JOIV : Int. J. Inform. Visualization, 7(4) - December 2023 2336-2343

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Grouping of Image Patterns Using Inceptionv3 for Face Shape Classification

Tonny Hidayat [a,*], Ika Asti Astuti [a] , Ainul Yaqin [a], Alexander Phuk Tjilen [b], Teguh Arifianto [c]

[a] *Faculty of Computer Science, Universitas AMIKOM Yogyakarta, Depok, Yogyakarta, 55281, Indonesia*
[b] *Faculty of Social and Political Science, Universitas Musamus, Merauke, Merauke, 99611, Indonesia*
[c] *Railway Electrical Technology, Politeknik Perkeretaapian Indonesia, Manguharjo, Madiun, 63132, Indonesia*
*Corresponding author: *tonny@amikom.ac.id*

***Abstract*—The human face is an extraordinary part where nearly everybody is not quite the same as each other. One perspective that should be visible plainly is the shape. Face shape grouping can be used for amusement, security, or excellence. One technique that can be utilized in picture grouping is the InceptionV3 model. InceptionV3 is the structure of the Convolutional Neural Network (CNN) created by Google, which can tackle picture examination and item discovery issues. This engineering is utilized to order face shapes into five classes: Round, Heart, Square, Oblong, and Oval. At that point, the Google Pictures dataset goes through the pre-handling stage, and the Shrewd Edge Identifier is applied to each picture. Hair turns into a commotion. Consider recognizing the side of the face because it does not make any difference what the hairdo resembles. What is important is the side of the face. When there is a dataset of elongated class and heart class with a comparable hairdo, InceptionV3 will identify the component and expect the two pieces of information to come from a similar class. The exchange learning strategy is done in preparation for the last Layer of ImageNet's InceptionV3 model. This strategy puts the high precision level with an exactness of 93% preparation and testing between 88% - 98%. InceptionV3 could arrange upwards of 692 from 747 datasets or around 92.65%. The most reduced information class is the heart class, where out of 150 information, InceptionV3 can characterize upwards of 130 information.**

***Keywords*— Face shape classification; InceptionV3; machine learning.**

Manuscript received 11 Apr. 2023; revised 22 Aug. 2023; accepted 9 Oct. 2023. Date of publication 31 Dec. 2023.

I. Introduction

There is a long way to go when regulate a business: for instance, the development of clients in a store and the items they view, places of creation, log jam, or stock that should be added [1]. In any case, humanly, we cannot watch out for a business continually, and this is where computer vision becomes possibly the most important factor [2], [3]. Computer imaging joins cameras, edge figuring, cloud-based processing, programming, and artificial intelligence (AI) to help frameworks "see" and distinguish objects [4]. Computer vision (CV) is an area of artificial intelligence (AI) that focuses on making it possible for computers to interpret and grasp visual data from the outside world [5], [6], much like how people see[7], perceive, and understand the visual world [7], [8]. It encompasses a range of techniques and methods for processing, analyzing, and making sense of visual data, such as images and videos [9], [10]. The primary goal of computer vision is to teach machines to extract meaningful insights and information from visual input, just as humans do [11], [12]. By being balanced with software developments in various sectors, Computer Vision technology can be useful in various ways [13], [14]. For example, it is applied to the security section for surveillance cameras [15], the entertainment section for camera filters [16], and others. To support the development of the software, it is necessary to research human face recognition. That is the reason this research takes the topic of face shape [17].

The human face is a unique part of the body, where each individual has different characteristics, although a small number of people have faces similar to others [18], [19]. One of the differences that can be grouped is the shape of the face. The grouping of face shapes can be categorized into five classes: round, square, heart, oblong, and oval [20].

There are several methods or ways of grouping human faces using the Computer Vision technique. One technique that can be used is the Convolutional Neural Network (CNN) [21], which is excellent at various image-processing tasks, including segmentation, object detection, and image categorization. They do not require manually engineered feature engineering because they automatically learn pertinent features from raw pixel data [22], [23]. CNN is extremely useful for various computer vision applications because of its ability to capture both low-level characteristics (like edges) and high-level semantics (like object IDs) in images [24], [25]. CNN itself consists of various nodes (neurons) that overlap to form one Layer (layers) and are connected to determine the information value of an input image.

InceptionV3 is one of the pre-trained CNN models Google AI developed, including 48 layers [26]. This network can classify images into 1000 objects, such as keyboard, mouse, pencil, and animal [27]. This aims research to (1) deploy InceptionV3 with a retrained model for facial shape classification and (2) measure the accuracy of the classification while using InceptionV3.

II. Materials and Method

Research conducted by [20] designed a novel framework for hairstyle recommendations based on face shape classification. The framework is based on the Support Vector Machine (SVM), while the face shape classification uses deep-learned (VGG-Face) features and a VGG-face fine-tuned version. To combine the three methods, this study uses two combination techniques: Vector Concatenation and Multiple Kernel Learning. The results show that the proposed method can achieve up to 70.33% accuracy. Research on the classification of face shapes was also carried out by [28] in a study on Face Shape Classifiers. In this study, the Dense Facial Landmarks API by Face++ was used to generate facial landmarks so that they could be calculated. The research divides face shape classes into 6, namely: Round, Square, Oblong, Diamond, Triangle, and Oval.

Another study [29] proposes a face-shape classification method based on Region Similarity, Correlation Coefficient, and Fractal Dimensions. The study was conducted using three different facial datasets. The results showed that, on average, there were 80% faces in the three datasets with an elliptical shape. Face Shape Classification from 3D human data using SVM was conducted by [30] using three stages. The stages are head segmentation, face plane identification, and face shape classification. With input in the form of 3D data, SVM is used to classify facial shapes into four classes: ellipse shape, long shape, round shape, and square shape. The study used 90 subjects as experimental material, which resulted in an accuracy of up to 73.68%. Thus, this current study used the Convolutional Neural Networks (CNNs) algorithm found in InceptionV3 to classify facial shapes into five classes: oblong, round, heart, square, and oval. In addition, this research also uses Canny Edge Detector for image pre-processing to be used.

A. Computer Vision

Computer Vision has been studied from various perspectives. This science has evolved from raw data recording to techniques and ideas that combine digital image processing, pattern recognition, machine learning, and computer graphics. Computer Vision helps experts analyze images and videos to obtain the necessary information and understand information about events or descriptions using multi-range application domain methods with big data [31], [32]. The main purpose of Computer Vision is to create models, extract data and information from images [33].

B. Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning method that is widely used to solve complex problems regarding image recognition and existing information. Neural Networks began to gain attention in the field of computer vision in 2012 when Alex Krizhevsky used them to win the ImageNet competition (annual Olympics on image classification) that year (Figure 1), recorded impressive statistics by successfully reducing the misclassification rate from 26% down to 15% [34].

Fig. 1 ImageNet 2012 Olympic stats

Before understanding Convolutional Neural Networks, it's good to discuss their predecessors, namely Artificial Neural Networks (ANNs) [35]. The basic structure of a neural network can be shown in Figure 2.

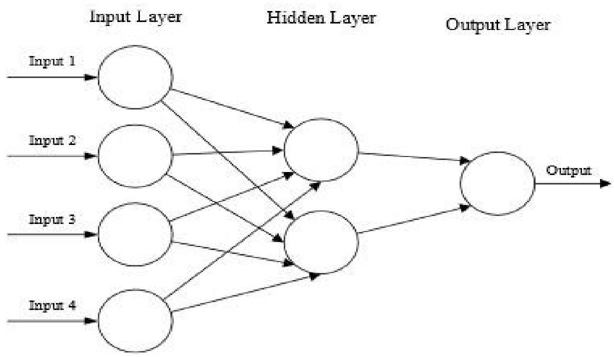

Fig. 2 The basic structure of Neural Networks

In the picture above, there are three layers of Feedforward Neural Networks (FNN), consisting of an Input layer, which is usually in the form of a multidimensional vector that is distributed to the hidden Layer [36]. Then, the hidden Layer will decide from the previous Layer and consider how the stochastic changes in itself harm or improve the final result, here, the learning process occurs. Several hidden layers stacked on each other are called Deep Learning processes [37].

Convolutional Neural Networks have similarities with the traditional ANN method, where there are neurons that will perform self-optimization through the learning process. The most striking difference between CNNs and ANNs is that CNNs are mainly used in the field of pattern recognition on an image [38], [39].

1) Convolutional Layer

Convolutional Layer plays a major role in running CNN[9]. The input layer provides the image to be classified and the output is a prediction class label, which is calculated using the extracted features of the image [40]. Before proceeding, it is necessary to know the comparison of humans and computers in viewing an image.

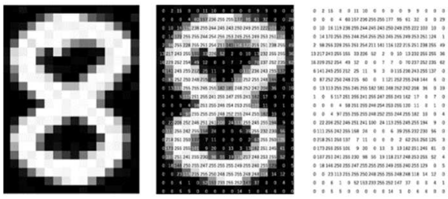

Fig. 3 How Computers See Images

In Figure 3, suppose there is an image of the number 8; the image on the far left is what humans see, while the image on the far right contains the values for each pixel because the image is grayscale, so a number unit can represent it. On the far right is how the computer sees figure 8 earlier.

The parameters used in the convolutional Layer are kernels or, commonly called filters [41]. These kernels are usually small matrices and can be self-determined. When the input data enters the Convolutional Layer, it combines each filter across the spatial dimensions of the input to produce an activation output [42].

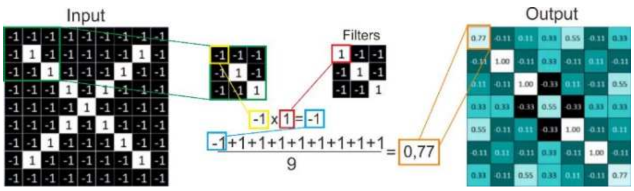

Fig. 4 Process on Convolution Layer

In Figure 4, it has been determined that the filter is a 3x3 matrix. Therefore, in the first 3x3 block, the input vector is taken (shown in process 1) and then calculated with the filter. The result of the calculation will be placed in the middle according to the position of the input vector that is being calculated. When finished, the process will shift one block to the right until it is finally finished in the lower right corner, this process is called Convolution Operation [43]. The output *aij* in the next Layer for the position (i,j) [44], is calculated after applying the convolution operation using the formula given as follows:

$$\alpha_{ij} = \sigma\left((W * X)_{ij} + b\right) \tag{1}$$

Where X is the input given to the Layer, W is the filter that shifts over the input, b is the bias, * represents the convolution operation, and is the non-linearity present in the network [45].

2) Pool Layer

Pooling Layer aims to reduce or shrink the dimensional representation gradually from the output generated by the Convolutional Layer to reduce the number of parameters and the complexity of the model computation process [46], [46]. CNN generally uses a 2x2 dimensional filter to perform pooling, which will select along the spatial dimensions of the input [47]. The pooling process can be seen in Figure 5.

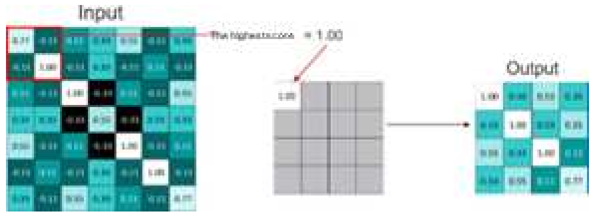

Fig. 5 Pooling Layer using MAX Value

There are several methods of pooling, including average pooling and max-pooling. In the example above, max-pooling has been carried out by determining the largest value for each pooling process carried out. Pooling will generate a new vector output [48].

3) Fully Connected Layer

The output of the Convolution Layer and Pooling layer is entered into the fully connected Layer, and the dot product of the weight vector and the input vector is calculated to get the final output [49]. Fully Connected Layer will generate a score for each class which will be used for classification[9].

4) Activation Function

Activation Function is a normalization function which aims to keep the vector values from being overloaded [50]. The trick is to replace all negative values with zero values, the ReLU function can be formulated as follows:

$$f(x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases} \tag{2}$$

Rectified Linear Units (ReLUs) is one of the computational units that can perform these calculations. ReLU works by selecting block-by-block from the output matrix shown in Figure 6.

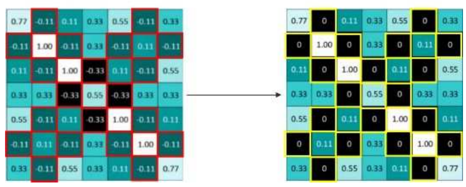

Fig. 6 Normalization using ReLU

C. Face Shape Classification

Classification of face shape is a rather complicated problem because everyone has their perspective in assessing which face shape a person has. Therefore, this study will quote from previous research on the grouping of face shapes. After conducting various studies, human faces can be categorized into six groups: round, oval, oblong, square, heart, and diamond [51].

D. InceptionV3

InceptionV3 is one of the pre-trained models in TensorFlow [26]. This is the advanced version of computer vision after Inception-vl, Inception-v2 in 2015. The model on InceptionV3 is trained using ImageNet datasets and contains information that can identify 1000 classes on ImageNet, the error (Figure 7).

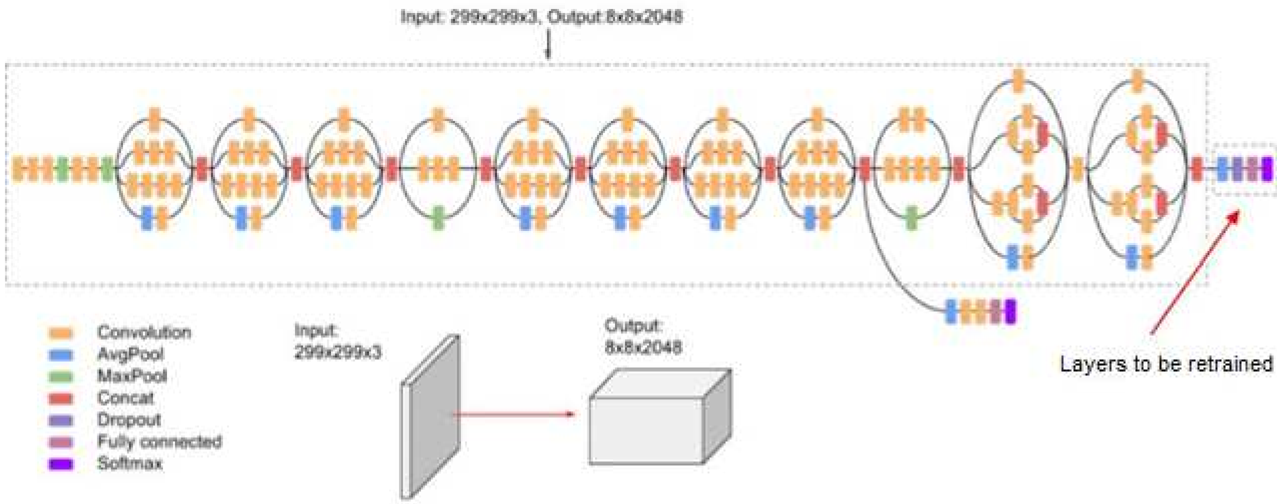

Fig. 7 InceptionV3's complete architecture

E. Canny Edge Detector

Canny Edge Detector is an edge detection method that uses a multi-stage algorithm to detect various edges in an image. This algorithm was developed by John F. Canny in 1986. The method in Canny Edge Detector is as follows [52]:

- Smoothing: Blurring an image using a Gaussian filter to remove Noise.
- Finding Gradients: Edges should be marked where the gradient of the image has a large magnitude.
- Non-maximum suppression: Only local maximum values should be marked as edges.
- Double Thresholding: Potential edges are determined by thresholding.
- Edge Detection: The final edge is determined by pressing all unconnected edges to a definite (strong) edge.

This research has three main stages, namely Image Pre-processing, training section, and testing section. A more complete explanation can be seen in Figure 8.

The research started from collecting facial datasets through search engines such as Google images, Yandex images, and Bing images. The dataset obtained is 747 images and has been grouped into five different face categories: oval, round, square, heart, and oblong. After that, the Image Pre-processing stage is carried out, where labels are added to the dataset based on the five previously mentioned categories, then change the dataset from RGB format to Grayscale to apply Canny Edge Detection.

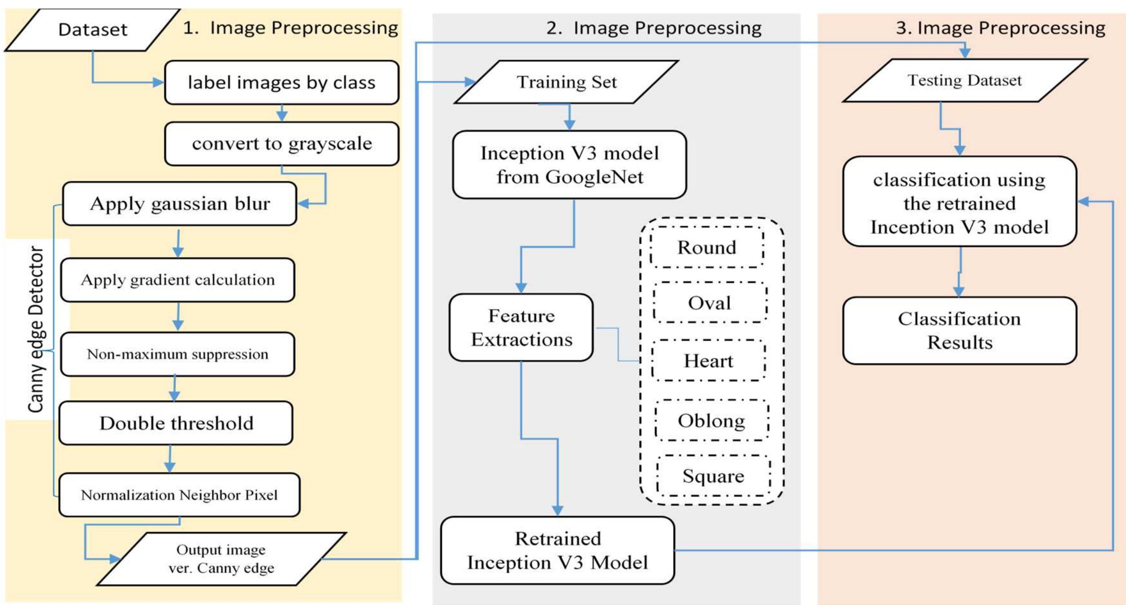

Fig. 8 Research Methodology

Furthermore, at the InceptionV3 Training stage, namely by retraining the last layers or layers in InceptionV3 using the transfer learning method utilizing the existing InceptionV3 model. A new model will be produced from the transfer learning process, namely a model of five predetermined face shapes. The classification results will be displayed using the plotlib feature of Python.

III. Results and Discussion

A. Image Pre Processing

This stage begins with collecting facial datasets by using the search engines Google images, Yandex images, and Bing images to search for photos based on the classification of face shapes. In addition, the researcher also uses several datasets used by previous researchers and categorizes them into five face-shape classifications (Figure 9) [53].

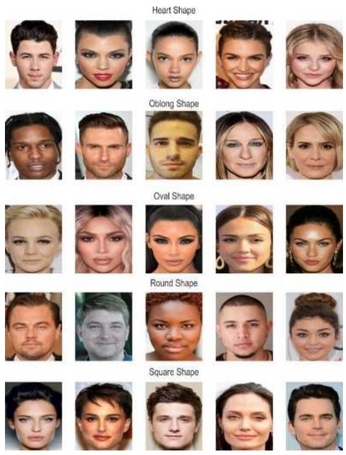

Fig. 9 Sample dataset of five face-shape classes

Before using the dataset as training and testing material, the first process is to do pre-processing, which aims to reduce unnecessary image features, such as dresses, hair, backgrounds, etc. Also, change the image format to Canny Edge. Figure 10 is a cropping process done manually because some of the auto-cropping features provided by Python cannot maintain some important features of the image.

Fig. 10 Initial photo and cropped photo

After that, the image format is changed from RGB to Grayscale (Figure 11).

Fig. 11 Before and after applying Grayscale

The next process is to convert the input image into an edge type using a canny edge detector. The process that occurs when the canny edge detector feature is applied is as follows[14]:

- Reduce Noise by applying Gaussian blur.
- Applying Gradient calculation to detect edges of 4 types, namely horizontal, vertical, and two diagonals.
- Perform Non-maximum suppression to thin the edges of the previous process by selecting the pixel with the highest value in the side direction.
- Applying a double threshold (threshold) to select strong, weak, and irrelevant pixels.
- Change the weak pixel edge to be strong based on the neighbor pixel by applying a lower threshold of 0.1 and an upper threshold of 0.3.

The above process can be seen in Figure 12.

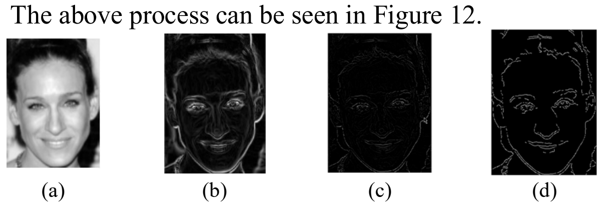

Fig. 12 Results of (a) Gaussian Blur (b) Gradient Calculation and (c) Non-Maximum Suppression Results (d) Final Results of Canny Edge Detection

B. Training Section

The data training process uses the transfer learning method from the InceptionV3 model that Google Net provided. TensorFlow has provided the script on its GitHub page to make it easier for researchers to retrain the last Layer. The training carried out by researchers uses the following conditions:

- Training Steps: 6000
- Learning Rate: 0.02
- Training Batch Size: 100
- Testing Percentage: 10
- Validation Percentage: 10
- Eval Step Interval : 10

When the training process is complete, it will output retrained_graph.pb, retrained_labels.txt, and bottleneck files. Retrained graph.pb is a new model in which they can classify five classes of face shapes. The file will later be used for data testing. Meanwhile, the bottleneck file contains important information from each photo that has been trained.

C. Testing Section

The data testing process uses the same dataset as the dataset used for the training process. The output of this testing process is the percentage of classification accuracy in the five classes of face shapes provided. The training process logs are stored in the /tmp/retrain_logs folder, which contains information about the training process that has been carried out. By using the Tensorboard, researchers can get information such as pictures Figure 13.

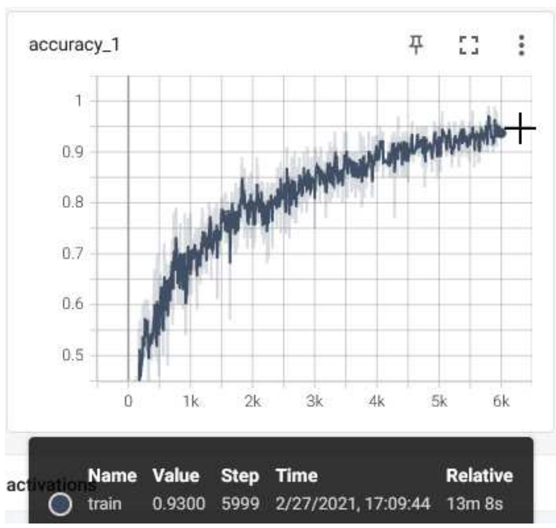

Fig. 13 The level of accuracy during training

From the graph above, it can be concluded that in the 5999th steps (out of a total of 6000), InceptionV3 can achieve an accuracy rate of 0.9300 or 93% with a record length of the training process which is 13 minutes 8 seconds.

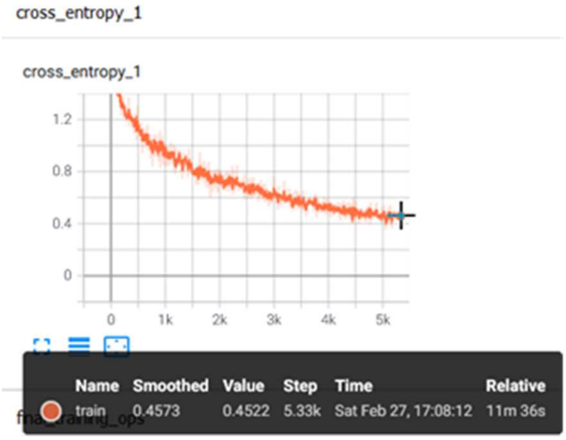

Fig. 14 Cross Entropy Loss rate during training

In addition, the value of cross-entropy loss during training is also obtained, which is 0.4573. The lower the cross-entropy value, the better the algorithm and dataset will be. Tests are carried out using the same dataset when conducting training in order to facilitate research. InceptionV3's accuracy percentage can be seen in Table I.

TABLE I
INCEPTION V3 ACCURACY PERCENTAGE IN FACE SHAPE CLASSIFICATION

Original Class	Class Prediction by InceptionV3							
	Heart	**Oblong**	**Oval**	**Round**	**Square**	**Prediction True**	**Amount data**	**Percentage**
Heart	130	5	5	9	1	130	150	86,667%
Oblong	0	146	0	2	1	146	149	97,985%
Oval	2	6	137	3	1	137	149	91,946%
Round	1	1	0	147	0	147	149	98,657%
Square	2	7	2	7	132	132	150	88%
		Total				**692**	**747**	**92.651%**

InceptionV3 in Table 1 can obtain very high accuracy values n the classification of face shapes with values between 86% - 98%. The processing time required for InceptionV3 has also been presented in Figure 15.

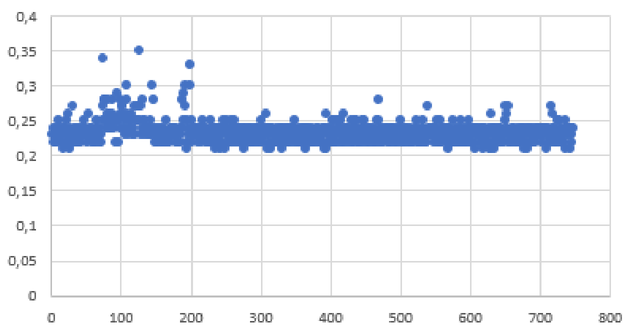

Fig. 15 InceptionV3 Processing Time (Second)

From Table 1, it can also be concluded that the shape of the face of the heart gets the lowest level of accuracy, which is 86.667%. Figure 16 is an example of heart-shaped data that is not grouped properly.


Fig. 16 Heart class dataset detected as spherical

This study uses Canny Edge Detection as a pre-processing method to eliminate image disturbances such as clothing and skin color. However, it turns out that some low-resolution images result in Canny Edge not being perfect in its feature extraction. In the picture above, some photos cannot be detected properly on the face, resulting in missing features.

Another annoyance that may result in misclassification is that several datasets with different classes have the same hairstyle, where the hair, when applied to Canny Edge, will produce the same features. This causes the emergence of new Noise in the data used.

In Table I, it can be concluded that the shape of the heart face is most misclassified into a round face shape with nine error data written (Figure 17) are some data on the shape of a heart that is detected as a round face as well as a sample dataset of the round face itself.

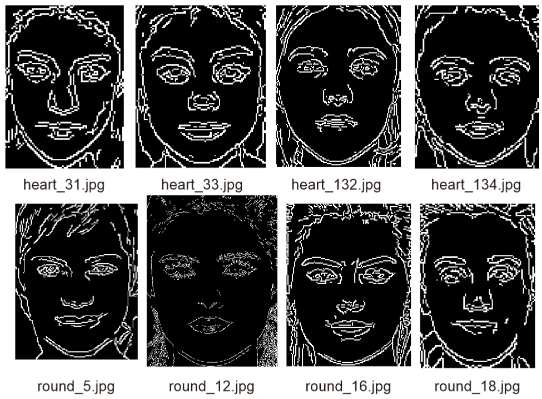

Fig. 17 Comparison of heart datasets classified as round

Based on the eight data examples, all data have a Double Chin element in the form of two parallel lines located on the chin. This condition causes the four heart face shape data to be classified as round faces.

IV. CONCLUSION

In the classification of face shapes, InceptionV3 can get a high accuracy value, which is between 86% - 98% in each class. Image quality and clarity greatly affect InceptionV3's level of accuracy in classifying. Some of the datasets used in this study have a small resolution, which makes it difficult for Canny Edge to detect the sides of the face. Hair is a noise factor in detecting the side of the face because it does not matter what the hairstyle is. What matters is the side of the face. If oblong class and heart class datasets have similar hairstyles, InceptionV3 will detect these features and assume that both data come from the same class. The Canny Edge

Detection method was applied in the image pre-processing process, which still produces some noise on certain data.

Therefore, it is recommended to use Facial Landmarks, which use key features on a face that are anatomically and geometrically significant, such as the corners of the eyes, the tip of the nose, the corners of the mouth, and so on, to represent a face's shape and position accurately. Then, using plot features in Python, particularly libraries like Matplotlib, can be a valuable aid in image pre-processing tasks like Image Visualization, Image Cropping and Region of Interest (ROI) Selection, Color Space Conversion, and so on.

ACKNOWLEDGMENT

We thank Universitas AMIKOM Yogyakarta for providing support and guidance in completing this research so that we can present it in a paper to be published and presented.

REFERENCES

[1] M.-H. Guo *et al.*, "Attention mechanisms in computer vision: A survey," *Comput. Vis. Media*, vol. 8, no. 3, pp. 331–368, Sep. 2022.

[2] S. Xu, J. Wang, W. Shou, T. Ngo, A.-M. Sadick, and X. Wang, "Computer Vision Techniques in Construction: A Critical Review," *Arch. Comput. Methods Eng.*, vol. 28, no. 5, pp. 3383–3397, Aug. 2021.

[3] T. M. Ward *et al.*, "Computer vision in surgery," *Surgery*, vol. 169, no. 5, pp. 1253–1256, May 2021.

[4] W.-H. Cheng, S. Song, C.-Y. Chen, S. C. Hidayati, and J. Liu, "Fashion Meets Computer Vision," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–41, May 2022.

[5] C. Zhang and Y. Lu, "Study on artificial intelligence: The state of the art and future prospects," *J. Ind. Inf. Integr.*, vol. 23, p. 100224, Sep. 2021.

[6] P. Stone *et al.*, "Artificial Intelligence and Life in 2030: The One Hundred Year Study on Artificial Intelligence," Oct. 2022.

[7] A. Taeihagh, "Governance of artificial intelligence," *Policy Soc.*, vol. 40, no. 2, pp. 137–157, Apr. 2021.

[8] S. Thiebes, S. Lins, and A. Sunyaev, "Trustworthy artificial intelligence," *Electron. Mark.*, vol. 31, no. 2, pp. 447–464, Jun. 2021.

[9] D. I. Patrício and R. Rieder, "Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review," *Comput. Electron. Agric.*, vol. 153, pp. 69–81, Oct. 2018.

[10] T. Hidayat, M. H. Zakaria, and A. N. C. Pee, "Survey of Performance Measurement Indicators for Lossless Compression Technique based on the Objectives," in *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, 2020, pp. 170–175.

[11] S. Paneru and I. Jeelani, "Computer vision applications in construction: Current state, opportunities & challenges," *Autom. Constr.*, vol. 132, p. 103940, Dec. 2021.

[12] D. Bhatt *et al.*, "CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope," *Electronics*, vol. 10, no. 20, p. 2470, Oct. 2021.

[13] A. B. M. R. Islam, "Machine Learning in Computer Vision," 2022, pp. 48–72.

[14] M. Afifi, A. Abdelhamed, A. Abuolaim, A. Punnappurath, and M. S. Brown, "CIE XYZ Net: Unprocessing Images for Low-Level Computer Vision Tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2021.

[15] J. Zhao, R. Masood, and S. Seneviratne, "A Review of Computer Vision Methods in Network Security," *IEEE Commun. Surv. Tutorials*, vol. 23, no. 3, pp. 1838–1878, 2021.

[16] S. Nie, L. Gu, Y. Zheng, A. Lam, N. Ono, and I. Sato, "Deeply Learned Filter Response Functions for Hyperspectral Reconstruction," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4767–4776, 2018.

[17] I. Adjabi, A. Ouahabi, A. Benzaoui, and A. Taleb-Ahmed, "Past, Present, and Future of Face Recognition: A Review," *Electronics*, vol. 9, no. 8, p. 1188, Jul. 2020.

[18] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, Mar. 2021.

[19] V. Chawda, V. Arya, S. Pandey, Shristi, and M. Valleti, "Unique Face Identification System using Machine Learning," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2020, pp. 701–706.

[20] K. Pasupa, W. Sunhem, and C. K. Loo, "A hybrid approach to building face shape classifier for hairstyle recommender system," *Expert Syst. Appl.*, vol. 120, pp. 14–32, Apr. 2019.

[21] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing," *ISPRS J. Photogramm. Remote Sens.*, vol. 173, pp. 24–49, Mar. 2021.

[22] J. P. Bohnslav *et al.*, "DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels," *Elife*, vol. 10, Sep. 2021.

[23] N. Memon, H. Parikh, S. B. Patel, D. Patel, and V. D. Patel, "Automatic land cover classification of multi-resolution dualpol data using convolutional neural network (CNN)," *Remote Sens. Appl. Soc. Environ.*, vol. 22, p. 100491, Apr. 2021.

[24] A. Moradzadeh, H. Teimourzadeh, B. Mohammadi-Ivatloo, and K. Pourhossein, "Hybrid CNN-LSTM approaches for identification of type and locations of transmission line faults," *Int. J. Electr. Power Energy Syst.*, vol. 135, p. 107563, Feb. 2022.

[25] S. M. Hassan, A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska, "Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach," *Electronics*, vol. 10, no. 12, p. 1388, Jun. 2021.

[26] Xiaoling Xia, Cui Xu, and Bing Nan, "Inception-v3 for flower classification," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, 2017, pp. 783–787.

[27] X. Cui *et al.*, "Assessing the effectiveness of artificial intelligence methods for melanoma: A retrospective review," *J. Am. Acad. Dermatol.*, vol. 81, no. 5, pp. 1176–1180, Nov. 2019.

[28] T. Alzahrani, W. Al-Nuaimy, and B. Al-Bander, "Integrated Multi-Model Face Shape and Eye Attributes Identification for Hair Style and Eyelashes Recommendation," *Computation*, vol. 9, no. 5, p. 54, Apr. 2021.

[29] N. K. Bansode and P. K. Sinha, "Face Shape Classification Based on Region Similarity, Correlation and Fractal Dimensions," *Int. J. Comput. Sci. Issues*, vol. 13, no. 1, pp. 24–31, Feb. 2016.

[30] P. Sarakon, T. Charoenpong, and S. Charoensiriwath, "Face shape classification from 3D human data by using SVM," in *The 7th 2014 Biomedical Engineering International Conference*, 2014, pp. 1–5.

[31] V. Wiley and T. Lucas, "Computer Vision and Image Processing: A Paper Review," *Int. J. Artif. Intell. Res.*, vol. 2, no. 1, p. 22, Jun. 2018.

[32] O. Cosido *et al.*, "Hybridization of Convergent Photogrammetry, Computer Vision, and Artificial Intelligence for Digital Documentation of Cultural Heritage - A Case Study: The Magdalena Palace," in *2014 International Conference on Cyberworlds*, 2014, pp. 369–376.

[33] O. H. Babatunde, L. Armstrong, J. Leng, and D. Diepeveen, "A survey of computer-based vision systems for automatic identification of plant species," *J. Agric. Informatics*, vol. 6, no. 1, Jan. 2015.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[35] O. A. Montesinos López, A. Montesinos López, and J. Crossa, "Fundamentals of Artificial Neural Networks and Deep Learning," in *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, Cham: Springer International Publishing, 2022, pp. 379–425.

[36] P. N. Suganthan and R. Katuwal, "On the origins of randomization-based feedforward neural networks," *Appl. Soft Comput.*, vol. 105, p. 107239, Jul. 2021.

[37] M. Q. Li, B. C. M. Fung, and A. Abusitta, "On the Effectiveness of Interpretable Feedforward Neural Network," in *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1–8.

[38] S. Kareem, Z. J. Hamad, and S. Askar, "An evaluation of CNN and ANN in prediction weather forecasting: A review," *Sustain. Eng. Innov.*, vol. 3, no. 2, pp. 148–159, Oct. 2021.

[39] T. Hidayat, M. H. Zakaria, and A. N. C. Pee, "Increasing the Huffman generation code algorithm to equalize compression ratio and time in lossless 16-bit data archiving," *Multimed. Tools Appl.*, vol. 82, no. 16, pp. 24031–24068, Jul. 2023.

[40] N. Sharma, V. Jain, and A. Mishra, "An Analysis Of Convolutional Neural Networks For Image Classification," *Procedia Comput. Sci.*, vol. 132, pp. 377–384, 2018.

[41] P. K. Roy, "Multilayer Convolutional Neural Network to Filter Low Quality Content from Quora," *Neural Process. Lett.*, vol. 52, no. 1, pp. 805–821, Aug. 2020.

[42] S. Qiu, X. Xu, and B. Cai, "FReLU: Flexible Rectified Linear Units for Improving Convolutional Neural Networks," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 1223–1228.

[43] A. Ajit, K. Acharya, and A. Samanta, "A Review of Convolutional Neural Networks," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, 2020, pp. 1–5.

[44] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," *Procedia Comput. Sci.*, vol. 132, pp. 679–688, 2018.

[45] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, Feb. 2017.

[46] G. J.L., B. Abraham, S. M.S., and M. S. Nair, "A computer-aided diagnosis system for the classification of COVID-19 and non-COVID-19 pneumonia on chest X-ray images by integrating CNN with sparse autoencoder and feed forward neural network," *Comput. Biol. Med.*, vol. 141, p. 105134, Feb. 2022.

[47] H. J. Jie and P. Wanda, "RunPool: A Dynamic Pooling Layer for Convolution Neural Network," *Int. J. Comput. Intell. Syst.*, vol. 13, no. 1, p. 66, 2020.

[48] P. Kim, "Convolutional Neural Network," in *MATLAB Deep Learning*, Berkeley, CA: Apress, 2017, pp. 121–147.

[49] K. Liu, G. Kang, N. Zhang, and B. Hou, "Breast Cancer Classification Based on Fully-Connected Layer First Convolutional Neural Networks," *IEEE Access*, vol. 6, pp. 23722–23732, 2018.

[50] G. Lin and W. Shen, "Research on convolutional neural network based on improved Relu piecewise activation function," *Procedia Comput. Sci.*, vol. 131, pp. 977–984, 2018.

[51] W. Sunhem and K. Pasupa, "An approach to face shape classification for hairstyle recommendation," in *2016 Eighth International Conference on Advanced Computational Intelligence (ICACI)*, 2016, pp. 390–394.

[52] D. Gaur, D. Mehrotra, and K. Singh, "A new approach to simulate the dynamic behavior of particulate matter using a canny edge detector embedded PIV algorithm," *Vis. Comput.*, Jul. 2022.

[53] A. E. Tio, "Face shape classification using Inception v3," no. November 2019, 2019.