

Penerapan *Dynamic Texture* Untuk Mendeteksi Kemunculan Api Pada Video (Desember 2014)

Nurdinsyah, Fahmi R.
Sarjana Informatika Telkom University
fahmi.nurdinsyah@gmail.com

Abstrak— Api memiliki banyak manfaat bagi manusia, namun adakalanya api memberikan bencana berupa kebakaran besar apabila ada kesalahan dalam menangani api. Oleh karena itu, informasi tentang adanya api secepat mungkin sangat penting untuk mencegah api menjadi bencana bagi manusia.

Jurnal ini membahas tentang bagaimana mendeteksi api yang ada di data video dengan menggunakan beberapa metode kombinasi. Yang pertama menggunakan algoritma yang dapat menentukan daerah di dalam video dimana ada daerah yang memiliki warna api. Dari daerah-daerah ini, piksel berwarna api diekstraksi menggunakan klasifikasi warna api. Yang terakhir, menggunakan klasifikasi *dynamic texture* untuk mengkonfirmasi bahwa kandidat terdeteksi api yang bergerak dan memiliki warna yang sama dengan api ini menggunakan *Local Binary Pattern Three Orthogonal Planes* untuk mengkonfirmasi apakah calon yang terdeteksi api ini memiliki ciri api yang tetap selama waktu berlalu (perbedaan waktu) pada video.

Dari Tugas Akhir ini diperoleh akurasi sistem deteksi api yaitu persentase perbandingan hasil deteksi yang akurat dengan yang hasil deteksi tidak akurat untuk kategori api dan bukan api. Tingkat keakuratan sistem yang diperoleh yaitu 81,25%

Kata Kunci— *Dynamic Texture*, Api, Deteksi, Warna, Video, Klasifikasi.

I. PENDAHULUAN

Alat untuk mendeteksi api secara otomatis telah ada sejak penemuan alarm asap yang pertama oleh Francis Upton di tahun 1890 (1). Setelah perkembangan teknologi yang lebih jauh dan maju di pertengahan tahun 1960-an, dan harga dari pendeteksi asap berkurang, alat-alat ini mulai ada di bangunan-bangunan di seluruh dunia, dan menjadi alat yang

sangat penting dan esensial seperti sekarang (1). Akan tetapi detektor asap memiliki beberapa keterbatasan yang signifikan yang membuat mereka kurang berguna di banyak situasi penting. Sebagai contoh, detektor asap hanya bekerja dengan baik di ruang yang tertutup seperti kantor dan rumah. Tapi di ruang yang terbuka luas detektor asap tidak akan bekerja dengan baik karena membutuhkan asap sampai tingkat tertentu untuk dapat membuat alarm dari detektor asap tersebut menyadari bahwa di sana ada asap. Hal yang sama juga terjadi

pada detektor panas yang membutuhkan jarak dan penghantar panas yang memadai.

Deteksi api berbasis video tidak akan memiliki kesulitan yang didapatkan dari hambatan berupa ruang yang detektor asap dan panas dapatkan. Kamera dapat mendeteksi dan menunjuk api dari jarak yang jauh saat api mulai menyala, sehingga api tersebut bisa dijinakkan sebelum api menjadi sulit untuk dikendalikan. Yang lebih jauh lagi, kamera dapat merekam video dalam area yang luas, yang sangat memungkinkan dapat mengurangi biaya dibandingkan dengan teknologi deteksi api yang lain. Deteksi api ini bahkan memiliki potensi untuk di tempatkan di platform yang bergerak seperti pesawat atau robot.

Deteksi api berbasis video ada banyak metode yang dipakai, namun dari metode-metode tersebut yang memiliki akurasi yang baik dan konsisten secara logis adalah menggunakan klasifikasi *dynamic texture* dimana ciri api ditangkap dalam bentuk gerak dan warna sekaligus dalam selang waktu tertentu (2). Salah satu metode ekstraksi ciri dari gambar atau *image* yang paling sukses adalah *Local Binary Pattern* dengan penambahan *Three Orthogonal Planes* untuk ekstraksi ciri dari video (2). Tujuan dari penulisan jurnal ini adalah merancang dan membangun program aplikasi yang berfungsi untuk mendeteksi adanya kemunculan api dalam data video yang diberikan menggunakan metode klasifikasi *dynamic texture* dengan metode *Local Binary Pattern Three Orthogonal Planes* dengan klasifikasi *Euclidean Distance*.

II. TEORI DASAR

A. Ruang Warna $YCbCr$

$YCbCr$, $Y'CbCr$, atau $Y Pb/Cb Pr/Cr$ adalah termasuk keluarga dari ruang warna yang dikenal sebagai transmisi primer (transmission primary) yang biasanya digunakan sebagai standar digital oleh televisi (3). Ruang warna $YCbCr$ memisahkan RGB ke informasi luminance dan chrominance yang biasanya berguna dalam aplikasi kompresi (baik digital maupun analog). Y' adalah komponen luma dan CB dan CR adalah komponen perbedaan chroma biru dan perbedaan chroma merah. Y' berbeda dengan Y yang merupakan luminance yang artinya intensitas cahaya tidak secara linear di-encode berdasarkan RGB primer terkoreksi gamma. Oleh karena itu sebuah nilai yang diekspresikan sebagai $Y'CbCr$ hanya dapat diprediksi apabila standar primer chromaticities RGB yang digunakan. Sinyal $Y'CbCr$ (sebelum dilakukan

penskalaan dan offset untuk mengubah sinyal ke bentuk digital) disebut dengan YPbPr dan dibuat dari sumber RGB yang disesuaikan gamma menggunakan dua variabel konstan terdefinisi KB dan KR seperti di bawah (11)

$$\begin{aligned} & \left(\begin{array}{c} \\ \\ \end{array} \right) \quad \left(\begin{array}{c} \\ \\ \end{array} \right) \\ & - \text{—————} \quad \left(\begin{array}{c} \\ \\ \end{array} \right) \\ & - \text{—————} \quad \left(\begin{array}{c} \\ \\ \end{array} \right) \end{aligned}$$

Dimana KB dan KR secara umum turunan dari definisi ruang korespondensi RGB (manipulasi matriks yang ekuivalen sering disebut sebagai matriks warna atau color matrix)

B. Dynamic texture

Dynamic atau *temporal textures* adalah tekstur yang bergerak (4). Ini mencakup pembagian kelas dari urutan video yang memperlihatkan beberapa sifat yang tetap yang terus melekat pada video selama waktu tertentu (5). Ada banyak jenis *dynamic texture* (DT) di dunia nyata, termasuk di dalamnya gelombang laut, asap, dedaunan, api, hujan dan angin puyuh. Deskripsi dan pengenalan dari DT sangat dibutuhkan, contohnya di sistem retrieval untuk video. Polana dan Nelson mengklasifikasikan gerak visual ke aktifitas, peristiwa gerak dan *dynamic texture*. (6).

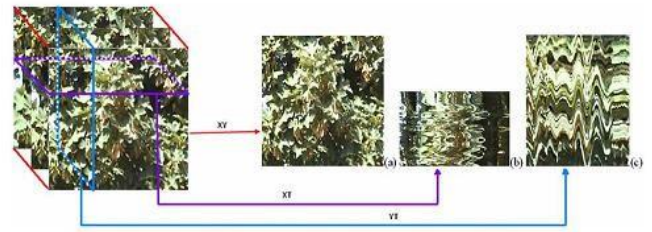
Kunci utama terkait pengambilan ciri *dynamic texture*, termasuk (a) Mengkombinasikan fitur gerak dengan fitur warna; (b) Memproses secara lokal untuk menangkap informasi transisi di ruang dan waktu, contohnya jalan lintasan yang terbakar oleh api yang kemudian berubah secara bertahap dari percikan api ke kebakaran besar; (c) Mendefinisikan fitur atau ciri yang kuat terhadap transformasi gambar seperti rotasi; (d) Tidak terpengaruh oleh variasi cahaya; (e) Kesederhanaan komputasi; (f) Uraian multi resolusi. Karena di masa lalu tidak ada metode yang dapat memenuhi semua persyaratan ini, maka ada pendekatan yang sederhana, secara teori dan komputasi, yaitu menggunakan local binary pattern (4).

C. Local Binary Pattern

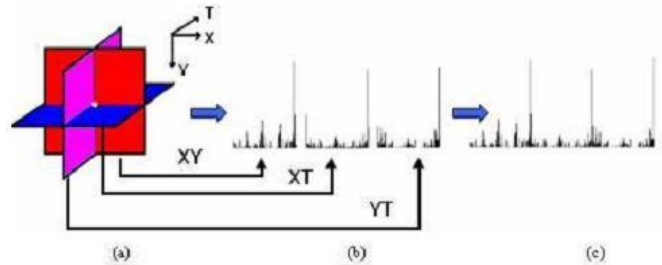
Local binary pattern (LBP) adalah jenis fitur yang digunakan untuk klasifikasi dalam computer vision (4). LBP adalah kasus khusus dari model Tekstur Spektrum diusulkan pada tahun 1990. LBP pertama kali dijelaskan pada tahun 1994. Sejak saat itu LBP telah menjadi fitur yang kuat untuk mengklasifikasi tekstur dan telah lebih jauh telah dipastikan bahwa ketika LBP dikombinasikan dengan deskriptor histogram of oriented gradients (HOG) akan meningkatkan kinerja deteksi dengan jauh lebih baik pada beberapa dataset.

D. Local Binary Pattern Three Orthogonal Layer

Untuk membuat VLBP secara komputasi cukup mudah dan sederhana untuk dikembangkan, operator berdasarkan kejadian yang sama di local binary pattern pada tiga bidang ortogonal (LBP-TOP) juga diperkenalkan. LBP-TOP memakai tiga bidang ortogonal: XY, XT dan YT, dan merangkai statistik biner pola kejadian lokal dalam tiga arah.



Gambar 2. 1 Menunjukkan contoh gambar dari tiga bidang LBP TOP (14)



Gambar 2. 2 (a)Menunjukkan tiga bidang *dynamic texture* (b)Menunjukkan histogram LBP dari tiap bidang (c)Menunjukkan ciri histogram ketiga bidang yang disambungkan (14)

XY merupakan informasi tampilan, sedangkan bidang XT memberikan kesan visual dari satu baris berubah dalam waktu dan YT menggambarkan gerakan satu kolom dalam ruang temporal. Kode LBP diekstraksi untuk semua pixel dari bidang XY, XT dan YT, dilambangkan sebagai XY-LBP, XT-LBP dan YT-LBP. Histogram dari bidang tersebut dihitung dan dirubah menjadi histogram tunggal dengan melakukan concatenation. Dalam representasi seperti itu, *dynamic texture* dikodekan oleh statistik satu tampilan (XY-LBP) dan dua temporal (XT-LBP dan YT-LBP) kejadian spasial yang bersamaan.

Dibandingkan dengan VLBP, tidak semua informasi volume yang diaplikasikan namun hanya ciri dari tiga bidang saja. Tekstur untuk frame rate yang berbeda akan menghasilkan tekstur bidang YT dan XT yang berbeda pula.

Histogram dari *dynamic texture* dapat didefinisikan sebagai

$$\sum * () + ()$$

Dengan n_j adalah jumlah label yang berbeda yang diproduksi operator LBP di bidang j^{th} ($j = 0 : XY, 1 : XT$ dan $2 : YT$), $f_i(x, y, t)$ menunjukkan kode dari piksel utama (x, y, t) di bidang j^{th} , dan $* + \{$

Ketika histogram *dynamic texture* dibandingkan dengan histogram *dynamic texture* lain dengan ukuran spasial dan temporal yang berbeda maka histogram harus dinormalisasi untuk mendapatkan deskripsi yang koheren.

$$\frac{}{\sum} ()$$

Dalam histogram ini, deskripsi *dynamic texture* secara efektif diperoleh berdasarkan LBP dari tiga bidang yang berbeda. Label dari bidang XY berisi informasi tentang tampilan muka, dan label dari bidang XT dan YT bergerak dalam arah horisontal dan vertikal. Ketiga histogram yang digabungkan

untuk membangun gambaran global *dynamic texture* dengan fitur spasial dan temporal. Normalisasi dapat diklasifikasikan api atau kebakaran hutan berdasarkan pada algoritma klasifikasi yang dalam makalah ini kami menggunakan jarak *eulclidean*.

E. Uniform LBP

Ide utama dari Uniform Local Binary Pattern adalah untuk menggunakan struktur intrinsik dari pola biner dalam Local Binary Pattern (5). Uniform LBP banyak digunakan

mengeliminasi data yang tidak relevan dan mengambil informasi yang berguna dengan menggunakan 58 nilai yang mungkin diambil.

Ojala et al. menemukan jika kebanyakan pola dari LBP merupakan uniform di tetangganya merupakan *uniform*. Sebuah pola disebut *uniform* jika dalam sebuah *sequence* terdapat maksimal dua buah perubahan dari 0 ke 1 atau

sebaliknya. Ketika menggunakan pola *uniform* semua pola *non-uniform* akan disimpan ke satu histogram baru. Ini membuat panjang dari *feature vector* jauh lebih pendek dan membuat kita mampu untuk mendefinisikan versi simple dari LBP.

F. Euclidean Distance

Dalam matematika *euclidean distance* atau metrik *euclidean* adalah jarak umum antara dua titik di ruang *euclidean* (1). Dengan jarak ini *euclidean* menjadi ruang metrik yang dapat kita ukur. Secara formal *euclidean distance* antara p dan q adalah

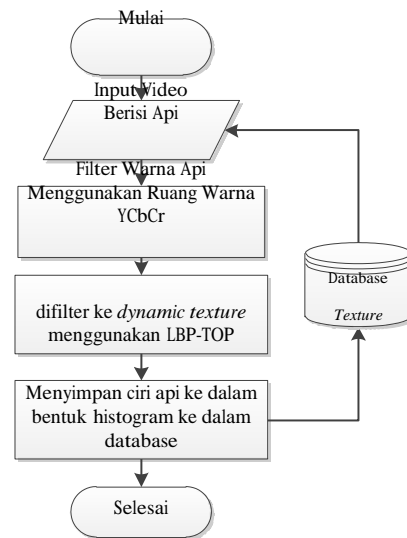
$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Dalam fase klasifikasi ketidaksamaan dalam histogram perlu untuk ditentukan seberapa jauh atau dekat. Dengan distribusi LBP, the simple *euclidean distance* sederhana dapat digunakan untuk tujuan ini.

III. PERANCANGAN SISTEM

Proses deteksi dari sistem deteksi api ini terdiri dari dua tahap utama yaitu proses *modelling* dan proses *testing*.

A. PROSES MODELLING DATABASE DYNAMIC TEXTURE

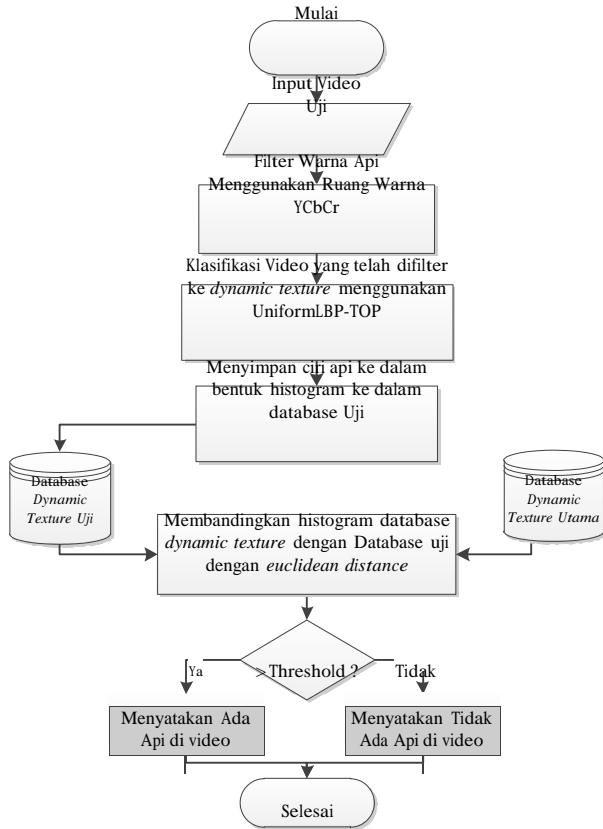


Gambar 2. 3 Proses Training Dynamic Texture dengan LBP-TOP

Proses pelatihan atau training adalah proses untuk mendapatkan ciri dari api dengan pertama menginput video yang berisi api. Setelah itu video yang berisi api ini akan dimasukkan ke dalam database *dynamic texture* dengan melakukan metode local binary pattern three orthogonal planes sehingga akan dihasilkan angka dalam bentuk biner yang merupakan ciri dari video yang berisi api tersebut. Angka dalam bentuk biner itu kemudian diubah dalam basis persepuluhan untuk memudahkan komputasi karena angka-angka tersebut akan dimasukkan ke dalam bentuk histogram. Histogram inilah ciri api yang disimpan dalam bentuk database *dynamic texture*. Karena penulis menggunakan model warna YCbCr maka akan ada tiga gray scale level values dalam database *dynamic texture*, gray scale untuk Y, Chroma Blue (Cb), dan Chroma Red (Cr) yang nantinya akan dijadikan alat perbandingan dan pengujian adanya api dalam video.

B. PROSES TESTING VIDEO

Proses pengujian adalah proses utama deteksi menggunakan database *dynamic texture*. Proses ini untuk mendapatkan ciri dari api dengan pertama menginput video yang berisi api hasil dari deteksi warna dan gerak yang telah dilakukan sebelumnya. Setelah itu video calon api akan diekstraksi cirinya dengan menggunakan metode local binary pattern three orthogonal planes dengan model warna YCbCr sehingga dihasilkan histogram yang menunjukkan ciri video calon api tersebut. Kemudian histogram hasil dari LBP-TOP akan dilakukan perbandingan dengan histogram database *dynamic texture* yang berisi ciri api dengan menggunakan *euclidean distance*. Apabila jarak *euclidean* jauh maka video calon api tersebut bukan api berdasarkan sistem deteksi, sedangkan jika jarak *euclidean* dekat maka video calon api tersebut adalah api berdasarkan sistem deteksi. Proses pengujian video akan dibagi tiap 15 frame, lalu di akhir video akan dijustifikasi apakah ada api atau tidak.



Gambar 2. 3 Proses Pengujian Video menggunakan dynamic texture dengan metode LBP-TOP

IV. HASIL PENGUJIAN SISTEM

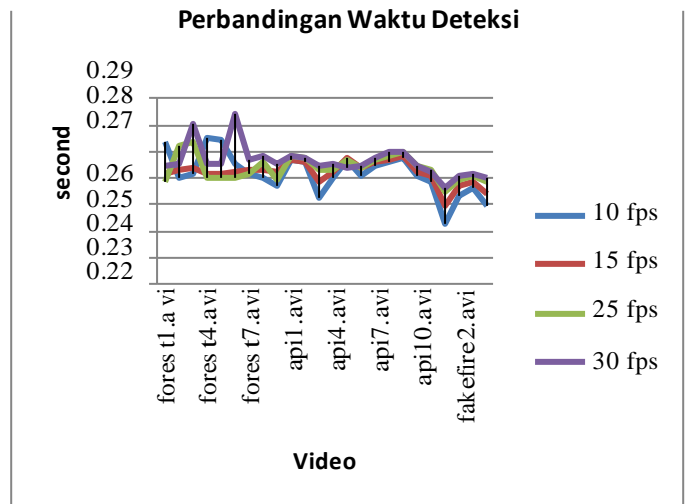
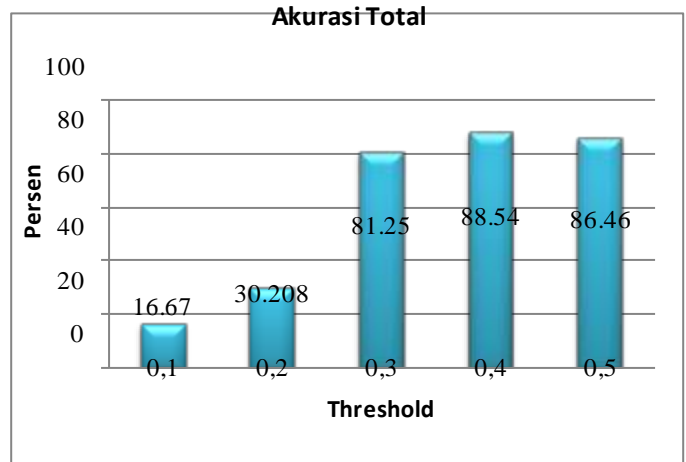
Ada dua skenario dalam jumlah ini :

1) Skenario Pertama

Skenario pertama akan melihat tingkat akurasi sistem deteksi api apabila diberi video berisi apidan bukan api dengan jarak euclidean 0,1 sampai dengan 0,5. Pengujian akan menghasilkan dua kesimpulan yaitu terdeteksi api atau tidak terdeteksi api. Setiap video akan dideteksi setiap 15 frame yang kemudian dijumlahkan hasil deteksinya. Apabila jumlah yang terdeteksi api setiap 15 frame lebih banyak maka video itu berisi api. Demikian pula sebaliknya, apabila jumlah yang terdeteksi bukan api lebih banyak maka video tersebut tidak terdeteksi api. Untuk video berisi api bernilai true jika terdeteksi sebagai api, sebaliknya pada video bukan api akan bernilai true jika terdeteksi sebagai bukan api. Dari pengujian tersebut akan dihitung tingkat akurasi rata-ratanya.

2) Skenario Kedua

Skenario kedua melihat performansi dari sistem deteksi api dengan menghitung waktu deteksi. Hasil dari performansi ini berupa waktu yang diperlukan untuk mendeteksi satu frame dengan cara membagi waktu total deteksi dengan frame yang dideteksi oleh sistem dalam bentuk detik per frame.



V. ANALISA HASIL PENGUJIAN

Dari data yang ditampilkan, maka rata-rata delay atau waktu untuk deteksi dari sistem deteksi api menggunakan dynamic texture adalah sebanyak 0.261747679 detik untuk 10 fps, 0.262186758 detik untuk 15 fps, 0.263825845 detik untuk 25 fps dan 0.26632267 detik untuk 30 fps. Kesimpulan yang dihasilkan adalah semakin tinggi frame rate-nya maka semakin lama proses deteksi per frame.

Dari hasil deteksi api yang dihasilkan, pada threshold 0,4 dan 0,5 merupakan tingkat deteksi dengan akurasi tertinggi, akan tetapi pada video bukan api (api palsu) tingkat akurasi pada kedua *threshold* itu sangat rendah dengan hanya memiliki 10 dan 3 video yang memiliki kebenaran akurasi (video bukan api dideteksi sebagai api). Hal berbeda ditunjukkan pada sistem dengan threshold 0,3 dimana tingkat akurasi untuk video bukan api sebesar 100% atau benar semua (terdeteksi sebagai bukan api). Mempertimbangkan hal tersebut maka *threshold euclidean distance* paling optimal untuk sistem ini adalah sebesar 0,3.

VI. KESIMPULAN

Berdasarkan analisis terhadap hasil pengujian tugas akhir ini, dapat diambil kesimpulan beberapa hal dibawah ini:

1. Aplikasi ini telah dapat bekerja dengan baik dengan tingkat akurasi untuk *threshold euclidean distance* 0,1 sebesar

16,67%, 0,2 sebesar 30,208%, 0,3 sebesar 81,25%, 0,4 sebesar 88,54% dan 0,5 sebesar 86,46%

2. Rata-rata *delay* atau waktu untuk deteksi dari sistem deteksi api menggunakan *Dynamic Texture* adalah sebanyak 0.261747679 detik untuk 10 fps, 0.262186758 detik untuk 15 fps, 0.263825845 detik untuk 25 fps dan 0.26632267 detik untuk 30 fps. Kesimpulan yang dihasilkan adalah semakin tinggi frame rate-nya maka semakin lama proses deteksi per frame.

3. Dari hasil deteksi api yang dihasilkan, pada *threshold* 0,4 dan 0,5 merupakan tingkat deteksi dengan akurasi tertinggi, akan tetapi pada video bukan api (api palsu) tingkat akurasi pada kedua *threshold* itu sangat rendah dengan hanya memiliki 10 dan 3 video yang memiliki kebenaran akurasi (video bukan api dideteksi sebagai api). Hal berbeda ditunjukkan pada sistem dengan *threshold* 0,3 dimana tingkat akurasi untuk video bukan api sebesar 100% atau benar semua (terdeteksi sebagai bukan api). Mempertimbangkan hal tersebut maka *threshold euclidean distance* paling optimal untuk sistem ini adalah sebesar 0,3 yaitu dengan akurasi 81,25%.

VII. REFERENSI

1. **Nicholas, True.** *Computer Vision Based Fire Detection.* University of California. San Diego : cseweb.ucsd.edu.
2. **Zhao, Guoying and Pietikainen, Matti.** *Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expression.* s.l. : IEEE Transaction on Pattern Analysis and Machine Intelligence, 2007.
3. **Keith, Jack.** *YCbCr to RGB Consideration.* s.l. : INTERSIL Corporation, 1997.
4. **Szummer, M. and Picard, R.W.** *Temporal Texture Modeling.* s.l. : Proc. IEEE Conf. Image Processing, 1996. pp. 823-826. Vol. 3.
5. **Doretto, G., et al., et al.** *Dynamic Textures.* s.l. : International Journal of Computer Vision, 2003. pp. 91-109. Vol. 51.
6. **Polana, R. and Nelson, R.** *Temporal Texture and Activity Recognition.* s.l. : Motion-based Recognition, 1997. pp. 87-115.
7. **Wikipedia.** Euclidean Distance. *Wikipedia.* [Online] November 17, 2014. [Cited: December 20, 2014.] http://en.wikipedia.org/wiki/Euclidean_distance.
8. **National Wildfire Coordinating Group.** *Glossary of Wildland Fire Terminology.* s.l. : Data Standards and Terminology Subcommittee, Oktober 2014.
9. **Beach, Lee Roy.** *Image Theory: an Alternative to Normative Decision Theory.* s.l. : n NA - Advances in Consumer Research eds. Leigh McAlister and Michael L. Rothschild, Provo, UT : Association for Consumer Research, 1993. pp. 235-238. Vol. 20.
10. **Turtinen, Markus, Maenpaa, Topi and Pietikainen, Matti.** *Texture Classification by Combining Local Binary Pattern Features and a Self-Organizing Map.* Finland : Machine Vision Group, University of Oulu, 2003.
11. **Tkalcic, Marko and Tasic, Jurij F.** *Colour Spaces - Perceptual, Historical and Applicational Background.* Ljubljana : Faculty of Electrical Engineering, University of Ljubljana, 2003.
12. **Skarbek, Wladyslaw and Koschan, Andreas.** *Colour Image Segmentation.* Technical Informatics, Technical University of Berlin. Berlin : Technical University of Berlin, 1994. p. 8, Survey.
13. **Mehta, Rakesh and Egiazarian, Karen.** *Rotated Local Binary Pattern (RLBP): Rotation invariant texture.* Department of Signal Processing, Tampere University of Technology. Tampere : Proceedings of International Conference on Pattern Recognition Applications and Methods, 2013.
14. **Joseph, Shelton, et al., et al.** *Genetic Based LBP Feature Extraction and Selection for Facial Recognition.* Greensboro : In: Proceedings of the 49th Annual Southeast Regional Conference. ACM, 2011. pp. 197-200.
15. **Huang, Di, et al., et al.** *Local Binary Patterns and Its Application to Facial Image Analysis : A Survey.* s.l. : IEEE Transaction Systems, Man and Cybernetics, 2011. pp. 765-781.
16. **Ford, Adrian and Roberts, Alan.** *Colour Space Conversion.* London : Westminster University, 1998.
17. **Beach, Lee Roy.** *Image Theory: an Alternative to Normative Decision Theory.* s.l. : n NA - Advances in Consumer Research eds. Leigh McAlister and Michael L. Rothschild, Provo, UT : Association for Consumer Research, 1993. pp. 235-238. Vol. 20.
18. **Public/Private Fire Safety Council.** *White Paper: Home Smoke Alarms and Other Fire Detection and Alarm Equipment.* s.l. : Public/Private Fire Safety Council, 2006. Technical report.
19. **UK Essay.** History And Properties Of Fire Alarms Engineering Essay. *UK Essays.* [Online] November 2013. [Cited: June 9, 2014.] <http://www.ukesays.com/essays/engineering/history-and-properties-of-fire-alarms-engineering-essay.php?ref=1>.
20. **National Wildfire Coordinating Group.** *Glossary of Wildland Fire Terminology.* s.l. : Data Standards and Terminology Subcommittee, Oktober 2014.
21. **University of Waikato.** Fire. *The Science Learning Hub (New Zealand).* [Online] November 19, 2009. [Cited: 12 2014, 16.] <http://sciencelearn.org.nz/Contexts/Fire/Science-Ideas-and-Concepts/What-is-fire>.