BAB III
TEORI PENDETEKSIAN WAJAH

3.1 Teori Viola Jones Detection

Teknik *face-detection* yang akan digunakan adalah teknik yang dikembangkan oleh Paul Viola dan Michael Jones-yang biasa dikenal sebagai pendeteksi Viola-Jones, dimana teknik ini dapat memproses citra dengan sangat cepat dengan tetap mencapai tingkat deteksi yang sangat tinggi (Viola and Jones, 2004). Ada tiga poin penting dalam aplikasi teknik ini, yaitu citra integral, algoritma AdaBoost, dan penggunaan *cascade*. Kelebihan dari teknik deteksi ini adalah dapat memproses dengan sangat cepat, tanpa membutuhkan spesifikasi komputer yang tinggi. Teknik ini dapat mencapai tingkatan *frame rate* yang tinggi hanya dengan menggunakan informasi yang muncul pada suatu citra *greyscale* tunggal.

3.2.1 Pembelajaran Mesin

Pembelajaran mesin atau *machine learning* adalah suatu teknik yang bertujuan untuk mengubah data mentah menjadi suatu informasi yang berguna (Bradski dan Kaehler, 2008). Setelah sebelumnya melakukan pembelajaran dari suatu kumpulan data, dengan pembelajaran mesin komputer dapat memberikan kesimpulan atau keputusan tentang data tersebut, seperti mendeteksi objek. Pembelajaran mesin mengubah data menjadi informasi dengan memisahkan aturan atau pola dari data tersebut. Bidang pembelajaran mesin ada karena dorongan pemikiran bahwa sistem dan algoritma komputer dapat meningkatkan performanya sendiri sejalan waktu (Sebe,et. al., 2005). Pembelajaran telah berevolusi dari sistem pembelajaran serba guna yang "*knowledge-free*", "*perceptron*", dan pendekatan keputusan-teoretikal untuk pembelajaran mesin, sampai pembelajaran simbolik dari pengetahuan tingkat-tinggi, dan jaringan saraf tiruan serta algoritma genetik. Dengan kemajuan terbaru dari perangkat lunak dan perangkat keras, muncul beragam aplikasi dari penelitian tentang pembelajaran mesin.

3.2.2 Citra Integral

Penggunaan representasi citra yang disebut "citra integral", yang memungkinkan fitur yang digunakan pada detektor teknik ini untuk dikomputasikan dengan sangat cepat. Citra integral hampir sama dengan jumlah dari keseluruhan tabel area yang digunakan pada *texture mapping* pada grafika komputer, dan merupakan struktur data yang memungkinkan penghitungan cepat dari daerah pada citra. Penghitungan ini sangat berguna pada banyak aplikasi, seperti komputasi dari Haar wavelets, yang digunakan pada pengenalan wajah dan algoritma sejenis. Pada teknik ini, digunakan suatu set dari fitur yang merupakan turunan dari fungsi Basis Haar, dan untuk menghitung fitur-fitur ini dengan sangat cepat pada berbagai ukuran, digunakanlah citra integral. Citra integral dapat dikomputasikan dari citra dengan menggunakan hanya beberapa operasi pada tiap *pixel*. Ketika sudah dikomputasikan, fitur-fitur sejenis-Haar ini dapat dikomputasikan pada berbagai ukuran atau lokasi dengan waktu yang konstan. Citra integral pada lokasi x,y mengandung jumlah total dari pixel diatasnya dan di sisi kiri dari x,y :

$$ii(x, y) = \sum_{x' \le x, y' \le y} i(x', y') \qquad (3.1)$$

Dimana *ii(x,y)* adalah citra integral dan *i(x,y)* adalah citra yang sebenarnya. Menggunakan pasangan ini :

$$s(x,y) = s(x,y-1)+i(x,y) \qquad (3.2)$$

$$ii(x,y) = ii(x-1,y)+s(x,y) \qquad (3.3)$$

(dimana *s(x,y)* adalah jumlah kumulatif baris, *s(x,-1)=0*, dan *ii(-1,y)=0*) citra integral dapat dihitung dalam satu kali pemrosesan citra sebenarnya.

Menggunakan citra integral, jumlah dari daerah persegi panjang di lokasi manapun dapat dihitung dengan empat array referensi, dan perbedaan dari jumlah dua daerah persegi panjang dapat dihitung dengan delapan referensi array.

3.2.3 Algoritma AdaBoost

Setelah penggunaan citra integral, kedua adalah penggunaan pengklasifikasi yang simple dan efisien yang dibangun dengan memilih sejumlah kecil dari fitur-

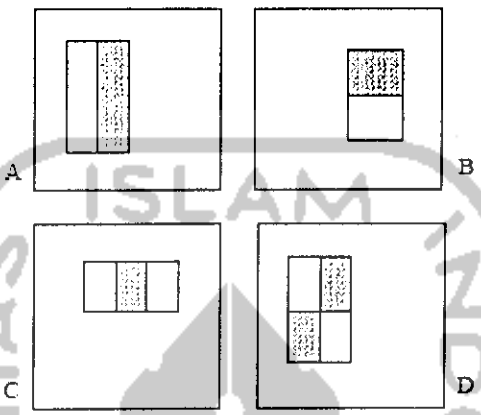
fitur penting yang berasal dari suatu pustaka yang sangat besar dari fitur-fitur potensial menggunakan AdaBoost. Untuk memastikan klasifikasi yang cepat, proses pembelajarannya harus mengecualikan suatu mayoritas besar dari fitur yang tersedia, dan memfokuskan pada suatu set kecil dari fitur yang kritikal. Pemilihan fitur ini dicapai menggunakan algoritma pembelajaran AdaBoost dengan membatasi setiap pengklasifikasi yang lemah untuk bergantung pada fitur tunggal. Sebagai hasilnya pada setiap tingkatan pada proses *boosting* untuk memilih pengklasifikasi lemah yang baru, dapat ditampilkan sebagai proses seleksi fitur.

3.2.4 Penggunaan Cascade

Ketiga adalah metode untuk mengkombinasikan pengklasifikasi yang lebih kompleks pada suatu struktur *cascade* yang mana dapat meningkatkan kecepatan detektor dengan sangat signifikan dengan lebih memfokuskan perhatian pada daerah citra yang lebih mungkin mengandung citra wajah, dimana teknik ini juga memungkinkan region latar belakang dari citra untuk diabaikan dengan cepat untuk lebih mengalokasikan komputasi atas region yang lebih mendekati wajah. Yang perlu dicatat pada pendekatan fokus ini adalah teknik ini seringkali memungkinkan untuk memberi keputusan dimana wajah mungkin muncul pada suatu citra. Proses yang lebih kompleks hanya dilakukan pada region-region ini. Kunci dari pendekatan ini adalah tingkat kesalahan negatif dari proses. Diimplementasikan pada komputer desktop biasa, deteksi wajah ini berjalan pada 15 frame per detik (Viola dan Jones, 2001). Pada intinya, teknik ini mengorganisasikannya sebagai *rejection cascade* atau susunan *cascade* penolakan dari node-node, dimana setiap node adalah pengklasifikasi pohon keputusan AdaBoost yang dirancang untuk mendapatkan tingkatan deteksi yang sangat tinggi (tingkat kesalahan negatif yang rendah, atau wajah yang luput terdeteksi) dengan mengorbankan sekitar 50% rasio penolakan (tingkat kesalahan positif, atau objek bukan wajah yang salah terdeteksi) (Bradski dan Kaehler, 2008).

3.2.5 Proses Klasifikasi dan Deteksi

Teknik deteksi objek ini mengklafikasikan citra berdasar nilai dari fitur-fitur sederhana. Banyak motivasi di balik penggunaan citra ini, daripada melakukan pengecekan *pixel* secara langsung.



Gambar 3.1 Fitur Persegi Panjang

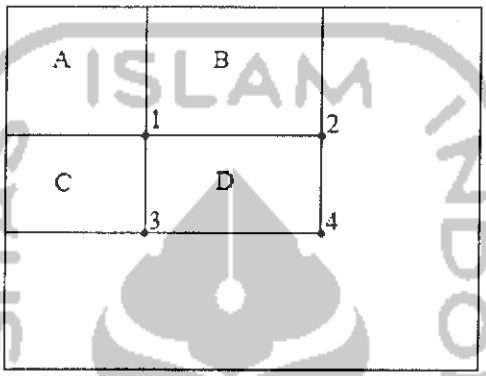
Gambar 3.1 (Viola dan Jones, 2001) menunjukkan contoh fitur persegi panjang menunjukkan hubungannya dengan *sub-window* deteksi terdekat. Jumlah dari *pixel* yang ada pada persegi panjang berwarna putih dipisahkan dari jumlah *pixel* persegi panjang berwarna abu-abu. Fitur dari dua persegi panjang ini ditunjukkan pada persegi panjang A dan B. Persegi panjang C menunjukkan fitur tiga persegi panjang, dan persegi panjang D menunjukkan fitur empat persegi panjang.

Alasan lain adalah karena fitur dapat bertindak untuk menerjemahkan domain pengetahuan *ad-hoc* yang merupakan sesuatu yang sulit untuk dipelajari menggunakan data pelatihan yang terbatas. Sistem ini juga mempunyai motivasi kritikal kedua tentang penggunaan fitur, yaitu sistem yang berbasis fitur beroperasi lebih cepat daripada sistem berbasis *pixel*.

Fitur sederhana yang digunakan merupakan turunan dari fungsi dasar Haar yang digunakan oleh Papageorgiou et al (Viola dan Jones, 2001). Lebih spesifiknya menggunakan tiga jenis fitur ini. Nilai dari fitur dua persegi panjang merupakan perbedaan antara jumlah keseluruhan *pixel* dalam lingkup dua daerah persegi panjang. Daerah yang mempunyai ukuran dan bentuk yang sama adalah berdekatan secara horizontal atau vertikal. Suatu fitur tiga persegi panjang menghitung jumlah dalam dua persegi panjang luar yang dipisahkan dari jumlah keseluruhan persegi

panjang pusat. Kemudian terakhir, fitur empat persegi panjang menghitung beda antara pasangan diagonal dari persegi panjang.

Dengan diberikan resolusi dasar dari detektor adalah 24*24 pixel, set dari fitur kotak ini berjumlah sangat besar, lebih dari 180.000. Tidak seperti basis Haar, set fitur kotak ini lebih dari lengkap.



Gambar 3.2 Penjumlahan Fitur

Jumlah dari keseluruhan *pixel* dalam persegi panjang D dapat dihitung dengan empat referensi array. Nilai dari citra integral pada lokasi ke-1 adalah jumlah dari pixel dalam persegi panjang A. Nilai dari lokasi ke-2 adalah A+B, pada lokasi ke-3 adalah A+C, dan pada lokasi ke-4 adalah jumlah A+B+C+D. Jumlah pada D saja dapat dihitung sebagai 4+1-(2+3) (Viola dan Jones, 2001).

Banyak metode pembelajaran mesin yang dapat dipakai pada fungsi klasifikasi ini, tetapi yang diterapkan disini adalah varian dari teknik AdaBoost (Adaptive Boost). Pada teknik AdaBoost yang sebenarnya, teknik ini digunakan untuk meningkatkan performa pada klasifikasi dari algoritma pembelajaran sederhana. Sedangkan pada sistem *face-detection* ini, AdaBoost digunakan untuk memilih sejumlah kecil fitur dan juga melatih unit pengklasifikasi. Alasan kenapa teknik AdaBoost digunakan, karena telah dibuktikan bahwa pada pelatihan pengklasifikasi kuat, tidak menghasilkan error, bahkan pada beberapa kali pelatihan. Lebih penting lagi, teknik ini telah terbukti meningkatkan performa generalisasi yang berhubungan pada pelatihan.

Menurut hipotesa Viola-Jones, sejumlah kecil dari fitur-fitur ini dapat dikombinasikan untuk membentuk pengklasifikasi yang efektif, dengan mengambil

contoh dari hasil 180.000 fitur persegi panjang yang diasosiasikan dengan setiap *sub-window* pada citra,yang mana komputasi dari semua ini sangat berat dan lamban.

Untuk mencari fitur yang dapat dikombinasikan ini, algoritma pembelajaran yang lemah digunakan untuk memilih fitur persegi panjang tunggal yang terbaik dalam memisahkan contoh positif dan negatif. Kemudian untuk setiap fitur, pembelajar lemah menentukan ambang batas optimal fungsi klasifikasi, seperti jumlah minimum dari contoh yang telah salah terpilih. Unit pengklasifikasi lemah hj(x) kemudian mengandung fitur fj, ambang batas [teta-j] dan paritas pj, menunjukkan arah dari pertidaksamaan :

$$h_j = \begin{cases} 1 & if\ p_j f_j(x) < p_j \theta_j \\ 0 & otherwise \end{cases} \qquad (3.4)$$

Disini x adalah *sub-window* 24x24 *pixel* dari citra. Namun pada prakteknya tidak ada fitur tunggal yang dapat melakukan tugas klasifikasi dengan error yang rendah, tingkatan errornya berkisar antara 0,1 dan 0,3.

Berikut adalah algoritma AdaBoost yang dipakai (Viola dan Jones, 2001):

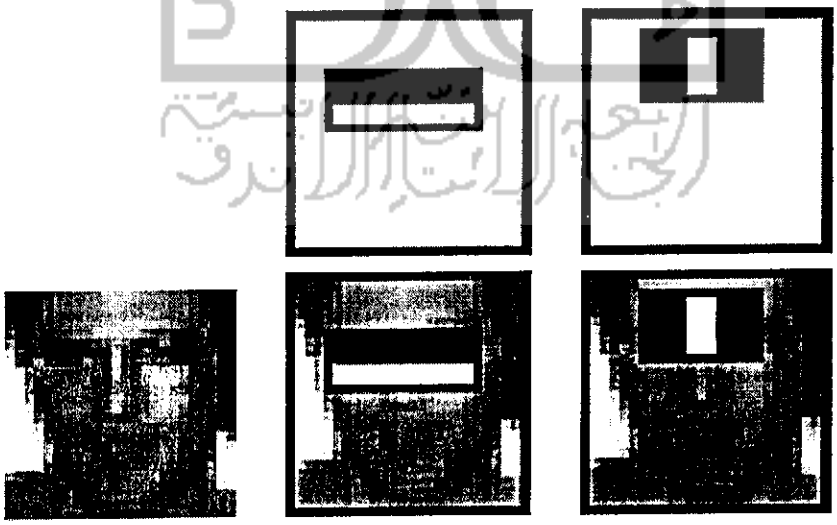
1. Dengan diberikan contoh citra $(x_1,y_1),\ldots,(x_n,y_n)$ dimana $y_i=0,1$ untuk contoh negatif dan positif berturut-turut.
2. Tentukan bobot $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ untuk $y_i = 0,1$ masing-masing, dimana m dan l masing-masing adalah nilai positif dan negatif.
3. Untuk $t=1,\ldots T$:
   a. Normalisasikan bobot,
   $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
   sehingga w_t merupakan distribusi probabilitas.
   b. Untuk setiap fitur j, latih sejumlah a pengklasifikasi hj untuk satu fitur. Kemudian *error* dievaluasi dengan melalui wt, $E_j = \sum_i w_i \mid h_j(x_i) - y_i \mid$.
   c. Pilih pengklasifikasi, h_t, dengan error terendah e_t.
   d. Perbarui bobot : $w_{t+1,i} = w_{t,i}\beta_t^{1-ei}$
   Dimana $e_i = 0$ jika contoh x_i diklasifikasikan dengan benar, $e_i = 1$ sebaliknya, dan $\beta_t = \frac{\epsilon t}{1-\epsilon t}$
4. Pengklasifikasi kuat yang terakhir adalah :

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

dimana $a_t = \log 1/\beta_t$

Dari algoritma diatas, setiap kali pelatihan memilih satu fitur dari 180.000 fitur yang potensial. Eksperimen awal menunjukkan bahwa pengklasifikasi wajah dari sisi depan yang dibangun dari 200 fitur menghasilkan tingkatan deteksi 95% dengan tingkatan kesalahan positif 1 dari 14084. Hasil ini cukup bagus, tetapi belum cukup untuk diterapkan pada tugas-tugas *real-time*. Pada lingkup komputasi, mungkin pengklasifikasi ini tergolong bekerja dengan cepat daripada sistem lain, dimana teknik ini hanya membutuhkan 0.7 detik untuk memindai suatu citra dengan luas 384x288 pixel. Namun teknik paling tepat untuk meningkatkan performa deteksi yaitu menambahkan fitur kepada pengklasifikasi, secara langsung berakibat pada penambahan waktu komputasi.

Untuk permasalahan deteksi wajah ini, fitur persegi panjang yang dipilih oleh algoritma AdaBoost cukup berarti dan dapat diterjemahkan dengan mudah. Fitur pertama memfokuskan pada daerah mata, yang seringkali lebih gelap daripada daerah hidung dan pipi. Fitur ini tidak begitu terpengaruh dengan ukuran dan lokasi dari citra wajah. Fitur kedua yang dipilih bergantung pada ciri bahwa mata lebih gelap daripada daerah diantara mata dan hidung.



Gambar 3.3 Fitur yang dipilih AdaBoost

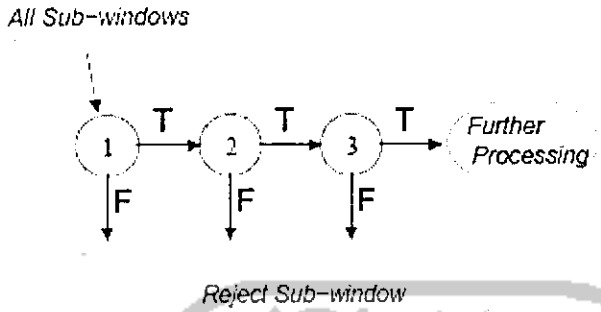
Gambar 3.3 (Viola dan Jones, 2001) menunjukkan fitur pertama dan kedua yang dipilih oleh algoritma AdaBoost. Dua fitur yang ada pada baris atas ditimpakan ke citra wajah yang dilatih pada baris bawah. Fitur pertama mengukur perbadaan

dalam intensitas antara daerah mata dan daerah di sekitar atas pipi. Fitur ini berdasar atas observasi bahwa daerah mata seringkali lebih gelap daripada pipi. Kemudian fitur kedua membandingkan intensitas daerah mata dengan intensitas bagian atas hidung.

Berikutnya adalah penggunaan algoritma untuk membangun suatu susunan *cascade* dari pengklasifikasi yang mencapai performa deteksi yang tinggi, dengan mengurangi waktu komputasi secara signifikan. Untuk dapat mencapai ini, diperlukan pengklasifikasi yang lebih kecil dan lebih efisien, serta telah mengalami *boost* yang dapat dibangun dengan melakukan penolakan pada banyak hasil *sub-window* negatif dan mendeteksi sebagian besar hasil yang positif, yang dapat dicapai dengan mengatur ambang batas dari pengklasifikasi hingga tingkat kesalahan negatifnya mencapai nol. Kemudian pengklasifikasi yang lebih sederhana digunakan untuk menolak sebagian besar *sub-window* sebelum pengklasifikasi yang lebih rumit dipanggil untuk mencapai tingkat kesalahan positif yang rendah.

Bentuk keseluruhan dari proses deteksi adalah pohon keputusan degenerasi, yang selama ini disebut "*cascade*", dapat dilihat pada gambar 3.3. Hasil positif dari pengklasifikasi pertama menyebabkan evaluasi dari pengklasifikasi kedua yang mana telah diatur agar dapat mencapai tingkat deteksi yang tinggi. Hasil positif dari pengklasifikasi kedua ini memanggil pengklasifikasi ketiga, dan begitu seterusnya. Hasil yang negatif akan menyebabkan dilakukan penolakan secara langsung pada *sub-window* yang bersangkutan.

Tingkatan pada *cascade* dibangun dengan melatih pengklasifikasi menggunakan algoritma AdaBoost dan kemudian mengatur ambang batas untuk meminimalisir tingkat kesalahan negatif. AdaBoost sendiri dirancang dengan tingkat error yang rendah atas data pelatihan. Secara umum, ambang batas yang rendah menghasilkan tingkat deteksi yang tinggi dan juga tingkat kesalahan positif yang tinggi.



Gambar 3.4 *Cascade* Deteksi

Gambar 3.4 (Viola dan Jones, 2001) adalah gambaran skematik dari susunan *cascade* deteksi, yang merupakan urutan pengklasifikasi yang diaplikasikan pada setiap *sub-window*. Pengklasifikasi yang sedang berjalan membuang sejumlah besar contoh negatif dengan pemrosesan yang sangat kecil. Kemudian lapisan berikutnya membuang hasil negatif yang lain, tetapi membutuhkan komputasi tambahan dan lebih rumit dari sebelumnya. Setelah beberapa tingkatan dari pemrosesan, *sub-window* yang ada telah berkurang banyak. Pemrosesan lebih lanjut dapat mencapai banyak bentuk, seperti penambahan lapisan dari *cascade* pada sistem deteksi ini.

Suatu unit pengklasifikasi yang bekerja dengan sangat baik dapat dibangun dari pengklasifikasi kuat dua-fitur, dengan cara mengurangi ambang batas untuk meminimalisir kesalahan negatif. Dengan menggunakan suatu set data pelatihan validasi, ambangnya dapat diatur agar tingkat deteksinya 100%, dengan tingkatan kesalahan positifnya 40%. Dijelaskan pada gambar 3, penggunaan dua-fitur ini.

Struktur dari *cascade* mencerminkan fakta bahwa dalam suatu citra, sebagian besar *sub-window* yang bekerja mempunyai hasil negatif. Karenanya, susunan *cascade* ini mencoba melakukan penolakan hasil negatif sebanyak mungkin pada tingkatan *cascade* yang paling awal.

Hampir mirip dengan pohon keputusan, pengklasifikasi berikutnya dilatih dengan menggunakan contoh yang dapat melalui tingkatan klasifikasi sebelumnya. Hasilnya, pengklasifikasi kedua menghadapi tugas yang lebih sulit daripada pengklasifikasi pertama. Contoh yang dapat melalui pengklasifikasi pertama memerlukan komputasi yang lebih sulit daripada contoh biasanya. Contoh-contoh lebih sulit yang dihadapi pengklasifikasi yang lebih dalam, mendorong kurva ROC

(*Receiver Operating Characteristic)* menjadi menurun. Dengan tingkat deteksi yang diberikan, pengklasifikasi yang lebih berikutnya mempunyai tingkatan kesalahan positif yang lebih tinggi.

Viola dan Jones mengorganisasikan setiap grup pengklasifikasi yang telah mengalami boost menjadi suatu node cascade penolakan. Setiap node mengandung keseluruhan cascade yang telah mengalami boost dari beberapa grup dari pohon keputusan yang telah dilatih pada fitur sejenis-Haar dari wajah dan bukan wajah (atau bisa juga objek lain). Umumnya, node-node ini diatur berurutan dari yang paling kompleks, sehingga komputasinya dapat diminimalisir karena yang dicoba adalah node yang paling sederhana komputasinya, ketika melakukan penolakan pada region dari citra. Biasanya, proses boost pada setiap node diatur sehingga mempunyai tingkatan deteksi yang sangat tinggi, yang berakibat banyak hasil kesalahan positif. Ketika dilatih pada wajah contohnya, hampir sebagian besar (99,9%) dari wajah ditemukan, tetapi juga banyak (sekitar 50%) dari yang bukan wajah, secara error, terdeteksi pada setiap node. Tapi ini tidak terlalu masalah, sebab menggunakan sekitar 20 node akan tetap menghasilkan tingkatan deteksi wajah sekitar 98% dengan tingkatan kesalahan positif hanya 0.5% (Bradski dan Kaehler, 2008).

Ketika aplikasi dijalankan, sebuah daerah pencarian dengan ukuran yang berbeda-beda ditelusuri secara keseluruhan citra aslinya. Pada prakteknya, 70-80% dari yang objek bukan wajah ditolak pada dua node pertama dari cascade penolakan, dimana setiap node menggunakan sekitar sepuluh cabang dari pohon keputusan (Bradski dan Kaehler, 2008).

Proses pelatihan *cascade* melibatkan dua tipe *'trade-off'*. Pada sebagian besar kasus, pengklasifikasi dengan banyak fitur akan lebih dapat mencapai tingkatan deteksi yang tinggi dan tingkat kesalahan positif yang rendah. Di sisi lain, hal ini menyebabkan waktu komputasi yang lebih tinggi. Pada prinsipnya dapat didefinisikan framework optimisasi dimana : i)jumlah dari tingkatan pengklasifikasi, ii)jumlah fitur pada setiap tingkatan, iii)ambang batas setiap tingkatan, merupakan *trade-off* yang bertujuan untuk meminimalisir jumlah fitur terevaluasi yang

diharapkan. Sayangnya, menemukan nilai optimal ini merupakan hal yang sangat sulit.

Pada prakteknya, *framework* yang sangat sederhana digunakan untuk memproduksi pengklasifikasi yang efektif yang bekerja dengan sangat efisien. Setiap lapisan pada cascade mengurangi tingkat kesalahan positifnya dan menurunkan tingkat deteksi. Suatu target dipilih untuk reduksi kesalahan positif dan penurunan tingkat deteksi. Setiap lapisan dilatih dengan cara menambahkan fitur sampai target deteksi dan tingkat kesalahan positif ditemukan, dan tingkat kesalahan positif ini ditentukan dengan melakukan tes menggunakan set validasi. Lapisan baru ditambahkan sampai keseluruhan target untuk tingkat kesalahan positif dan deteksi dapat tercapai.