

Penerapan Metode *Median Filtering* untuk Optimasi Deteksi Wajah pada Foto Digital

Application of Median Filtering Method for Optimizing Face Detection in Digital Photos

Sunardi¹, Anton Yudhana², Setiawan Ardi Wijaya*³

^{1,2,3}Universitas Ahmad Dahlan

e-mail: sunardi@mti.uad.ac.id¹, eyudhana@ee.uad.ac.id², setiawanardiwijaya@gmail.com³

Abstrak

Deteksi wajah pada foto digital bertujuan untuk mendapatkan area wajah pada foto tersebut. Sering terjadi masalah saat mendeteksi wajah yakni terdapat banyak noise pada foto digital sehingga berpengaruh pada saat proses pendeteksian. Maka dari itu pada penelitian ini menerapkan metode *median filtering* untuk memperbaiki foto digital dengan mengurangi noise pada foto yang akan diuji. Keuntungan menggunakan filter ini adalah menjaga tepi agar tetap halus sekaligus mengurangi *noise* yang bertujuan untuk meningkatkan kualitas citra. Kemampuan metode ini diukur menggunakan parameter dari *Mean Square Error* (MSE) dan *Peak Noise to Signal Ratio* (PNSR). Ketika nilai MSE yang didapat bernilai rendah dan nilai PNSR yang didapat bernilai tinggi maka hasil tersebut dapat dikatakan baik. Metode yang digunakan untuk mendeteksi wajah pada riset ini ialah *viola-jones* dipilih karena salah satu prosedur deteksi wajah dengan tingkatan akurasi tinggi serta kemampuan komputasi yang baik. Sampel yang digunakan pada penelitian ini berjumlah 10 foto digital yang diambil menggunakan kamera *smartphone* Samsung A11 dengan resolusi 13 mega piksel kamera belakang dan 8 mega piksel kamera depan dengan format JPEG. Hasil yang didapat pada penelitian ini adalah nilai *error* terbaik dalam MSE 0,9517 dan PSNR 24,2804 berdasar hasil tersebut dapat disimpulkan bahwa metode *mean filtering* layak untuk digunakan pada kasus deteksi wajah pada foto digital ini.

Kata Kunci: *median filtering*, deteksi wajah, *pre-processing*.

Abstract

*Face detection in digital photos aims to get the face area in the photo. Problems often occur when detecting faces, namely there is a lot of noise in digital photos so that it affects the detection process. Therefore, in this study apply the median filtering method to improve digital photos by reducing noise in the photos to be tested. The advantage of using this filter is to keep the edges smooth while reducing noise which aims to improve image quality. The ability of this method is measured using the parameters of Mean Square Error (MSE) and Peak Noise to Signal Ratio (PNSR). When the MSE value obtained is low and the PNSR value obtained is high, the results can be said to be good. The method used to detect faces in this research, *viola-jones*, was chosen because it is one of the face detection procedures with a high level of accuracy and good computational ability. The sample used in this study consisted of 10 digital photos taken using a Samsung A11 smartphone camera with a resolution of 13 mega pixels and 8 mega pixels in JPEG format. The results obtained in this study are the best error values in MSE 0.9517 and PSNR 24.2804. Based on these results, it can be concluded that the mean filtering method is feasible to be used in the case of face detection in this digital photo.*

Keywords: *Median filtering*, *Face detection*, *Pre-processing*.

Pendahuluan

Perkembangan teknologi masa kini sangatlah cepat, salah satunya adalah pada *software* citra digital[1][2], Sistem pengenalan wajah manusia merupakan salah satunya, sebelum sistem dapat mengenali wajah manusia maka sistem harus mendeteksi bagian wajah[3][4]. Namun sering terjadi masalah saat mendeteksi foto yakni terdapat banyak noise pada foto tersebut yang berpengaruh pada saat pemrosesan[5].

Penelitian ini bertujuan untuk mengurangi noise pada foto yang akan diproses dengan menerapkan metode *median filtering* serta menguji metode tersebut apakah layak atau tidak digunakan dalam kasus deteksi wajah pada foto digital ini. Metode *median filtering* dipilih karena metode ini baik dalam melakukan perbaikan noise pada citra dan menghasilkan citra yang lebih fokus disebabkan penggantian nilai piksel menggunakan nilai tengah dari semua nilai yang ada[6][7][8]. Pada penelitian sebelumnya yang juga

*) *Penulis Korespondensi:* setiawanardiwijaya@gmail.com

menggunakan metode *median filtering* didapatkan hasil yang baik[9][10]. Kemampuan metode ini diukur menggunakan *Mean Square Error* (MSE) dan *Peak Noise to Signal Ratio* (PNSR). Semakin kecil nilai MSE yang diperoleh atau mendekati 0 maka semakin bagus hasil yang didapatkan, sedangkan semakin tinggi nilai PNSR yang diperoleh maka semakin bagus hasil yang didapatkan[11][12]. Selain itu MSE dan PNSR dipilih karena parameter ini mudah didapatkan dan banyak digunakan dalam penelitian yang berkaitan dengan citra digital [13][14][15][16][17][18].

Langkah-langkah deteksi wajah yang pertama adalah input citra guna memasukkan citra *Red Green Blue* (RGB) kedalam sistem pemroses, lalu citra RGB dikonversi menjadi citra *grayscale*[19]. Setelah itu diterapkan *median filtering* pada citra *grayscale* dengan tujuan mengurangi *noise* dan memperhalus citra sehingga mempermudah saat pemrosesan.

Metode yang dipakai ketika mendeteksi wajah adalah *Viola-Jones*. Metode ini dipilih karena salah satu prosedur deteksi wajah dengan tingkatan akurasi tinggi serta kemampuan perhitungan yang baik[3][20]. Prosedur Viola- Jones memakai fitur Haar selaku deskriptor, kemudian mengkombinasikan *Integral Image* serta *AdaBoost* untuk mendapatkan dan melakukan seleksi nilai fitur serta membentuk *Cascade Classifier*. *Classifier* tersebut yang hendak dipakai buat mencari wajah pada foto digital[5].

Foto yang digunakan pada penelitian ini menggunakan jenis format *Joint Photographic Experts Group* (JPEG), yang merupakan algoritma *lossy compress* dengan kata lain akan menurunkan kualitas pada citra[15][21][22]. Foto ialah cerminan dari objek yang dapat diambil menggunakan kamera yang menghasilkan foto sama persis dengan kondisi objek saat di ambil[23]. Foto diambil menggunakan kamera *smartphone* Samsung A11 dengan resolusi 13 mega piksel kamera belakang dan 8 mega piksel kamera depan. Foto yang digunakan sebagai sampel pada penelitian ini adalah gambar *landscape* yang jelas dan hanya terdapat satu wajah orang didalamnya.

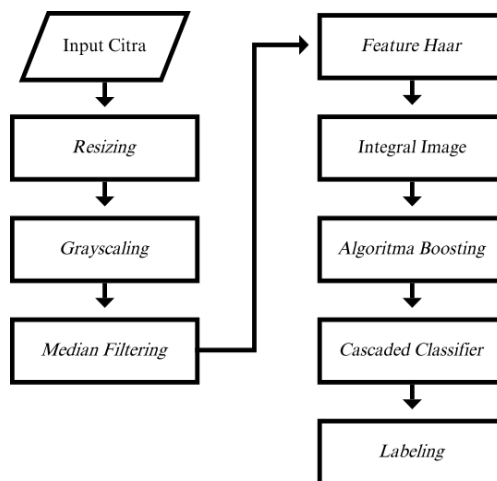
Penelitian ini menggunakan laptop Asus Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz memori 8 GB menggunakan sistem operasi Windows 10 pro dan aplikasi yang digunakan adalah Matlab a13. Matlab digunakan karena aplikasi ini didukung dengan perangkat lunak matematika, grafis, dan kemampuan pemrograman. Aplikasi ini memiliki fungsi bawaan untuk melakukan banyak operasi[24][25][26].

Metode Penelitian

Objek penelitian yang diteliti adalah *Median Filtering* untuk Deteksi Wajah pada Foto Digital. Parameter yang digunakan untuk mengukur kemampuan *Median Filtering* pada input citra berbeda adalah MSE dan PNSR. Parameter pengukuran disesuaikan dengan lingkup penelitian.

Perancangan Sistem

Penelitian ini menggunakan metode *Median Filtering* yang masuk kedalam *Pre-processing*. Penelitian ini memiliki empat tahapan, yaitu akuisisi citra, *pre-processing*, *Viola-Jones*, dan *labeling* seperti yang dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Input Citra

Pada proses ini, citra diinputkan kedalam aplikasi pemroses citra, yaitu MATLAB. Citra yang diinputkan berformat .JPEG dan bermode RGB.

Resizing

Hasil pada tahapan resizing ini merupakan citra sampel yang telah mengalami perubahan ukuran citra. Tujuan dari resizing citra adalah agar pemrosesan citra lebih cepat.

Grayscale

Tahap ini merupakan awal dari pemrosesan, citra yang awalnya berbentuk RGB diubah menjadi berskala keabuan menggunakan *grayscale* process. Konversi citra RGB ke grayscale menggunakan rumus (1).

$$I_y = 0,333F_r + 0,5F_g + 0,1666F_b \quad (1)$$

F_r merupakan intensitas untuk *Red*, sedangkan F_g merupakan intensitas untuk *Green*, dan F_b adalah intensitas untuk *Blue*, sedangkan I_y merupakan intensitas abu-abu yang setara gambar level RGB.

Median filtering

Proses perbaikan citra dilakukan supaya *noise* yang ada pada citra dapat berkurang. Cara mencari *median* yakni dengan mengurutkan nilai piksel dari nilai terkecil sampai terbesar berdasarkan delapan ketetanggaan seperti pada tabel 1, setelah itu diambil nilai tengahnya. Secara matematis, filter dapat dirumuskan sebagai berikut:

Tabel 1. Contoh piksel delapan ketetanggaan

P1	P2	P3
P8	(x,y)	P4
P7	P6	P5

$$\text{Median}(x,y) = P1, P2, P3, P4, P5, P6, P7, P8 \quad (2)$$

Feature Haar

Mekanisme deteksi wajah Viola-Jones mengklasifikasikan foto sesuai pada nilai fitur sederhana. terdapat sebagian alasan untuk menggunakan fitur daripada piksel secara langsung. Alasan yang sangat masuk akal yakni jika fitur dapat digunakan untuk mengkodekan pengetahuan domain ad-hoc yang tidak mudah dalam pengenalan terhadap data latih yang terbatas jumlahnya.

Integral image

Guna mempermudah proses menghitung nilai asal tiap fitur Haar disetiap lokasi gambar dipergunakan teknik yang dianggap citra integral. Secara keseluruhan integral memiliki makna menambah bobot, yaitu nilai-nilai piksel yang akan dibubuhi ke dalam gambar asli.

Algoritma boosting

Prosedur pemecahan *Boosting* mengkombinasikan banyak citra yang kurang tajam (*weak classifiers*) untuk menjadi citra yang lebih tajam (*strong classifiers*) dengan memberi bobot kepada citra *weak classifiers*.

Cascaded classifier

Memberi bobot pada *weak classifier* dan menggabungkan banyak *classifier* yang lemah untuk menghasilkan *classifier* yang lebih kuat. Lemah disini berarti urutan filter di *classifier* hanya menerima jawaban benar lebih sedikit. Filter disetiap level mengklarifikasi citra yang sebelumnya telah dilakukan filter. Bila salah satu filter tadi tidak berhasil, wilayah pada citra diklarifikasikan sebagai bukan wajah.

Labeling

Setelah ditahap *Cascade classifier* didapatkan filter dengan bobot paling besar lalu letakkan pada proses pertama, bertujuan untuk menghapus daerah gambar bukan wajah secepat mungkin. Selanjutnya menampilkan objek sampel gambar yang telah terdeteksi wajah ataupun bukan wajah, dengan memberi label dengan garis merah jika objek tersebut dianggap sebagai daerah wajah.

Pengujian sistem

Hasil proses *median filtering* dihitung *error*-nya menggunakan MSE dan PSNR karena kedua metode tersebut biasa digunakan untuk mengevaluasi suatu metode, sejalan dengan tujuan dari penelitian ini yakni menguji metode *median filtering* layak atau tidak digunakan pada kasus ini. Ketika hasil PSNR bernilai tinggi dan hasil MSE bernilai

rendah atau mendekati 0 maka citra yang didapat amat halus untuk persepsi mata[12]. Berikut adalah formula untuk MSE.

$$MSE = \frac{\sum_{a=0}^{x-1} \sum_{b=0}^{y-1} (M(a,b) - N(a,b))^2}{x*y} \quad (3)$$

M(a,b) adalah nilai citra hasil *Median Filtering* dan N(a,b) adalah citra referensi. PSNR adalah rasio kemungkinan maksimum *power* sinyal menjadi kemungkinan maksimum *power noise*[12]. Berikut adalah rumus menghitung PSNR:

$$PSNR = 10 \log_{10} \frac{Max_i^2}{MSE} \quad (4)$$

Max merupakan nilai piksel tertinggi pada sebuah citra biasanya bernilai 255.

Hasil dan Pembahasan

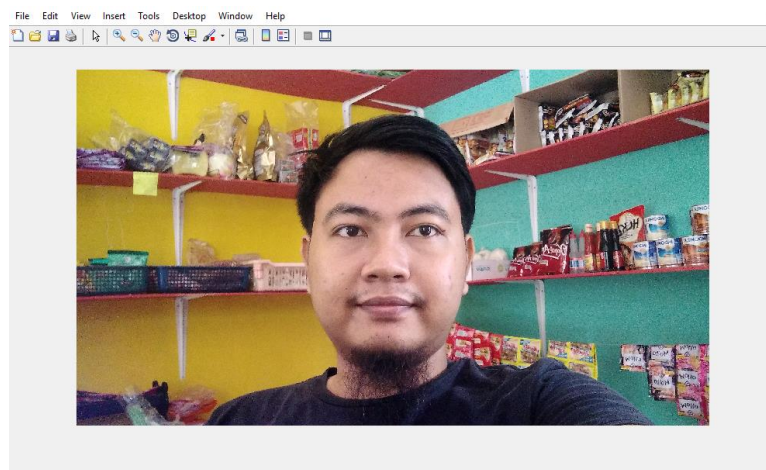
Penelitian ini menggunakan 10 foto yang terdiri dari lima foto diambil pada waktu siang hari dan lima foto diambil pada waktu malam hari, Foto diambil menggunakan kamera *smartphone* Samsung A11 dengan resolusi 13 mega piksel kamera belakang dan 8 mega piksel kamera depan. Pada tahap ini dilakukan pengujian deteksi wajah menggunakan metode *Median Filtering*. Pengujian dilakukan menggunakan aplikasi MATLAB dengan dukungan hardware Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz memori 8 GB. Berikut adalah langkah-langkah menggunakan aplikasi MATLAB untuk menguji metode *Median Filtering*.

Input citra

Tahap pertama adalah melakukan input citra menggunakan kode program 1 dan hasilnya pada gambar 2.

Kode Program 1. Kode input citra

```
%akuisisi citra
%input citra
img= ('C:\Users\acer\Documents\MATLAB\sampel\DSC_0017.jpg');
J = imread(img);
```



Gambar 2. Hasil input citra

Pada gambar 2 dapat dilihat bahwa citra sudah berhasil ditampilkan dengan format RGB.

Resizing

Tahap selanjutnya adalah melakukan *resizing* citra yang bertujuan untuk memperkecil ukuran citra agar mempercepat saat diproses, kode program 2 dan hasilnya dapat dilihat pada gambar 3.

Kode Program 2. Kode *resizing* citra

```
%resize gambar
I = imresize(I,0.6,'bilinear');
```

Name	Value
FDetect	1x1 CascadeObjectDetector
G	918x1632 uint8
I	918x1632x3 uint8
J	1836x3264x3 uint8
K	918x1632 uint8
MSE	21.3083
PSNR_tamp	10.7799
x	[560 223 491 491]

Gambar 3. Perbandingan hasil *resizing*

Pada gambar 3 dapat dilihat perbedaan ukuran antara citra awal J dan citra yang telah di-*resize* I.

Grayscale

Perubahan citra RGB menjadi citra berskala keabuan dilakukan melalui proses *grayscale* dengan kode program 3 dan hasilnya dapat dilihat pada gambar 4.

Kode Program 3. Kode *grayscale*

```
%preprocessing
%grayscale
G = rgb2gray(I);
```

**Gambar 4.** Hasil dari *grayscale*

Pada gambar 2 dapat dilihat citra yang sebelumnya berwarna RGB pada gambar 4 sudah diubah menjadi warna keabuan.

Median filtering

Pada tahap ini, citra berskala keabuan diperbaiki menggunakan *median filtering* untuk menghilangkan *noise* yang ada pada citra menggunakan kode program 4 dan hasilnya dapat dilihat pada gambar 5.

Kode Program 4. Kode *median filtering*

```
%median filter
K = medfilt2(G);
```



Gambar 5. Hasil dari *median filtering*

Pada gambar 5 dapat dilihat hasil dari proses *median filtering* bahwa *noise* dari citra tersebut sudah berhasil dihilangkan.

Deteksi menggunakan viola-jones

Ada beberapa proses pada deteksi menggunakan Viola-Jones, tahapannya diuraikan seperti berikut.

Feature Haar

Tahap ini menggunakan data latih dari citra yang tidak tajam sebagai bagian dari proses pengelompokan citra. pengelompokan citra dilakukan berdasar nilai sebuah fitur.

Integral image

Guna mempermudah proses menghitung nilai asal tiap fitur Haar disetiap lokasi gambar dipergunakan teknik yang dianggap citra integral. Secara keseluruhan integral memiliki makna menambah bobot, yaitu nilai-nilai piksel yang akan dibubuhi ke dalam gambar asli

Algoritma boosting

Tahap ini menggabungkan lebih dari satu citra yang kurang tajam (*weak classifiers*) untuk menjadi citra yang lebih tajam (*strong classifiers*) dengan memberi bobot kepada citra yang kurang tajam.

Cascaded classifier

Memberi bobot pada weak classifier dan menggabungkan banyak *classifier* yang lemah untuk menghasilkan *classifier* yang lebih kuat. Lemah disini berarti urutan filter di *classifier* hanya menerima jawaban benar lebih sedikit. Filter disetiap level mengklarifikasi citra yang sebelumnya telah dilakukan filter. Bila salah satu filter tadi tidak berhasil, wilayah pada citra diklarifikasikan sebagai bukan wajah.

Beberapa proses ini diperingkas sebagai suatu *function* pada aplikasi Matlab menggunakan kode program 5.

Kode Program 5. Kode deteksi wajah

```
%deteksi
FDetect = vision.CascadeObjectDetector;
figure, imshow(I), title('Detected faces');
x = step(FDetect, K);
```

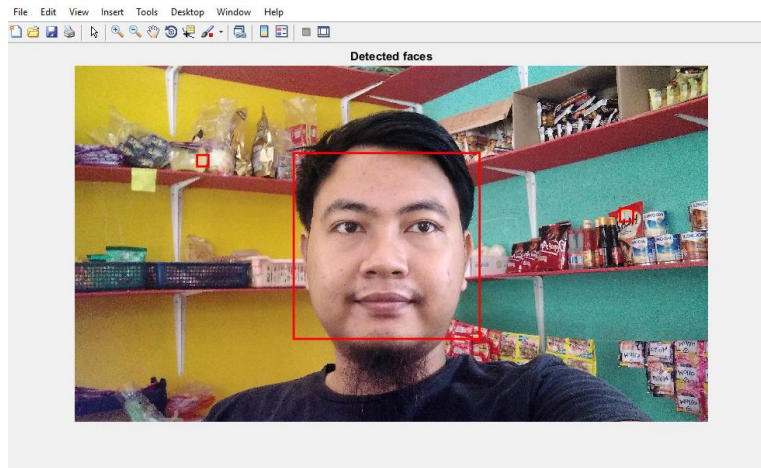
Kode program 5 merupakan *function* yang sudah disediakan oleh Matlab.

Labeling

Pada tahap ini, wajah yang sudah berhasil dideteksi menggunakan metode Viola-Jones masih belum ditandai. Untuk menandai bagian yang dideteksi digunakan proses *labeling* seperti dapat dilihat pada kode program 6 dan hasilnya pada gambar 6.

Kode Program 6. Kode labeling

```
%labeling
for i = 1:size(x,1)
    rectangle('Position', x(i,:), 'LineWidth', 2, 'LineStyle', '-
', 'EdgeColor', 'r');
end
```



Gambar 6. Hasil dari proses *labeling*

Hasil Pengujian Median Filtering

Pada tahap ini dilakukan pengujian metode *Median Filtering* untuk mengukur nilai *error* dari metode tersebut. Pengujian ini dilakukan menggunakan MSE dan PNSR dengan Kode program 7.

Kode Program 7. Kode menghitung MSE dan PNSR



```
%menghitung MSE
MSE = immse(K, G);
display(MSE);
%menghitung PNSR
PSNR_tamp = 10*(log10(255/MSE));
display(PSNR_tamp);
```

Berdasarkan kode program 7, *metode median filtering* pada sampel ini mendapatkan hasil MSE 23,9770, PNSR 10,2675, dan kecepatan proses 3,937866 detik.

Pembahasan

Dari hasil metode *Median Filtering* yang didapat, selanjutnya dilakukan pengujian nilai *error* menggunakan MSE dan PNSR. Hal ini dilakukan untuk mengetahui nilai *error* yang paling kecil diantara kedua kondisi yaitu kondisi waktu siang hari dan malam hari. Hasil pengujian menggunakan aplikasi Matlab disajikan pada tabel 2.

Tabel 2. Hasil hitung MSE dan PNSR

No	Sampel	Waktu	MSE	PNSR	Kecepatan (detik)
1		Siang 1	16,3104	11,9408	3,869012
2		Siang 2	19,0076	11,2761	3,906015

3		Siang 3	30,9790	9,1547	3,917720
4		Siang 4	8,6014	14,7197	4,123559
5		Siang 5	7,0168	15,6040	3,254818
6		Malam 1	0,9517	24,2804	3,013215
7		Malam 2	7,6298	15,2403	3,910288
8		Malam 3	4,2319	17,8000	3,933778
9		Malam 4	9,9018	14,1083	3,819817
10		Malam 5	4,9167	17,1487	3,550787

Tabel 2 menjelaskan hasil dari deteksi wajah menggunakan metode *median filtering* pada dua kondisi yaitu siang dan malam. Dari semua sampel dengan dua kondisi tersebut berhasil dideteksi, namun nilai MSE dan PNSR yang didapat berbeda-beda dikarenakan perbedaan pencahayaan, kondisi objek saat difoto dan objek-objek pendukung yang ada didalam foto tersebut. Jika dirata-rata hasil yang didapat berdasarkan kondisi siang hari adalah 16,38304 untuk MSE dan 12,53906 untuk PNSR, sedangkan pada kondisi malam didapatkan 5,52638 untuk MSE dan 17,71554 untuk PNSR. Berdasarkan hasil yang didapat metode *median filtering* bekerja dengan baik pada kondisi waktu malam hari karena nilai rata-rata MSE yang didapat lebih rendah daripada nilai PNSR.

Kesimpulan

Hasil dari penelitian menunjukkan bahwa metode *median filtering* 100% bekerja, hal ini dibuktikan dengan 10 sampel foto yang berhasil dideteksi dengan baik. Akan tetapi pada beberapa sampel foto ada objek lain yang bukan wajah juga ikut terdeteksi, hal itu dapat terjadi karena kondisi perbedaan pencahayaan, kondisi objek saat difoto dan objek-objek pendukung yang ada didalam foto tersebut.

Sedangkan hasil pengujian metode *median filtering* menggunakan MSE dan PNSR Jika dirata-rata hasil yang didapat berdasarkan kondisi siang hari adalah 16,38304 untuk MSE dan 12,53906 untuk PNSR, sedangkan pada kondisi malam didapatkan 5,52638 untuk MSE dan 17,71554 untuk PNSR. Sedangkan dari semua

sampel foto didapatkan nilai *error* terbaik yaitu MSE 0,9517 dan PSNR 24,2804. Ketika posisi PSNR meningkat atau bernilai tinggi dan posisi MSE menurun maka citra yang didapat sangat halus untuk persepsi mata. Metode *median filtering* telah sukses diaplikasikan untuk mendeteksi wajah pada foto dan metode ini dapat digunakan dengan baik sebagai acuan pada deteksi wajah.

Daftar Pustaka

- [1] M. R. Wankhade and N. M. Wagdarikar, "Feature Extraction of Edge Detected Images," *Int. J. Comput. Sci. Mob. Comput.*, vol. 6, no. 6, pp. 336–345, 2017.
- [2] A. Eleyan and M. S. Anwar, "Multiresolution Edge Detection Using Particle Swarm Optimization," *Int. J. Eng. Sci. Appl.*, vol. 1, no. 1, pp. 11–17, 2017.
- [3] M. V. Alyushin, V. M. Alyushin, and L. V. Kolobashkina, "Optimization of the data representation integrated form in the viola-jones algorithm for a person's face search," *Procedia Comput. Sci.*, vol. 123, pp. 18–23, 2018, doi: 10.1016/j.procs.2018.01.004.
- [4] N. T. Deshpande and S. Ravishankar, "Face Detection and Recognition using Viola-Jones algorithm and fusion of LDA and ANN," *IOSR J. Comput. Eng.*, vol. 18, no. 6, pp. 1–6, 2016, [Online]. Available: <https://pdfs.semanticscholar.org/c5cf/c1f5a430ad9c103b381d016adb4cba20ce4e.pdf>.
- [5] Syafira, A. Rizkita, and G. Ariyanto, "Sistem Deteksi Wajah Dengan Modifikasi Metode Viola Jones," *Emit. J. Tek. Elektro*, vol. 17, no. 1, pp. 26–33, 2017, doi: 10.23917/emit.v17i1.5964.
- [6] Z. M. Ramadan, "Optimum image filters for various types of noise," *Telkomika (Telecommunication Comput. Electron. Control.*, vol. 16, no. 5, pp. 2458–2464, 2018, doi: 10.12928/TELKOMNIKA.v16i5.10508.
- [7] R. R. K. Al-taie, "A REVIEW PAPER : DIGITAL IMAGE FILTERING PROCESSING," vol. 3, no. 9, pp. 1–11, 2021.
- [8] A. Suneetha and E. Srinivasa Reddy, "Robust Gaussian noise detection and removal in color images using modified fuzzy set filter," *J. Intell. Syst.*, vol. 30, no. 1, pp. 240–257, 2020, doi: 10.1515/jisys-2019-0211.
- [9] A. Kaur, "Mingle Face Detection using Adaptive Thresholding and Hybrid Median Filter," *Int. J. Comput. Appl.*, vol. 70, no. 10, pp. 13–17, 2013, doi: 10.5120/11997-7883.
- [10] R. Kumar, C. Shao, and P. Kaur, "An improved adaptive weighted mean filtering approach for metallographic image processing," *J. Intell. Syst.*, vol. 30, no. 1, pp. 470–478, 2021, doi: 10.1515/jisys-2020-0080.
- [11] D. G. Chachlakis, T. Zhou, F. Ahmad, and P. P. Markopoulos, "Minimum Mean-Squared-Error autocorrelation processing in coprime arrays," *Digit. Signal Process. A Rev. J.*, vol. 114, p. 103034, 2021, doi: 10.1016/j.dsp.2021.103034.
- [12] P. Pinki and R. Mehra, "Estimation of the Image Quality under Different Distortions," *Int. J. Eng. Comput. Sci.*, vol. 5, no. 17291, pp. 17291–17296, 2016, doi: 10.18535/ijecs/v5i7.20.
- [13] D. Umamaheswari and E. Karthikeyan, "Comparative analysis of various filtering techniques in image processing," *Int. J. Sci. Technol. Res.*, vol. 8, no. 9, pp. 109–114, 2019.
- [14] M. A. Abdillah, A. Yudhana, and A. Fadlil, "Compression Analysis Using Coiflets, Haar Wavelet, and SVD Methods," *JUITA J. Inform.*, vol. 9, no. 1, p. 43, 2021, doi: 10.30595/juita.v9i1.8559.
- [15] I. Riadi, A. Fadlil, and T. Sari, "Image Forensic for detecting Splicing Image with Distance Function," *Int. J. Comput. Appl.*, vol. 169, no. 5, pp. 6–10, 2017, doi: 10.5120/ijca2017914729.
- [16] E. A. Pambudi, E. S. Wijaya, and A. Fauzan, "Improved Sauvola Threshold for Background Subtraction on Moving Object Detection," *Int. J. Softw. Eng. Comput. Syst.*, vol. 5, no. 2, pp. 78–89, 2019, doi: 10.15282/ijsecs.5.2.2019.6.0062.
- [17] N. Senthilkumaran and S. Vaithegi, "Image Segmentation By Using Thresholding Techniques For Medical Images," *Comput. Sci. Eng. An Int. J.*, vol. 6, no. 1, pp. 1–13, 2016, doi: 10.5121/cseij.2016.6101.
- [18] A. Kaur, U. Rani, and G. S. Josan, "Modified Sauvola binarization for degraded document images," *Eng. Appl. Artif. Intell.*, vol. 92, no. March, p. 103672, 2020, doi: 10.1016/j.engappai.2020.103672.
- [19] A. Yudhana, Sunardi, and S. Saifullah, "Segmentation comparing eggs watermarking image and original image," *Bull. Electr. Eng. Informatics*, vol. 6, no. 1, pp. 47–53, 2017, doi: 10.11591/eei.v6i1.595.
- [20] P. Irgens, C. Bader, T. Lé, D. Saxena, and C. Ababei, "An efficient and cost effective FPGA based implementation of the Viola-Jones face detection algorithm," *HardwareX*, vol. 1, pp. 68–75, 2017, doi: 10.1016/j.ohx.2017.03.002.

-
- [21] M. Zairi, T. Boujiha, and A. Ouelli, "Improved JPEG image watermarking in data compression domain using block selection strategy," *EAI Endorsed Trans. Internet Things*, vol. 6, no. 24, p. 168690, 2021, doi: 10.4108/eai.8-2-2021.168690.
- [22] S. Battiato, O. Giudice, F. Guarnera, and G. Puglisi, "Estimating Previous Quantization Factors on Multiple JPEG Compressed Images," *Eurasip J. Inf. Secur.*, vol. 2021, no. 1, 2021, doi: 10.1186/s13635-021-00120-7.
- [23] S. Saifullah, S. Sunardi, and A. Yudhana, "Analisis Perbandingan Pengolahan Citra Asli Dan Hasil Cropping Untuk Identifikasi Telur," *J. Tek. Inform. dan Sist. Inf.*, vol. 2, no. 3, pp. 341–350, 2016, doi: 10.28932/jutisi.v2i3.512.
- [24] S. Attaway, *Introduction to MATLAB*. 2019.
- [25] A. Vagga, S. Aherrao, H. Pol, and V. Borkar, "Flow visualization by Matlab® based image analysis of high-speed polymer melt extrusion film casting process for determining necking defect and quantifying surface velocity profiles," *Adv. Ind. Eng. Polym. Res.*, no. xxxx, 2021, doi: 10.1016/j.aiepr.2021.02.003.
- [26] M. Coblenz, "MATVines: A vine copula package for MATLAB," *SoftwareX*, vol. 14, p. 100700, 2021, doi: 10.1016/j.softx.2021.100700.