Fire Detection

Langkah-langkah pengerjaan :

1. Deteksi Gerak

Deteksi gerak digunakan untuk menentukan pixel bergerak dari suatu gambar, deteksi gerak ini dilakukan dengan cara membandingkan antara frame yang dilihat saat ini dengan frame sebelumnya. Deteksi gerak dilakukan karena salah satu sifat api yang memiliki gerak yang dinamis. Dalam pengerjaan TA ini saya menggunakan library yang sudah ada. Pada library yang saya gunakan, harus memasukkan berapa learning rate yang digunakan. Pada kasus ini saya menggunakan learning rate 0. Hal ini dikarenakan inginnya detail yang saya dapatkan dari api tersebut.



Figure . api yang bergerak

1. Probabilitas Warna

Dalam asumsi pengerjaan, api yang akan dikenali adalah api yang memiliki warna merah ( berada dalam range warna merah ) antara kuning hingga merah menggunakan probabilitas gaussian dan menetapkan threshold dari dataset api yang saya ambil. Dataset yang saya ambil berasal dari gambar api. Ada 11 gambar api yang saya jadikan sebagai data api. Untuk threshold yang digunakan adalah 10-8 , angka tersebut didapatkan dengan menghitung rata-rata dari dataset api yang saya gunakan. Dalam pengerjaan TA ini, saya membuat fungsi sendiri untuk menentukan apakah suatu pixel termasuk dalam kandidat api atau tidak.



Figure . probabilitas warna api

1. Keterangan Cahaya Gambar

Salah satu sifat api adalah mengeluarkan cahaya sendiri, dari salah satu sifat ini saya menggunakannya sebagai salah satu cara untuk menghilangkan pixel – pixel yang bukan api. Membandingkan nilai pixel tersebut dengan nilai rata-rata nilai yang ada pada sekeliling objek tersebut ( menggunakan region growing untuk mendapatkan ROI objek dan nilai rata-rata sekitar objek ). Apabila nilai pixel yang dicurigai api lebih kecil ( lebih gelap ) dari sekelilingnya maka pixel tersebut akan di hilangkan dari pixel yang dicurigai sebagai pixel api. Dalam metode ini, gambar diubah kedalam luminance map agar terlihat keterangan cahanyanya. Dalam pengerjaan TA ini saya membuat sendiri fungsi untuk menentukan keterangan cahaya gambar



Figure . cahaya

1. Variasi warna objek ( ROI )

Warna api dalam satu *region* memiliki warna yang berbeda-beda ( gradasi warna ). Pada metode yang saya gunakan, saya menghitung ROI objek dan menghitung nilai standar deviasi dari region tersebut. Jika nilai standar deviasi ROI dari objek melebihi threshold, maka akan dinyatakan sebagai region api, jika tidak maka akan di hilangkan sebagai kandidat api. Pada TA ini saya membuat fungsi sendiri dalam menentukan variasi warna dari suatu ROI.



Figure . ROI

1. Perubahan warna Pixel

Asumsi yang saya gunakan adalah gerakan api yang acak membuat nilai dari suatu pixel selama beberapa frame akan berbeda. dari asumsi ini, pixel yang dicurigai api akan dilihat perubahan warna gray dengan 9 frame sebelumnya. Jika perubahan pixel tersebut tidak terlalu besar dan tidak terlalu kecil, maka pixel tersebut dianggap sebagai api. Dalam pengerjaan TA ini saya membuat sendiri fungsi untuk menentukan perubahan dari pixel.

1. Perubahan Titik Tengah Region

Titik tengah dari region api akan berbeda pada tiap frame. Hal ini dikarenakan sifat api yang bergerak dinamis sehingga titik tengah dari region api akan terus bergerak. Pada kasus ini saya menganggap bahwa api mempunyai perubahan titik tengah yang ekstrim sehigga memiliki perubahan titik tengah yang besar. Menggunakan 9 frame sebelumnya dan 1 frame saat ini untuk menentukan jarak titik tengah api tiap framenya. Titik tengah pada 1 frame bisa lebih dari 1 tergantung berapa banyaknya ROI yang ada. Pada pengerjaan TA ini saya membuat sendiri fungsi untuk menentukan titik tengah dari suatu region



1. Klasifikasi

Klasifikasi dilakukan untuk memverifikasi suatu pixel apakah pixel api atau bukan. Pada klasifikasi ini menggunakan wavelet untuk mendapatkan fitur suatu gambar. Ekstraksi fitur dilakukan dengan cara melihat nilai wavelet dari titik kandidat yang diurigai api selama 9 frame sebelumnya dan 1 frame saat ini. menggunakan svm sebagai klasifier. Pada pengerjaan TA ini saya menggunakan library yang sudah ada.