

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang berhubungan dengan metode *convolutional neural network* sudah banyak dilakukan oleh peneliti-peneliti sebelumnya diantaranya yang berjudul “Penerapan Algoritma *Convolutional Neural Network* Untuk Klasifikasi Citra Ekspresi Wajah Manusia Pada *MMA Facial Ekspression Dataset*”. Pada penelitian ini akan dilakukan klasifikasi citra ekspresi wajah dengan menerapkan *convolutional neural network* pada *dataset MMA facial ekspression*. Dimana data dibagi menjadi 2 kelas, yaitu *happy* dan *sad*. Berdasarkan hasil pengujian dapat disimpulkan bahwa CNN dapat melakukan klasifikasi citra ekspresi wajah manusia dengan baik menggunakan optimizer SGD dengan nilai akurasi 63% [9].

Penelitian kedua menggunakan “Algoritma *Convolutional Neural Network* Untuk Klasifikasi Tanaman *Aglaonema* Berdasarkan Citra Daun”. Sistem ini dibuat dengan *convolutional neural network* (CNN) dengan menggunakan arsitektur *ResNet5v2*. Sistem ini menggunakan *dataset* sebanyak 1960 citra gambar dari empat jenis *aglaonema* yang berbeda dengan ciri-ciri yang hampir sama yang sering ditemukan dijual dipasaran yaitu *red anjamani*, *red majesty* dan *black maroon*, sementara pada *model* tidak ber-*background* menghasilkan akurasi *testing* sebesar 71% [10].

Penelitian ketiga “Pengenalan Ekspresi Wajah Menggunakan Metode SVM dengan Transformasi *Fourier* Dan *PCA*”. Dalam penelitian ini dilakukan pengenalan ekspresi menggunakan metode SVM dengan ekstraksi ciri transformasi *fourier* dan direduksi menggunakan *PCA*. Tahapan penelitian dimulai dengan mengambil *dataset FER2013*, dilanjutkan dengan pengkonversian data piksel menjadi gambar, selanjutnya gambar diekstraksi menggunakan transformasi *fourier* lalu direduksi menggunakan *PCA*, setelah itu diklasifikasi menggunakan SVM dimana hasil pengujian menggunakan data original mendapatkan model terbaik *kernel polinomial* derajat dengan $c = 10$ yang memberikan akurasi sebesar 38% untuk citra gambar bagus dan 32.666% untuk

citra gambar dengan *noise salt and pepper*, sementara itu hasil terbaik pengujian menggunakan data dengan peningkatan kontras mendapatkan *model* terbaik pada *kernel polinomial* derajat dengan $c = 1$ memberikan hasil akurasi sebesar 36.666% untuk citra gambar bagus dan 33.333% untuk citra gambar dengan *noice salt and pepper* menggunakan data *testing*[11].

Penelitian keempat “Sistem Deteksi Wajah Keamanan Pintu Menggunakan Metode *Convolutional Neural Network* (CNN) Berbasis Arduino”. Dalam penelitian ini sistem yang akan digunakan adalah teknologi *biometrik* pengenalan wajah atau deteksi wajah, yaitu menggunakan ciri-ciri dan fisik manusia menggunakan kamera atau *webcam* untuk menangkap wajah manusia. Sistem deteksi wajah dirancang untuk mendeteksi wajah seseorang dengan menggunakan metode *convolutional neural network* (CNN). Identifikasi wajah yang dilakukan yaitu menangkap fitur-fitur pada wajah seperti posisi yang berbeda, jarak pandang wajah ke kamera dan gaya ekspresi wajah. Hasil pengujian menunjukkan sistem dapat mengetahui atau mengenali wajah yang sudah terdaftar dan yang belum terdaftar sesuai harapan yang diinginkan. Sistem deteksi wajah untuk keamanan pintu dengan metode *convolutional neural network* (CNN) berbasis arduino, dapat meningkatkan keamanan pintu dengan memperoleh tingkat akurasi sebesar 76,6% dengan jarak maksimal 30 cm dalam kondisi pencahayaan terang dan gelap[12].

Penelitian kelima “Implementasi Algoritma *Convolutional Neural Network* Untuk Klasifikasi Varietas Pada Citra Daun Sawi Menggunakan Keras”. Penelitian ini menghasilkan data uji coba dari klasifikasi citra pada sayuran varietas sawi yaitu sawi pokcoy, sawi putih, dan sawi caisim menghasilkan nilai akurasi sebesar 83%, *recall* 80% dan *presisi* 89%. Ketika data di *training* dilakukan perbedaan dengan jumlah perbandingannya maka hasil yang didapatkan adalah sama yaitu 83% [13].

Kesimpulan dari pemaparan jurnal terkait diatas menunjukkan bahwa metode *convolutional neural network* cocok diterapkan untuk pengenalan citra dan pengenalan ekspresi wajah, metode *convolutional neural network* juga memiliki tingkat keakurasian yang lebih baik dibandingkan dengan metode SVM untuk pengenalan citra. Sehingga menjadi pilihan baik untuk penelitian pengenalan ekspresi wajah menggunakan metode *convolutional neural network*. Berikut perbedaan penelitian terkait dapat dilihat pada tabel 2.1.

Tabel 2. 1 Perbandingan penelitian terkait

No	Penulis	Judul	Hasil	Perbedaan	Tujuan
1.	Tinaliah Triana, Elizabeth 2021	Penerapan Algoritma <i>Convolutional Neural Network</i> Untuk Klasifikasi Citra Ekspresi Wajah Manusia Pada MMA <i>Facial Ekspression Dataset</i>	Menghasilkan tingkat akurasi 63%	Penggunaan 2 <i>dataset</i> yaitu <i>happy</i> dan <i>sad</i>	Untuk mengklasifikasi citra ekspresi wajah menggunakan algoritma <i>convolutional neural network</i>
2.	Satrio Muhammad, Agung Toto Wibowo, 2021	Klasifikasi Tanaman Aglaonema Berdasarkan Citra Daun Menggunakan Metode <i>Convolutional Neural Network</i> (CNN)	Menghasilkan tingkat akurasi <i>testing</i> sebesar 71%	Penggunaan metode CNN untuk klasifikasi tanaman aglaonema	Mengenali tanaman aglaonema berdasarkan citra daun menggunakan metode <i>convolutional neural network</i>

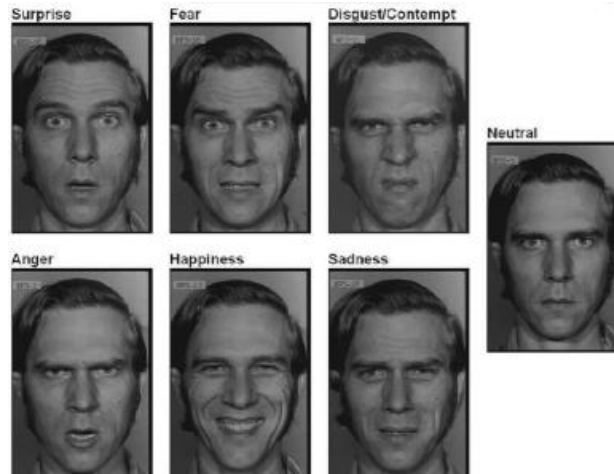
No	Penulis	Judul	Hasil	Perbedaan	Tujuan
3.	Ricky Julianto, Derry Alamsyah, 2021	Pengenalan Ekspresi Wajah Menggunakan Metode SVM dengan Transformasi Fourier Dan PCA	Menghasilkan akurasi terbesar 36.5%	Metode ekstraksi ciri menggunakan transformasi fourier dan PCA	Penerapan metode SVM dan PCA untuk mengenali ekspresi wajah
4.	Kiki Wahyuddin, Deden Wahiddin, Dwi Sulistya Kusumaningrum, 2023	Sistem Deteksi Wajah Keamanan Pintu Menggunakan Metode <i>Convolutional Neural Network</i>	Menghasilkan akurasi sebesar 76,6%	Pengujian pada penelitian tersebut menggunakan arduino	Sebagai alat keamanan pintu dengan mendeteksi wajah
5.	Ahmad Kurniadi, Kusrini, Moh. Fal Sadikin, 2023	Implementasi <i>Convolutional Neural Network</i> Untuk klasifikasi Varietas Pada Citra Daun Sawi Menggunakan Keras	Menghasilkan akurasi sebesar 83%	Penggunaan metode CNN untuk mengklasifikasi citra gambar sawi	Mengklasifikasikan daun sawi menggunakan <i>convolutional neural network</i>

2.2 Landasan Teori

2.2.1 Ekspresi Wajah

Ekspresi wajah (*facial expression recognition*) merupakan salah satu penerapan pada *deep learning* di bidang pengolahan citra yang berguna untuk mendeteksi emosi manusia menggunakan algoritma *neural network*. Dalam penerapan *facial expression recognition* secara umum untuk mendeteksi emosi, langkah yang digunakan adalah dengan cara melakukan dari nilai piksel setiap emosi dan secara urutan untuk mendapatkan informasi dalam *frame dataset*[14]. Saat ini, teknologi *face recognition*, sudah digunakan di berbagai bidang seperti kepolisian hingga pemerintahan. Dibidang kepolisian, *face recognition* digunakan untuk indentifikasi forensik, sedangkan diperusahaan metode ini digunakan untuk akses keamanan pada area tertentu. Bahkan baru-baru ini, pemerintah *United States* menggunakan *face recogniton* untuk mencari dan mengidentifikasi anak yang hilang pada internet.

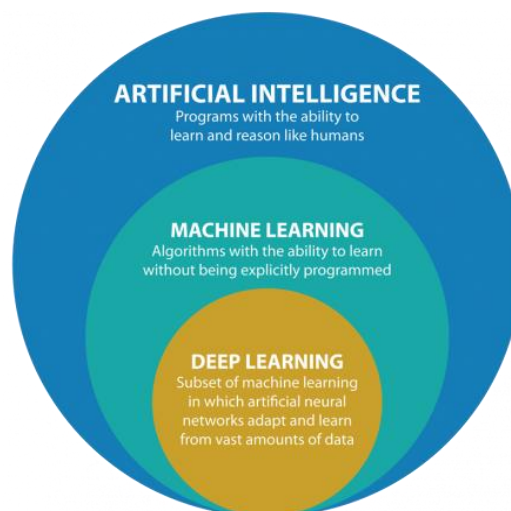
Cara kerja metode *face recognition* ini yaitu dengan mengambil gambar wajah seseorang, mengekstraksi gambar tersebut, kemudian membandingkannya dengan gambar yang ada pada *database* dan mengidentifikasi apakah cocok atau tidak. Karena metode ini menggunakan komputasi dalam pemrosesannya, maka tidak bisa disamakan dengan mata manusia. Oleh karena itu, hasilnya sangat bergantung pada kualitas gambar dan algoritma *recognition* yang digunakan. Saat ini banyak peneliti yang berlomba-lomba untuk menemukan algoritma *recognition* yang mampu memberikan hasil terbaik. Salah satu tantangan utama dari *face recognition* ini adalah kemampuan dan akurasi algoritma dalam mengidentifikasi dan memverifikasi seseorang dengan ribuan *database* yang memiliki banyak sudut pandang kamera, pencahayaan, *non-rigid deformation* dan kondisi pencitraan. Masalah tentang akurasi dalam *face recognition* itulah yang saat ini masih terus dicari solusinya, demi mendapatkan akurasi terbaik yang mendekati kemampuan visual manusia (encyclopedia.com, 2019). Emosi dapat dilihat dari perubahan pada raut wajah, seperti kerutan pada kening dan kedipan mata (L.Pt. Purnamaningsih, Ni Kt. Suarni, 2019). Contoh ilustrasi gambar citra emosi pada wajah ditunjukkan pada gambar 2.1 berikut:



Gambar 2. 1 Contoh citra emosi(Fabri, 2004)

2.2.2 *Artificial Intelligence*

Kecerdasan buatan (*Artificial Intelligence*) merupakan salah satu bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik manusia, bahkan bisa lebih baik dari pada yang dilakukan manusia. Kecerdasan buatan juga merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal, dalam pandangan manusia adalah cerdas. Salah satu teknologi kecerdasan buatan adalah sistem pakar yang merupakan program komputer yang dapat meniru proses pemikiran pengetahuan pakar untuk menyelesaikan suatu masalah yang spesifikasi[15]. *Artificial Intelligence* bisa diibaratkan payung yang lebih luas, dimana *machine learning* dan *deep learning* berada dalam lingkungannya.



Gambar 2. 2 Hubungan AI, *Machine Learning*, *Deep Learning*(ilmuti.org)

Kecerdasan buatan sebenarnya sudah dimulai sejak musim panas tahun 1956. Pada waktu itu sekelompok pakar komputer dan peneliti dari disiplin ilmu lain dari berbagai akademi, industri serta berbagai kalangan berkumpul di *Dartmouth College* untuk membahas potensi komputer dalam rangka menirukan atau mensimulasi kepandaian manusia. Beberapa ilmuwan yang terlibat adalah allen newel, herbert simon, marvin miskey, oliver selfridge, dan john mccarthy. Sejak saat itu, para ahli mulai bekerja keras untuk membuat, mendiskusikan, merubah dan mengembangkan sampai mencapai titik kemajuan yang penuh. Mulai dari laboratorium sampai pada pelaksanaan kerja nyata[1].

2.2.3 *Machine Learning*

Machine learning atau dikenal dengan pembelajaran mesin adalah ilmu komputer yang bisa bekerja tanpa diprogram secara eksplisit. Banyak peneliti berpikir bagaimana cara untuk membuat kemajuan menuju AI terhadap tingkat manusia. *Machine learning* ini merupakan kecerdasan buatan yang mempelajari bagaimana membuat data. *Machine learning* bisa disingkat dengan ML. Ini dibutuhkan untuk menerapkan teknik yang cepat dan kuat dalam menemukan masalah baru. Secara definisi, *machine learning* merupakan ilmu atau studi yang mempelajari tentang algoritma dan *model* statistik yang digunakan oleh sistem komputer untuk melakukan *task* tertentu tanpa intruksi eksplisit. *Machine learning* bergantung pada pola dan kesimpulan. Untuk mendapatkan pola dan kesimpulan tersebut, algoritma *machine learning* menghasilkan *model* matematika yang didasari dari data sampel yang sering disebut dengan *training* data[16].

Pengertian *machine learning* menurut putra (2019) adalah teknik untuk melakukan inferensi (menitik beratkan ranah hubungan variabel) terhadap data dengan pendekatan matematis. Inti *machine learning* adalah untuk membuat *model* (*matematis*) yang merefleksikan pola-pola data. *Machine learning* memungkinkan

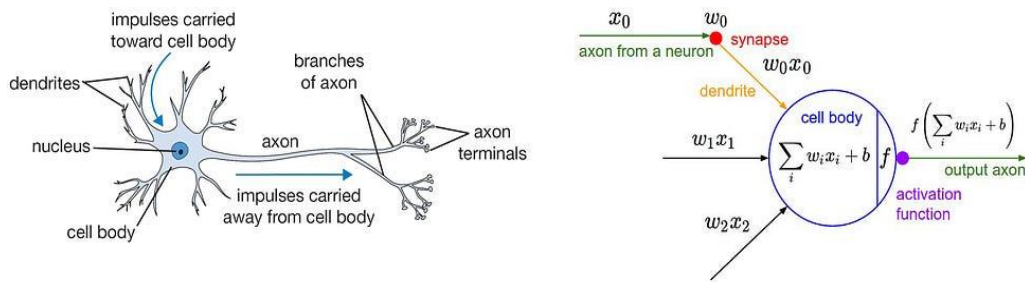
komputer atau suatu program dapat menemukan pengetahuan tanpa diprogram secara eksplisit[17]. Dalam *machine learning*, komputer menganalisis dan mempelajari pola dan informasi tersembunyi dari data pelatihan untuk melakukan tugas tertentu, seperti pengenalan pola, prediksi, klasifikasi, atau pengambilan keputusan. *Machine learning* menggunakan sebuah algoritma yang akan membuat

komputer untuk belajar dan melakukan tugasnya tanpa harus adanya intruksi dari pengguna. Algoritma ini bekerja dengan cara membangun sebuah *model* dari masukan agar dapat menghasilkan suatu prediksi atau pengambilan keputusan berdasarkan data yang ada.

2.2.4 *Artificial Neural Network*

Jaringan syaraf tiruan atau sering disebut *neural network* merupakan teknik komputasi yang didasari pada jaringan syaraf biologis yang terdapat di dalam otak manusia. Elemen pemrosesan di dalam jaringan syaraf tiruan yang dikenal dengan *neuron* terhubung ke elemen pemrosesan lainnya. *Neuron-neuron* tersebut diatur dalam lapisan atau vektor, dengan *output* dari satu lapisan berfungsi sebagai *input* ke lapisan berikutnya. Jaringan syaraf tiruan (JST) terdiri dari lapisan *input*, *hidden*, dan *output* dengan *neuron* yang saling terhubung untuk mensimulasikan otak manusia. Seperti jaringan syaraf pada otak manusia, JST juga mempunyai kemampuan untuk beradaptasi terhadap *input* yang diberikan melalui proses belajar[18].

Algoritma *Artificial neural network* (ANN) muncul pertama kali ketika manusia ingin menggabungkan antara kemampuan otak manusia yang mampu untuk belajar dan komputer yang memiliki kemampuan untuk memproses banyak data dan menyimpannya. Karena banyak sekali masalah yang tidak mampu oleh algoritma biasa. *Artificial neural network* (ANN) merupakan suatu *model* komputasi paralel yang meniru fungsi dari sistem jaringan syaraf biologi otak manusia. Dalam otak manusia terdiri dari milyaran *neuron* yang saling berhubungan. Hubungan ini disebut *synapses*. Komponen *neuron* terdiri dari satu intel sel yang akan melakukan pemrosesan informasi, satu akson (*axon*) dan minimal satu *dendrit*. Informasi yang masuk akan diterima oleh *dendrit*. Selain itu, *dendrit* juga menyertai *akson* sebagai keluaran dari suatu pemrosesan informasi.

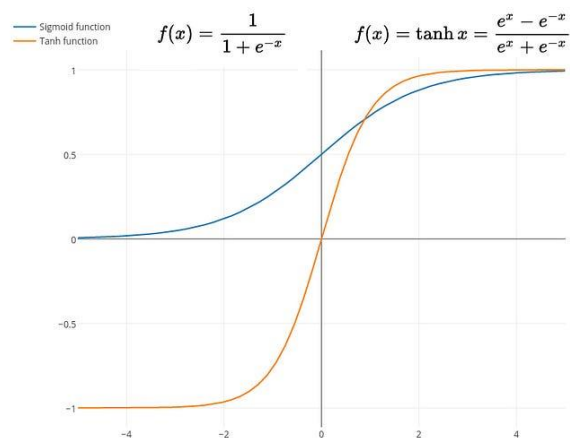


Gambar 2. 3 Ilustrasi neuron (Stanford Course)

Tiap *neuron* menerima *input* dan melakukan operasi dot dengan sebuah *weight*, menjumlahkannya (*weight sum*) dan menambahkan *bias*. Hasil dari operasi ini akan dijadikan parameter dari *activation function* yang akan dijadikan *output* dari *neuron* tersebut. Secara umum terdapat 2 jenis *activation function*, *linear* dan *non-linear activation function*. Secara umum, menurut artikel yang ditulis oleh Samuel sena (medium.com, 2017) ada 2 *activation function* yang sering digunakan dalam jaringan *neural network*, diantaranya:

1. Fungsi *Sigmoid* and *Tanh* (*Non-Linear*)

Fungsi *sigmoid* dan fungsi *tanh* merupakan salah satu fungsi yang sering digunakan untuk fungsi aktivasi pada *neural network multilayer*. Kedua fungsi aktivasi ini, sama-sama fungsi aktivasi *non-linear* yang fungsi ini biasanya digunakan untuk klasifikasi 2 class atau kelompok data.



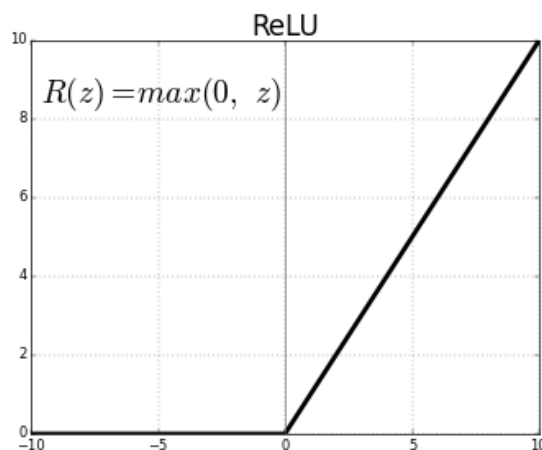
Gambar 2. 4 Grafik fungsi aktivasi sigmoid & tanh (Stanford Course)

Fungsi *sigmoid* mempunyai rentang antara 0 hingga 1 sedangkan rentang dari fungsi *tanh* adalah -1 hingga 1. Kedua fungsi ini hampir sama, hanya saja fungsi *tanh* merupakan pengembangan dari fungsi *sigmoid*. Tapi kedua fungsi ini

memiliki kelemahan yaitu, dapat mematikan *gradient*, ketika aktivasi dari *neuron* mengeluarkan nilai yang berada pada *range* 0 atau satu, dimana *gradient* di wilayah ini hampir bernilai 0. Kemudian *output* dari *sigmoid* tidak *zerocontered*.

2. Fungsi ReLu (*non-linear*)

ReLu atau *Rectified linear Unit* menjadi salah satu *activation function* yang populer belakangan ini, *vincent vanhoucke* dalam *course deep learningnya* di *udacity* mengatakan bahwa ReLu merupakan *activation function* favorit para *engineer* yang malas. Karena ReLu pada intinya hanya membuat pembatas pada bilangan nol, artinya apabila $z \leq 0$ maka $z = 0$ dan apabila $z \geq 0$ maka $z = z$.



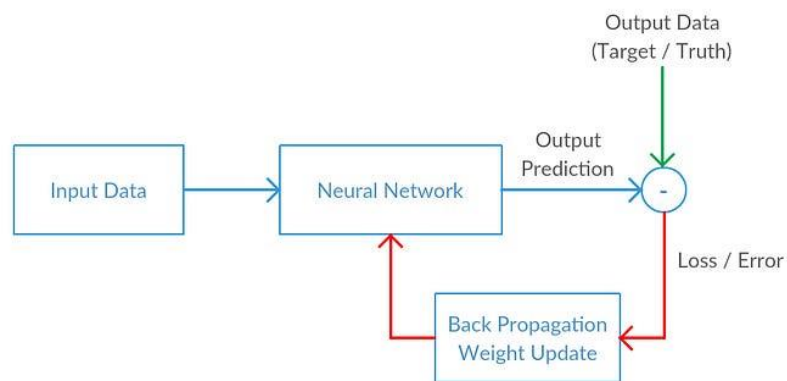
Gambar 2. 5 Grafik fungsi aktivasi ReLu (Stanford Course)

ReLu bisa diimplementasikan hanya dengan membuat pembatas (*threshold*) pada bilangan nol. Fungsi aktivasi ini memiliki kelebihan yaitu dapat mempercepat proses konfigurasi yang dilakukan dengan *Stochastic Gradient Descent* (SGD) jika dibandingkan dengan fungsi *sigmoid* dan *tanh*. Namun aktivasi ini juga memiliki kelemahan yaitu aktivasi ini bisa menjadi rapuh pada proses unit tersebut mati.

2.2.5 *Backpropagation*

Backpropagation yaitu sebuah metode atau algoritma pembelajaran yang termasuk ke dalam *supervised learning* yang memiliki banyak lapisan atau disebut dengan *multilayer perceptron* yang biasa digunakan oleh *perceptron* dalam melakukan perubahan bobot yang saling terhubung dengan *neuron* di dalam *hidden layer*[19]. *Neural network* merupakan suatu *model* komputasi yang sistemnya mengikuti syaraf otak manusia. Dan salah satu ciri khas dari otak

manusia adalah melakukan pembelajaran berdasarkan pengalaman. *Neural network* juga begitu, *model* ini membutuhkan proses pembelajaran untuk mengenali pola dari data yang dipelajarinya. Pembelajaran ini bertujuan untuk melakukan suatu proses dalam menentukan nilai bobot (*weight*) yang tepat untuk masing-masing *input*. Proses untuk melakukan pembelajaran ini biasanya terjadi pada saat proses pelatihan data (*training data*) biasanya disebut dengan *backpropagation*. *Backpropagation* merupakan salah satu dari metode pelatihan jaringan syaraf, dimana ciri dari metode ini adalah meminimalkan *error* pada *output* yang dihasilkan oleh jaringan. Jadi *backpropagation* ini bekerja dengan cara melakukan *update* nilai bobot pada *neuron* di *layer* sebelumnya.



Gambar 2. 6 *Backpropagation* (Stanford Course)

Seperti pada gambar diagram diatas tugas dari metode *backpropagation* adalah dengan melakukan *update* bobot berdasarkan nilai *error* yang didapatkan dari perbandingan antara nilai *output* dan target *output*. Proses *training* menggunakan *backpropagation* secara garis besar terdiri dari 2 tahap ; tahap maju (*forward pass*) dan tahap mundur (*backward pass*).

2.2.6 *Deep Learning*

Deep learning merupakan sebuah algoritma *neural network* yang menggunakan metadata sebagai *input* dan mengolah *input* tersebut menggunakan sekumpulan fungsi transformasi *non-linier* yang ditata berlapis-lapis dan mendalam. Pada *deep learning* terdapat *hidden layer* (lapisan tersembunyi) yang bertugas untuk melatih serangkaian fitur unik berdasarkan *output* dari jaringan sebelumnya[20]. Teknik *deep learning* adalah teknik *machine learning* yang

mengajarkan komputer agar memiliki kemampuan alami manusia, yaitu belajar dari pengalaman. Dengan adanya teknologi *deep learning*, produk-produk seperti *self-driving car*, *face recognition* atau *voice recognition* dapat dihasilkan. Ini adalah kunci untuk mengontrol suara di perangkat konsumen seperti ponsel, tablet, TV, dan lainnya. Teknik *deep learning* akhir-akhir ini mendapatkan banyak perhatian dari para peneliti karena pencapaian hasilnya yang bagus. Komputer akan memodelkan pembelajaran untuk klasifikasi langsung dari gambar, teks atau suara ketika menggunakan *deep learning*. Penggunaan *deep learning* ini dapat mencapai akurasi ketepatan yang tinggi, terkadang melebihi kinerja manusia pada beberapa kasus. *Model* dilatih dengan menggunakan sejumlah data yang besar dan dipadukan dengan arsitektur *neural network* yang mengandung banyak lapisan. Meskipun istilah *deep learning* pada *neural network* sudah ada sejak 2006, tapi beberapa tahun belakang ini istilah ini dimunculkan lagi. Hal itu dikarenakan 2 hal, yang pertama, metode *deep learning* membutuhkan *dataset* dalam jumlah besar, dengan begitu kemampuan akurasi yang tinggi dari *deep learning* ini baru terlihat benar-benar nyata. Alasan kedua, *deep learning* dengan *dataset* yang besar membutuhkan perangkat komputasi dengan performa yang tinggi untuk meningkatkan kecepatan pemrosesan. Kedua alasan inilah yang menjadi sebab metode *deep learning* kini menjadi sangat digemari oleh peneliti, karena saat perangkat komputasi seperti GPU dengan performa yang tinggi dan *cloud computing* yang semakin berkembang, membuat metode ini cepat berkembang.

Salah satu metode *deep learning* yang memanfaatkan *deep neural network* adalah *convolutional neural network* (CNN). Nama tersebut diambil dari operasi terpenting yang ada pada metode tersebut yaitu *convolutional*, yang kemudian digabung dengan metode *neural network*. CNN ini merupakan *model* terbaru dari algoritma *deep neural network* yang dikembangkan untuk memproses data dalam bentuk 2D data, seperti gambar. Berbeda dari algoritma pengolahan citra lainnya, algoritma CNN ini menghilangkan ekstraksi fitur secara manual. Jadi tidak perlu untuk mengekstraksi fitur untuk mengklasifikasi gambar tersebut. Fitur-fitur yang relevan tidak perlu dilatih terlebih dahulu, dan mempelajari fitur dari sebuah gambar ketika jaringan tersebut dijalankan. Ekstraksi fitur otomatis ini membuat

model deep learning memiliki akurasi tinggi untuk melakukan tugas *computer vision* seperti klasifikasi objek[21].

2.2.7 Confusion Matriks

Confusion matrixs merupakan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (*model*) dengan hasil klasifikasi sebenarnya. Untuk menghitung performa suatu *model* diperlukan parameter pengukuran yaitu tingkat akurasi, *recall*, dan *presisi*. *Confusion matrix* adalah tabel yang menyatakan klasifikasi jumlah data uji yang benar dan jumlah data uji yang salah[22]. Contoh *confusion matrix* untuk klasifikasi *biner* ditunjukkan pada tabel 2.2 berikut:

Tabel 2. 2 *Confusion Matriks*

Data		Aktual	
		True	False
Prediksi	True	TP	FP
	False	FN	TN

Kinerja klasifikasi dapat dihitung dengan menggunakan rumus berikut ini:

- a. Nilai akurasi (*acc*) merupakan presentasi prediksi berhasil atau bernilai benar. Dapat dihitung dengan rumus sebagai berikut:

$$Acc = \frac{TP+TN}{TP+FP+TN+FN}$$

- b. *Precision* Positif (*pp*) merupakan presentase prediksi data positif benar. Dapat dihitung dengan rumus sebagai berikut:

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

- c. *Precision* Negatif (*pn*) merupakan presentase prediksi data negatif benar. Dapat dihitung dengan rumus sebagai berikut:

$$PN = \frac{TN}{TN+FN}$$

- d. *Recall* Positif (*rp*) merupakan presentase data positif yang diprediksi sebagai positif. Dapat dihitung dengan rumus sebagai berikut:

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

- e. *Recall* Negatif (*rp*) merupakan presentase data negatif yang diprediksi sebagai negatif. Dapat dihitung dengan rumus sebagai berikut:

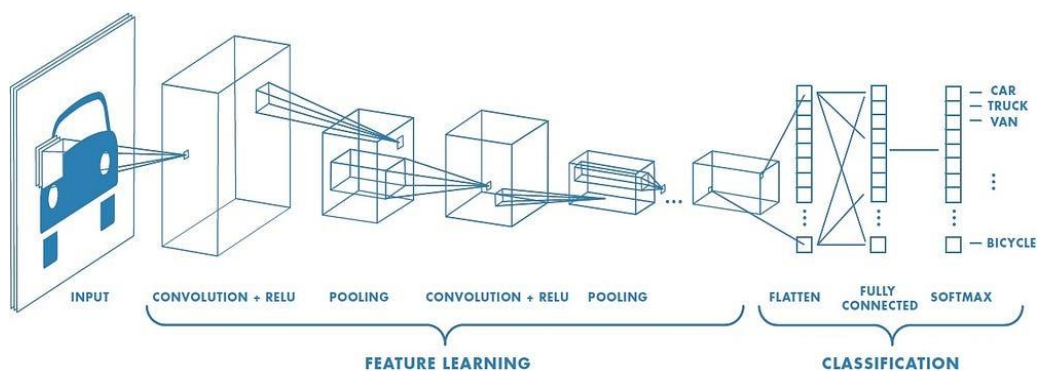
$$RN = \frac{TN}{TN+FP}$$

2.2.8 Citra digital

Citra digital adalah gambar dua dimensi yang bisa ditampilkan pada layar komputer sebagai himpunan/ diskrit nilai digital yang disebut *pixel/ picture elements*. Dalam tinjauan secara matematis, citra merupakan fungsi kontinu dari intensitas cahaya dalam bidang 2 dimensi. Citra digital merupakan citra $f(x,y)$ dimana dilakukan diskritisasi koordinat spasial/ *sampling* dan diskritisasi tingkat kwantisasi (kecemerlangan/ keabuannya). Citra digital adalah suatu fungsi intensitas cahaya $f(x,y)$, dimana nilai x dan nilai y adalah koordinat spasial. Nilai fungsi tersebut di setiap titik (x, y) adalah tingkat kecemerlangan citra di titik tersebut. Citra digital adalah sebuah *matriks* dimana *indeks* baris maupun kolomnya menyatakan sebuah titik pada citra tersebut dan elemen *matriksnya* (yang dikenal sebagai elemen gambar/ *picture element/ pixel/ pels*) menyatakan tingkat keabuan di titik tersebut[23].

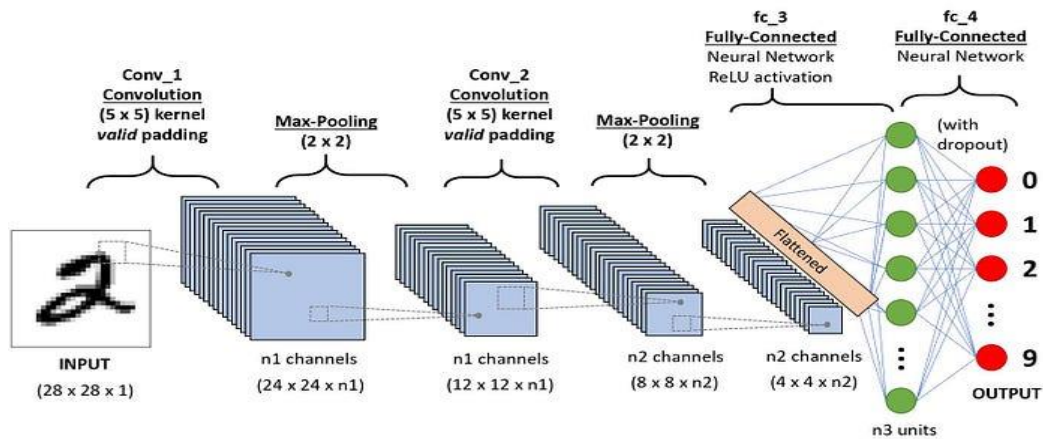
2.2.9 Convolutional Neural Network

Convolutional neural network (CNN) adalah salah satu metode paling populer digunakan untuk *deep learning*, sebuah *machine learning* yang *model* pembelajarannya dikhususkan untuk melakukan klasifikasi langsung pada 2 dimensi seperti gambar, video, teks, atau suara, namun pada penerapannya CNN juga dapat digunakan untuk tipe data 1 dimensi dan 3 dimensi[24]. Algoritma CNN akan sangat berguna khususnya ketika digunakan untuk mencari pola pada suatu gambar kemudian mengenali objek pada gambar tersebut. Bukan hanya pada objek atau benda saja, CNN ini sebenarnya juga bisa digunakan untuk mengenali wajah yang selama ini perlu segmentasi untuk meningkatkan akurasi[6].



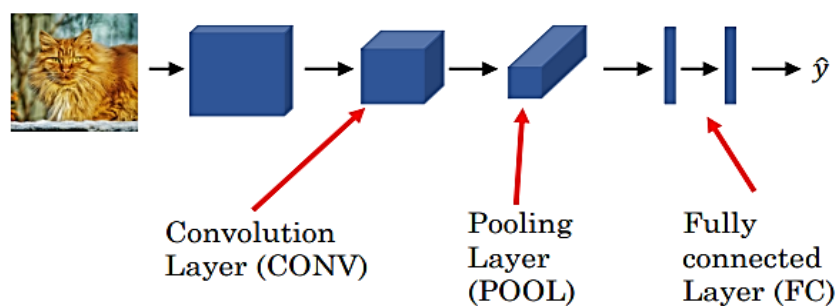
Gambar 2. 7 Arsitektur CNN (Stanford Course)

CNN dapat disebut juga jaringan syaraf tiruan yang melibatkan konvolusi (CNN = ANN + *convolution*).



Gambar 2. 8 Gambar Urutan CNN

CNN pertama kali dikembangkan dengan nama *NeoCognitron* oleh kunihiko fukushima, seorang peneliti dari NHK *Broadcasting Science Research Laboratories*, kinuta, setagaya, Tokyo, Jepang. Konsep tersebut kemudian dimatangkan oleh Yann LeChun[25], seorang peneliti dari AT&T Bell *Laboratories* di Holmdel, New Jersey, USA. Model CNN dengan nama *LeNet* berhasil diterapkan oleh LeChun pada penelitiannya mengenai pengenalan angka dan tulisan tangan. Pada tahun 2012, Alex Krizhevsky dengan penerapan CNN miliknya berhasil menjuarai kompetisi *ImageNet Large Scale Visual Recognition Challenge 2012*. Metode CNN terbukti berhasil mengungguli metode *machine learning* lainnya seperti SVM pada kasus klasifikasi objek pada citra[26].



Gambar 2. 9 Gambar lapisan CNN

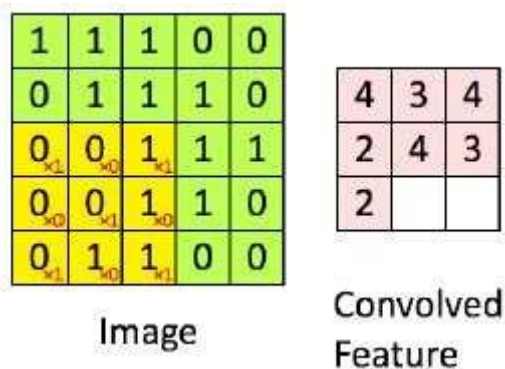
CNN terdiri dari tiga jenis layer yaitu *convolutional layer*, *layer pooling*, dan layer *fully connected*[27]. *Convolutional layer* memiliki sekumpulan filter yang dapat menggabungkan seluruh *input* gambar dan menghasilkan berbagai

jenis *feature map*. *Feature map* adalah sebuah *output* yang berupa *map* yang dihasilkan oleh proses konvolusi. *Layer pooling* mengikuti *layer convolutional* dan digunakan untuk mengurangi ukuran spasial *feature map* dan beban pada komputasi jaringan. *Average pooling* dan *max pooling* adalah dua strategi pengambilan *downsampling nonlinier* yang paling umum digunakan untuk menerjemahkan invarian. *Layer fully connected* (FC) umumnya digunakan di akhir jaringan untuk memastikan bahwa semua *neuron* di *layer* tersebut sepenuhnya terhubung ke *activation* di *layer* sebelumnya dan untuk mengaktifkan *feature map 2D* untuk diubah menjadi *feature map 1D* untuk fitur representasi dan klasifikasi lanjutan[9].

Secara umum model CNN terdiri dari 4 komponen, yaitu *convolution layer*, *pooling layer*, fungsi aktivasi, dan *fully connected layer*:

1. *Convolution Layer*

Convolution Layer melakukan operasi konvolusi pada *output* dari *layer* sebelumnya. *Layer* tersebut adalah proses utama yang mendasari sebuah CNN. Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada *output* fungsi lain secara berulang. *Convolutional layer* adalah tahap yang dilakukan untuk melakukan filter terhadap citra, dimana filter itu akan bergeser ke seluruh bagian gambar, dan menghasilkan *output* yang disebut *feature map*. Proses *convolution layer* dapat dilihat pada gambar 2.8 berikut:



Gambar 2. 10 Operasi Konvolusi (Stanford Course)

Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra *input*. Konvolusi akan menghasilkan transformasi *linear* dari data *input* sesuai informasi spasial pada data. Bobot pada *layer* tersebut

menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN.

a. *Stride*

Stride merupakan sebuah parameter untuk menyatakan jumlah pergeseran pada sebuah filter. Apabila *stride* bernilai 1, maka pada proses konvolusi *kernel* bergeser sebanyak 1 *pixel* secara *horizontal* lalu *vertikal*. Jika *stride* bernilai 2, *kernel* akan bergeser sebanyak 2 *pixel* secara *horizontal* lalu *vertikal*. Jika nilai *stride* yang digunakan semakin kecil, informasi yang diperoleh dari sebuah citra *input* akan semakin detail walaupun waktu komputasi yang dibutuhkan semakin besar. Meskipun demikian, dengan nilai *stride* yang kecil dan informasi yang diperoleh semakin detail bukan berarti bahwa performansi yang dihasilkan oleh sistem akan semakin baik[28].

b. *Padding*

Padding adalah parameter yang menentukan jumlah *pixels* (berisi nilai 0) yang akan ditambahkan di setiap sisi dari *input*. Hal ini digunakan dengan tujuan untuk memanipulasi dimensi *output* dari *convolutional layer (feature map)*. Tujuan dari penggunaan *padding* adalah dimensi *output* dari *conv. layer* selalu lebih kecil dari *inputnya* (kecuali penggunaan 1x1 filter dengan *stride* 1). *Output* ini akan digunakan kembali sebagai *input* dari *conv. layer* selanjutnya, sehingga makin banyak informasi yang terbuang. Dengan menggunakan *padding*, kita dapat mengatur dimensi *output* agar tetap sama seperti dimensi *input* atau setidaknya tidak berkurang secara drastis. Sehingga kita bisa menggunakan *conv. layer* yang lebih dalam/*deep* sehingga lebih banyak *features* yang berhasil di-*extract*. Meningkatkan performa dari *model* karena *conv. filter* akan fokus pada informasi yang sebenarnya yaitu yang berada diantara *zero padding* tersebut. Untuk menghitung dimensi dari *feature map* kita bisa gunakan rumus[29]:

$$output = \frac{W-N+2P}{S} + 1 \dots\dots\dots(4)$$

W = Panjang/Tinggi Input

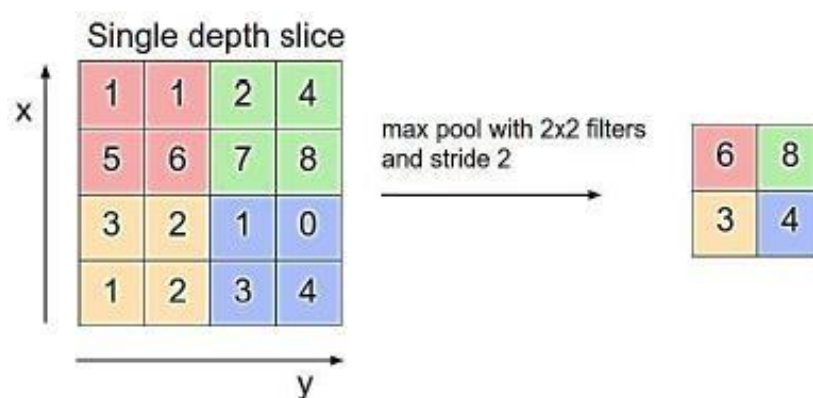
N = Panjang/Tinggi Filter

P = Zero Padding

S = Stride

2. Pooling Layer

Pooling layer dilakukan untuk mengurangi dimensi *feature map* sehingga mempercepat komputasi[30]. Pada dasarnya *pooling layer* ini terdiri dari sebuah filter dengan ukuran tertentu yang akan dioperasikan dengan *stride* tertentu ke *feature map* hasil *convolutional layer*. *Max pooling* akan mengambil nilai terbesar untuk menyusun *matriks* baru berdasarkan citra yang telah direduksi. Pada penerapannya *pooling layer* yang biasa digunakan di *pooling layer* ini yaitu *average pooling* dan *max pooling*. Perbedaan dari 2 metode ini adalah pada nilai rata-rata dari *matriks* yang mengalami operasi *pooling*. Sedangkan *max pooling* adalah mengambil nilai tertinggi.



Gambar 2. 11 Operasi *Max Pooling* (Stanford Course)

Berdasarkan ilustrasi diatas, pada *feature map* hasil konvolusi dilakukan operasi *max pooling*. Operasi untuk mengambil nilai tertinggi pada batasan ukuran filter tertentu. Dari operasi *max pooling* dengan *filter* 2x2 dan *stride* 2 tersebut didapatkan *feature map* baru dengan ukuran yang lebih kecil yaitu 2x2.

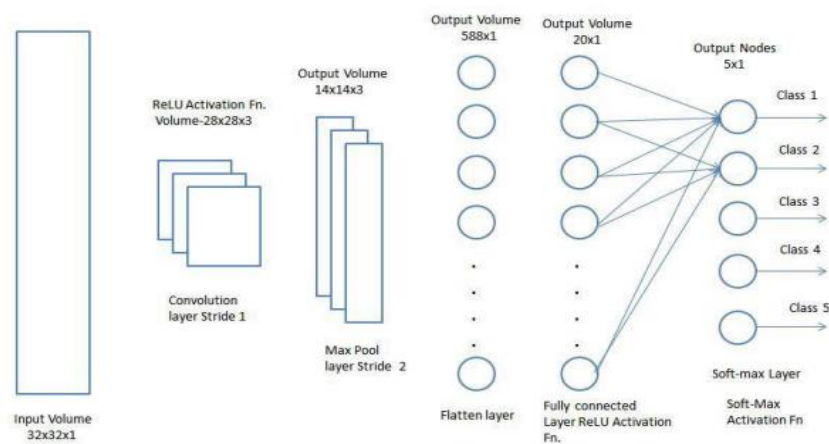
3. Activation Function (Fungsi Aktivasi)

Fungsi aktivasi sangat umum digunakan dalam *neural network*. Alasan utama menggunakan fungsi aktivasi adalah agar *neural network* mengenali data *non-linear*, karena *output* yang dihasilkan dari *neural network* jarang sekali bersifat *linear*. Fungsi aktivasi adalah fungsi *non-linear* yang membuat sebuah jaringan dapat mentransformasi data *input* ke dimensi yang lebih tinggi yang memungkinkan dilakukan klasifikasi. Pada CNN terdapat fungsi aktivasi digunakan, yaitu fungsi *sigmoid*. Fungsi *sigmoid* berfungsi mentransformasi nilai dari *input* x menjadi antara 0 dan 1 dengan bentuk distribusi fungsi[9]. Fungsi *sigmoid* memiliki bentuk, yaitu:

$$\sigma(x) = \frac{1}{(1+\theta-x)} \dots \dots \dots (5)$$

4. Fully Connected Layer

Fully connected layer yang bertujuan untuk menghubungkan semua *neuron* serupa seperti yang dimiliki oleh *multilayer perceptron* (MLP), yang bertujuan untuk mentransformasi dimensi data agar data dapat diklasifikasikan secara *linear*. Pada dasarnya lapisan ini sama halnya dengan lapisan *neural network* biasa, bisa dalam bentuk *single net* ataupun MLP. Tapi sebelum dilakukan proses klasifikasi, *feature map* yang dihasilkan dari *feature learning* tersebut masih berbentuk multi dimensial *array*, sehingga perlu mengubahnya menjadi bentuk *vector*, teknik ini disebut dengan *flatten*. *Flatten* merupakan teknik untuk *reshape feature map* menjadi sebuah *vektor* agar bisa digunakan sebagai *input* dari *fully connected layer*. Jadi *input* dari *fully connected layer* terdiri dari satu *neuron* hasil *reshape feature map* tadi menjadi *vektor*. Setelah dilakukan *flatten*, semua bobot tersebut akan diklasifikasi sesuai dengan banyaknya kelas[25]. Proses *fully connected layer* dapat dilihat pada gambar 2.10 berikut:



Gambar 2. 12 Proses Klasifikasi *Fully Connected Layer*

2.3 Kelengkapan Sistem

2.3.1 Python

Python adalah bahasa pemrograman tingkat tinggi yang sering digunakan dalam berbagai bidang seperti pengembangan perangkat lunak, analisis data, kecerdasan buatan, pengembangan web, dan lain-lain. *Python* dikembangkan oleh guido van rossum pada tahun 1991 dengan tujuan menyediakan sintaksis yang

jasas dan mudah dibaca, serta memungkinkan para pengembang untuk menulis kode yang lebih terstruktur dan ekspresif.

Salah satu fitur yang tersedia pada *python* adalah sebagai bahasa pemrograman dinamis lainnya, *python* umumnya digunakan sebagai bahasa *script* meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan bahasa *script*. *Python* dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan diberbagai *platform* sistem operasi[2].

Python adalah bahasa pemrograman yang sering digunakan dalam implementasi CNN karena menyediakan *library* dan kerangka kerja yang kuat untuk membangun, melatih, dan menggunakan model CNN. *Python* menyediakan fleksibilitas dan kemudahan dalam mengolah data, melatih *model*, dan menerapkan *model* dalam tugas pengolahan data spasial seperti pengenalan gambar dan video. *Python* memiliki beberapa *library* populer untuk *deep learning*, seperti *TensorFlow*, *Keras*, dan *PyTorch*, yang menyediakan API yang mudah digunakan untuk membangun, melatih, dan menggunakan model CNN. *Library* ini menyediakan berbagai fungsi dan algoritma yang dioptimalkan untuk tugas-tugas seperti pengenalan gambar, segmentasi, klasifikasi, dan deteksi objek. *TensorFlow* dan *keras* adalah *modul* yang populer untuk *deep learning*. *TensorFlow* adalah kerangka kerja yang kuat untuk pembelajaran mesin dan *deep learning*, sementara *keras* adalah antarmuka tingkat tinggi yang memudahkan pembuatan dan pelatihan model *deep learning*.

2.3.2 Anaconda

Anaconda adalah *platform* distribusi *python* yang dikembangkan oleh *anaconda, Inc.* *Platform* ini dirancang untuk menyederhanakan *instalasi* dan pengelolaan paket, serta membuat lingkungan pemrograman yang konsisten untuk pengembangan aplikasi *python*. *Anaconda* mencakup distribusi *python* itu sendiri, bersama dengan banyak paket, modul, dan alat tambahan yang sering digunakan dalam pengembangan ilmiah dan analisis data. *Anaconda* juga menyediakan *Anaconda Navigator*, sebuah antarmuka *grafis* yang memudahkan dalam menjelajahi dan mengelola lingkungan, paket, dan proyek *python*. Selain itu,

anaconda juga menyertakan *jupyter notebook*, lingkungan pengembangan interaktif yang populer untuk penulisan dan menjalankan kode *python*[31].

Anaconda merupakan salah satu aplikasi yang berfungsi sebagai distribusi bahasa pemrograman *Python* dan *R* yang memiliki sifat *open source*. *Python* banyak dimanfaatkan untuk berbagai perhitungan ilmiah, yang di dalamnya berupa *machine learning*, pengolahan data dengan ukuran besar, analisis prediksi, dan lain sebagainya. *Anaconda* memiliki tujuan untuk dapat menyederhanakan berbagai proses manajemen *package* ataupun *deployment*. *Anaconda* memiliki jumlah distribusi lebih dari 1500 *package* yang populer dan dapat diakses oleh berbagai *platform* sistem operasi seperti halnya *windows*, *linux*, dan *MacOS*.

Anaconda sangat populer dalam komunitas pengembangan data *science* dan analisis data karena menyediakan banyak paket populer seperti *NumPy*, *pandas*, *scikit-learn*, *matplotlib*, dan banyak lagi. *Anaconda* memudahkan pengguna untuk menginstal, mengatur, dan memulai proyek *python* yang berkaitan dengan ilmu data, analisis statistik, pembelajaran mesin, kecerdasan buatan, dan bidang lainnya.

2.3.3 Jupyter Notebook

Jupyter Notebook adalah lingkungan pengembangan yang diluncurkan pada tahun 2015 yang merupakan singkatan dari *Julia*, *Python*, dan *R*. *Jupyter Notebook* dapat digunakan dalam pembuatan software berbasis web dengan menggunakan bahasa pemrograman apapun, kelebihan inilah yang menjadi kemudahan bagi para programmer untuk berinovasi dengan menggunakan *Jupyter Notebook*.

2.3.4 TensorFlow

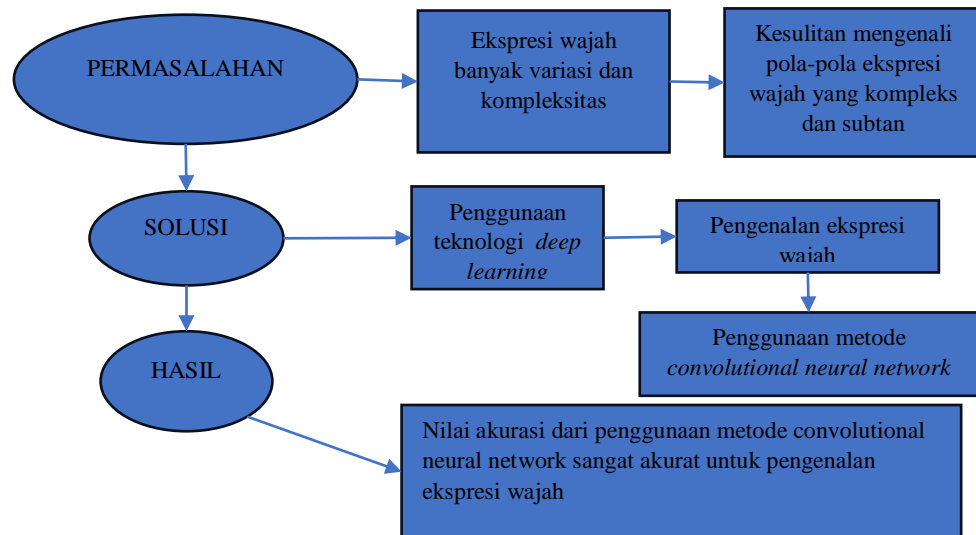
TensorFlow adalah *open source library* untuk *machine learning* yang di *release* oleh *Google* yang mendukung beberapa bahasa pemrograman. Dalam proses *transfer learning*. *Tensorflow* berperan untuk memproses *inception-v3 model* untuk di *training* ulang menggunakan data yang baru dan kemudian menghasilkan *classifier* dengan komputasi yang cepat dan akurasi yang baik. *Tensorflow* dapat digunakan pada semua sistem operasi

Tensorflow adalah *library* yang populer untuk *deep learning* dalam *python*. Ini menyediakan kerangka kerja yang kuat untuk membangun dan melatih model

neural network yang kompleks. *Keras*, di sisi lain, adalah API tingkat tinggi yang memudahkan pembuatan dan pelatihan model *deep learning*. *Anaconda* dapat memiliki beberapa kegunaan dalam penelitian pengenalan ekspresi wajah menggunakan *convolutional neural network* (CNN). *Anaconda* menyediakan manajer paket *conda* yang memungkinkan untuk menginstal, mengelola, dan mengatur paket-paket *python* yang dibutuhkan untuk penelitian pengenalan ekspresi wajah menggunakan CNN. Dan mudah menginstal dan memperbarui paket seperti *TensorFlow*, *Keras*, *OpenCV*, *scikit-learn*, dan *library* lainnya yang diperlukan untuk membangun dan melatih model CNN. [2].

2.4 Kerangka Pemikiran

Berdasarkan pengamatan yang penulis lakukan terdapat masalah pada pengenalan ekspresi wajah. Ekspresi wajah manusia memiliki banyak variasi dan kompleksitas. Metode tradisional seringkali kesulitan dalam memodelkan dan mengenali pola-pola ekspresi yang kompleks dan subtan. *Convolutioal neural network*, dengan kemampuannya dalam mengekstraksi fitur hierarkis dari data gambar, dapat mengatasi kekompleksan dan variasi ekspresi wajah dengan lebih baik. Pengenalan ekspresi wajah dapat membantu dalam pengenalan emosi manusia. Dengan mengidentifikasi ekspresi wajah, sistem dapat mengenali emosi seperti kebahagiaan, sedih, marah, takut, jijik, terkejut dan netral. Hal ini dapat digunakan dalam berbagai aplikasi, seperti pengenalan emosi dalam interaksi manusia dan komputer, pengawasan kesehatan mental, pengembangan robotika sosial, mengenali kepuasan individu terhadap layanan dan lain-lain. Gambar 2.11 berikut menunjukkan kerangka berpikir yang digunakan pada penelitian ini:



Gambar 2. 13 Kerangka Pemikiran

1. Permasalahan

Eksresi wajah manusia memiliki banyak variasi dan kompleksitas. Metode tradisional seringkali kesulitan dalam memodelkan dan mengenali pola-pola ekspresi yang kompleks dan subtan. *Convolutional neural network*, dengan kemampuannya dalam mengekstraksi fitur hierarkis dari data gambar, dapat mengatasi kekompleksan dan variasi ekspresi wajah dengan lebih baik. Pengenalan ekspresi wajah dapat membantu dalam pengenalan emosi manusia dan pengenalan emosi dalam interaksi manusia dan komputer dan mengenali kepuasan individu terhadap layanan. Dengan mengidentifikasi ekspresi wajah, sistem dapat mengenali emosi seperti kebahagiaan, sedih, marah, takut, jijik, terkejut dan netral.

2. Solusi

Teknologi *deep learning* dengan menggunakan arsitektur jaringan saraf tiruan berlapis-lapis seperti *convolutional neural networks* (CNN), dapat menjadi alternatif dalam pengenalan ekspresi wajah. CNN mampu mengidentifikasi emosi seseorang dengan lebih akurat dan konsisten. Penulis memberikan solusi untuk identifikasi emosi dengan menggunakan metode CNN dalam pengenalan ekspresi wajah. Metode yang penulis gunakan pada penelitian ini adalah *convolutional neural network*. Metode tersebut akan diuji terlebih dahulu untuk mengevaluasi tingkat akurasi dalam pengenalan ekspresi wajah.

3. Hasil

Hasil akurasi dari penggunaan metode *convolutional neural network* dalam pengenalan ekspresi wajah sangat akurat, maka program hasil penelitian ini dapat digunakan dalam berbagai aplikasi, seperti pengenalan emosi dalam interaksi manusia dan komputer, pengawasan kesehatan mental, pengembangan robotika sosial, mengenali kepuasan individu terhadap layanan dan lain-lain.