BAB 2

DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian Ivan Hartono, Agustinus Noertjahyana, Leo Willyanto Santoso pada tahun 2022 yang berjudul " Deteksi Masker Wajah dengan Metode *Convolutional Neural Network*" meneliti tentang pembuatan sistem untuk menerapkan protokol kesehatan yang sudah dibuat oleh pemerintah yang bertujuan dalam membantu kinerjadari petugas keamanan fasilitas umum untuk memeriksa apakah masyarakat sudah menggunakan masker dengan benar atau tidak dengan menggunakan arsitektur VGG16Net dan untuk mendeteksi wajah menggunakan SSD*Resnet10*. *Penelitian ini menggunakan dataset* dengan total data sebanyak 2.095 *data* gambar yang diperoleh dari *github* /chandrikadeb7 "*Face Mask Detection*" dengan perbandingan data *training*sebesar 75% dari jumlah keseluruhan *dataset* yang dimiliki dan data *testing* dengan ukuran sebesar 25% dari semua *dataset* yang ada. Hasil performa dari VGG16Net dan SSD*Resnet* 98% untuk akurasi dan *f1-score* sebesar 98%. Saran dalam penelitian ini agar dapat menambah jumlah *dataset* yang lebih bervariasi dan menguji konfigurasi pada VGG16Net untuk meningkatkan akurasi[6].

Parmonang R. Togatorop, Ahmad Fauzi pada tahun 2019 dengan penelitiannya yang berjudul "Klasifikasi Penggunaan Masker Wajah Menggunakan *Squeeznet*" membahas tentang banyaknya penelitian terkait pendeteksian penggunaan masker yang sudah banyak dilakukan. Penelitian ini menggunakan *deep transfer learning model* menggunakan algoritma YoloV3 untuk mendeteksi wajah dan Darknet-53 untuk *backbone.* Penulis menggunakan *single shot multibox detector* dan *MobileNetV2* serta melakukan pendeteksian secara *realtime* menggunakan *Transfer Learning* penggunaan *Naïve Bayes* (NB) dan *Support Vector Machine* (SVM) pada saat proses klasifikasi untuk melihat performa *Squeeznet*. Hasil yang didapat dengan menggunakan *Naïve Bayes* (NB) akurasi 0,958, presisi 0,981, *recall* 0,938, sedangkan dengan menggunakan *Support Vector Machine* (SVM) akurasi sebesar 0,992, presisi 0,994 dan *recall* 0,990 [7].

Menurut penelitian yang dilakukan oleh Tri Septiana Nadia Puspita Putri, Mohamad Al Fikih, Novendra Setyawan melakukan penelitian pada tahun 2020 tentang "*Face Mask Detection Covid19 Using Convolutional Neural Network*" yang membahas tentang penerapan sistem deteksi masker dengan menggunakan pengolahan citra. Perancangan sistem dalam penelitian ini menggunakan kombinasi klasifikasi dari pendeteksian objek, gambar dan pelacakan objek untuk mengembangkan sistem deteksimasker dalam bentuk gambar maupun *video*. *Dataset* diambil dengan berbagai macam variasi antara lain gambar yang memakai hijab, topi maupun yang tidak memakai atribut apapun, dan juga gambar yang berasal dari berbagai negara antara lain asia, eropa, dan amerika. Penelitian ini menghasilkan nilai akurasi sebesar 0,9933% dan *training loss* 0,0213%[8].

Penelitian yang dilakukan oleh Muhammad Aminullah pada tahun 2021 dengan judul "Alat Deteksi Masker Dengan Metode *Convolutional Neural Network* Untuk Tunanetra Pada *Era New Normal*" melakukan penelitian pembuatan alat untuk mendukung penyandang tunanetra dalam menghadapi *covid-19*. Dalam perancangan ini penelitian dilakukan dengan menggunakan metode CNN dan penggunaan *tensorflow* yang berfungsi untuk *framework deep learning* bertujuan untuk mengenali dan mengklasifikasikan suatu objek. Sistem deteksi masker dan jarak di implementasikan pada Rasberry Pi dan mendapat FPS sebesar 0,33. Hasil pengujian dapat diterima dengan bantuan *earphone* dan pengasuh dapat melakukan pemantauan melalui sistem pengawasan yang ada pada alat ini secara *real-time* antara lain dapat diipantau suhu badan, keadaan dan lokasi keberadaan penyandang tunanetra dengan mengguunakan aplikasi *smartphone* pada penelitian ini juga dilengkapi dengan API *Google Maps* untuk menampilkan lokasi penyandang tunanetra[16].

Nyoman Purnama, Putu Kusuma Negara melakukan penelitian tentang "Deteksi Masker Pencegahan *Covid19* Menggunakan *Convolutional Neural Network* Berbasis Android" pada tahun 2021, membuat sebuah perbandingan dua metode optimasi pada *deep learning* yaitu *adam* dan *gradient descent* dan tujuan dari dilakukan proses pengujian untuk mengetahui nilai *recall*, *presisi* dan akurasi setiap *optimizer*. Pengujian diproses menggunakan perangkat dengan berbasis *android* dan penggunaan bahasa pemograman *java* serta *framework mobilenetv2.* Penelitian yang

dilakukan ini menghasilkan akurasi sebesar 90% pada saat menggunakan *adam* dan saat menggunakan optimasi *gradient descent* menghasilkan angka sebesar 80% [17].

Penelitian dengan judul "*Convolutional Neural Network* Arsitektur *MobileNet-V2* Untuk Mendeteksi Tumor Otak" yang dilakukan oleh Widi Hastomo, Sugiyanto dan Sudjiran pada tahun 2021 melakukan sebuah penelitian untuk memprediksi penyakit tumor otak yang diderita oleh pasien yang diprediksi dari kemampuan pembacaan *image* dari peralatan CT *Scanner* dengan menggunakan metode CNN. Penggunaan arsitektur *MobilenetV2* sebagai arsitektur yang digunakan pada penelitian ini memiliki fungsi untuk digunakan sebagai *training* data dan *testing* data dengan total 2.870 *image*tumor otak. Penelitian yang dilakukan menghasilkan sebuah nilai *training* akurasi sebesar 97% dan nilai 94% untuk nilai *testing* serta nilai disetiap akurasi pada setiap klasifikasi yang dilakukan antara lain *glioma* sebesar 99%, *meningioma* sebesar 85%, *no tumor* sebesar 99% dan *pituaty* sebesar 96%. Hasil akurasi yang dilakukan dalam penelitian kali ini mendapatkan hasil yang sangat baik dan menghasilkan sebuah modelyang memiliki manfaat untuk mendiagnosa pasien dengan cepat, murah dan akurat[18].

Penelitian tentang pendeteksian masker wajah *covid-19* menggunakan metode CNN yang akan dilakukan penulis tidak terlepas dari hasil penelitian yang telah disebutkan yang selanjutnya dipaparkan pada tabel 2.1.

Tabel 2.1 Tinjauan Pustaka Penelitian Terdahulu

No	Jurnal	Keterangan
1	Mohammad Ivan Hartono, Agustinus Noertjahyana, Leo Willyanto Santoso yang berjudul "*Deteksi Masker Wajah dengan Metode Convolutional Neural Network*" tahun 2022.	Pembuatan sistem deteksi masker wajah menggunakan metode *Convolutinal neural network* menggunakan arsitektur VGG16Net dan pendeteksi wajah menggunakan ssd *resnet10*.

No	Jurnal	Keterangan
2	Parmonangan R. Togatorop, Ahmad Fauzi dengan penelitiannya yang ber judul ”Klasifikasi Penggu naan Masker Wajah Menggunakan *Squeezenet*” pada tahun 2019.	Klasifikasi Penggunaaan masker wajah menggunakan *Squeezenet*. Pe*model*an klasifikasi dilakukan dengan menggu- nakan *SqueezeNet* untuk proses ekstraksi fiturdan *Naïve Bayes, Support Vector Machine* untuk proses klasifikasi.
3	Tri Septiana Nadia Puspita Putri, Mohamad Al Fikih, Novendra Setyawan tentang “*Face Mask Detection Covid19 Using Convolutional Neural Network*” tahun 2020	Membahas tentangpenerapan sistem deteksi masker dengan menggunakan pengolahan citra. Perancangan sistemdalam penelitian ini menggunakan kombinasi klasifikasi dari pendeteksian objek, gambar dan pelacakan objek untuk mengembangkan sistem deteksi masker dalam bentuk gambar maupun video. *Dataset* diambil dengan berbagai macamvariasi antara lain gambaryang memakai hijab, topi maupun yang tidakmemakai atribut apapun, dan juga gambar yangberasal dari berbagai negaraantara lain asia, eropa, dan amerika. Penelitian ini menghasilkan nilai akurasi sebesar 0,9933% dan training loss 0,0213%.
4	Penelitian yang dilakukan oleh Muhammad Aminullah dengan judul“Alat Deteksi Masker	Pembuatan alat untuk mendukung penyandang tunanetra dalam menghadapi *covid-19*. Penelitian ini menggunakan CNN dengan meng-

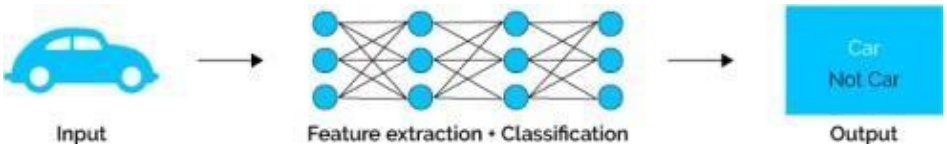
No	Jurnal	Keterangan
	Dengan Metode *Convolutional Neural Network* Untuk Tunanetra Pada Era New Normal" tahun 2021	gunakan *tensorflow* sebagai *framework deep learning* yang dapat mengenali dan mengklasifikasikan suatu objek. Sistem deteksi masker dan jarak di implementasikan pada Rasberry Pi dan mendapat FPS sebesar 0,33. Hasil pengujian dapat diterima melalui *earphone* dan sistem pengawasan pada alat ini antara lain pengasuh dapat melakukan monitoring secara real- time suhu badan, keadaan dan lokasi keberadaan penyandang tunanetra melalui aplikasi smartphone dan pada aplikasi dilengkapi API *Google Maps* untuk menampilkan lokasi penyandang tunanetra.
5	Nyoman Purnama, Putu Kusuma Negara melakukan penelitian tentang "Deteksi Masker Pencegahan *Covid19* Menggunakan *Convolutional Neural Network* Berbasis *Android*" tahun 2021	Membuat sebuah perbandingan dua metode optimasi pada deep learning yaitu *adam* dan *gradient descent* dan tujuandari dilakukan proses penguji untukmengetahui nilai *recall,* presisi dan akurasi setiap *optimizer*. Pengujian diproses menggunakan perangkat dengan berbasis *android* dan penggunaanbahasa pemograman *java* serta *framework mobilenetv2.* Penelitian yang dilakukan ini meng hasilkan akurasi sebesar 90% pada

No	Jurnal	Keterangan
		saat menggunakan *adam* dan saat menggunakan optimasi *gradient descent* menghasilkan angka sebesar 80%
6	Penelitian dengan judul "*Convolutional Neural Network* Arsitektur *MobileNet-V2* Untuk Mendeteksi Tumor Otak" yang dilakukan oleh Widi Hastomo, Sugiyanto dan Sudjiran tahun 2021	Melakukan penelitian untuk memprediksi penyakit tumor otak yang diderita oleh pasien yang diprediksi dari kemampuan pembacaan *image* dari peralatan CT Scanner dengan menggunakan metode CNN. Penggunaan arsitektur MobilenetV2 memiliki fungsi untuk digunakan sebagai *training* data dan *testing* data dengan total 2.870 *image* tumor otak. Penelitian yang dilakukan menghasilkan sebuah nilai *training* akurasi sebesar 97% dan nilai 94% untuk nilai *testing* serta nilai disetiap akurasi pada setiap klasifikasi yang dilakukan antara lain *glioma* sebesar 99%, *meningioma* sebesar 85%, *no tumor* sebesar 99% dan *pituaty* sebesar 96%. Hasil akurasi yang dilakukan mendapatkan hasil yang sangat baik dan menghasilkan sebuah model yang memiliki manfaat untuk mendiagnosa pasien dengan cepat, murah dan akurat.

2.2 DASAR TEORI

2.2.1 Deep Learning

Salah satu cabang dari *Machine Learning* yang tersusun dari algoritma pemodelan abstraksi tingkat tinggi pada *data* dengan menggunakan sekumpulan fungsi transformasi *non-linear* dengan berlapis-lapis dan mendalam yaitu pengertian dari *Deep Learning*. Pada pengaplikasiannya *deep learning* banyak digunakan untuk *speech recognition* pada ponsel pintar, analisa *video* dan citra, 12(dua belas) klasifikasi teks dan sebagainya. Teknik dan algoritma dalam *deeplearning* dapat digunakan untuk kebutuhan *supervised learning* (pembelajaran terarah), *unsupervised learning* (pembelajaran tak terarah), dan s*emi-supervised learning* (semi-terarah). Struktur dan jumlah jaringan saraf pada algoritmanya sangatbanyak bisa mencapai ratusan lapisan. Terdapat dua istilah penting dalam pembangunan *model* yaitu pelatihan dan pengujian. Pelatihan atau *training* adalah proses konstruksi *model* sedangkan pengujian atau *testing* merupakan proses menguji kinerja dari *model* hasil pembelajaran. Pelatihan biasanya menggunakan sebuah kumpulan *data* atau biasa disebut *dataset* yang berupa sampel *data* dalam statistika, dapat berupa citra [9].



Gambar 2.1 Arsitektur *Deep Learning*[15]

Seperti yang terlihat pada Gambar 2.1 Arsitektur Deep Learning[15] pada deeplearning tahap *feature extraction* dan *classification* berada dalam satu tahapan [15] .

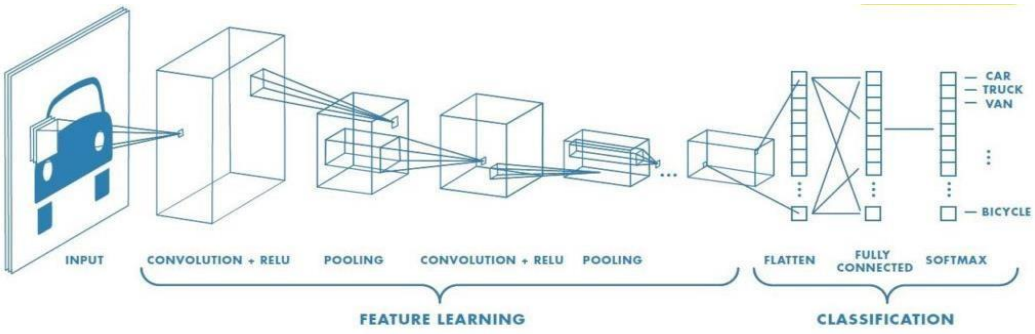
2.2.2 Coronavirus Disease 2019 (Covid-19)

Coronavirus Disease 2019 (Covid-19) mempunyai pengertian sebuah penyakit dapat menular dengan penyebabnya yaitu *Severe Acute Respiratory Syndrome Coronavirus* 2 atau dengan singkatan SARS-CoV2 [1]. *Virus* jenis baru yang biasa dikenal dengan nama *covid-19* yaitu *virus* corona yang tidak pernah ter

identifikasi sebelumnya dalam tubuh manusia. Ciri-ciri kemunculan ataupertanda dan gejala saat terpapar infeksi *virus* ini antara lain sakit batuk, demam,dan parahnya hingga sesak napas. *Virus* ini juga dapat menimbulkan penyakit berat diantaranya sindrom pernapasan akut , gagal ginjal, pneumonia, dan bahkan dapat menyebabkan kematian. Informasi tentang *virus covid-19* pertamakalinya diketahui berasal dari Kota Wuhan, Provinsi Hubei, China pada 31 Desember 2019. Pada tanggal 30 Januari 2020, *World Health Organization* (WHO) menetapkan bahwa kejadian tersebut sebagai Kedaruratan Kesehatan Masyarakat yang Meresahkan Dunia (KKMMD) dan pada tanggal 11 Maret 2020 *Covid-19* ditetapkan sebagai pandemi, sebab *virus* ini dapat menular sangatcepat dan sangat berbahaya [1].

2.2.3 Convolutional Neural Network (CNN)

Pengembangan dari *MultilayerPerceptron* (MLP) yang di *desain* untuk mengolah *data* dua dimensi merupakan pengertian dari *Convolutional Neural Network (CNN)*. CNN dapat mengetahui informasi dari suatu objek seperti citra,teks, potongan suara dan sebagainya berupa *data*. Namun paling banyak digunakan pada bidang pemrosesan citra.



Gambar 2.2 Alur Kerja CNN [6]

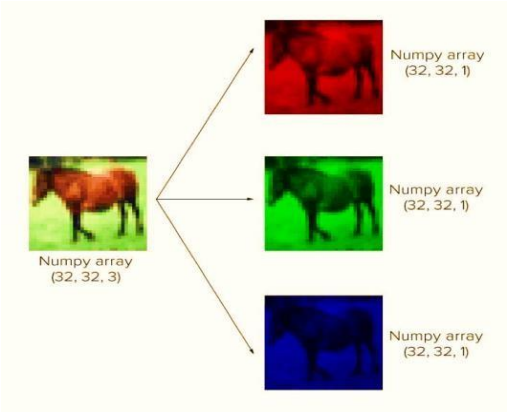
Gambar 2.2 memperlihatkan sebuah alur dari proses CNN ketika melakukanpemrosesan citra dengan *feature learning* menggunakan *convolutionrelu* kemudian *pooling* hingga proses klasifikasi citra dengan menggunakan*flatten, fully connected* dan *softmax* sehingga citra dapat diklasifikasikan ke kategori tertentu berdasarkan nilai keluarannya.

Alur Kerja CNN seperti pada Gambar 2.2 Alur Kerja CNN dapat

dikategorikan memiliki lima komponen utamapada *layer* atau lapisannya yaitu sebagai berikut :

1. *Input Layer*

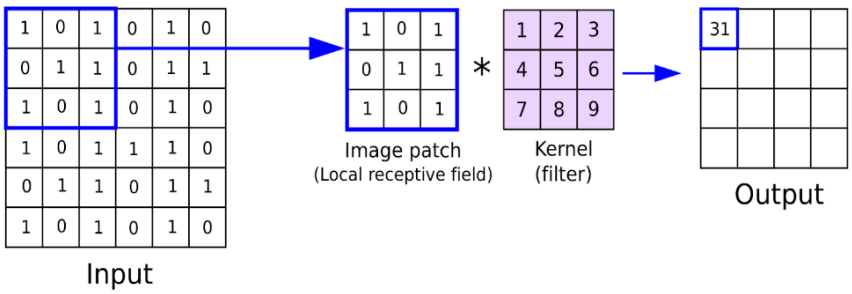
Lapisan masukkan dapat berupa sebuah citra RGB (*Red, Green, Blue*) dengan ukuran32x32 piksel yang sebenarnya merupakan sebuah *multidimensional array* dengan ukuran 32x32x3. Nilai 3 terakhir merupakan jumlah dari kanal. Contoh dari citra tersebut ditunjukkan pada Gambar 2.3.



Gambar 2.3 *Citra Input Layer* [6]

2. *Convolutional Layer*

Lapisan ini merupakan lapisan yang pertama kali menerima masukkan citra langsung pada arsitektur.Pada lapisan ini juga melakukan kombinasi *linier filter* terhadap daerah lokal seperti operasi konvolusi. Seperti layaknya *citra, filter* lapisan pada proses konvolusi memiliki ukuran tinggi, lebar dan tebal tertentu. Alur padalapisan konvolusi dapat digambarkan pada gambar 2.4.



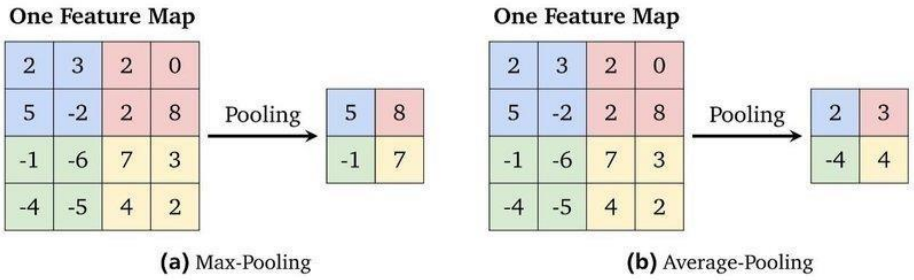
Gambar 2.4 Alur *Convolutional Layer* [6]

3. Activation Layer

Lapisan aktivasi mempunyai pengertian lapisan dimana *feature map* dimasukan ke dalam fungsi aktivasi. mengubah nilai yang ada pada *feature map* pada *range* tertentu sesuai dengan fungsi aktivasi yang digunakan adalah fungsi dari lapisan aktivasi. Tujuan dilakukan hal tersebut untuk meneruskan nilai yang menampilkan fitur dominan dari citra yang masuk ke lapisan berikutnya. Terdapatbeberapa fungsi aktivasi yang umum untuk digunakan, namun dalam penelitian hanya menggunakan aktivasi *ReLu* dan *Softmax.*

4. Pooling Layer

Selanjutnya masukan dari lapisan aktivasi akan menuju *pooling layer*, kemudian lapisan ini akan mengurangi parameternya. *Pooling* juga bisa disebut sebagai *subsampling* atau *downsampling* yang berfungsi untuk mengurangi dimensi dari *feature map* tanpa menghilangkan informasi penting di dalamnya. Proses padalapisan ini cukup sederhana, dimana dengan menentukan ukuran *down sampling* yang akan digunakan pada *feature map*, sebagai contoh 2x2 yang selanjutnya akandilakukan proses *pooling* pada *feature map*. Ada beberapa macam proses dari *pooling* antara lain *max pooling*, *mean pooling*, dan *sum pooling*. Contoh proses *pooling* ditunjukan pada gambar 2.5.



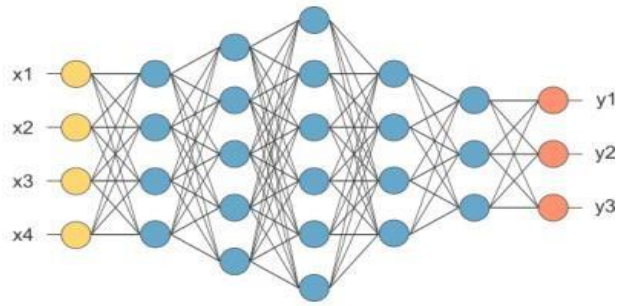
Gambar 2.5 Proses *Pooling* [6]

Setelah dilakukan *pooling layer*, maka dapat diketahui tujuan dari *pooling layer* tersebut yaitu untuk mengurangi dimensi dari *feature map*. Sehingga proses ini gambar 2.5 matriks *feature map* 4x4 dengan proses *pooling* 2x2 akan mempercepat komputasi karena parameter yang harus diperbaharui semakin sedikitdan mengatasi

overfitting.

5. Fully Connected Layer

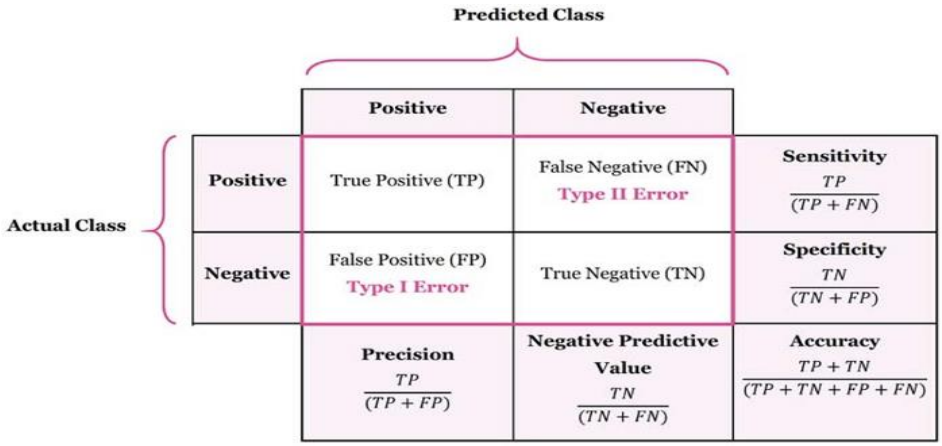
Dilapisan ini bekerja setelah melakukan beberapa lapisan diatas dan menghasilkan *pooling layer* yang digunakan untuk *input*-an untuk *fully connected layer*. Lapisan ini mempunyai kesamaan struktur dengan *Artificial Neural Network* (ANN) yang memiliki lapisan *input*-an, lapisan tersembunyi, dan lapisan *output* yang masing-masing memiliki *neuron* yang saling berhubungan dengan *neuron* yang berada dalam lapisantetangganya. Dalam konsep sebelum *pooling* digunakanuntuk input, hasil *pooling* terlebih dahulu diubah menjadi vektor (x1, x2, x3, x4, dan seterusnya) yang akan diproses ke dalam *fully connected layer* yang nantinya pada lapisan terakhir *fully connected layer* akan digunakan fungsi ReLu atau *softmax*untuk menentukan klasifikasi dari citra masukan yang dari lapisan masukan CNN.Untuk lebih jelas contoh *fully connected layer* pada Gambar 2.6.



Gambar 2.6 *Fully Connected Layer* [6]

2.2.4 Confusion Matrix

Confusion Matrix adalah salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. *Confusion Matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang sebenarnya. Pengukuran kinerja dengan *confusion matrix*, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi yang ditampilkan dalam gambar 2.7.



Gambar 2.7 *Confusion Matrix* [20]

Penjelasan gambar 2.7 tersebut adalah *True Positive* (TP) adalah kasus di mana model klasifikasi yang dibuat dengan benar diprediksi *positif*. *False Negative* (FN) adalah kasus di mana model klasifikasi diprediksi salah, tetapi sebenarnya *positif*. Ini juga dianggap sebagai kesalahan Tipe II. *False Positif* (FP) adalah *positif* palsu kasus dimana model klasifikasi diprediksi positif, tetapi sebenarnya negatif. Ini juga dianggap sebagai kesalahan Tipe I. *True Negative* (TN) adalah kasus di mana model klasifikasi memprediksi negatif dengan benar. Setelah hasil dari masalah klasifikasi telah diterima, *matrix* klasifikasi dapat menghitung nilai *Sensitivitas, Spesifisitas*, Akurasi, Nilai Prediktif *Negatif*, dan *Presisi*. Dimana sensitivitas yang biasa disebut *True Positif Rate* atau *Recall* adalah rasio prediksi benar *positif* dibandingkan dengan keseluruhan data yang benar *positif*. *Spesifisitas* juga dikenal sebagai *True Negative Rate* merupakan kebenaran memprediksi *negatif* dibandingkan dengan keseluruhan data *negatif*. Akurasi merupakan rasio prediksi benar (*positif* dan *negatif*) dengan keseluruhan data. Nilai *Prediktif Negatif* merupakan hasil yang diberi label dengan benar sebagai salah. *Presisi* merupakan rasio prediksi benar *positif* dibandingkan dengan keseluruhan hasil yang diprediksi *positf*. *F1-Score* merupakan perbandingan rata-rata *presisi* dan *recall* yang dibobotkan *F1 Score* = 2 * *(Recall*Precission*) / (*Recall + Precission*) [20].

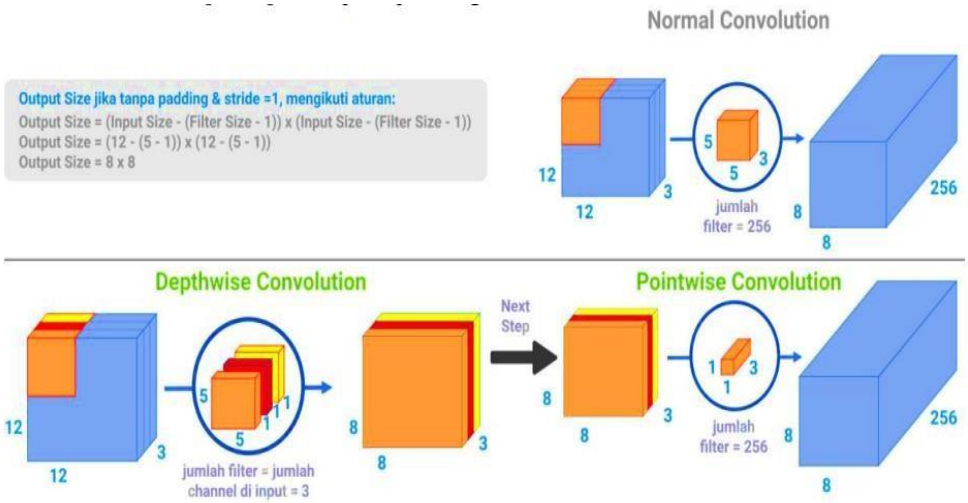
2.2.5 Arsitektur CNN *MobilenetV2*

Berbagai macam arsitektur CNN mulai banyak bermunculan, dan masing-masing arsitektur berlomba-lomba agar memperoleh skor akurasi tertinggi yang di

training terhadap *ImageNet*. Pengertian dari *ImageNet* sendiri merupakan sekumpulan *dataset* gambar yang berjumlah sangat besar dan dirancang oleh akademisi yang ditujukan untuk penelitian *computer vision*.

Dalam penelitian ini, pemilihan arsitektur CNN *MobileNetV2* dikarenakan skor akurasinya cukup tinggi dan fungsi utamanya ialah perbandingan jumlah *training parameters* yang kecil dibandingkan dengan arsitektur CNN yang lain sehingga kebutuhan akan komputasinya jauh lebih ringan dan juga *model size MobileNetV2* cukup kecil dengan ukuran 14MB dan performansi yang cukup baik sehingga kedepannya pada saat *model* akan di *deploy* kedalam sebuah *real app*, seperti dibuat menjadi sebuah aplikasi *android* maupun aplikasi berbasis *website* ringan dan berukuran kecil.

Sebuah perancangan yang dibuat oleh *Google* dan memiliki *layer* khusus yang disebut dengan *depthwise separable convolution* ialah *MobileNet*. Layer ini memiliki fungsi untuk mereduksi komputasi sehingga menghasilkan ukuran *model* yang lebih kecil [10].



Gambar 2.8 *Normal Convolution vs Depthwise Separable Convolution* [10]

Penjelasan gambar 2.8 *filter* melakukan sebuah konvolusi terhadap *input image* dan juga membentuk sebuah *feature map* atau *output image* pada *layer normal convolutional*, dan dalam *depthwise convolution*, jumlah *filter* sama dengan jumlah *channel* yang ada pada *input image*. Setiap *filter* nantinya melakukan

konvolusi terhadap masing-masing *channel* pada *input image* (atau dengan kata lain, secara tidak langsung melakukan konvolusi terhadap seluruh *channel* pada *input image* seperti yang dilakukan pada *normal convolution*). Kemudian agar *output image* dengan *depthwise convolution* sama dengan *output image* yang dihasilkan dengan *normal convolution*,akan dilakukan *pointwise convolution* dan hasil dari tahap *depthwise convolution* akan dikonvolusi lagi dengan filter 1x1, filter ini memiliki kedalaman yang sama dengan jumlah *channel* di *output image* sebelumnya.

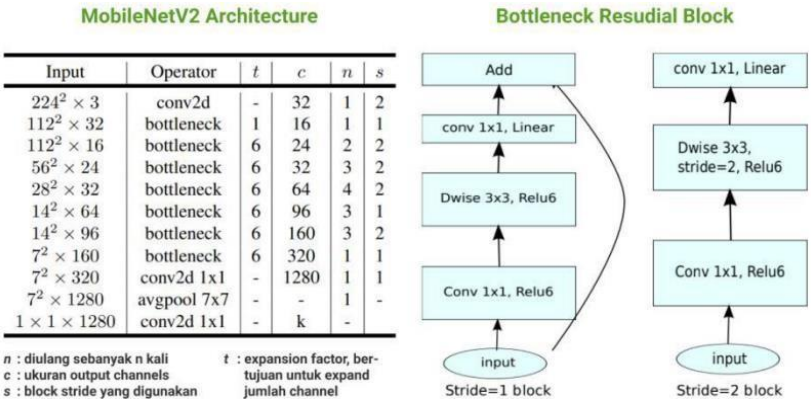
Perbandingan jumlah komputasi antara *normal convolution* dibandingkan dengan *depthwise separable convolution* berdasarkan contoh gambar 2.7 *Normal Convolution vs Depthwise Separable Convolution* [10].

1. *Normal Convolution*

Ada 256 *filter* berukuran 5x5x3 yang bergerak sebanyak delapan kali delapan jumlah pergeseran *filter* 5x5 dari kiri atas ke kanan bawah terhadap *input image* 12x12). Yang mempunyai arti ketika proses konvolusi 256x5x5x3x8x8=1.228.800 total perkalian yang dilakukan.

2. *Depthwise Separable Convolution*

Dalam tahap *depthwise convolution* terdapat 3(tiga) *filter* 5x5x1 yang bergeser sebanyak delapan kali delapan dengan arti 3x5x5x1x8x8=4.800 yang kemudian, dalam *pointwise convolution* terdapat 256 filter 1x1x3 yang bergeser sebanyak delapan kali delapan juga memiliki arti 256x1x1x3x8x8=49.152 yang jika ditotal perkalian yang dilakukan saat proses konvolusi pada *depthwise separable convolution* ini yaitu 4.800+49.152=52.952. Dapat disimpulkan bahwa jumlah 52.952 hanya kurang lebih sekitar 22,7% dari 1.228.800 pada *normal convolution*.

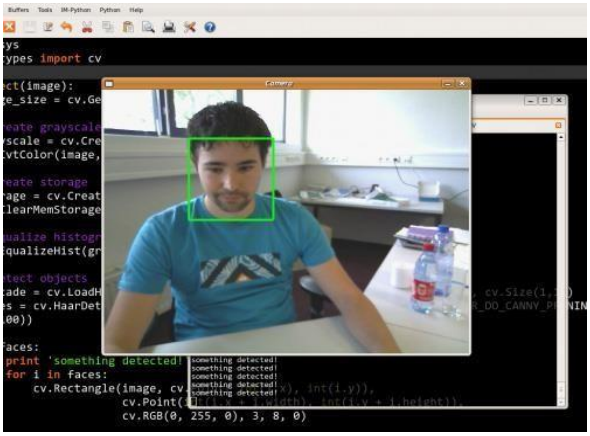


Gambar 2.9 Arsitektur *MobileNetV2* [10]

MobileNetV2 merupakan perkembangan lebih lanjut dari *MobileNet* dimana ditambahkan fitur baru yaitu sebuah blok dalam arsitektur yang didalamnya menggunakan *depthwise separable convolution* [10].

2.2.6 Modul DNN *OpenCV*

Model caffe yang didasarkan pada *Single Shot-Multibox Detector* (SSD) dan menggunakan arsitektur *ResNet*-10 sebagai tulang punggungnya merupakan penjelasan Modul *Deep Neural Network* (DNN) *OpenCV*. *OpenCV* membuat pendeteksian pengenalan wajah dapat dilakukan dengan menggunakan *model* pendeteksi *wajah deep learning* yang sudah terlatih. Penggunaan dari modul DNN *OpenCV* dengan *model caffe* membutuhkan *prototxt* yang mengartikan arsitektur *model ResNet10* dan *caffe model* merupakan sebuah *file* yang berisi bobot untuk lapisan sebenarnya. Pendeteksian wajah dengan menggunakan *OpenCV* pada gambar 2.10 Deteksi dengan *OpenCV* [11].



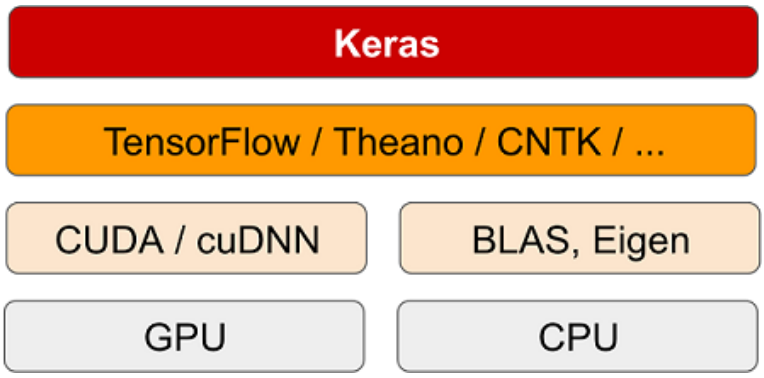
Gambar 2.10 Penerapan deteksi dengan *OpenCV* [14]

2.2.7 Google Colaboratory

Google Colab atau biasa disebut *Google Colaboratory* adalah sebuah produk *Google* berbasis *cloud* yang bisa diakses atau digunakan secara gratis. Perbedaan dengan produk *Google* lainnya yaitu pada *coding environment Google Colab* bisa diaplikasikan dengan bahasa pemrograman *Python* dengan format "*notebook*" yang memiliki kesamaan dengan *Jupyter Notebook,* yang seakan-akan *Google* memberikan fasilitas untuk para *programmer* atau *researcher* untuk menggunakan sebuah komputer secara gratis tetapi dengan spesifikasi yang tinggi [13].

2.2.8 Framework Tensorfow dan *Keras*

Cukup banyak *framework* yang dapat digunakan untuk pengguna *deep learning*, antara lain ada *framework TensorFlow* dan *Keras*. Kedua *framework* tersebut merupakan yang paling banyak digunakan dalam dunia *deep learning*. *TensorFlow* merupakan sebuah platform sumber terbuka untuk *deep learning*. Ini merupakan sebuah *library* dengan API level tinggi, dengan begitu TensorFlow dapat membantu dalam pembuatan *neural network* dalam skala yang besar. *framework deep learning* untuk *python* yang menyediakan cara mudah untukmendefinisikan dan melatih hampir semua jenis *model deep learning* pengertian dari Keras. Awal dikembangkan *Keras* mempunyai tujuan untuk mempercepat kemungkinan bereksperimen.



Gambar 2.11 Perangkat Lunak dan Perangkat Keras *Deep Learning*[15]

Penjelasan gambar 2.11 menjelaskan bahwa *TensorFlow, Keras* dapat berjalandengan mulus di CPU dan GPU, ketika berjalan pada CPU, *TensorFlow*

sendiri akanmelibatkan *library* tingkat rendah untuk operasi tensor yang disebut dengan *Eigen* edangkan pada GPU, *TensorFlow* akan melibatkan *library* operasi *deep learning* yang dioptimalkan dengan baik menggunakan cuDNN atau biasa disebut dengan *library* NVIDIA CUDA *Deep Neural Network* [12].

2.2.9 Spyder

Spyder adalah *development environment* terintegrasi ilmiah yang ditulis dengan *python*. *Software* ini dirancang untuk dan oleh para *data scientist* yang terintegrasi dengan beberapa *library python* seperti Mat*plot*lib, SciPy, NumPy, Pandas, Cython, I*Python*, SymPy, dan *software open source* lainnya. *Spyder* tersedia melalui distribusi *Anaconda* di Windows, MacOS, dan Linux. *Software* ini dapat diakses secara gratis dengan berbagai fitur, seperti dapat menjalankan kode *python* berdasarkan sel, baris, atau *file*, menyediakan *histogram* dan *plot time series*, penyelesaian kode otomatis dengan pemisah *horizontal* atau *vertikal*, dan lain sebagainya [14].