Face Recognition Using Fisherface and Support Vector Machine Method

Syachrul Qolbi Nur Septi[ab1], Intan Nurma Yulita[ac2], Herlina Napitupulu[ab2]
[a]*Department of Mathematics, Universitas Padjadjaran*
Sumedang, Indonesia
[1]syachrul17001@mail.unpad.ac.id
[2]herlina@unpad.ac.id
[b]*Research Center for Artificial Intelligence and Big Data, Universitas Padjadjaran*
Sumedang, Indonesia
[c]*Department of Computer Science, Universitas Padjadjaran*
Sumedang, Indonesia
[3]intan.nurma@unpad.ac.id

2021 International Conference on Artificial Intelligence and Big Data Analytics (ICAIBDA) | 978-1-6654-0890-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICAIBDA53487.2021.9689738

***Abstract*— COVID-19 is declared as a pandemic by WHO and until now COVID-19 pandemic remains a problem in 2021. Many efforts have been made to reduce the spreading virus, one way to reduce its spread is by wearing a mask but most people often ignore it. Monitoring large groups of people becomes difficult by the government or the authorities. Face recognition, a biometric technology, is based on the identification of a face features of a person. This paper describes a face recognition using Fisherface and Support Vector Machine method to classify face mask dataset. Face recognition using Fisherface method is based on Principal Component Analysis (PCA) and Fisher's Linear Discriminant (FLD) method or also known as Linear Discriminant Analysis (LDA). The algorithm used in the process for feature extraction is Fisherface algorithm while classification using Support Vector Machine method. The results show that for face recognition on face mask dataset using cross validation with 10 fold, the average percentage accuracy is 99.76%.**

Keywords— PCA, LDA, Fisherface, Support Vector Machine, Face Recognition

I. INTRODUCTION

COVID-19 is declared as a pandemic by WHO and until now COVID-19 pandemic remains a problem in 2021 [1]. Many efforts have been made to reduce its spread virus, one way to reduce its spread is by wearing a mask [2]. People are forced by law to wearing masks in public places and wherever they interact with others since the new normal but many of them ignore it [3], so it is more difficult to monitor large groups of people by the government or the authorities [4].

Face is the easiest part of the human body and is often used to distinguish individual identities. From face, humans can be recognized and recognized easily and quickly [5]. Therefore, the face is used as a person's tool or commonly referred to as Facial Recognition (Face Recognition) [6]. Face Recognition is a computer application that is capable of detecting, tracking, identifying, or verifying human faces from an image or video captured using a digital camera. Currently, face recognition is an attraction for computer vision observers and researchers to continue developing this system.

In general, image recognition is divided into two types, namely: feature-based and image-based. In a feature-based, the extracted feature components such as eyes, mouth, nose, ears, etc. are modeled to determine the relationship between the feature components. While the image-based uses image pixels which represented by certain methods such as Principal Component Analysis, Wavelet Transformation, etc. which is used to train and classify images [7].

Feature extraction is a process to obtain characteristics that distinguish image samples from other image samples. The feature extraction technique is the key in solving pattern recognition problems. An example of a feature extraction method is the Principal Component Analysis (PCA) which is used for face recognition and introduced by Turk and Pentland (1991). Although Principal Component Analysis (PCA) is a fairly well-known technique in image recognition, in reality, Principal Component Analysis (PCA) has problems in handling very large data so that processing time from recognition becomes long and accuracy decreases rapidly as the amount of data increases [8].

In 1997, Belheumeur introduced the Fisherface method for face recognition. This method is a combination of Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) methods. Principal Component Analysis (PCA) is used to solve singular problems by reducing the dimensions of the data before it is used to carry out the Linear Discriminant Analysis (LDA) process [6]. Belhumeur et al's research using the Fisherface method showed better results than the PCA or LDA method itself because the greater the ratio, the more feature vectors produced the less sensitive to changes in expression and changes in light so that it can produce a good classification.

One of the statistical methods that can be applied to perform classification is the Support Vector Machine (SVM). SVM is a technique for finding a hyperplane that can separate two data sets of two different classes. SVM has advantages including in determining the distance using a support vector so that the computing process becomes fast [9].

This paper will describe the research on face recognition using face mask dataset. Finally, we introduce the general evaluation criteria of face recognition.

II. MATERIALS AND METHOD

A. System Design

Face recognition using the Fisherface and SVM method is designed to recognize face images by classifying the feature extraction results with SVM method. This process is expected to determine whether the image to be tested is classified correctly or not. In this research, 5000 face images with and without a mask are used in *.png format which is generated using style GAN-2 [10].

B. Process Design

1) Data Preprocessing

The face image must be preprocessed first. This stage converts RBG image to grayscale. Conversion of face image from RGB to grayscale with size 128×128 pixels. Furthermore, the image is divided into training image (training dataset) and test image (testing dataset).

2) Feature Extraction

Features extraction produce feature image of the people face with and without wearing a mask. Fisherface method is chosen which is a merger between PCA and LDA methods.

III. Results

In this section, we discussed about the algorithm and results of face recognition using Fisherface and SVM method. In general, the stage of the face recognition process in this study can be seen in Fig. 1.

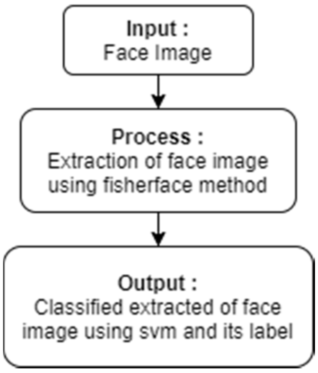

Fig. 1. Stage of process

A. Image Data

Here is a sample of images of people with and without wearing a mask which is generated using style GAN-2. The images can be seen in Fig. 2.

Fig. 2. An example of image people with and without a mask

Here is a sample of images after data preprocessing. The images can be seen in Fig. 3.

Fig. 3. An example of image people with and without mask after data preprocessing

B. Feature Extraction Method (Fisherface)

1) PCA Algorithm

- Take training data images and their labels, then each image is transformed into a row vector through data preprocessing, then each row vector is made a matrix so that a matrix will be obtained, each row represents a different image ($\boldsymbol{X}$) with the associated image label ($\boldsymbol{Y}$).
- Calculate the average matrix ($\boldsymbol{\Psi}$), which is the average value of all training data images, as in (1).

$$\psi_k = \frac{1}{M}\sum_{m=1}^{M} x_{mk}; \qquad (1)$$
$$m = 1,2,3,\ldots,M$$

Here is an example of average image people with and without mask. The images can be seen in Fig. 4.

Fig. 4. An example of average image people with and without mask

- Calculate the difference matrix ($\boldsymbol{\Phi}$), which is the difference between all training data images and their average value, as in (2).

$$\phi_{mk} = x_{mk} - \psi_k \qquad (2)$$

- Calculating the total scatter matrix ($\boldsymbol{S_T}$), by multiplying the difference matrix by the transpose, as in (3).

$$s_{t_{kk}} = \sum_{m=1}^{M} \phi_{km}\phi_{mk} \qquad (3)$$

- Calculate the eigenvalue (λ_{PCA}) and eigenvector ($\boldsymbol{v_{PCA}}$) of the total scatter matrix, as in (4).

$$\boldsymbol{S_T v_{PCA}} = \lambda_{PCA}\boldsymbol{v_{PCA}} \qquad (4)$$

- Sort the eigenvectors based on the eigenvalues of each eigenvector.
- Take the required $N_{v_{PCA}} - C$ eigenvectors where $N_{v_{PCA}}$ is the number of eigenvectors and C is the number of classes.
- Take the selected set of eigenvectors then each eigenvector is transformed into a row vector, then each row vector is made a matrix so that a matrix that each row represents each eigenvector ($\boldsymbol{W_{PCA}}$) will be obtained.
- Calculate the eigenface weight matrix ($\boldsymbol{\Omega_{PCA}}$), by multiplying the difference matrix by the $\boldsymbol{W_{PCA}}$ transpose matrix, as in (5).

$$\omega_{PCA_{ml}} = \sum_{k=1}^{K} \phi_{mk} w_{PCA_{kl}}; \qquad (5)$$
$$l = 1,2,3,\ldots,L$$

2) LDA Algorithm

- Take the eigenface weight matrix ($\boldsymbol{\Omega_{PCA}}$) and the label matrix ($\boldsymbol{Y}$) to be used as input.

- Calculates the overall mean matrix ($\boldsymbol{\Psi}^{\boldsymbol{\Omega}_{PCA}}$) and the average on the same label ($\boldsymbol{\Psi}_{\boldsymbol{c}}^{\boldsymbol{\Omega}_{PCA}}$) from the eigenface weight matrix $\boldsymbol{\Omega}_{\boldsymbol{PCA}}$, as in (6) and (7).

$$\psi_l^{\Omega_{PCA}} = \frac{1}{M}\sum_{m=1}^{M} \omega_{PCA_{ml}}; \quad l = 1,2,3,\dots,L \tag{6}$$

$$\psi_{cl}^{\Omega_{PCA}} = \frac{1}{M^{(c)}}\sum_{m^{(c)}=1}^{M^{(c)}} \omega_{PCA_{m^{(c)}l}}; \quad m^{(c)} = 1,2,3,\dots,M^{(c)} \tag{7}$$

- Calculating the distribution between classes scatter matrix ($\boldsymbol{S}_{\boldsymbol{B}}$) from the eigenface weight matrix $\boldsymbol{\Omega}_{\boldsymbol{PCA}}$, as in (8).

$$\boldsymbol{S}_{\boldsymbol{b}} = \sum_{c=1}^{M^{(c)}} M^{(c)} \left(\boldsymbol{\Psi}_{\boldsymbol{c}}^{\boldsymbol{\Omega}_{PCA}} - \boldsymbol{\Psi}^{\boldsymbol{\Omega}_{PCA}}\right)\left(\boldsymbol{\Psi}_{\boldsymbol{c}}^{\boldsymbol{\Omega}_{PCA}} - \boldsymbol{\Psi}^{\boldsymbol{\Omega}_{PCA}}\right)^T \tag{8}$$

- Calculating the distribution within classes scatter matrix ($\boldsymbol{S}_{\boldsymbol{W}}$) from the eigenface weight matrix $\boldsymbol{\Omega}_{\boldsymbol{PCA}}$, as in (9).

$$\boldsymbol{S}_{\boldsymbol{w}} = \sum_{c=1}^{C}\sum_{m=1}^{M^{(c)}} \left(\boldsymbol{\Omega}_{PCA_{m^{(c)}}} - \boldsymbol{\Psi}_{\boldsymbol{c}}^{\boldsymbol{\Omega}_{PCA}}\right)\left(\boldsymbol{\Omega}_{PCA_{m^{(c)}}} - \boldsymbol{\Psi}_{\boldsymbol{c}}^{\boldsymbol{\Omega}_{PCA}}\right)^T \tag{9}$$

- Calculate the eigenvalue (λ_{LDA}) and eigenvector ($\boldsymbol{v}_{\boldsymbol{LDA}}$) of the $\boldsymbol{S}_{\boldsymbol{B}}$ and $\boldsymbol{S}_{\boldsymbol{W}}$ matrix, as in (10).

$$(\boldsymbol{S}_{\boldsymbol{w}})^{-\mathbf{1}}\boldsymbol{S}_{\boldsymbol{b}}\boldsymbol{v}_{\boldsymbol{FLD}} = \lambda_{FLD}\boldsymbol{v}_{\boldsymbol{FLD}} \tag{10}$$

- Sort the eigenvectors based on the eigenvalues of each eigenvector.
- Take the required $C - 1$ eigenvectors where C is the number of classes.
- Take the selected set of eigenvectors then each eigenvector is transformed into a row vector, then each row vector is made a matrix so that a matrix that each row represents each eigenvector ($\boldsymbol{W}_{\boldsymbol{LDA}}$) will be obtained.
- Calculating the Fisherface matrix ($\boldsymbol{W}_{\boldsymbol{OPT}}$), by multiplying the matrix $\boldsymbol{W}_{\boldsymbol{LDA}}$ by $\boldsymbol{W}_{\boldsymbol{PCA}}$. matrix, as in (11).

$$w_{OPT_{l^*k}} = \sum_{l=1}^{L} w_{FLD_{l^*l}} w_{PCA_{lk}}; \quad k = 1,2,3,\dots,K \tag{11}$$

Here is an example of Fisherface of image people with and without a mask. The images can be seen in Fig. 5.

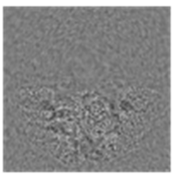

Fig. 5. An example Fisherface of image people with and without a mask

- Calculate the Fisherface weight matrix ($\boldsymbol{\Omega}_{\boldsymbol{LDA}}$), by multiplying the difference matrix by the $\boldsymbol{W}_{\boldsymbol{OPT}}$ transpose matrix, as in (12).

$$\omega_{FLD_{ml^*}} = \sum_{k=1}^{K} \phi_{mk} w_{OPT_{kl^*}}; \quad l^* = 1,2,3,\dots,L^* \tag{12}$$

C. Classification Algorithm

- Test data is taken from the results of data preprocessing and then stored into an $\boldsymbol{X}^{\boldsymbol{TEST}}$ matrix and image labels are stored in a $\boldsymbol{Y}^{\boldsymbol{TEST}}$ matrix where y_u is a label for $\boldsymbol{X}_{\boldsymbol{t}}^{\boldsymbol{TEST}}$ images.
- Calculate the difference matrix ($\boldsymbol{\Phi}^{\boldsymbol{TEST}}$), which is the difference between all test data images and their average value, as in (2).
- Calculate the Fisherface weight matrix ($\boldsymbol{\Omega}_{\boldsymbol{LDA}}^{\boldsymbol{TEST}}$), by multiplying the difference matrix by the $\boldsymbol{W}_{\boldsymbol{OPT}}$ transpose matrix, as in (12).
- *Classify each test data Fisherface weight ($\boldsymbol{\Omega}_{\boldsymbol{LDA}}^{\boldsymbol{TEST}}$) using the SVM model based on the training data Fisherface weight ($\boldsymbol{\Omega}_{\boldsymbol{LDA}}$).*

D. Analysis SVM Model

Data training and testing are divided into several main scenarios. Each scenario has a different purpose in measuring the performance of different scenario. Testing the classification model using the 10-fold cross validation technique, means that 5000 data is divided into 10 folds (parts). A combination of 9 different folds is combined and used as training data, the remaining 1-fold is used as testing data. In this section, the test results of each scenario are presented to see the performance support vector machine method in face recognition of people with and without wearing a mask. The performance of each scenario is measured based on calculation results accuracy, precision, and recall.

1) Kernel Function Performance Comparison

This stage aims to obtain the most optimal kernel function to be applied in the final model prediction in this study. Kernel functions to be used in this test scenario are Linear Kernel, Polynomial Kernel, Radial Base Function (RBF) Kernel and Sigmoid Kernel. Comparison of test results is seen based on accuracy, precision, and recall on every kernel function.

TABLE I. KERNEL FUNCTION COMPARISON ACCURACY

Fold	Kernel Function Accuracy (%)			
	Linear	Polynomial	RBF	Sigmoid
1	99.80%	99.80%	99.80%	99.80%
2	99.60%	99.80%	99.60%	99.60%
3	99.80%	99.80%	99.80%	99.80%
4	99.80%	100%	99.80%	99.80%
5	100%	100%	100%	100%
6	99.80%	99.80%	99.80%	99.80%
7	99.80%	99.80%	99.80%	99.80%
8	99.40%	99.40%	99.40%	99.40%
9	99.60%	99.60%	99.60%	99.60%
10	99.60%	99.60%	99.60%	99.60%

Table I shows that the comparison of performance prediction accuracy each scenario using SVM with various kernel functions has a sufficient accuracy value good. From the calculation of 10-fold cross-validation for this accuracy, the highest average percentage accuracy is obtained from the SVM prediction with the Polynomial Kernel function by 99.76%, followed by the use of Linear Kernel, RBF Kernel, and Sigmoid Kernel with 99.72% accuracy percentage.

Table II shows that the comparison of performance prediction precision for each scenario using SVM with various kernel functions has a sufficient precision value good. From the calculation of 10-fold cross-validation for this precision, the highest average percentage precision is obtained from the SVM prediction with the Polynomial Kernel function by 99.79%, followed by the use of Linear Kernel, RBF Kernel, and Sigmoid Kernel with 99.76% precision percentage.

TABLE II. KERNEL FUNCTION COMPARISON PRECISION

Fold	Kernel Function Precision (%)			
	Linear	**Polynomial**	**RBF**	**Sigmoid**
1	100%	100%	100%	100%
2	99.20%	99.60%	99.20%	99.20%
3	99.60%	99.60%	99.60%	99.60%
4	100%	100%	100%	100%
5	100%	100%	100%	100%
6	100%	100%	100%	100%
7	100%	100%	100%	100%
8	99.60%	99.60%	99.60%	99.60%
9	99.60%	99.60%	99.60%	99.60%
10	99.60%	99.60%	99.60%	99.60%

Table III shows that the comparison of performance prediction recalls each scenario using SVM with various kernel functions has a sufficient recall value good. From the calculation of 10-fold cross-validation for this recall, the highest average percentage recall is obtained from the SVM prediction with the Polynomial Kernel function by 99.72%, followed by the use of Linear Kernel, RBF Kernel, and Sigmoid Kernel with 99.68% recall percentage.

2) C Parameter Performance Comparison

This stage aims to obtain the most optimal regularization parameter (C) to be applied in the final model prediction in this study. Regularization parameters (C) to be used in this test scenario are $C = 1, C = 5, C = 10$. Comparison of test results is seen based on accuracy, precision, and recall on every regularization parameter (C).

TABLE III. KERNEL FUNCTION COMPARISON RECALL

Fold	Kernel Function Recall (%)			
	Linear	**Polynomial**	**RBF**	**Sigmoid**
1	99.60%	99.60%	99.60%	99.60%
2	100%	100%	100%	100%
3	100%	100%	100%	100%
4	99.60%	100%	99.60%	99.60%
5	100%	100%	100%	100%
6	99.60%	99.60%	99.60%	99.60%
7	99.60%	99.60%	99.60%	99.60%
8	99.20%	99.20%	99.20%	99.20%
9	99.60%	99.60%	99.60%	99.60%
10	99.60%	99.60%	99.60%	99.60%

TABLE IV. REGULARIZATION PARAMETER (C) COMPARISON ACCURACY

Fold	Regularization Parameter (C) Accuracy (%)		
	$C = 1$	$C = 5$	$C = 10$
1	99.80%	99.80%	99.80%
2	99.80%	99.80%	99.80%
3	99.80%	99.80%	99.80%
4	100%	100%	100%
5	100%	100%	100%
6	99.80%	99.80%	99.80%
7	99.80%	99.80%	99.80%
8	99.40%	99.40%	99.40%
9	99.60%	99.60%	99.60%
10	99.60%	99.60%	99.60%

TABLE V. REGULARIZATION PARAMETER (C) COMPARISON PRECISION

Fold	Regularization Parameter (C) Precision (%)		
	$C = 1$	$C = 5$	$C = 10$
1	100%	100%	100%
2	99.60%	99.60%	99.60%
3	99.60%	99.60%	99.60%
4	100%	100%	100%
5	100%	100%	100%
6	100%	100%	100%
7	100%	100%	100%
8	99.60%	99.60%	99.60%
9	99.60%	99.60%	99.60%
10	99.60%	99.60%	99.60%

Table IV shows that the comparison of performance prediction accuracy for each scenario using SVM with various regularization parameter (C) has a sufficient accuracy value good. From the calculation of 10-fold cross validation for this accuracy, the average percentage accuracy is obtained from the SVM prediction with the regularization parameter $C = 1, C = 5, C = 10$ by 99.76% accuracy percentage.

Table V shows that the comparison of performance prediction precision for each scenario using SVM with various regularization parameters (C) has a sufficient precision value good. From the calculation of 10-fold cross-validation for this precision, the highest average percentage precision is obtained from the SVM prediction with the regularization parameter $C = 1, C = 5, C = 10$ by 99.80% accuracy percentage.

TABLE VI. REGULARIZATION PARAMETER (C) COMPARISON RECALL

Fold	Regularization Parameter (C) Recall (%)		
	$C = 1$	$C = 5$	$C = 10$
1	99.60%	99.60%	99.60%
2	100%	100%	100%
3	100%	100%	100%
4	100%	100%	100%
5	100%	100%	100%
6	99.60%	99.60%	99.60%
7	99.60%	99.60%	99.60%
8	99.20%	99.20%	99.20%
9	99.60%	99.60%	99.60%
10	99.60%	99.60%	99.60%

Table VI shows that the comparison of performance prediction recall each scenario using SVM with various regularization parameter (C) has a sufficient recall value good. From the calculation of 10-fold cross validation for this recall, the highest average percentage recall is obtained from the SVM prediction with the regularization parameter $C = 1, C = 5, C = 10$ by 99.72% accuracy percentage.

E. System Result

To determine whether the system is running well and properly it is necessary to test the model with the following process.

1) Training Process

The first process is the training process. This process aims to generate the weight of each image of training images ($\boldsymbol{W_{OPT}}$).

2) Image Recognition Process

This process is to recognize the test image with SVM method. From the analysis SVM model, we choose Polynomial Kernel, and $C = 1$. The goal is how accurate the system is to recognize the test image.

3) Image Recognition Results

The following is the results of face which can be seen in Table VII.

TABLE VII. FACE RECOGNITION FINAL RESULTS

Fold	Accuracy (%)	Precision (%)	Recall (%)
1	99.80%	100%	99.60%
2	99.80%	99.60%	100%
3	99.80%	99.60%	100%
4	100%	100%	100%
5	100%	100%	100%
6	99.80%	100%	99.60%
7	99.80%	100%	99.60%
8	99.40%	99.60%	99.20%
9	99.60%	99.60%	99.60%
10	99.60%	99.60%	99.60%

IV. CONCLUSION

Face recognition using Fisherface and SVM methods using face mask dataset produces a high accuracy. This is indicated by the value of the average prediction accuracy of the test data obtained by 99.76%, average prediction precision by 99.79%, and average prediction recall by 99.72%. Face recognition using Fisherface and SVM methods is not only capable of performing an introduction to the test face images with different color components of the training image and a sketch of the original image. This method also work to noise induced images and the blurring effect on the images.

ACKNOWLEDGMENT

The Author thanks to the Research Center for Artificial Intelligence and Big Data Universitas Padjadjaran Indonesia that supported the study. This work also supported by Conference Grant funded by Directorate of Research & Community Service, Universitas Padjadjaran Contract No: 2574/UN6.3.1/TU.00/2021.

REFERENCES

[1] Y. H. Lin, C. H. Liu, and Y. C. Chiu, “Google searches for the keywords of ‘wash hands’ predict the speed of national spread of COVID-19 outbreak among 21 countries,” Brain. Behav. Immun., vol. 87, pp. 30–32, 2020.

[2] O. O. Fadare and E. D. Okoffo, “Covid-19 face masks: A potential source of microplastic fibers in the environment,” Sci. Total Environ., vol. 737, p. 140279, 2020.

[3] S. Setiati and M. K. Azwar, “COVID-19 and Indonesia,” Acta Med. Indones., vol. 52, no. 1, pp. 84–89, 2020.

[4] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, “A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic,” Meas. J. Int. Meas. Confed., vol. 167, p. 108288, 2021.

[5] F. Mahmud, M. T. Khatun, S. T. Zuhori, S. Afroge, M. Aktar, and B. Pal, “Face recognition using Principle Component Analysis and Linear Discriminant Analysis,” in 2nd International Conference on Electrical Engineering and Information and Communication Technology, iCEEiCT 2015, 2015, pp. 1–4.

[6] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection,” IEEE Transactions on pattern analysis and machine intelligence, vol. 19, no. 7, pp. 711–720, 1996.

[7] M. Anggo and La Arapu, “face recognition using Fisherface method,” in Journal of Physics: Conference Series, vol. 1028, no. 1, p. 12119, 2018

[8] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” in Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991, pp. 586–587.

[9] C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.

[10] P. Kottarathil, Face Mask Lite Dataset. Kaggle Data. Kaggle, 2020.