International Journal of Artificial Intelligence Research
Vol 6, No 2, December 2022
ISSN: 2579-7298

# Application of Mean Filtering Method for Optimizing Face Detection in Digital Photos

Sunardi[a,1], Anton Yudhana[a,2], Setiawan Ardi Wijaya[b,3,*]

[a] *Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia*
[b] *Master Program of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia*
[1]*sunardi@mti.uad.ac.id;* [2] *eyudhana@ee.uad.ac.id;* [3] *setiawanardiwijaya@gmail.com*
** corresponding author*

ARTICLE INFO

*Article history:*
Received 02 April 2022
Revised 21 May
Accepted 24 June 2022

*Keywords:*
Mean Filtering
Viola-Jones
MSE
PNSR
Confusion Matrix

ABSTRACT

Face detection in digital photos aims to get the face area in the digital photo. Usually, a lot of noise occurred when detecting faces in digital photos. This study applies the mean filtering method to improve digital photos by reducing noise. The accuracy of the mean filtering method is calculated using a confusion matrix, while the ability of this method is measured using the parameters of Mean Square Error (MSE) and Peak Noise to Signal Ratio (PNSR). Viola-Jones method was used to detect faces in this research. This method was chosen because it is one of the face detection procedures with high accuracy and good computational ability. Testing the mean filtering method obtained the lowest MSE of 9.33, while the highest PNSR of 14.37. The accuracy obtained by the mean filtering method using confusion is 90%. 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 digital photos.



## I. Introduction

The development of today's technology is very fast, one of which is in digital image software [1][2]. The human face recognition system is one of them, before the system can recognize a human face, the system must detect the face [3][4]. However, there problems often occur when detecting photos, namely, there is a lot of noise in the photo. This study solves this problem by applying the mean filtering method and testing whether the method is feasible or not to be used in the case of face detection in digital photos. The mean filtering method was chosen because this method is good at correcting noise in the image and produces a more focused image due to the replacement of pixel values using the average value of all existing values [5][6][7][8]. Noise can be caused by physical interference with the acquisition device or intentionally due to improper processing. An example is an unwanted random appearance of black or white spots in an image. Previous face detection studies used the median filtering method and got pretty good results [9][10], therefore this study tried to use another method, namely, mean filtering. The accuracy of this method is measured using a confusion matrix. This method uses several criteria to measure it, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) making it easier during the calculation process [11][12][13]. The ability of this method is measured using Mean Square Error (MSE) and Peak Noise to Signal Ratio (PNSR). The smaller the MSE value obtained, the better the results obtained, while the higher the PNSR value obtained, the better the results obtained [14][15]. In addition, MSE and PNSR were chosen because these parameters are easy to obtain and are widely used in research [16][17][18][19][20][21].

The first face detection steps are image input to input the Red Green Blue (RGB) image into the processing system, then the RGB image is converted to a grayscale image [22]. After that, mean filtering is applied to the grayscale image with the aim of reducing noise and refining the image so that it makes processing easier. The method used when detecting faces is Viola-Jones. This method was chosen because it is a face detection procedure with high accuracy and good computational ability

[3][23]. The Viola-Jones procedure uses the Haar feature as a descriptor, then combines Integral Image and AdaBoost to obtain and select feature values and form a Cascade Classifier. This classifier will be used to find faces in digital photos [24]. The image that has been converted to a matrix value will go through a process to get the location of the area, after which the face is labeled with a red line.

The image used in this study uses the Joint Photographic Experts Group (JPEG) format, which is a lossy compression algorithm, in other words, it will reduce the quality of the image [25][26][27]. An image is a reflection of an object that can be taken using a camera that produces photos that are exactly the same as the condition of the object when it was taken [28]. The image was taken using the Samsung A11 smartphone camera with a resolution of 13 megapixels (MP). The image used as a sample in this study is a clear landscape image and there are only one to three people's faces in it.

This research uses an Asus laptop with 8 GB of core i3 memory using the Windows 10 Pro operating system and the application used is MATLAB a13. MATLAB is used because this application is supported by mathematical software, graphics, and programming capabilities. This app has built-in functions to perform multiple operations [29][30][31]. This paper will be divided into four-part. Part 1, the background of this research and some related previous studies. Part 2, contains flow diagrams and explanations of the mean filtering method applied to face detection in digital photos. Part 3, experimental results from using the mean filtering method. Part 4, draws conclusions about the use of the mean filtering method for face detection in digital photos and whether it is good enough and feasible based on error calculations.

## II. Methods

### *A. Preprocessing*

Preprocessing is a process used to increase the level of precision and accuracy in the classification made.

### *B. Mean Filtering*

Mean filtering is one of the filtering methods that is widely used because this method is good at correcting noise in the image and produces a more focused image due to the replacement of pixel values using the average value of all existing values [5]. The following is the formula for calculating the filtering mean.

$$g(y,\ x) = \ \frac{1}{9}\sum_{p=-1}^{1}\sum_{q=-1}^{1} f(y+p,\ x+q) \quad (1)$$

Table 1. Example of eight neighboring pixels

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

Table 1 is an example of 3x3 pixels, and how to calculate them using formula (2) as follows:

$$g(y,\ x) = \ \frac{1}{9}\ (P1, P2, P3, P4, P5, P6, P7, P8) \quad (2)$$

So the initial value of f(y, x) is changed to the value that gets g(y, x)

### *C. Viola-Jones*

To detect facial parts in digital photos, the viola-jones method is used. In this method there are four main parts, namely:

- Haar feature. This feature was named by Alfred Haar, a Hungarian mathematician in the 19$^{th}$ century. Haar features are determined by showing a square with dark and light sides, sometimes one side lighter than the other, such as the edge of the eyebrows [32][33]. The center may be shinier than the surrounding squares, which can be interpreted as a nose. The dark side and the bright side of the Haar feature are shown in Figure 1.

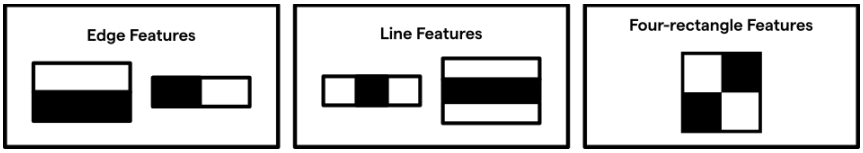


Fig. 1. Haar Feature

Figure 1 shows the dark and light sides to help the engine understand what the image is. The dark side is worth -1 and the light side is +1 each feature will result in one value. In addition, features that are also important for face detection are horizontal and vertical features that describe the shape of the eyebrows and nose on the machine, respectively.

- Integral Image. In this section, the feature values are calculated intensively because the number of pixels will be larger when in large features. An integral image is used to perform intensive calculations quickly so that it can be understood that the features of a number of features match the criteria [32][33]. An example of calculating the integral image is shown in Figure 2.

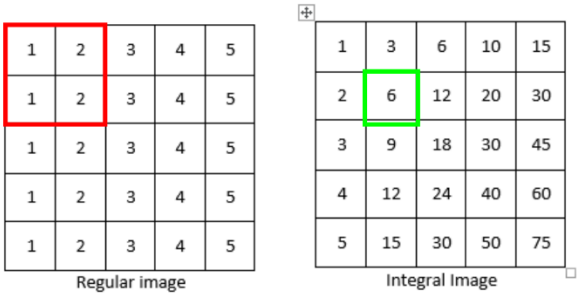


Fig. 2. Example of Integral Image

In Figure 2, there are two images, namely regular and integral images. The value in the integral image shown by the green line is obtained by adding up the regular image values according to the red line.

- AdaBoost. This algorithm is actually able to determine FP and TN in the data from a given image, thus making it more accurate [32][33].

Fig. 3. Example of Feature

$$F(x) = a_1 f_1(x) + a_2 f_2(x) + \cdots + a_n f_n(x) \quad (3)$$

Figure 3 is an example of the features that determine the success rate, with f1, f2, and f3 as features and a1, a2, and a3 as the weights of each feature. Each feature is known as a weak classifier while the equation f(x) is called a strong classifier. To get a strong classifier, one must have a combination of two or three weak classifiers. As it continues to be added it will become stronger and stronger. This is called assemblies, this is how adaptive boosting comes into play [32][33]. To train the machine to identify these features given information, and then train it to learn from the information to predict. The algorithm sets a minimum threshold to determine whether something can be clarified as a feature or not [32][33]. This algorithm shrinks the image to 24x24 and looks for trained features in the image. It takes a lot of facial image data to be able to see features in different and varied forms, therefore Viola and Jones gave 4960 facial image data and 9544 non-face image data. This is done to train the machine [32][33].

- Cascaded classifiers. This is one of the shortcuts to increase the speed and accuracy of the model. Starting with taking the sub-window then taking the most important or best features and making sure these features are in the image in the sub-window. If there is no sub-window

then it will not be processed. If there is then the number of features it has will continue and reject the sub-window without using the feature. The process will take a lot of time because you have to do it on every feature, therefore to speed up the machine in processing, cascading is used [32][33].

### *D. Evaluation of Results*

To measure the accuracy of this filtering method using a confusion matrix, this method uses several criteria to measure it, namely TP, TN, FP, and FN making it easier during the calculation process. [11][12][13]. Here is the formula for calculating accuracy:

$$accuracy = \frac{TP+TN}{TP+TN+FP} = \% \quad (4)$$

While the ability of this method is measured using Mean Square Error (MSE) and Peak Noise to Signal Ratio (PNSR). The smaller the MSE value obtained, the better the results obtained, while the higher the PNSR value obtained, the better the results obtained [14][15]. Here is the formula for MSE:

$$MSE = \frac{\sum_{a=0}^{x-1} \sum_{b=0}^{y-1} (M(a,b)-N(a,b))}{x*y} \quad (5)$$

M(a, b) is the value of the Mean Filtering image and N(a, b) is the reference image. PSNR is the ratio of the highest possible power signal to the highest possible power noise [15]. The formula for PNSR is as follows.

$$PNSR = 10\log_{10} \frac{Max_I^2}{MSE} \quad (6)$$

Max is the highest pixel value in an image, usually 255.

### *E. Proposed Method*

The method proposed in this study is the use of mean filtering for face detection in digital photos. The flowchart for this research method can be seen in Figure 4.

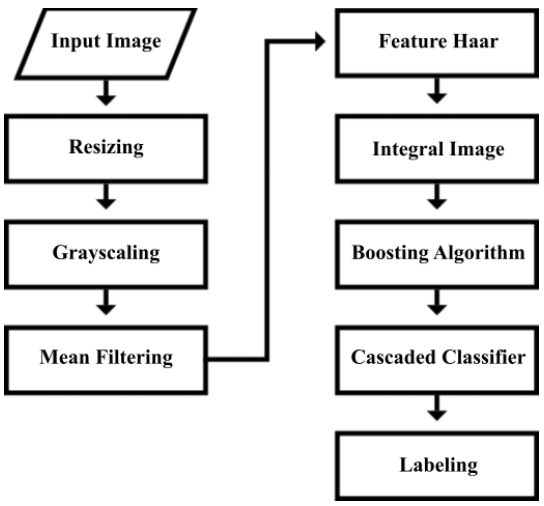


Fig. 4. Research flow

The steps used in this research method:

- Input photos into the application used to process photos, namely MATLAB.
- The resizing stage is the initial photo whose size is changed with the aim of speeding up processing.
- Convert photos that were originally in RGB form to grayscale using the gray scaling process. Convert RGB image to grayscale using formula (7).

  $$Iy = 0.333Fr + 0.5Fg + 0.1666Fb \quad (7)$$

  Fr is the intensity for Red, Fg is the intensity for Green, and Fb is the intensity for Blue, while Iy is the gray intensity which is equivalent to an RGB level image.
- Improve the image using the mean filtering method so that the noise in the image can be reduced. The way to find the Mean Filtering is by sorting the pixel values based on eight neighbors, after that the number of pixel values is added up and divided by the number of pixel values [5][7].
- Classify photos according to simple feature values. There are several reasons to use features instead of pixels directly. A very plausible reason is if the feature can be used to encode ad-hoc domain knowledge which is not easy to recognize against the limited amount of training data [32][33].
- In order to simplify the process of calculating the original value of each Haar feature at each image location, the integral image technique is used. Overall integral has the meaning of adding weight, namely the pixel values that will be added to the original image [32][33].
- The Boosting solving procedure combines many images that are less sharp (weak classifiers) to become sharper images (strong classifiers) by giving weight to the images of weak classifiers [32][33].
- Give weight to the weak classifier and combine many weak classifiers to produce a stronger classifier. Weak here means that the filter sequence in the classifier only accepts fewer correct answers. Filters at each level clarify the image that was previously filtered. If one of these filters is not successful, the region in the image is classified as non-face [32][33].
- The image that has been converted into a matrix value will go through a process to get the location of the area, after which the face is labeled with a red line.

## III. Results and Discussion

This study used 30 photos consisting of 10 photos with one face, 10 photos with two faces, and 10 photos with three faces in them. These photos were taken using the Samsung A11 smartphone camera with a resolution of 13 MP on the rear camera and 8 MP on the front camera. At this stage, face detection testing is carried out using the Mean Filtering method. Tests were carried out using the MATLAB application with hardware support for Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz 8 GB memory. The test results using the MATLAB application are presented in Table 2.

Table 2. MSE and PNSR results

| No | Sample | Description | MSE | PNSR | Speed (second) |
|---|---|---|---|---|---|
| 1 |  | TP | 14.92 | 12.33 | 3.61 |
| 2 |  | TP | 19.59 | 11.15 | 4.90 |
| 3 |  | TN | 11.79 | 13.35 | 4.88 |

| | | | | | |
|---|---|---|---|---|---|
| 4 |  | TN | 20.00 | 11.05 | 4.66 |
| 5 |  | TN | 13.13 | 12.88 | 4.36 |
| 6 |  | TP | 33.05 | 8.87 | 4.78 |
| 7 |  | TN | 35.35 | 8.58 | 4.90 |
| 8 |  | TN | 51.58 | 6.94 | 4.88 |
| 9 |  | TP | 16.13 | 11.99 | 5.13 |
| 10 |  | TN | 19.56 | 11.15 | 4.04 |
| 11 |  | TN | 90.53 | 4.50 | 3.62 |
| 12 |  | TP | 72.96 | 5.43 | 4.54 |
| 13 |  | TN | 9.33 | 14.37 | 3.72 |
| 14 |  | TN | 23.39 | 10.38 | 3.50 |
| 15 |  | TN | 21.31 | 10.78 | 3.85 |
| 16 |  | TP | 31.73 | 9.05 | 3.64 |
| 17 |  | TN | 27.41 | 9.69 | 3.89 |
| 18 |  | TP | 30.98 | 9.16 | 3.53 |
| 19 |  | TP | 66.66 | 5.83 | 4.78 |
| 20 |  | TP | 10.55 | 13.83 | 3.71 |
| 21 |  | TN | 124.19 | 3.12 | 0.66 |
| 22 |  | FP | 26.01 | 9.91 | 3.39 |
| 23 |  | TP | 11.63 | 13.41 | 3.55 |
| 24 |  | TP | 102.95 | 3.94 | 3.46 |
| 25 |  | TP | 48.11 | 7.24 | 1.74 |

| | | | | | |
|---|---|---|---|---|---|
| 26 |  | TP | 19.59 | 11.15 | 2.91 |
| 27 |  | FP | 12.47 | 13.11 | 6.66 |
| 28 |  | TN | 74.18 | 5.36 | 5.71 |
| 29 |  | FP | 44.80 | 7.55 | 1.70 |
| 30 |  | TP | 15.11 | 12.27 | 3.12 |

Based on Table 2, the results from Mean Filtering are obtained, from 30 sample images obtained results with three criteria. First, 14 samples with TP, namely the face in the image were detected. Second, 13 samples with TN, namely the face on the image. The image has been detected, but other objects have also been detected. Third, 3 samples with FP, namely the face in the image failed to be detected. This can occur due to lighting conditions, the condition of the object when photographed, and the supporting objects in the photo. To calculate the accuracy of the Mean Filtering method using a confusion matrix with the following equation.

$$accuracy = \frac{TP+TN}{TP+TN+FP} = \frac{14+13}{14+13+3} = \frac{27}{30} = 0.9 = 90\% \quad (8)$$

Based on the calculations obtained a high accuracy of 90%, while the lowest MSE value was obtained in the 13th sample with a value of 9.33. The highest PNSR value was in the 13th sample with a value of 14.37. The fastest time was obtained in the 21st sample with a time of 0.66 seconds. Previous face detection studies used the median filtering method and got pretty good results [9][10]. Meanwhile, based on the results obtained in this study, the Mean Filtering method works well to reduce noise caused by physical disturbances in the acquisition tool or intentionally due to improper processing. An example is an unwanted random appearance of black or white spots in an image.

## IV. Conclusion

The results of the study show that the Mean Filtering method can work well, this is evidenced by the accuracy obtained, which is 90% based on the confusion matrix. However, in some sample photos, there are other objects that are not faces that are also detected, this can happen due to lighting conditions, the condition of the object when being photographed and the scroll-down objects in the photo.

From the results of testing the Mean Filtering method using MSE and PNSR on 30 sample images, the lowest MSE results in the 13th sample with a value of 9.33, while the highest PNSR value in the 13th sample with a value of 14.37. The accuracy of the Mean Filtering method is measured using a confusion matrix with an accuracy of 90%. Based on these results, it can be concluded that the Mean Filtering Method has succeeded in reducing noise caused by physical disturbances in the acquisition tool or due to inappropriate photo-taking processes and has been successfully applied to detect faces in digital photos.

## References

[1] 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.

[2] 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.

[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] A. Wedianto, H. L. Sari, and Y. S. H, "Analisa Perbandingan Metode Filter Gaussian, Mean Dan Median Terhadap Reduksi Noise," *J. Media Infotama*, vol. 12, no. 1, pp. 21–30, 2016, doi: 10.37676/jmi.v12i1.269.

[6] 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.

[7] J. Choudhary and A. Choudhary, "Enhancement in Morphological Mean Filter for Image Denoising Using GLCM Algorithm," *Int. J. Comput. Theory Eng.*, vol. 13, no. 4, pp. 134–137, 2021, doi: 10.7763/ijcte.2021.v13.1302.

[8] S. A. Shaban and D. L. Elsheweikh, "Blood group classification system based on image processing techniques," *Intell. Autom. Soft Comput.*, vol. 31, no. 2, pp. 817–834, 2022, doi: 10.32604/iasc.2022.019500.

[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] S. Sunardi, A. Yudhana, and S. A. Wijaya, "Penerapan Metode Median Filtering untuk Optimasi Deteksi Wajah pada Foto Digital," *J. Innov. Inf. Technol. Appl.*, no. x, pp. 1–11, 2019.

[11] S. Maharani, H. Ridwanto, H. Rahmania Hatta, D. Marisa Khairina, and M. Rivani Ibrahim, "Comparison of TOPSIS and MAUT methods for recipient determination home surgery," *IAES Int. J. Artif. Intell.*, vol. 10, no. 4, p. 930, 2021, doi: 10.11591/ijai.v10.i4.pp930-937.

[12] K. Jankowska and P. Ewert, "Effectiveness Analysis of Rolling Bearing Fault Detectors Based On Self-Organising Kohonen Neural Network – A Case Study of PMSM Drive," *Power Electron. Drives*, vol. 6, no. 1, pp. 100–112, 2021, doi: 10.2478/pead-2021-0008.

[13] V. Balaji and S. K. S. Raja, "Recommendation learning system model for children with autism," *Intell. Autom. Soft Comput.*, vol. 31, no. 2, pp. 1301–1315, 2022, doi: 10.32604/iasc.2022.020287.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] I. Riadi, A. Yudhana, and W. Y. Sulistyo, "Analisis Perbandingan Nilai Kualitas Citra pada Metode Deteksi Tepi," *Rekayasa Sist. dan Teknol. Inf.*, vol. 4, no. 2, pp. 345–351, 2020.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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/emitor.v17i1.5964.

[25] 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.

[26] 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.

[27] 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.

[28] S. Saifullah, S. Sunardi, and A. Yudhana, "Analisis Perbandingan Pengolahan Citra Asli Dan Hasil Croping Untuk Identifikasi Telur," *J. Tek. Inform. dan Sist. Inf.*, vol. 2, no. 3, pp. 341–350, 2016, doi: 10.28932/jutisi.v2i3.512.

[29] S. Attaway, *Introduction to MATLAB*. 2019. doi: 10.1016/b978-0-12-815479-3.00001-5.

[30] 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.

[31] M. Coblenz, "MATVines: A vine copula package for MATLAB," *SoftwareX*, vol. 14, p. 100700, 2021, doi: 10.1016/j.softx.2021.100700.

[32] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," no. February, 2001, doi: 10.1109/CVPR.2001.990517.

[33] P. Viola and M. Jones, "Robust Real-Time Face Detection Robust Real-time Face Detection," no. June, pp. 2–3, 2014, doi: 10.1023/B.