

BAB II

TINJAUAN PUSTAKA

2.1 Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif [13]. Python sendiri merupakan bahasa pemrograman tingkat tinggi (*high level language*) yang dikembangkan oleh Guido van Rossum pada tahun 1989 dan diperkenalkan untuk pertama kalinya pada tahun 1991. Python dirancang untuk memberikan kemudahan kepada programmer baik dari segi efisiensi waktu maupun kemudahan dalam pengembangan program dan dalam hal kompatibilitas dengan sistem. Python dapat digunakan untuk membuat program *standalone* dan pemrograman skrip (*scripting programming*).

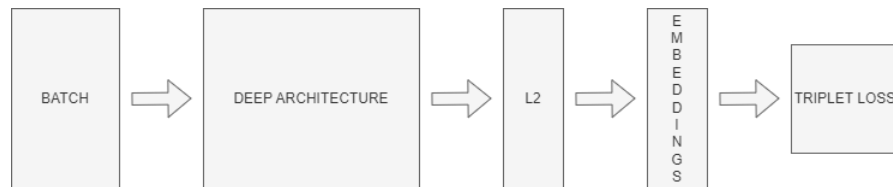
Dalam bahasa Python sendiri terdapat bahasa tingkat rendah (*low level language*) yang berhubungan dengan bahasa mesin atau *assembly*. Python memiliki beberapa kelebihan yaitu:

1. pemrograman tingkat tinggi (*high level language*)
2. mudah dipelajari
3. mudah digunakan
4. mudah dalam pengembangan
5. manajemen memori dinamis
6. pemrograman berorientasi objek (*Object Oriented Programming*)
7. platform independent
8. bersifat open source

2.2 FaceNet

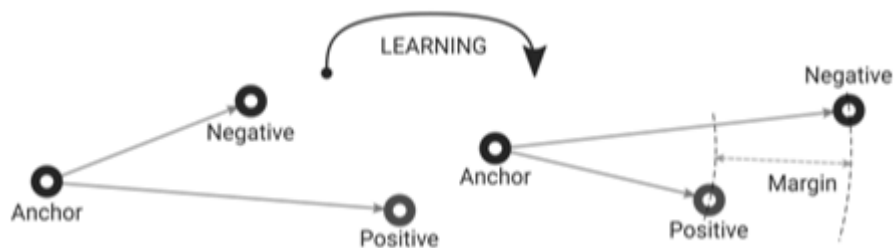
FaceNet merupakan sistem pengenalan wajah yang dikembangkan oleh peneliti Google. *FaceNet* memiliki akurasi hampir sempurna sebesar 99,63% [14]. *FaceNet* mengekstrak fitur wajah menjadi vektor menggunakan arsitektur *Deep Convolutional Neural Network* yang dilatih dengan fungsi triplet loss yang membuat vektor dengan identitas yang sama menjadi semakin serupa (jarak semakin kecil), sedangkan vektor dengan identitas yang berbeda menjadi semakin tidak serupa (jarak semakin jauh), dan identitas dari wajah yang sama akan memiliki nilai vektor yang sama [15]. Masukan dari *FaceNet* berupa gambar dan keluarannya berupa vektor sebanyak 128 elemen. Vektor wajah atau *face embedding* yang dihasilkan dapat memetakan kemiripan wajah yang memiliki kedekatan posisi pada *embedding space*, wajah yang serupa cenderung memiliki jarak yang lebih dekat dengan 0 sedangkan yang tidak serupa memiliki jarak yang lebih jauh.

Jadi bisa diartikan bahwa *face embedding* adalah sebuah hasil dari ekstraksi fitur-fitur berkualitas tinggi dari gambar wajah yang menghasilkan sejumlah elemen vektor yang merepresentasikan fitur-fitur tersebut. Selanjutnya *face embedding* yang sudah dihasilkan akan diklasifikasi menggunakan SVM.



Gambar 2.1 *FaceNet*

Secara umum, ekstraksi fitur menggunakan model *FaceNet* seperti pada Gambar 2.1, citra masukan akan masuk ke dalam arsitektur *deep learning*, kemudian dinormalisasi L2 dan hasilnya berupa fitur wajah (*face embedding*) yang dilatih menggunakan *Triplet*. Metode latihan *triplet loss* memiliki tiga unsur utama, yaitu *anchor*, *positive*, dan *negative*. *Triplet loss* ini bekerja dengan meminimalkan jarak antar-*anchor* secara positif dan memaksimalkan jarak antar-*anchor* secara negatif, dimana *positive* ini memiliki identitas yang sama dengan *anchor* dan *negative* memiliki identitas yang berbeda dengan *anchor* [16]. Proses latihan bisa dilihat pada Gambar 2.2.

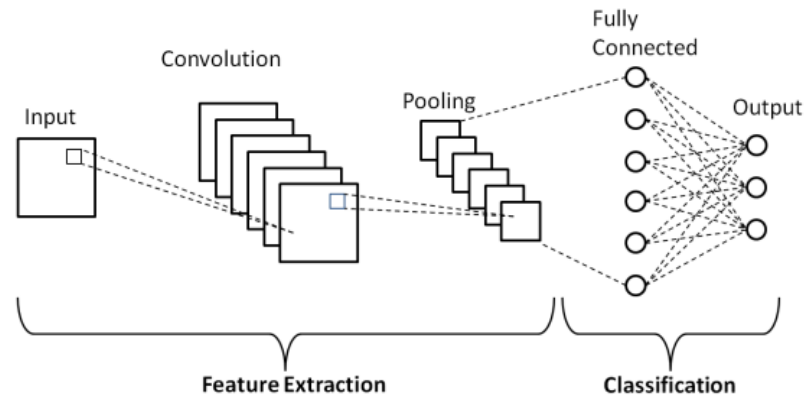


Gambar 2.2 *Triplet Loss Learning*

2.3 CNN

Convolutional Neural Network (CNN) adalah salah satu arsitektur jaringan saraf tiruan yang termasuk dalam *Deep Learning* dan didesain khusus untuk menangani data berbentuk larik CNN menerapkan operasi konvolusi pada satu atau lebih lapisannya yang terinspirasi oleh sistem saraf biologis. Pada CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, sehingga metode ini cocok untuk pemrosesan dengan input berupa gambar. CNN terdiri dari beberapa *layer* dimana setiap *layer* memiliki tujuannya masing-masing. *Layer convolutional* mempelajari beberapa set dari *convolutional filter*. *Layer* aktivasi diterapkan di atas

layer convolutional untuk mendapatkan transformasi *non-linear*. *Layer pooling* (penyatuan) digunakan untuk membantu mengurangi dimensi spasial dari volume input. *Layer fully-connected* merupakan layer terakhir yang diterapkan untuk mendapatkan prediksi. Arsitektur dasar CNN dapat dilihat pada Gambar 2.3.

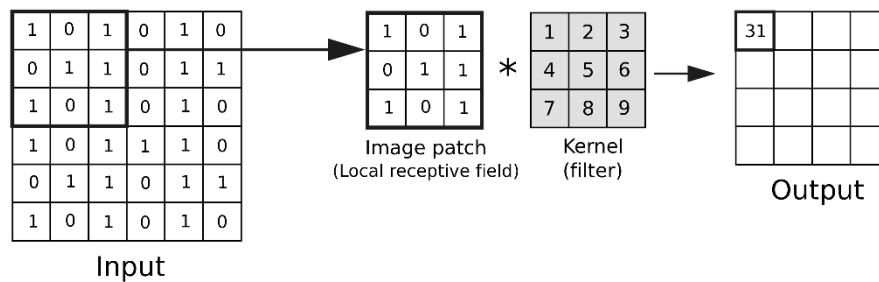


Gambar 2.3 Arsitektur CNN

Terdapat beberapa *layer* yang digunakan untuk membangun sebuah arsitektur CNN:

2.3.1 Convolutional Network

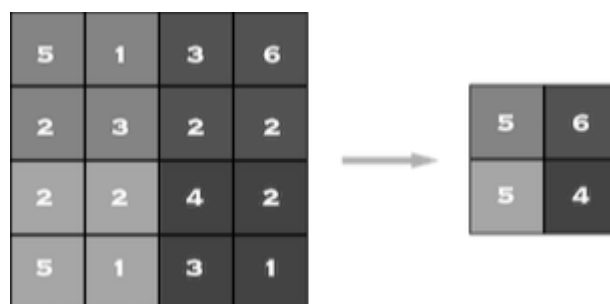
Convolutional Network atau jaringan konvolusi adalah jenis layer penting dalam *Convolutional Neural Network* (CNN). Penggunaannya yang paling umum adalah untuk mengekstraksi fitur dalam gambar, di mana ia menggunakan *filter* untuk memindai gambar, beberapa piksel sekaligus, dan menghasilkan *feature map* yang mengklasifikasikan setiap fitur yang ditemukan. Filter sendiri adalah suatu matriks 2 dimensi dengan ukuran tertentu, seperti 1x1, 3x3, dan seterusnya, yang digunakan sebagai operasi perhitungan terhadap citra masukan. Filter harus memiliki ukuran ganjil, dikarenakan filter harus memiliki koordinat tengah. Proses *layer* konvolusi dapat dilihat pada Gambar 2.4.



Gambar 2.4 Convolutional Layer

2.3.2 Pooling Layer

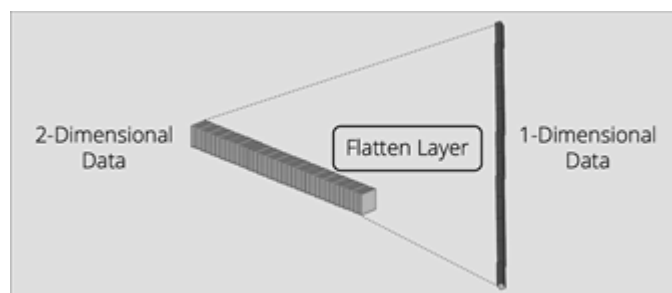
Pooling Layer merupakan *layer* yang biasanya digunakan untuk mengurangi kompleksitas dari komputasi tanpa menghilangkan fitur utamanya. Dengan kata lain, *layer* ini menggabungkan nilai piksel untuk mengurangi kompleksitas dan ini tidak mempengaruhi kinerja model. Pooling ini dapat dipilih didasarkan pada pemilihan jumlah maksimum, rata-rata, atau total nilai piksel. Metode *pooling* yang paling populer adalah *MaxPooling*, yang dapat dicontohkan seperti pada Gambar 2.5.



Gambar 2.5 Pooling Layer

2.3.3 Flatten Layer

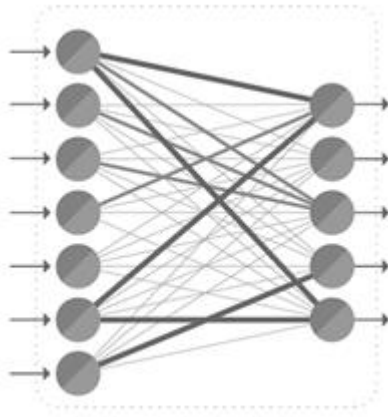
Flatten Layer berguna seperti jembatan yang menghubungkan antara *Convolutional Network* dan *Fully Connected Layer*. *Layer* ini biasa dipakai dikarenakan gambar dan output dari *Convolutional Network* berupa 2 dimensi, sedangkan di sisi lain *Fully Connected Layer* hanya menerima 1 dimensi input. Oleh karena itu *Flatten Layer* dipakai untuk mengubah data 2 dimensi menjadi 1 dimensi berupa vektor. Ilustrasinya seperti pada Gambar 2.6.



Gambar 2.6 Flatten Layer

2.3.4 Fully Connected Layer

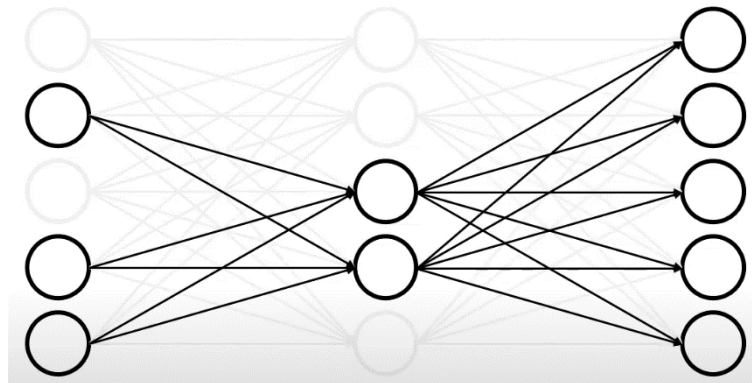
Fully Connected Layer adalah layer yang menghubungkan setiap neuron dalam satu layer ke layer berikutnya seperti yang dapat dilihat pada Gambar 2.7.



Gambar 2.7 Fully Conncted Layer

2.3.5 Dropout Layer

Dropout adalah teknik regularisasi jaringan syaraf dimana beberapa neuron akan dipilih secara acak dan tidak dipakai selama pelatihan. *Dropout* mengacu kepada menghilangkan neuron yang berupa *hidden* mapun *layer* yang *visible* di dalam jaringan. Salah satu kegunaan dari *dropout* sendiri adalah untuk mengatasi *overfit*, karena dengan menggunakan *dropout* terkadang model akan melihat fitur-fitur yang belum dilihat dikarenakan neuron-neuron yang menjadi fitur utamanya dihilangkan dengan *dropout layer*. Ilustrasinya dapat dilihat pada Gambar 8.



Gambar 2.8 Dropout Layer

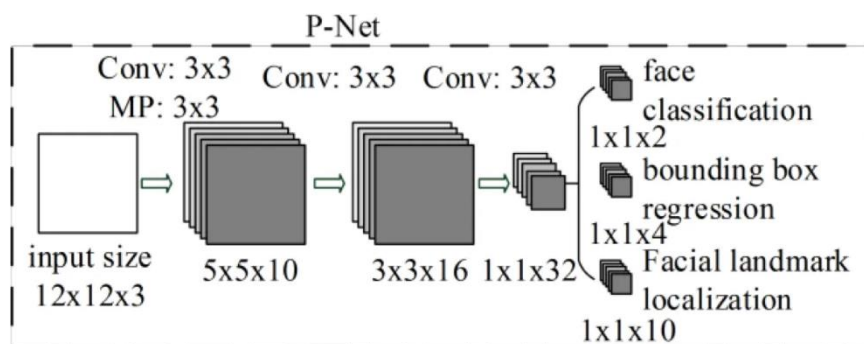
2.4 Multi-Task Cascaded Convolutional Neural Networks (MTCNN)

MTCNN adalah salah satu variasi dari metode *Deep Convolutional Neural Network* yang biasa digunakan untuk mengenali atau mendeteksi wajah. MTCNN terdiri dari 3 jaringan konvolusional: P-Net, R-Net, dan O-Net. Di jaringan P-Net. Ketiga tahapan tersebut mampu mengenali wajah dan lokasi *landmark*, seperti mata, hidung, dan mulut [17]. Pada saat MTCNN memproses sebuah gambar, proses ini akan melakukan operasi perubahan ukuran gambar untuk menskalakan gambar asli ke skala yang berbeda untuk menghasilkan piramida gambar, kemudian gambar dari skala yang berbeda akan dikirim ke 3 tahap jaringan untuk pelatihan yang berguna untuk mendeteksi wajah [17][18].

2.4.1 Tahapan Jaringan Dalam MTCNN:

2.4.1.1 Proposal Network (P-Net)

Proposal Network (P-Net) adalah jaringan pertama di MTCNN. P-Net bertugas untuk menghasilkan kandidat wilayah yang mungkin mengandung wajah dalam gambar masukan. Kandidat dan kotak vektor regresi dibatasi dan diperoleh melalui P-Net, kemudian kotak vektor regresi digunakan untuk mengkalibrasi kandidatnya. P-Net mengambil gambar masukan dan menghasilkan proposal wajah serta titik *landmark* (koordinat titik-titik penting pada wajah) sebagai keluaran. Setelah itu, *non-maximum suppression* (NMS) akan menggabungkan kandidiat yang saling tumpang tindih (*overlapped*) [19]. Struktur P-Net dapat dilihat pada Gambar 2.9.

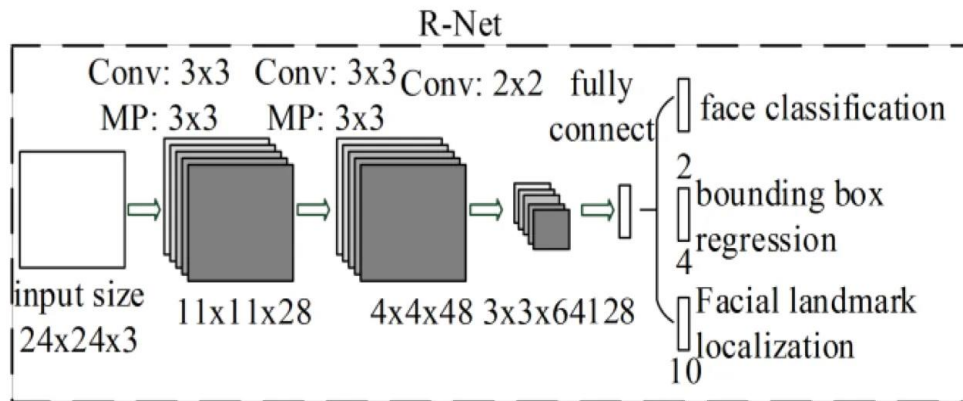


Gambar 2.9 P-Net

2.4.1.2 Refine Network (R-Net)

Refine Network adalah jaringan kedua dalam algoritma MTCNN. Setelah proposal wajah dihasilkan oleh P-Net, proposal-proposal tersebut akan mengurangi jumlah kandidat dan melakukan kalibrasi atau penyaringan lebih lanjut dengan kotak vektor regresi untuk menggabungkan kandidat yang tumpang tindih

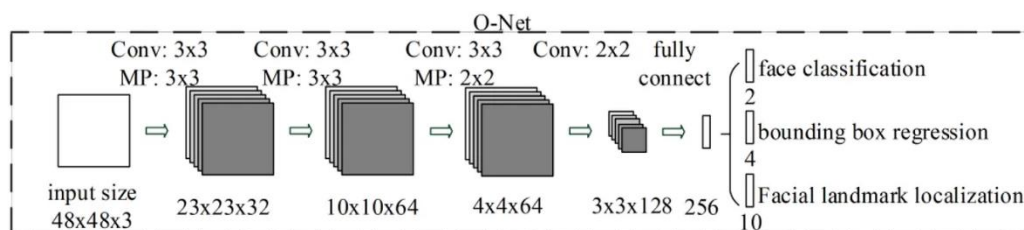
(*overlapped*). Dalam hal ini, R-Net akan menghasilkan titik *landmark* yang lebih akurat daripada P-Net [19]. Struktur R-Net dapat dilihat pada Gambar 2.10.



Gambar 2.10 R-Net

2.4.1.3 Output Network (O-Net)

Output Network adalah jaringan ketiga dan terakhir dalam algoritma MTCNN. O-Net adalah CNN yang lebih efisien untuk mengoptimalkan hasil dari R-Net. R-Net akan menampilkan lima posisi *landmark* wajah utama untuk mata, hidung, dan mulut [19]. Struktur O-Net dapat dilihat pada Gambar 2.11.



Gambar 2.11 O-Net

2.4.2 Tugas utama MTCNN

2.4.2.1 Face/non-face classification

Merupakan masalah klasifikasi binary yang akan menentukan ada atau tidaknya wajah pada suatu gambar yang menggunakan *cross-entropy loss*. *Cross-entropy loss* sendiri merupakan *loss function* yang banyak digunakan pada tugas klasifikasi. *Binary cross-entropy* mengukur perbedaan antara dua distribusi probabilitas untuk variabel acak tertentu atau serangkaian peristiwa. Dalam *case* ini *binary cross-entropy* akan menentukan antara 2 probabilitas yaitu: ada wajah atau

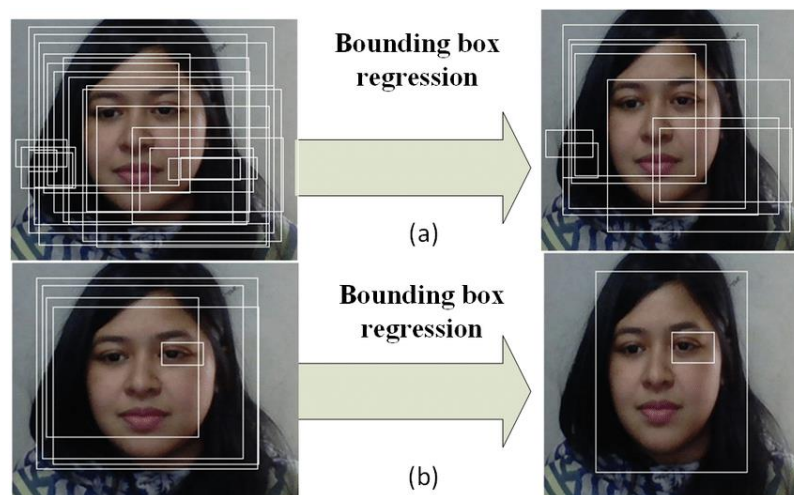
tidak ada wajah dalam suatu gambar. Untuk perhitungannya dapat dilihat pada Formula 0-1.

$$H(P, Q) = -\sum P(x) \log(Q(x)) \quad (2-1)$$

Dimana H adalah fungsi *cross-entropy*, P merupakan distribusi target dan Q adalah perkiraan distribusi target.

2.4.2.2 Bounding-box regression

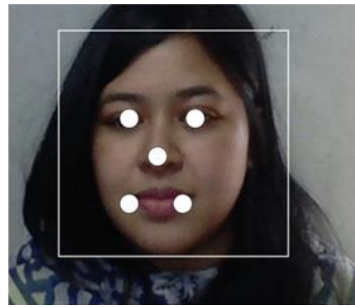
Merupakan masalah regresi yang mampu diselesaikan oleh MTCNN yang menggunakan euclidean loss / euclidean distance. Euclidean loss mampu membedakan satu wajah dengan wajah yang lain dengan cara diberi kotak pada sekitaran wajah. Contoh bounding-box dapat dilihat pada Gambar 2.12.



Gambar 2.12 Bounding-box regression

2.4.2.3 Facial landmark localization

Merupakan masalah regresi yang menggunakan *euclidean loss*. Berfungsi untuk menentukan titik *landmark* penting yaitu: mata kiri, mata kanan, ujung mulut kiri, ujung mulut kanan, dan hidung. Contoh *facial landmark* dapat dilihat pada Gambar 2.13.



Gambar 2.13 *Facial landmark*

2.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang pattern recognition. SVM adalah metode learning machine yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *input space* [20]. Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada *input space*. Pattern yang merupakan anggota dari dua buah kelas: +1 dan -1 dan berbagi alternatif garis pemisah (*discrimination boundaries*). *Margin* adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing kelas. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM [21].

2.6 Penelitian Terdahulu

Tahapan ini digunakan sebagai dasar dalam penyusunan kerangka berfikir penelitian.

Tabel 2.1 Penelitian terlebih dahulu

No	Judul	Tahun, Penulis	Metode	Hasil/Kesimpulan
1.	Implementasi <i>Convolutional Neural Network</i> (CNN) untuk <i>illumination-invariant face recognition</i> menggunakan <i>dataset</i>	2021, Jonathan Imago Dei Gloriawan	FaceNet, ResNet DL, Pose Invariant, ResNet50, InceptionV3	Tujuan dari penelitian ini adalah mencari yang terbaik diantara 5 metode untuk pengenalan wajah. Dataset yang digunakan berupa <i>dataset extended yale</i>

	<i>extended yale face database B</i>			<i>database B.</i> Hasilnya FaceNet dan ResNet50 mendapatkan akurasi sebesar 98.29 dan 97.95.
2	Face Recognition Based on MTCNN and FaceNet	2021, Rongrong Jin, Hao Li, Jing Pan, Wenxi Ma, and Jingyu Lin	FaceNet dan MTCNN	Membandingkan arsitektur dalam FaceNet yaitu <i>ZeilerFergus</i> dengan 1×1 konvolusi dan normalisasi, ResNet50, dan ResNet101. Hasil akurasi yang didapat sebesar 89.52%; 90.16%; 92.86%.
3.	Pengembangan Aplikasi Android Presensi Kehadiran Realtime menggunakan Pengenalan Wajah dengan Model Facenet	2022, Natanniel Eka Christyanto, Eriq Muhammad Adams Jonemaro, Novanto Yudistira	FaceNet	Tujuan dari penelitian ini adalah meningkatkan sistem proses presensi yang awalnya masih secara manual (kertas) menjadi lebih modern dengan membuat aplikasi kehadiran di android menggunakan FaceNet. Hasil akurasi berupa 100% yang dicoba dengan 15x percobaan.
4.	Pengenalan Ekspresi Wajah secara Realtime	2022, Luky Vianika Sari, Aziz Musthafa, Triana Harmini	Haarcascade Classifier dan FaceNet	Dataset merupakan <i>self-build</i> , didapatkan dari foto pribadi

	menggunakan Transfer Learning pada FaceNet			menggunakan kamera dan <i>OpenCV</i> . Hasil akurasi yang didapat sebesar 93%.
5.	Sistem Presensi Mahasiswa Menggunakan Face Recognition Dengan Metode Facenet Pada Android	2022, Evelyn, Rudy Adipranata, Kartika Gunadi	FaceNet, L2Norm	Dataset berjumlah 100 data, yang terdiri dari 5 foto per orang. Hasil akurasi hanya sebesar 58%.
6.	Design and Development of Student Attention System using Face Recognition Based on MTCNN and FaceNet	2023, Moch. Maulana A. C., Hudiono, Yoyok Hero Prasetyo Isnomo	FaceNet dan MTCNN	Dapat membedakan wajah orang kembar dengan tingkat akurasi sebesar 90%.

Berdasarkan penelitian-penelitian sebelumnya, kombinasi FaceNet dan MTCNN selalu menjadi pilihan utama karena performanya yang bagus dan menghasilkan nilai akurasi yang memuaskan. Akan tetapi pada penelitian terdahulu belum ditemukan pemanfaatan *face recognition* untuk mendeteksi kecurangan pada saat pengerjaan kuis online.