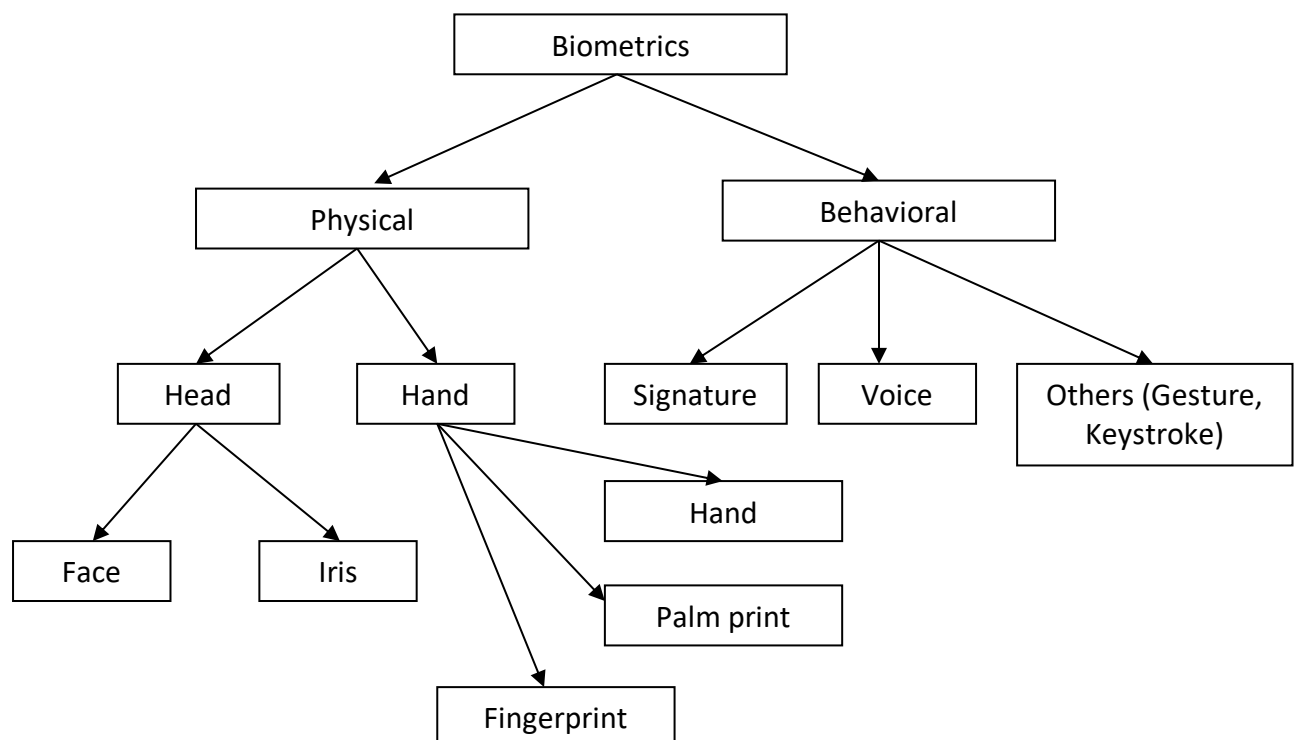


BAB II Landasan Teori

II.1 Otentikasi Biometrik

Proses identifikasi dan otentikasi merupakan kemampuan untuk memverifikasi dan mengkonfirmasi suatu identitas. Proses ini diselesaikan dengan menggunakan satu atau satuan kombinasi dari tiga teknik identifikasi tradisional: sesuatu yang anda miliki; sesuatu yang anda ketahui; atau sesuatu yang merupakan anda (J.Ashbourn, 2000). Otentikasi biometrik dikenal sebagai identifikasi otomatis, atau verifikasi identitas, dari suatu individu menggunakan suatu fitur biological yang memiliki *physiological* karakteristik seperti sebuah sidik jari, geometri tangan, scan retina, scan iris dan pengenalan wajah atau sesuatu yang dilakukan yang memiliki karakteristik perilaku, seperti sebuah tanda tangan (J.L. Wayman & Alyea L., 2000).

Menurut Woodward (J.D. Woodward, 2001) umumnya, karakteristik fisik dan perilaku yang digunakan oleh *biometrics* termasuk kedalam taxonomy berikut ini :



Gambar 2.1 Karakteristik Fisik dan Perilaku pada *Biometrics*

II.2 Deep Learning

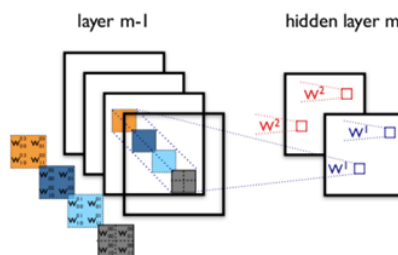
Deep learning merupakan bagian dari metode pembelajaran sistem yang direpresentasikan oleh fakta. *Deep Learning* merupakan salah satu cabang *machine learning* (ML) yang menggunakan *Deep Neural Network* untuk menyelesaikan permasalahan pada domain ML. Pengetahuan yang diperolehnya merupakan pengetahuan yang *supervised*, *semi-supervised* dan *unsupervised* (A. Alahi & okayGoel, et al, 2016). *Deep* mengenal suatu gaya dengan melakukan pemrosesan fakta dan gaya komunikasi dalam mesin biologis yang mengerikan, termasuk pengkodean saraf yang mencoba untuk menguraikan berbagai rangsangan dan respon dari syaraf terkait di dalam pikiran (V. Badarinarayan, et al, 2015). Algoritma pada *deep learning* memiliki fitur yang unik yaitu sebuah fitur yang mampu mengekstraksi secara otomatis. Hal ini berarti algoritma yang dimilikinya secara otomatis dapat menangkap fitur yang relevan sebagai keperluan dalam pemecahan suatu masalah. Algoritma semacam ini sangat penting dalam sebuah kecerdasan buatan karena mampu mengurangi beban pemrograman dalam memilih fitur yang eksplisit. Dalam jaringan saraf tiruan tipe *Deep Learning* setiap lapisan tersembunyi bertanggung jawab untuk melatih serangkaian fitur unik berdasarkan output dari jaringan sebelumnya. Algoritma ini akan menjadi semakin kompleks dan bersifat abstrak ketika jumlah lapisan tersembunyi (*hidden layer*) semakin bertambah banyak. Jaringan saraf yang dimiliki oleh *deep learning* terbentuk dari hirarki sederhana dengan beberapa lapisan hingga tingkat tinggi atau banyak lapisan (*multi layer*). Berdasarkan hal itulah *deep learning* dapat digunakan untuk memecahkan masalah kompleks yang lebih rumit dan terdiri dari sejumlah besar lapisan transformasi non-linier. Berikut jenis *deep learning*:

- *Unsupervised learning*: tipe ini digunakan pada saat label dari variable target tidak tersedia dan korelasi nilai yang lebih tinggi harus dihitung dari unit yang diamati untuk menganalisis polanya.

- *Hybrid deep networks*: pendekatan tipe ini bertujuan agar dapat dicapai hasil yang baik dengan menggunakan pembelajaran yang diawasi untuk melakukan analisis pola atau dapat juga dengan menggunakan pembelajaran tanpa pengawasan.

II.3 Convolutional Neural Network (CNN)

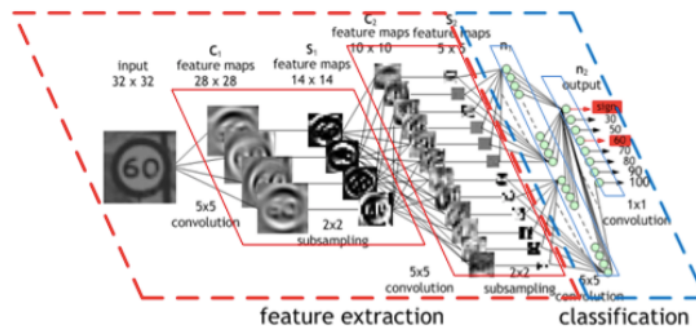
Convolutional Neural Network adalah salah satu metode *machine learning* dari pengembangan *Multi Layer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena dalamnya tingkat jaringan dan banyak diimplementasikan dalam data citra. CNN memiliki dua metode; yakni klasifikasi menggunakan feedforward dan tahap pembelajaran menggunakan backpropagation. Cara kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi. Operasi linear pada CNN menggunakan operasi konvolusi, dengan bobot yang tidak lagi satu dimensi saja, namun berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi seperti pada Gambar.2. Dimensi bobot pada CNN adalah: *neuron input x neuron output x tinggi x lebar*



Gambar 2.2 Proses konvolusi pada CNN

CNN merupakan pengembangan lebih lanjut dari MLP karena menggunakan metode yang mirip dengan dimensi yang lebih banyak. Di algoritma CNN ini, input dari layer sebelumnya bukan array 1 dimensi melainkan array 2 dimensi. Jika di analogikan dengan fitur dari wajah manusia, layer pertama merupakan refleksi goresan-goresan

berbeda arah, pada layer kedua fitur seperti bentuk mata, hidung, dan mulut mulai terlihat, hal ini karena dilakukan pooling/penggabungan dari layer pertama yang masih berupa goresan-goresan, pada layer ketiga akan terbentuk kombinasi fitur-fitur mata, hidung, dan mulut yang nantinya akan disimpulkan dengan wajah orang tertentu.



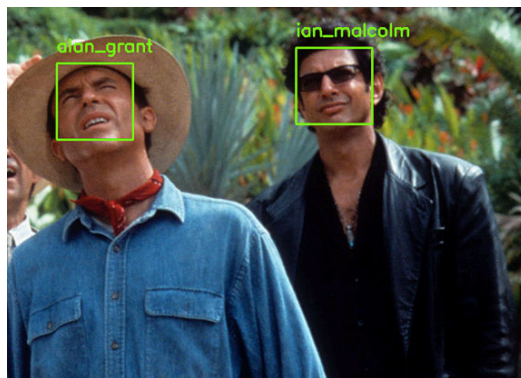
Gambar 2.3 Contoh arsitektur pada CNN

Sama halnya dengan Neural Network pada umumnya, CNN memiliki beberapa lapisan tersembunyi (hidden layers) dari sebuah input berupa vektor tunggal. Pada gambar 2.2, dengan input berupa citra yang dijadikan vektor tunggal 32×32 . Di tiap hidden layer, terdapat beberapa neuron layaknya empat feature maps C1 pada gambar tersebut. Neuron-neuron pada C1 dihubungkan dengan neuron di S1, dan seterusnya. Lapisan terakhir yang terhubung dengan lapisan-lapisan tersembunyi sebelumnya disebut dengan output layer dan merepresentasikan hasil akhir klasifikasi kelas. Seperti n2 yang ditunjukkan oleh Gambar 2.1 yang merepresentasikan hasil pada output layer, seperti 30, 50, 60, dll. Keseluruhan skala dalam objek sangat penting agar input tidak kehilangan informasi spasialnya yang akan diekstraksi fitur dan diklasifikasikan. Hal ini akan menambah tingkat akurasi dan optimum algoritma CNN. Seperti pada kubus yang memiliki skala pada panjang, lebar, dan tinggi. Jika hanya menggunakan Neural Network biasa, mungkin hanya memuat skala panjang dan tinggi. Namun CNN bisa memuat semua informasi dari keseluruhan skala yang bisa mengklasifikasikan objek

dengan lebih akurat karena bisa menggunakan skala lebarnya juga (yang mungkin tidak akan terlihat oleh Neural Network lainnya yang berdimensi dua). Dengan CNN diharapkan pengklasifikasian objek pada penelitian ini dapat menghasilkan hasil yang lebih akurat.

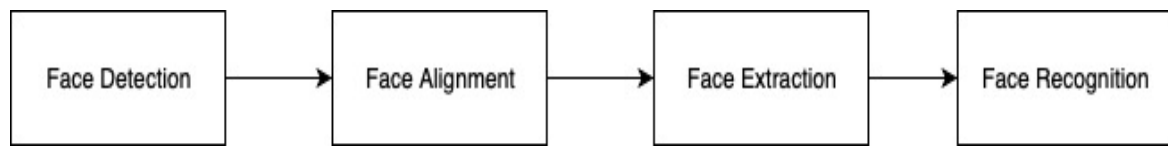
II.4 Face Recognition

Pengenalan wajah adalah salah satu ilmu yang terdapat di dalam *computer vision*, di mana sebuah komputer dapat menganalisa suatu citra wajah yang terdapat di dalam sebuah gambar dan dapat menemukan identitas atau data diri dari citra wajah tersebut dengan membandingkan terhadap data-data citra wajah yang sudah disimpan sebelumnya di dalam *database*.



Gambar 2.4 Face Recognition

Umumnya, *face recognition* dilakukan dari sisi depan dengan pencahayaan yang merata ke seluruh wajah. Saragih (2007) mengatakan bahwa *face recognition* adalah satu teknik identifikasi teknologi biometrik dengan menggunakan wajah individu yang bersangkutan sebagai parameter utamanya. Secara garis besar, proses pengenalan wajah terdiri dari empat proses utama, yaitu deteksi wajah (*face detection*), *alignment* (*face landmarks* dan *face positioning*), ekstraksi ciri atau wajah (*face* atau *feature extraction*), pengenalan wajah (*face recognition*).



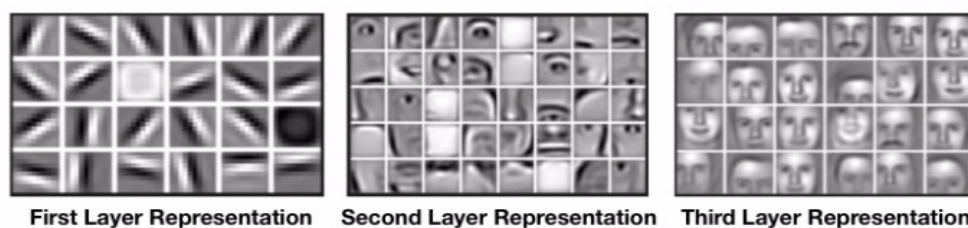
Gambar 2.5 Proses Pengenalan Wajah

Face detection merupakan langkah awal untuk melakukan identifikasi wajah atau *face recognition*. Pendeteksi wajah yang ideal seharusnya mampu untuk mengidentifikasi dan menemukan lokasi dan luas semua wajah yang ada di dalam sebuah gambar tanpa memperhatikan pose, skala wajah, orientasi, umur seseorang, dan ekspresi (Li dan Jain, 2005). *Face detection* mempunyai beragam algoritma yang dapat diterapkan dengan tujuan untuk memberikan kecepatan dan akurasi dalam mendeteksi wajah seperti Haar *Cascade Face Detector*, HOG *Face Detector*, dan *Convolutional Neural Network (CNN) Cascade*. Setelah wajah di deteksi, proses selanjutnya adalah *face alignment*. *Face alignment* bertujuan untuk mendapatkan akurasi yang lebih baik dan tinggi untuk lokalisasi dan normalisasi citra wajah, karena deteksi wajah menyediakan batas lokasi dan skala dari setiap citra wajah yang dapat terdeteksi. Setelah wajah dilakukan normalisasi, dilakukanlah *face extraction* yang digunakan untuk mengambil data yang efektif dan berguna untuk memisahkan antara citra-citra wajah dan orang-orang yang berbeda satu sama lain. Tahap terakhir adalah *face recognition* atau pencocokan wajah. Proses ini dilakukan dengan cara melakukan pencocokan fitur yang telah diekstraksi dengan citra wajah yang telah tersimpan sebagai *database* wajah

II.5 Multi-task Cascaded Convolutional Neural Network (MTCNN)

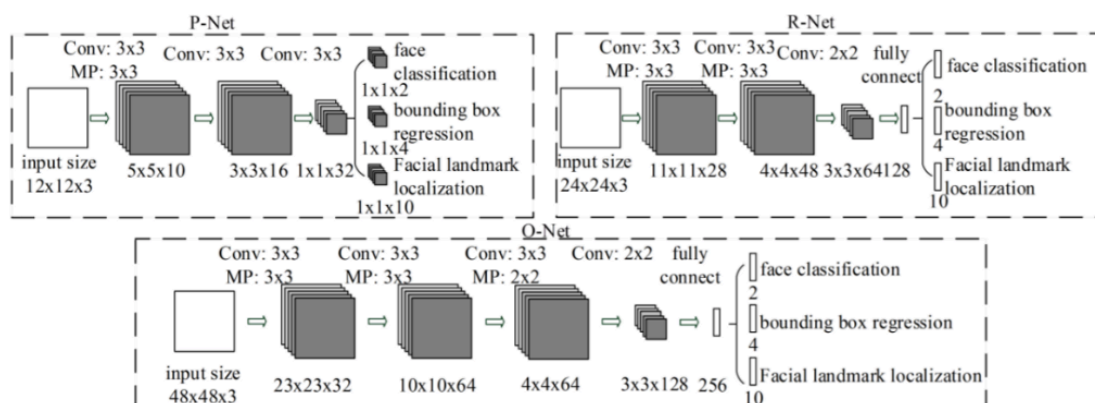
Metode konvensional seperti Haar *cascade* dan HOG berfokus pada menemukan descriptor fitur terbaik untuk menggambarkan pola visual wajah secara langsung. Namun, perkembangan *deep learning* belakangan ini mendorong upaya lebih jauh dengan memaksakan

mesin untuk mempelajari pola secara otomatis. Pembelajaran pola secara otomatis ini dapat dicapai dengan penggunaan *deep learning*, khususnya menggunakan *convolutional neural network*. Konvolusi merupakan integral yang mengukur seberapa banyak dua fungsi saling tindih ketika satu fungsi melewati fungsi lainnya, atau menganggap bahwa konvolusi merupakan cara menggabungkan dua fungsi dengan mengalikannya. Dalam *convolutional neural network*, filter yang dilalui cukup banyak melalui satu tambalan gambar. Setiap filter akan mengambil pola yang berbeda dan menghasilkan peta fitur. Semakin dalam lapisan CNN yang berjalan, pola yang lebih kompleks (peta fitur) akan diambil. Peta fitur (*feature map*) ini akan menjadi deskriptor fitur yang dapat digunakan untuk mengklasifikasikan apakah gambar merupakan wajah atau tidak.



Gambar 2.6 Peta Fitur dari Convolutional Neural Network

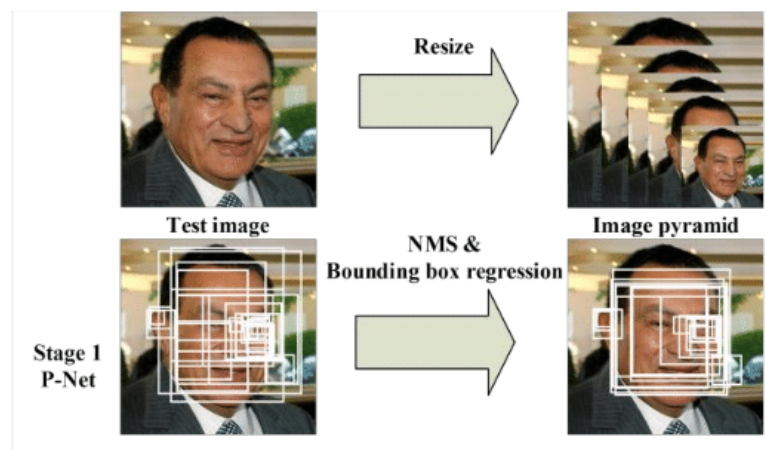
Multi-task cascade convolutional neural network (MTCNN) merupakan *framework* yang diterapkan oleh Kaipeng Zhang (Zhang, et al, 2016) yang mana dalam *framework* ini dilakukan penggabungan *face detection* dan *face alignment*.



Gambar 2.7 Arsitektur MTCNN

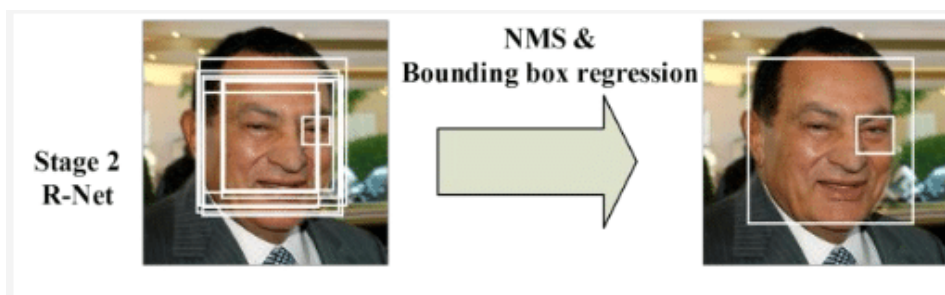
Multi-task cascade convolutional neural network (MTCNN) ini mengusulkan 3 tahapan untuk melakukan penggabungan tersebut. 3 tahapan tersebut adalah :

- a. Tahapan pertama, *fully convolutional network* yang disebut dengan Proposal Network (P-Net) digunakan untuk mendapatkan daerah yang diusulkan dan vektor regresi dari *bounding box*. Vektor regresi yang diperoleh digunakan untuk mengkalibrasi daerah yang diusulkan dan kemudian menerapkan *non-maxima suppression* (NMS) untuk menggabungkan daerah yang sangat tumpang tindih (*overlapping*)



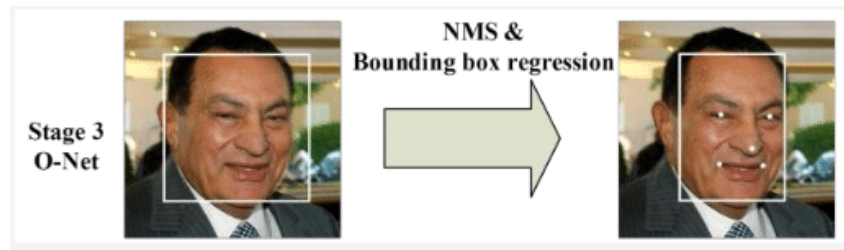
Gambar 2.7 Tahapan P-Net pada MTCNN

- b. Setelah mendapatkan hasil dari p-net, maka akan diumpankan ke CNN lain yang disebut dengan *Refine Network* (R-Net). *Refine Network* (R-Net) ini akan menolak kandidat-kandidat palsu, melakukan kalibrasi lain dengan regresi *bounding box* dan juga melakukan penggabungan kandidat NMS



Gambar 2.8 Tahapan R-Net pada MTCNN

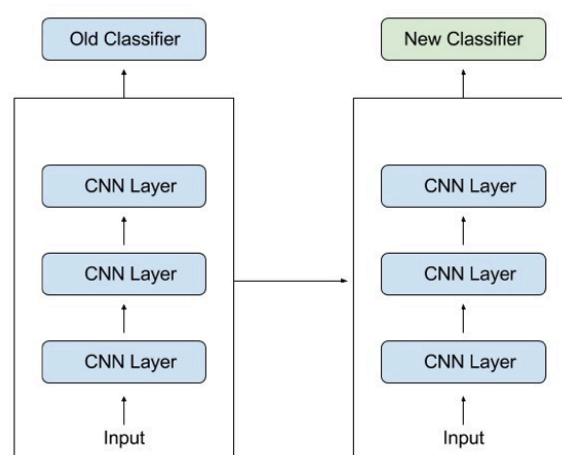
- c. Tahapan ketiga, mirip seperti tahapan yang kedua dan dinamakan dengan *Output Network* (O-Net). Tahapan ini memberikan deskripsi wajah secara detail, dan juga memberikan hasil 5 tanda posisi wajah



Gambar 2.9 Tahapan O-Net pada MTCNN

II.6 *Transfer Learning*

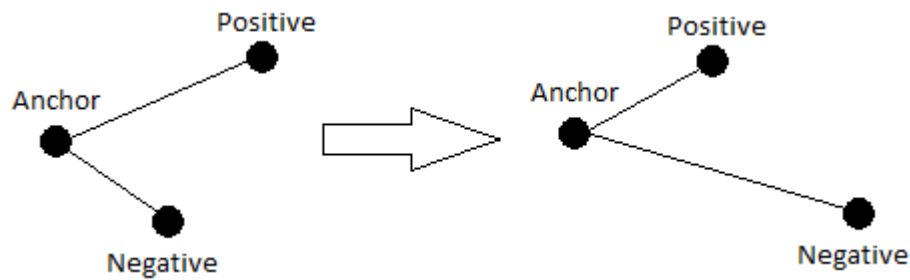
Transfer Learning adalah metode yang populer dalam *computer vision* yang memungkinkan untuk membangun model yang akurat dengan cara yang lebih efisien (Rawat dan Wang, 2017). Dengan *transfer learning*, proses pembelajaran tidak dimulai dari awal, tetapi mulai dari pola yang telah dipelajari saat memecahkan masalah yang berbeda. Hal ini dapat memanfaatkan pembelajaran sebelumnya dan menghindari memulai dari awal. Penggunaan *transfer learning* ini berguna ketika model yang telah dilatih sebelumnya mempunyai hambatan seperti pencahayaan yang terlalu terang pada wajah.



Gambar 2.10 Alur Proses *Transfer Learning*

II.6.1 Triplet Loss

Triplet loss merupakan sebuah algoritma yang diterapkan dalam OpenFace sebagai salah satu komponen dari proses stochastic gradient descent pada saat training (Amos et al., 2016). Ilustrasi mengenai mekanisme dari Triplet Loss dapat dilihat pada Gambar 2.8



Triplet Loss memiliki input berupa triplet yang terdiri atas anchor, positive, dan negative. Dalam bentuk notasi, input ini dapat ditulis sebagai $\{X_a, X_p, X_n\}$ dimana X_a merupakan anchor, X_p merupakan positive, dan X_n merupakan negative. Triplet Loss bertujuan agar jarak antara anchor dengan positive images menjadi lebih dekat daripada dengan negative images (Schroff et al., 2015). Secara formal, definisi dari Triplet Loss dapat dilihat pada Formula (2.1):

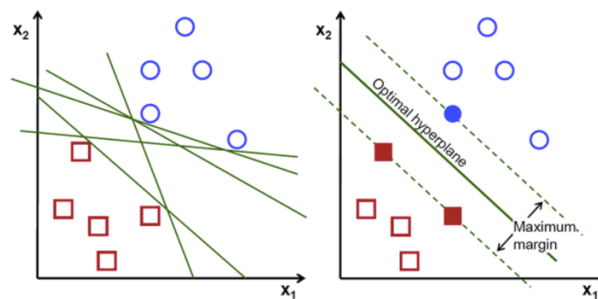
$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 \quad (2.1)$$

Pada notasi tersebut dapat dilihat adanya parameter α . Parameter ini dikenal sebagai margin yang pada penerapannya bernilai 0.2. Hal ini bertujuan agar selisih antara anchor-positive pair dengan anchor-negative pair bernilai relatif jauh, dimana hal ini pada umumnya akan memberikan performa yang baik dalam proses training. Sebaliknya, triplet pairs yang memiliki nilai diluar margin tersebut akan diabaikan karena memiliki resiko yang dapat menyebabkan proses training menjadi gagal (collapsed) dengan terjadinya early convergence.

Dengan demikian, dapat disimpulkan bahwa dalam Triplet Loss tidak semua triplet pairs yang tersedia dapat digunakan.

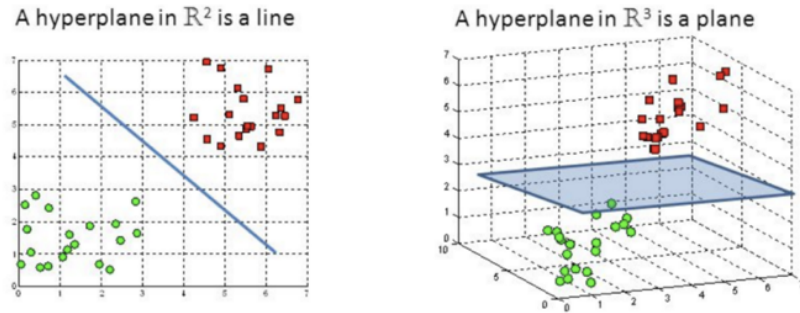
II.7 Support Vector Machine (SVM) Classifier

Support Vector Machine (SVM) merupakan metode *machine learning* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* yang memisahkan dua buah *class* pada input *space* (Byun, et al, 2003).



Gambar 2.11 Hyperplane pada SVM

Untuk memisahkan dua kelas dari titik data, ada banyak kemungkinan *hyperplanes* yang dapat dipilih. Tujuannya adalah untuk menemukan *plane* yang mempunyai maximum margin, contoh: jarak maksimum antara titik data pada setiap kelas. Memaksimalkan jarak margin memberikan penguatan sehingga titik data selanjutnya dapat diklasifikasikan dengan lebih akurat. *Hyperplanes* adalah batas keputusan yang membantu untuk mengklasifikasikan titik data. titik data yang jatuh di kedua sisi *hyperplane* dapat dikaitkan dengan kelas yang berbeda dan juga dimensi *hyperplane* tergantung pada jumlah fitur. Jika jumlah fitur input adalah 2, maka *hyperplane* hanyalah sebuah garis. Jika jumlah fitur input adalah 3, maka *hyperplane* menjadi bidang dua dimensi.



Gambar 2.12 *Hyperplane* pada 2D dan 3D

Untuk mendapatkan *hyperplane* yang terbaik, maka dapat ditemukan dengan menggunakan rumus berikut :

$$\max L_D = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j x_i x_j \quad (2.2)$$

Dengan $\{X_1, \dots, X_n\}$ adalah *dataset* input, dan $y_i \in \{+1, -1\}$ adalah lebel kelas dari data X_i . Dengan hasil tersebut akan terdapat nilai untuk setiap data *training*. Data *training* yang memiliki nilai $\alpha_i > 1$ adalah *support vector* sedangkan sisanya memiliki nilai $\alpha_i = 0$. Menurut Campbell, *support vectors* adalah titik data yang lebih dekat ke *hyperplane* dan mempengaruhi posisi dan orientasi *hyperplane* (C, Campbell, 2011). Dengan menggunakan vector-vektor pendukung ini, margin *classifier* dapat dimaksimalkan untuk memberikan hasil klasifikasi yang akurat. Dalam memaksimalkan fungsi margin, maka nilai *output* diambil dari fungsi linear dan jika *output* lebih besar dari 1 ditambahkan satu kelas, tetapi jika *output* -1 maka diklasifikasikan dengan kelas yang lain.

II.8 Tinjauan Pustaka

Beberapa penelitian sudah dilakukan untuk memberikan tingkat akurasi yang tinggi dalam mendeteksi wajah pada kasus yang berbeda-beda. Li, et al mengusulkan menggunakan *cascade* CNN untuk mendeteksi wajah, tetapi membutuhkan kalibrasi *bounding box* dari deteksi wajah dengan biaya komputasi extra dan mengabaikan kolerasi yang melekat antara lokalisasi *face landmark* dan regresi *bounding box* (Li, et al 2015). Untuk meningkatkan akurasi dan *speed* dalam pendeteksian wajah, Zhang, et al mengusulkan *framework* baru untuk mengintegrasikan *face detection* dan *face alignment* menggunakan CNN *cascade* dengan *multi-task learning*. CNN yang diusulkan terdiri dari tiga tahapan, yaitu : menghasilkan kandidat dengan cepat melalui CNN, memperbaiki hasil dengan menolak sejumlah besar kandidat yang bukan wajah melalui CNN *complex*, dan tahapan yang terakhir adalah membuat hasil *output* berupa 5 poin wajah pada *bounding box* (Zhang, et al, 2015). Berikut ini adalah hasil perbandingan dari *speed* dan akurasi dari MTCNN (*Multi-task Convolutional Neural Network*) dan CNN (*Convolutional Neural Network*) :

Tabel 2.1 Tingkat Akurasi MTCNN vs CNN

Group	CNN vs MTCNN	300 x Forward Propagation	Validation Accuracy
Group 1	12-Net (CNN)	0.38s	94.4%
	P-Net (MTCNN)	0.031s	94.6%
Group 2	24-Net (CNN)	0.738s	95.1%
	R-Net (MTCNN)	0.458s	95.4%
Group 3	48-Net	3.577s	93.2%
	O-Net	1.347s	95.4%

Kemudian, untuk mengatasi hambatan-hambatan pendeteksian wajah seperti pencahayaan yang terlalu terang pada wajah, dilakukanlah *transfer learning*. Schroff, et al melakukan penelitian menggunakan *FaceNet* (Schroff, et al, 2015). *FaceNet* merupakan *pre-trained* CNN yang melakukan *embaddings* menggunakan 128d dan merupakan model sekali pakai (*one-shot*) yang secara langsung mempelajari pemetaan dari gambar wajah ke *euclidean space* yang ringkas di mana jarak secara langsung berhubungan dengan ukuran kesamaan wajah. Setelah ruang diproduksi, tugas-tugas seperti *face recognition*, verifikasi, dan pengelompokan dapat dengan mudah diimplementasikan menggunakan teknik standar dengan *embeddings FaceNet* sebagai vektor fitur. *FaceNet* mencapai akurasi 99,63 dengan menggunakan set data *Labeled Faces in the Wild* (LFW) yang berisi 13.000 gambar wajah berlabel dari 1680 orang.

Selanjutnya, pada tahapan *classifier* untuk meningkatkan akurasi klasifikasi dilakukan dengan melatih *classifier multiclass* SVM pada fitur yang diekstraksi oleh basis *convolutional*. Penelitian dilakukan oleh Tang dengan menerapkan standar *softmax* pada fungsi *multiclass* SVM yang memberikan 10 kemungkinan kelas, penelitian oleh tang ini memberikan nilai akurasi sebesar 99.0% (Tang, 2013).

Dari tinjauan studi literature yang telah dilakukan, *Multi-task convolutional neural network* akan digunakan untuk mendeteksi wajah pada penerapan sistem absensi karyawan. Kemudian, untuk mengurangi hambatan pencahayaan pada wajah akan diterapkan *pre-trained* CNN menggunakan *FaceNet* dan langkah terakhir adalah mengklasifikasikan hasil *input-an* wajah dengan menggunakan linear SVM.