pada akhir CNN akan menggunakan fungsi softmax (Gunawan et al., 2018).

Ilustrasi arsitektur *Convolutional Neural Network* dapat dilihat pada gambar 5 berikut:



Gambar 5 Ilustrasi arsitektur *Convolutional Neural Network (CNN)*

(Gunawan et al., 2018).

Tahap pertama dalam arsitektur CNN adalah tahap konvolusi. Kemudian dilanjutkan fungsi aktivasi menggunakan fungsi aktivasi *ReLU (Rectifier Linear Unit)*, kemudian dilanjutkan dengan proses *pooling*. Proses ini diulang terus menerus sampai didapatkan peta fitur yang cukup untuk dilanjutkan ke *fully connected neural network*, sehingga dapat

dihasilkan output class. Penjelasan secara detail dari tahap arsitektur CNN tersebut dijelaskan sebagai berikut:

2.6.1. Convolutional Layer

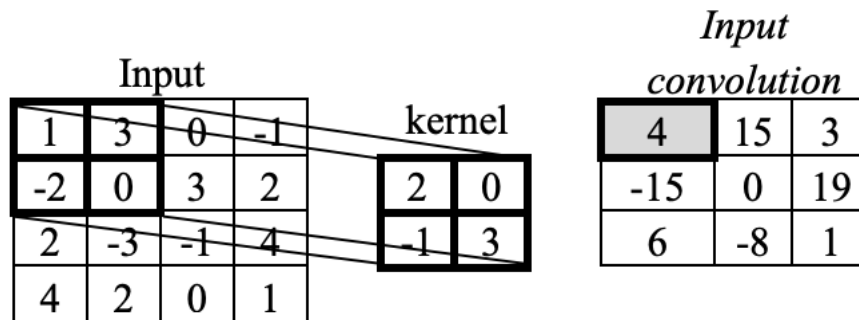
Convolution layer merupakan proses utama yang mendasari jaringan arsitektur CNN dan terdiri atas kernel. Kernel-kernel pada lapisan ini disebut filter konvolusi. Kernel berfungsi mempelajari fitur-fitur lokal pada *feature map*. Tahap *convolutional layer* melakukan operasi konvolusi pada output dari layer sebelumnya. Konvolusi adalah istilah matematis dimana pengaplikasian sebuah fungsi pada output fungsi lain secara berulang. Persamaan konvolusi merupakan persamaan pada dua fungsi argument bernilai riil. Operasi konvolusi $s(t)$ dapat ditunjukkan pada persamaan 1 berikut:

$$s(t) = \sum I(a).K(t-a) \quad (1)$$

Dimana $I(a)$ adalah input dan $K(a)$ adalah kernel. Input *convolution layer* merupakan citra yang direpresentasikan menjadi sebuah matriks. Operasi konvolusi menghasilkan nilai tinggi dan rendah pada posisi tertentu pada feature map. Posisi tertentu dari konvolusi kernel, merupakan perkalian untuk setiap nilai pada sel kernel dan nilai piksel gambar yang tumpang tindih dengan sel kernel (Khan et al., 2018). Adapun operasi perkaliannya menggunakan persamaan 2 berikut:

$$h(i, j) = \sum \sum W(a, b) * X(i+a-1, j+b-1) \quad (2)$$

Dimana operator *sigma* (\sum) digunakan untuk menjumlahkan hasil perkalian antara setiap elemen kernel W dengan nilai piksel yang sesuai dalam citra input X . Variabel a dan b digunakan sebagai indeks untuk mengakses elemen kernel dan nilai piksel yang tumpang tindih dengan kernel. Indeks a berjalan dari -1 hingga 1 dan indeks b berjalan dari -1 hingga 1. Sebagai contoh, ilustrasi pada proses convolution layer dengan nilai stride 2 dapat dilihat pada gambar 6.



Gambar 6 Ilustrasi proses *convolutional layer* (Khan et al., 2018)

Adapun perhitungan nilai pada input *convolution* yang diperoleh dengan menggunakan persamaan 2 adalah sebagai berikut:

$$\begin{aligned}
 h_{1,1} &= (1.2) + (3.0) + (-2. -1) + (0.3) = 4 \\
 h_{1,2} &= (3.2) + (0.0) + (0. -1) + (3.3) = 15 \\
 h_{1,3} &= (0.2) + (-1.0) + (3. -1) + (2.3) = 3 \\
 h_{2,1} &= (-2.2) + (0.0) + (2. -1) + (-3.3) = -15 \\
 h_{2,2} &= (0.2) + (3.0) + (-3. -1) + (-1.3) = 0 \\
 h_{2,3} &= (3.2) + (2.0) + (-1. -1) + (4.3) = 19 \\
 h_{3,1} &= (2.2) + (-3.0) + (4. -1) + (2.3) = 6 \\
 h_{3,2} &= (-3.2) + (-1.0) + (2. -1) + (0.3) = -8 \\
 h_{3,3} &= (-1.2) + (4.0) + (0. -1) + (1.3) = 1
 \end{aligned}$$

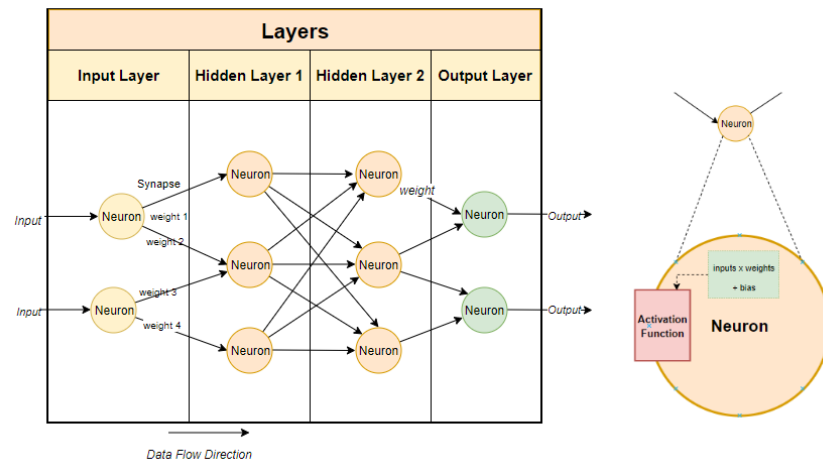
Dari perhitungan di atas, di peroleh matriks input convolution sebagai berikut:

$$\begin{bmatrix} 4 & 15 & 3 \\ -15 & 0 & 19 \\ 6 & -8 & 1 \end{bmatrix}$$

(Khan et al., 2018).

2.6.2. *Weight* dan *Bias*

Ketika input ditransmisikan antar *neuron*, *weights* diterapkan dan diteruskan ke fungsi aktivasi bersama dengan *bias*. Untuk lebih jelasnya perhatikan ilustrasi weights dan bias pada gambar 7.



Gambar 7 Ilustrasi *weights* dan *bias* pada *layer* (Malik, 2019)

Weights mengontrol sinyal (atau kekuatan koneksi) antara dua neuron. Dengan kata lain, bobot menentukan seberapa besar pengaruh input terhadap output. Saat jaringan neural dilatih pada set pelatihan, jaringan akan di inialisasi dengan satu set *weights*. *Weights* kemudian dioptimasi selama periode pelatihan untuk menghasilkan *weights* yang optimal.

Bias adalah nilai konstanta yang ditambahkan ke *layer* input dengan *weights*. *Bias* digunakan untuk mengimbangi hasil dan untuk menggeser hasil fungsi aktivasi ke nilai positif atau negatif. Dalam proses perhitungannya, neuron akan menghitung weighted sum dari input dengan persamaan:

$$Y = \sum(\text{weight} * \text{input}) + \text{bias} \quad (3)$$

Dimana input merupakan :

$$x1, x2, \dots, xn \quad (4)$$

Dan weight adalah

$$w1, w2, \dots, wn \quad (5)$$

Maka weighted sum dapat dihitung dengan persamaan :

$$Y = (x1 * w1) + (x2 * w2) + \dots + (xn * wn) + \text{bias} \quad (6)$$

Setiap input x_i dikalikan dengan bobot w_i yang sesuai, dan hasil perkalian tersebut dijumlahkan dengan tambahan bias. Hasil akhir dari rumus tersebut adalah nilai Y yang merupakan *weighted sum* dari *input* dengan bobot dan *bias* yang telah ditentukan.

2.6.3. Fungsi Aktivasi

Fungsi aktivasi merupakan operasi matematik yang dikenakan pada sinyal output y . Fungsi aktivasi berfungsi menentukan apakah suatu neuron aktif atau tidak berdasarkan weighter sum dari suatu input. Beberapa jenis fungsi aktivasi yang sering digunakan pada *deep learning* adalah *sigmoid*, *Tanh*, *algebraic sigmoid*, *ReLU*, *noisy ReLU*, *Leakly ReLU/PReLU*, *Randomized Leakly ReLU*, dan *Eksponential Linear Unit*. Penelitian ini menggunakan dua fungsi aktivasi yaitu fungsi aktivasi *ReLU* dan fungsi aktivasi *softmax*. Penjelasan dari kedua fungsi tersebut adalah sebagai berikut:

1. *ReLU*

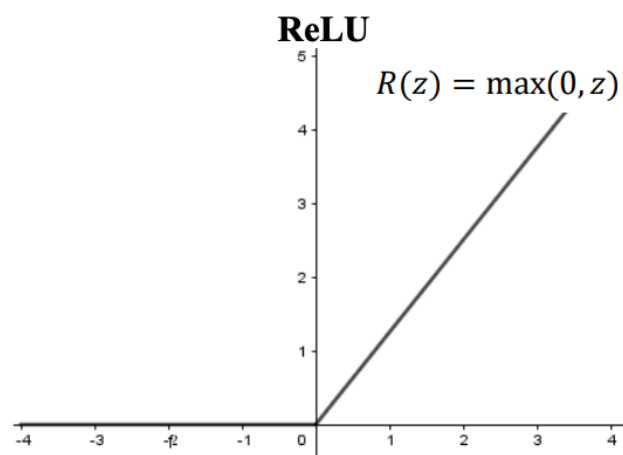
Fungsi aktivasi *Rectifier Linear Unit (ReLU)* merupakan fungsi aktivasi sederhana yang memiliki kepentingan praktis khusus karena perhitungannya yang cepat. Kelebihan fungsi aktivasi *ReLU* dibandingkan dengan fungsi aktivasi lain adalah sebagai berikut:

- Fungsi aktivasi *ReLU* merupakan fungsi aktivasi default ketika mengembangkan multilayer perceptron dan *convolutional neural network*.
- Fungsi aktivasi *ReLU* mengatasi masalah *gradient descent* yang hilang, yang memungkinkan model belajar lebih cepat dan berkinerja lebih baik.
- Menemukan cara melatih jaringan dengan lebih cepat, sehingga mengurangi kemungkinan terjadinya *overfitting*.

Fungsi aktivasi *ReLU* memetakan input ke 0 jika negatif dan mempertahankan nilainya jika positif. Representasi fungsi *ReLU* adalah sebagai berikut:

$$f(x) = \{ x, \text{ jika } x \geq 0, 0, \text{ jika } x < 0 \} \quad (7)$$

Adapun grafik nya dapat dilihat pada gambar 8.



Gambar 8 Grafik Fungsi Aktivasi *ReLU* (Khan et al., 2018).

2. Softmax

Fungsi aktivasi *softmax* merupakan fungsi input vektor dari bilangan *real* K , yang kemudian dinormalkan menjadi distribusi probabilitas yang terdiri atas probabilitas K yang proporsional ke eksponensial input. Komponen vektor pada softmax memiliki interval $(0,1)$. Fungsi *softmax* merupakan lapisan yang menghubungkan antara *fully connected layer* dengan *dense connection*. *Softmax* berfungsi untuk menghitung probabilitas pada setiap kelas target yang memungkinkan dan akan membantu menentukan kelas target pada input yang diberikan. Nilai *softmax* berada pada rentang probabilitas output dari 0 hingga 1 dan jumlah semua probabilitas sama dengan satu. Definisi fungsi *softmax* $\sigma = \mathbb{R}^K \rightarrow \mathbb{R}^K$ dituliskan sebagai persamaan 8 berikut:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (8)$$

dimana $i = 1, K$ dan $z = (z_1, \dots, z_K) \in \mathbb{R}^K$. Adapun contoh ilustrasi fungsi aktivasi *softmax* dapat dilihat pada gambar 9 sebagai berikut:

$$\begin{array}{ccc} \text{Logits score} & \text{Softmax} & \text{Probabilitas} \\ y = \begin{cases} 2.0 \\ 1.0 \\ 0.1 \end{cases} & \longrightarrow \sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} & \longrightarrow \begin{cases} P=0.7 \\ P=0.2 \\ P=0.1 \end{cases} \end{array}$$

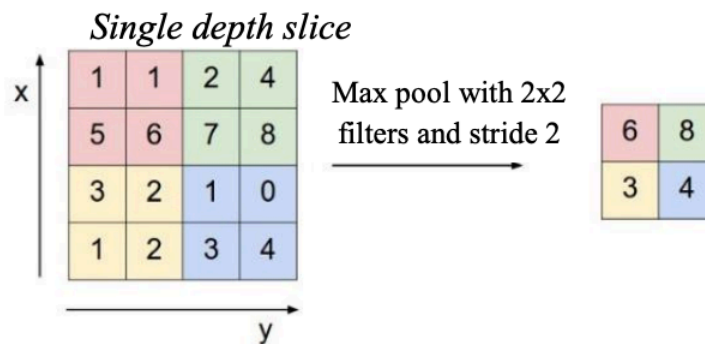
Gambar 9 ilustrasi fungsi aktivasi *softmax*

Logits score menunjukkan lapisan neuron terakhir sebagai output mentah pada lapisan terakhir *neural network* sebelum proses aktivasi berlangsung. Setelah output diproses dengan *softmax* akan menghasilkan nilai probabilitas dengan jumlah 1 (Khan et al., 2018).

3. Pooling layer

Pooling merupakan pengurangan ukuran matriks dengan menggunakan operasi *pooling*. *Pooling Layer* biasanya berada setelah *conv layer*. Pada dasarnya *pooling layer* terdiri dari sebuah filter dengan ukuran dan stride tertentu yang akan secara bergantian bergeser pada seluruh area *feature map*. Dalam *pooling layer* terdapat dua macam *pooling* yang biasa digunakan yaitu *average pooling* dan *max pooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata, sedangkan pada *max-pooling* adalah nilai maksimal. Lapisan *pooling* yang dimasukkan di antara lapisan konvolusi secara berturut-turut dalam arsitektur CNN dapat secara progresif mengurangi ukuran volume output pada *feature map*, sehingga mengurangi jumlah parameter pada jaringan untuk mengendalikan

overfitting. Lapisan pooling bekerja pada setiap tumpukan *feature map* dan melakukan pengurangan pada ukurannya. Bentuk lapisan *pooling* umumnya dengan menggunakan filter dengan ukuran 2x2 yang diaplikasikan dengan langkah (stride) sebanyak dua dan beroperasi pada setiap irisan dari inputnya. Berikut ini adalah contoh gambar operasi *max pooling*:



Gambar 10 Operasi *max pooling* (Khan et al., 2018)

Output dari proses *pooling* adalah sebuah matriks dengan dimensi yang lebih kecil dibandingkan dengan citra awal. Lapisan *pooling* akan beroperasi pada setiap irisan kedalaman volume input secara bergantian. Operasi max-pooling pada gambar 10 di atas menggunakan ukuran filter 2x2. Masukan pada proses tersebut berukuran 4x4. Dari masing-masing 4 angka pada input operasi tersebut diambil nilai maksimalnya kemudian dilanjutkan membuat ukuran output baru menjadi ukuran 2x2.

4. Fully Connected

Layer Fully connected layer merupakan sebuah lapisan dimana semua neuron aktivasi dari lapisan sebelumnya terhubung dengan neuron lapisan selanjutnya. Pada dasarnya lapisan ini biasanya digunakan pada MLP (*Multi Layer Perceptron*) yang mempunyai tujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Perbedaan *fully connected layer* dengan konvolusi biasa adalah neuron pada lapisan konvolusi terhubung hanya ke daerah tertentu sedangkan *fully connected* memiliki neuron yang semuanya terhubung (Khan et al., 2018). Namun keduanya masih menggunakan operasi dot.

5. Crossy Entropy Loss Function

Loss Function merupakan salah satu komponen penting dalam *neural network*. *Loss* menggambarkan kemungkinan kesalahan yang dihasilkan oleh model. *Loss Function* yang

baik adalah fungsi yang diharapkan menghasilkan error yang paling rendah. Ketika suatu model memiliki kelas yang cukup banyak, perlu adanya cara untuk mengukur perbedaan antara probabilitas hasil hipotesis dan probabilitas kebenaran yang asli. *Categorical Cross Entropy* merupakan salah satu pilihan terbaik untuk menghitung nilai loss pada permasalahan *multi-class classification*. *Categorical Cross Entropy* (CE) biasa juga disebut *Softmax Loss* yang merupakan gabungan dari *softmax activation* dan *cross-entropy loss*. Berikut merupakan rumus *Categorical Cross Entropy* (Gomez, 2018).

$$CE = - \sum(y * \log(p)) \quad (9)$$

Di mana:

- CE adalah nilai Cross Entropy
- Σ adalah operasi penjumlahan
- y adalah label yang sebenarnya dalam bentuk one-hot encoding (vektor dengan nilai 1 di indeks kelas yang benar dan 0 di indeks kelas lainnya)
- p adalah prediksi model dalam bentuk probabilitas kelas-kelas yang mungkin

Rumus ini menghitung perbedaan antara distribusi probabilitas prediksi model (p) dan distribusi probabilitas yang sebenarnya (y) dengan menggunakan logaritma negatif dari probabilitas prediksi yang benar.

6. *Precision*, *Recall* dan *Akurasi*

Precision, *recall* dan akurasi digunakan untuk mengukur kinerja sistem, *Precision* adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. *Recall* merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. Akurasi merupakan tingkat kedekatan antara nilai prediksi dengan nilai aktual.

2.7 Confusion Matrix

Confusion matrix menurut Han dan Kamber (2011) dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan analisis apakah *classifier* tersebut baik dalam mengenali tuple dari kelas yang berbeda. Nilai dari *True Positive* dan *True-Negative* memberikan informasi ketika *classifier* dalam melakukan klasifikasi data bernilai benar, sedangkan *False-Positive* dan *False Negative* memberikan informasi ketika *classifier* salah dalam melakukan klasifikasi data.

		Predicted class		Total
		<i>yes</i>	<i>no</i>	
Actual class	<i>yes</i>	<i>TP</i>	<i>FN</i>	<i>P</i>
	<i>no</i>	<i>FP</i>	<i>TN</i>	<i>N</i>
Total		<i>P'</i>	<i>N'</i>	<i>P + N</i>

Gambar 11 *Confusion Matrix* menampilkan total *positive* dan *negative tuple*

(Fibrianda and Bhawiyuga, 2018)

TP (*True Positive*) → Jumlah data dengan nilai sebenarnya positif dan nilai prediksi positif

FP (*False Positive*) → Jumlah data dengan nilai sebenarnya negatif dan nilai prediksi positif

FN (*False Negative*) → Jumlah data dengan nilai sebenarnya positif dan nilai prediksi negatif

TN (*True Negative*) → Jumlah data dengan nilai sebenarnya negatif dan nilai prediksi negatif