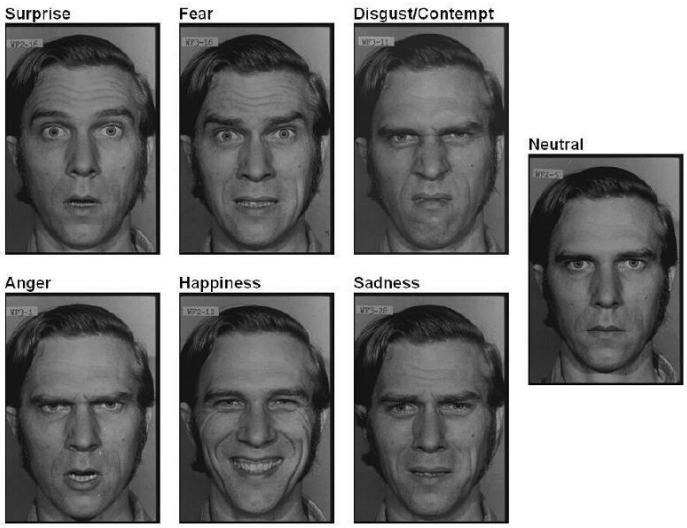
BAB 2
LANDASAN TEORI

2.1 Wajah

Wajah merupakan bagian depan dari kepala pada manusia yang meliputi wilayah dari dahi hingga dagu, termasuk rambut, dahi, alis, pelipis, mata, hidung, pipi, mulut, bibir, gigi, kulit, termasuk dagu. Wajah terutama digunakan untuk ekspresi wajah, penampilan, serta identitas seorang manusia. (Arti kata wajah - Kamus Besar Bahasa Indonesia (KBBI) Online, n.d.)

2.2 Ekspresi Wajah

Ekspresi wajah merupakan salah satu bentuk bahasa non-verbal yang dilakukan manusia saat bersosialisai. Ekspresi wajah dapat menyampaikan emosi yang sedang dirasakan [25]. Ekspresi wajah dibagi ke dalam 7 ekspresi dasar yaitu Bahagia, Sedih, Terkejut, Takut, Marah Muak dan Netral.



Gambar 2.1 Ekspresi Dasar Manusia

2.3 Citra Digital

Citra digital merupakan sekumpulan larik (aray) yang berisi nilai-nilai *real* dan kompleks atau biasa disebut *pixel* yang dituangkan pada deretan *bit-bit* tertentu dan nilainya merepresentasikan gray *level* (keabuan) atau spasial di titik tersebut [4].

Adapun jenis-jenis dari citra digital adalah:

1. Citra biner (*monochrome*): citra monochrome merupakan citra yang mempunyai dua kemungkinan nilai piksel yaitu hitam dan putih atau sering disebut dengan citra hitam putih (*Black and White*).
2. Citra skala keabuan (*grayscale*): sebuah citra yang terdiri dari tiga kanal utama yaitu *Red, Green, dan Blue* atau biasa disingkat dengan RGB pada citra grayscale nilai bagian *red = green = blue*. Dengan kata lain citra ini menjadikan nilai dari RGB menjadi satu kanal atau mempunyai nilai yang sama. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas.
3. Citra warna (24 bit) : citra warna memiliki kerterbalikan dari citra *Grayscale* dimana Citra ini memiliki masing-masing *layer* yaitu *red layer, green layer, blue layer*. Sistem warna *Red Green Blue* (RGB) menggunakan sistem tampilan grafik kualitas tinggi (*high quality raster graphic*) yaitu mode 24 bit.

Citra digital tersusun atas elemen-elemen bilangan berhingga yang memiliki lokasi tertentu. Citra digital direpresentasikan dalam bentuk matrik berukuran M x N seperti dinyatakan dalam Persamaan 1.

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix} \quad (1)$$

Kedua ruas dari persamaan ini merupakan cara ekivalen dalam menyatakan citra digital. Ruas kanan dari Persamaan 1 merupakan matrik bilangan ril. Setiap elemen dalam matrik ini disebut sebagai image element, picture element, pixel, atau pel [13]

2.4 Citra *Grayscale*

Citra *grayscale* merupakan citra digital yang terdiri dari satu nilai kanal di setiap pikselnya, dengan kata lain setiap bagian dari *Red*, *Green*, dan *Blue* (RGB) memiliki nilai yang sama dan nilai tersebut menunjukan tingkat intensitas sebuah citra. Warna yang dimiliki oleh citra grayscale adalah warna hitam, keabuan dan putih. Tingkat keabuan merupakan warna abu yang memilki tingkatan mendekati warna hitam atau putih.

Citra *grayscale* sering disebut juga dengan citra beraras keabuan. Citra ini memiliki nilai tunggal dengan nilai intensitas berkisar antara 0 sampai 255. Nilai intensitas yang mendekati 255 memiliki derajat keabuan yang semakin terang begitu juga sebaliknya nilai yang mendekati 0 akan semakin gelap. Proses untuk membuat citra grayscale dilakukan dengan cara meratakan nilai piksel dari tiga nilai *Red, Green, Blue* (RGB) menjadi satu nilai. Tapi karena nilai RGB dianggap tidak seragam dalam hal kemampuan kontribusi pada kecerahan, maka ada cara konversi yang lebih tepat dengan mnggunakan persamaan 1.

$$Y = 0{,}299R + 0{,}587G + 0{,}114B \quad (1)$$

Di mana Y adalah nilai kecerahan suatu piksel pada citra *grayscale*, dengan persentase 29,9% warna merah, 58,7% dari warna hijau, dan 11,4 dari warna biru [13].

Citra yang diubah ke bentuk citra *grayscale* digunakan untuk menyederhanakan model citra sehingga dapat mempermudah dalam melakukan pengolahan citra.

2.5 Facial Expression Recognition

Facial expression recognition atau pengenalan ekspresi wajah merupakan suatu teknologi dari komputer yang memungkinkan untuk melakukan identifikasi dan verifikasi ekspresi wajah seseorang melalui sebuah citra digital. Pengenalan ekspresi wajah dilakukan dalam beberapa tahap yaitu:

1. Penentuan jumlah citra ekpsresi wajah yang akan digunakan
2. Pengambilan citra ekspresi wajah

3. Penghilangan noise yang terdapat pada citra
4. Pengolahan fitur pada citra
5. Pengklasifikasian ekspresi wajah berdasarkan kelas

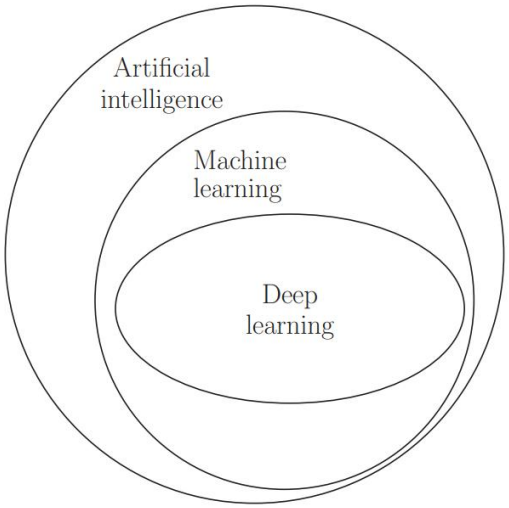
Setelah tahap tersebut telah dilakukan, hasilnya adalah citra ekspresi wajah yang telah diklasifikasi berdasarkan 7 kategori ekspresi wajah yaitu : Bahagia, Sedih, Terkejut, Takut, Marah Muak dan Netral [14].

2.6 Deep Learning

Deep learning merupakan salah satu subbidang dari kecerdasan buatan yang berfokus pada pembuatan model jaringan saraf besar yang mampu mengambil keputusan berbasis data dengan akurat. *Deep learning* sangat cocok digunakan pada data yang kompleks dan pada dataset yang besar atau banyak

Saat ini sebagian besar perusahaan online menggunakan deep learning, antara lain Facebook dalam menganalisis teks dalam percakapan online. Google, Baidu, Microsoft juga menggunakan deep learning dalam pencarian gambar, dan machine transalation. Semua smart phone telah memiliki sistem *deep learning* yang berjalan pada sistemnya. Sebagai contoh, *deep learning* sekarang telah menjadi standar dalam *speech recognition* dan juga *face detection* pada kamera digital.

Deep learning telah muncul dari penelitian dalam *Artificial Intelligence dan Machine Learning*. Gambar 2.2 mengilustrasikan hubungan antara *Artificial Intelligence, Machine Learning dan Deep Learning.*



Gambar 2.2 Hubungan antara Artificial Intelligence, Machine Learning dan Deep Learning

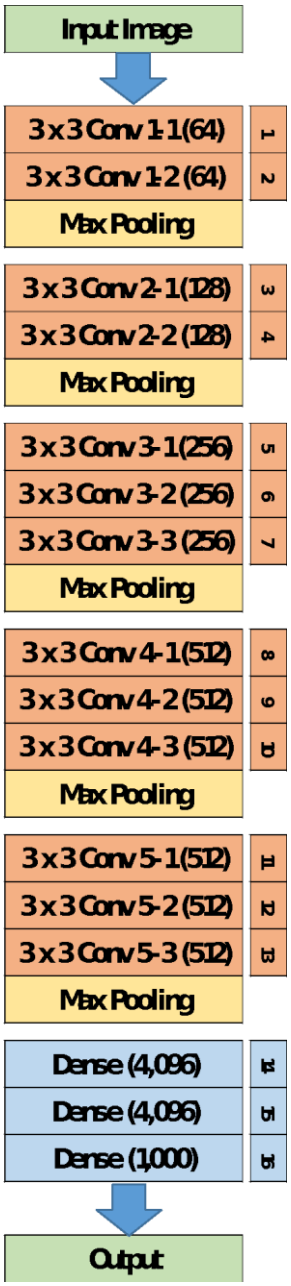
Deep Learning memungkinkan sistem untuk mengambil keputusan berdasarkan data dengan mengidentifikasi dan mengekstrasi pola dari dataset besar untuk menghasilkan keputusan yang akurat [15].

2.7 CNN

CNN merupakan kepanjangan dari *Convolutional Neural Network*, yang menggunakan operasi matematika yang disebut convolution pada struktur jaringan saraf, sehingga dinamakan konvolusi. *Convolutional network* dapat membuat CNN digunakan sebagai struktur dua dimensi dari input data. Saat ini, penerapan jaringan ini digunakan sebagai *Image Recognition* dan *Speech Recognition* dan bisa mendapatkan hasil yang baik. Jika dibandingkan dengan algortima lainnya, Convolutinal *Neural Network* memiliki tahap pemrosesan gambar yang lebih sedikit dan memiliki kelebihan tersendiri [6].

2.8 VGG16

VGG16 merupakan model CNN yang memanfaatkan *convolutional laye*r dengan spesifikasi *convolutional filter* yang kecil (3×3). Dengan ukuran *convolutional filter* tersebut, kedalaman *neural network* dapat ditambah dengan lebih banyak lagi *convolutional layer*. Model VGG16 mempunyai 19 *layer* yang terdiri dari 16 *convolutional layer* dan 3 *fully-connected layer* [24].



Gambar 2.3 Arsitektur VGG16

2.9 MTCNN

MTCNN kepanjangan dari *Multi-Task Cascaded Convolutional Neural Networks* yang merupakan arsitektur jaringan yang diperoleh dengan melakukan *Multiple Task Cascading* terhadap CNN. Ini merupakan algoritma yang menggabungkan area deteksi wajah dengan lokasi titik kunci. Jaringan arsitektur dibagi menjadi tiga lapisan yaitu P-NET, R-NET, dan O-NET [6].

Algoritma MTCNN dapat menyelesaikan tiga tugas pada waktu yang bersamaan: *face detection, face border regression and face feature point positioning.* Karena ketiga tugas tersebut membutuhkan label pelatihan yang berbeda, maka *loss function* yang berbeda diperlukan.

face detection menggunakan *two-class cross-entropy loss function*

$$L_i^{det} = -\left(y_i^{det}\log(p_i) + \left(1 - y_i^{det}\right)(1 - \log(p_i))\right) \quad (3)$$

P_i mewakilki kemungkinan bahwa sampel X_i adalah wajah manusia, dan y_i^{det} adalah 0 atau 1

The bounding box regression dan *key point tasks* menggunakan *L2 Loss Function*

$$L_i^{box} = ||\hat{y}_i^{box} - y_i^{box}||_2^2 \quad (4)$$

Dimana $\hat{y}_i^{box}$ adalah *regression box of network output* dan y_i^{box} adalah *ground-truth coordinate.* Ada 4 titik koordinat *ground truth* dimana nilai $y_i^{landmark} \in \mathbb{R}^4$

$$L_i^{landmark} = ||\hat{y}_i^{landmark} - y_i^{landmark}||_2^2 \quad (5)$$

Diantaranya $\hat{y}_i^{landmark}$ adalah koordinat titik kunci atau *key point coordinates* dari keluaran jaringan, dan $y_i^{landmark}$ adalah *ground-truth coordinate.* Ada 5 landmark pada wajah yaitu: bagian mata kiri, mata kanan, hidung, mulut ujung kiri, dan mulut ujung kanan dimana nilai $y_i^{landmark} \in \mathbb{R}^{10}$

Jadi total *Loss Function* adalah

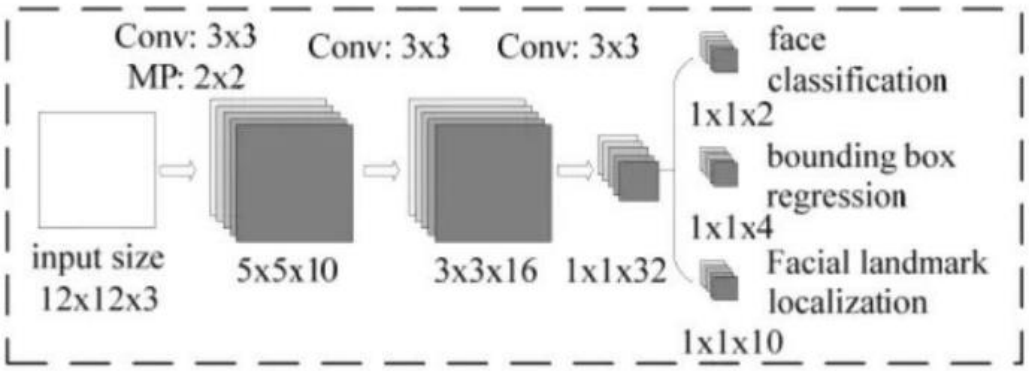
$$loss = min \sum_{i=1}^{N} \sum_{j \in \{det, bbox, landmark\}} \alpha_j \beta_i^j L_i^j \quad (6)$$

Dimana N merupakan jumlah sampel *training*, α_j mewakili pentingnya perbedaan tasks, β_i^j bernilai antara 0 dan 1 dan L_i^j mempunyai *Loss Function* yang

berbeda dalam sampel pelatihan yang berbeda. Pada P-Net dan R-Net nilai a_{det} = 1, a_{box} = 0.5, $a_{landmark}$ = 0.5, dan pada O-Net nilai a_{det} = 1, a_{box} = 0.5, $a_{landmark}$ = 1.

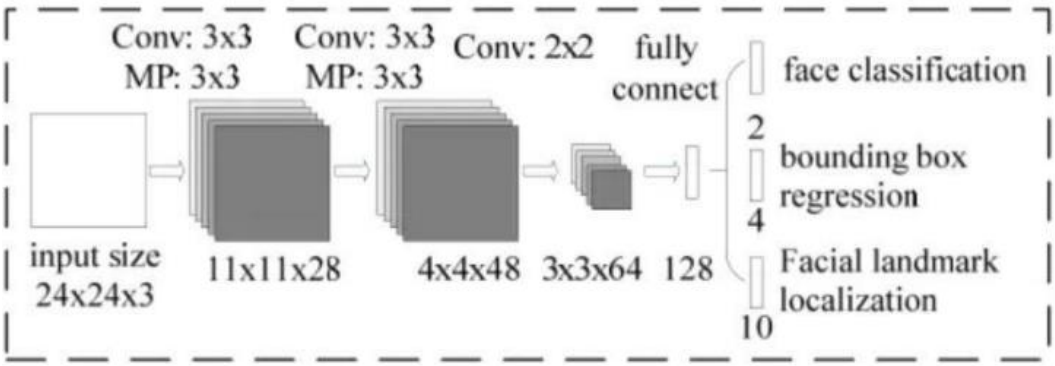
Tahap pertama dalam algoritma MTCNN adalah preprocessing data. Algoritma MTCNN merespon perubahan ukuran wajah dengan membangun piramida gambar, mengubah skala ukuran asli ke beberapa ukuran melalui faktor skala terentu, dan membangun gambar piramida sebagai input dari *network cascade architecture*.

Pada tahap pertama, semua gambar dalam piramida diperoleh melalui *shallow full convolutional neural network PNet* untuk mendapatkan kandidat *face frame* dan *face frame regression* yang digunakan untuk memperbaiki posisi kandidat *face frame*, sehingga dapat menghasilkan kandidat *face frame* dengan cepat. Kemudan NMS (*Non-Maximum Suppression*) algoritma digunakan untuk menggabungkan kandidat *face frame* dengan tingkat tumpang tindih yang tinggi.



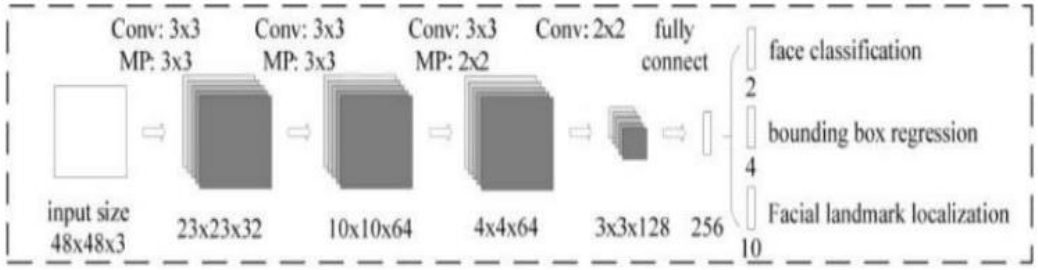
Gambar 2.4 Struktur P-net

Pada tahap kedua, kandidat face frame dihasilkan pada tahap pertama sebagai input dari R-net. R-net lebih kompleks dari P-net dalam struktur jaringan yang dapat menghapus sebagian besar kandidat *face frame* yang salah dan kemudian digunakan vektor *face frame regression* untuk menyempurnakan posisi calon *face frame*. Kemudian menggunakan algoritma NMS untuk mengurangi *face frame*.



Gambar 2.5 Strukutur R-net

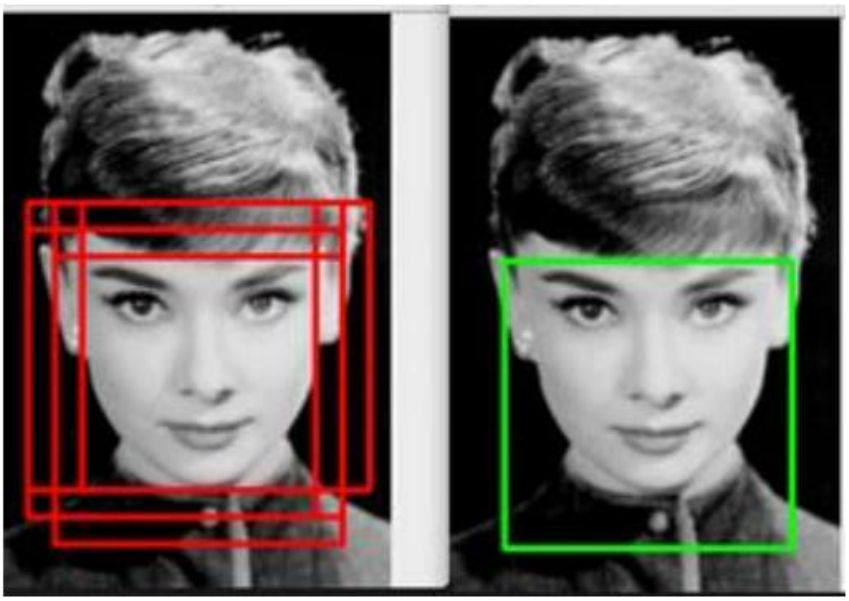
Proses tahap ketiga dan kedua serupa, dan keduanya menggunakan output proses sebelumnya sebagai input untuk proses sekarang. Sesuaikan posisi face frame saat menghapus kandidat frame yang salah. Dan memberikan informasi koordinat dari lima titik fitur wajah. Jaringan struktur O-net lebih kompleks dari pada R-net, dan hasilnya lebih akurat [16].



Gambar 2.6 Struktur O-net

2.10 Non-Maximum Suppression (NMS)

Proses dari algoritma MTCNN adalah untuk menghasilkan banyak bingkai persegi dengan ukuran berbeda berdasarkan wajah melalui *multi-scale transformation.* Untuk mendeteksi wajah dengan baik algoritma NMS digunakan dalam algoritma MTCNN sebagai *local maximum search* seperti pada gambar 2.7



Gambar 2.7 Contoh NMS

Tujuannya adalah untuk menghapus banyak kandidat face frame yang bertepatan dari target yang sama untuk menemukan batas sasaran yang optimal. Pertama kerangka kandidat gambar dengan tingkat kepercayaan yang tinggi dipilih, kemudian nilai IoU (*Intersection over Union*) yang dimiliki kandidat frame dihitung dengan kandidat frames maksimum. Dengan menetapkan nilai threshold, semua nilai kandidat frame yang memiliki nilai IoU lebih besar dari ambang batas akan dihapus untuk menghindari terjadinya *cross target* yang berulang pada kandidat frame [17].

2.11 IoU (Intersection over Union)

Intersection over Union merupakan angka yang mengukur tingkat tumpang tindih antara dua kotak. Pada kasus deteksi objek dan segmentasi, IoU mengevaluasi tumpang tindih wilayah *Ground Truth* dan *Prediction*.

Fungsi menghitung nilai IoU dapat didefinisikan dengan:

$$IoU = \frac{area(BBprediksi \cap BBgroundtruth)}{area(BBprediksi \cup BBgroundtruth)} \quad (7)$$

Gambar 2. 8 Contoh IoU Yang Baik

Gambar 2. 9 Contoh IoU Yang Cukup Baik

Gambar 2. 10 Contoh IoU Yang Kurang Baik

2.12 Python

Python merupakan bahasa pemograman yang dikembangkan oleh Guido van Rossum pada tahun 1990. Python adalah bahasa interpretatif yang dapat digunakan di berbagai platform dengan filosofi perancangan yang difokuskan pada tingkat keterbacaan kode dan merupakan salah satu bahasa pemograman populer untuk *Data Science, Machine Learning*, dan *Internet of Things* (IoT).

Bahasa pemograman *Python* utamanya mendukung multi paradigma pemrograman namun tidak dibatasi pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis.

2.13 OpenCV

Open-Source Computer Vision Library (OpenCV) adalah library yang dikhususkan untuk pengembangan computer vision. OpenCV dirilis dibawah lisensi permitif BSD yang lebih bebas dari pada GPL, dan memberikan kebebasan sepenuhnya untuk dimanfaatkan secara komersil tanpa perlu menggunakan kode sumbernya. OpenCV mendukung bahasa pemograman Python, Java, C++ dan C, dan juga sistem operasi Windows, Linux, Mac Os, Android dan iOS. OpenCV didesain dengan berfokuskan pada efisiensi pada aplikasi secara real-time [11].

2.14 Library Numpy

Numpy adalah sebuah library pada Python yang berguna untuk melakukan perhitungan scientific. Di dalamnya terdapat paket-paket untuk melakukan operasi terhadap objek array yang berupa matriks dengan N-dimensi, aljabar linear, dan lainnya.

2.15 Library Tensorflow

Tensorflow adalah sebuah framework komputasional untuk membuat model *machine learning* [20]. Tensorflow merupakan *end-to-end open source* untuk *machine learning*. Tensorflow memiliki flesibilitas ekosistem pada tool, library dan sumber data yang memungkinkan peneliti mendorong *state-of-the-art* yang

terdapat pada ML dan pengembangan yang mudah serta menggunakan aplikasi bertenaga ML. Beberapa hal yang membuat tensorflom menjadi salah satu library ML yang layak digunakan [21]:

1. Mudah dalam pemodelan Tensorflow menawarkan berbagai level abstrasi sehingga pengguna dapat memilih apa yang mereka butuhkan. Membuat dan melatih model dengan menggunakan *high-level* keras API, dimana memudahkan dalam memulai 30 *Tensorflow* dan mesin *learning* secara mudah. Jika yang dibutuhkan adalah flesibilitas, dimana eksekusi yang cepat dan *intuitive* debuging dapat menggunakan Distribusi *strategy API* untuk distribusi *training* pada beberapa perangkat yang berbeda tanpa merubah definisi model.

2. Robust ML dimana saja TensorFlow selalu menyediakan jalur langsung ke produksi. Baik itu di *server*, perangkat tepi, atau *web*, *TensorFlow* memungkinkan Anda melatih dan menggunakan model Anda dengan mudah, apa pun bahasa atau platform yang Anda gunakan.

3. Eksperimen yang kuat untuk penelitian Membuat dan melatih *state-of-the-art model* tanpa mengorbankan kecepatan dan performa. fleksibilitas dan kontrol dengan fitur-fitur seperti *Keras Functional API dan Model Subclassing API* untuk pembuatan topologi kompleks. Untuk membuat prototipe yang mudah dan *debugging* cepat, gunakan eksekusi yang cepat.

2.16 Library MTCNN

MTCNN merupakan *library* yang dibuat oleh Iván de Paz Centeno berdasarkan paper dari Zhang, K et al. (2016). *Library* ini digunakan untuk melakukan deteksi wajah dengan memberi *bounding box* dan *landmark* pada bagian wajah.

2.17 Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta

bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace Visual Studio Code (seperti C++, C#, Python, Go, Java, dst). Fitur-fitur yang disediakan oleh *Visual Studio Cod*e, diantaranya Intellisense, *Git Integration*, *Debugging*, dan fitur ekstensi yang menambah kemampuan teks editor.

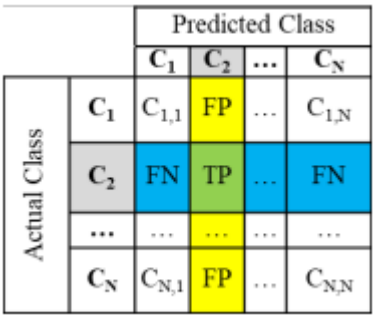
2.18 Google Colab

Google Colab adalah *notebook* Jupyter *cloud* yang digunakan untuk mengajarkan pembelajaran mesin dengan menulis kode dengan menggunakan bahasa pemograman *python* yang dapat dikerjakan melalui browser. Karya ini memperlkenalkan ekstensi Colab untuk mengajarkan *logic circuit design*, bahasa *Verilog*, prosesor dan arsitektur GPU (*Graphics Processing Unit*). *Google colab* memungkinkan untuk melakukan eksperimen pada website tanpa ketergantungan akan *hardware* komputer [18].

2.19 Confusion Matrix

Confusion matrix merupakan salah satu metode yang digunakan untuk mengukur kinerja sebuah metode klasifikasi. *Confusion matrix* pada dasarnya mempunyai infomasi yang membandingkan hasil klasifikasi yang telah dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya [19].

Berdasarkan jurnal *Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem multiclass confusion matrix* dapat dibuat dengan gambar 2.11 [28].



Gambar 2. 11 Multiclass Confusion Matrix

Mengacu pada gambar 2.11 maka TP (True Positive) dapat ditentukan jika kelas sebenarnya sama dengan hasil prediksi, sedangkan untuk FP ditentukan berdasarkan suatu kelas yang kelas sebenarnya yang diprediksi kelas tersebut pada kategori lain. Sedangkan FN merupakan total kelas sebenarnya yang salah diprediksi.

Tabel 2.1 Tabel Confusin Matrix Ekspresi Wajah

Kelas Sebenarnya	Hasil Prediksi						
	Marah	Jijik	Takut	Bahagia	Sedih	Terkejut	Netral
Marah	TP	FN/FP	FN/FP	FN/FP	FN/FP	FN/FP	FN/FP
Jijik	FN/FP	TP	FN/FP	FN/FP	FN/FP	FN/FP	FN/FP
Takut	FN/FP	FN/FP	TP	FN/FP	FN/FP	FN/FP	FN/FP
Bahagia	FN/FP	FN/FP	FN/FP	TP	FN/FP	FN/FP	FN/FP
Sedih	FN/FP	FN/FP	FN/FP	FN/FP	TP	FN/FP	FN/FP
Terkejut	FN/FP	FN/FP	FN/FP	FN/FP	FN/FP	TP	FN/FP
Netral	FN/FP	FN/FP	FN/FP	FN/FP	FN/FP	FN/FP	TP

Kemudian persamaan yang akan digunakan untuk menguji metode multi klasifikasi yang digunakan pada penelitian ini adalah sebagai berikut:

$$Akurasi = \frac{\sum_{i=1}^{N} TP(c_i)}{\sum_{i=1}^{N}\sum_{i=1}^{N} c_{i,j}} * 100\% \quad (8)$$

$$Presisi\ Kelas = \frac{TP(c_i)}{TP(c_i)+FP(c_i)} * 100\% \quad (9)$$

$$Recall\ Kelas = \frac{TP(c_i)}{TP(c_i)+FN(c_i)} * 100\% \quad (10)$$

Keterangan:

TP = True Positive

FP = False Positive

FN = False Negative