Application of Convolution Techniques with Bounding Box for Buddha Statue Face Image Processing

Linda Marlinda[1*], Marysca Shintya Dewi[2]
[1]Informatika, Universitas Nusa Mandiri, Jakarta, Indonesia
[2]Teknik Mesin, Universitas Dian Nusantara, Jakarta, Indonesia
Email address: linda.ldm@nusamandiri.ac.id[1], marysca.shintya.dewi@dosen.undira.ac.id[2]

***Abstract*— *Face recognition on Buddha statues is a significant challenge in cultural heritage preservation research, especially when the images have a high degree of similarity, low quality, or uneven lighting. This makes identifying faces on Buddha statues in museums or historical sites difficult. This study aims to develop an effective method for detecting faces on Buddha statues using a convolutional neural network (CNN) combined with the bounding box technique to improve detection accuracy. The bounding box technique is applied to reduce the area of analysis and improve the efficiency of the face detection process. In addition, variations in the kernel size and parameter stride of CNN are analyzed to obtain optimal results in face recognition with distorted images. The experimental results show that combining CNN with a bounding box significantly improves face detection accuracy on Buddha statues, even on images with uneven lighting and low quality. This technique is superior to conventional face detection methods under difficult image conditions. This study contributes to developing more accurate face-detection techniques for cultural heritage preservation. Applying this method can improve face recognition on historical artifacts with low image quality and poor lighting and opens up opportunities for further research in technology-based cultural conservation.*

***Keywords*— *Convolution, Image, Buddha Statue, Bounding Box*

I. Introduction

Digital image processing plays a vital role in various fields, including facial recognition, cultural heritage preservation, and object detection[1][2]. Images, by their very nature, represent objects or scenes and can be classified into two categories: visible and invisible[3][4]. The focus here is on digital images, which are usually represented as a matrix of pixel intensities, which allows them to be processed using computational techniques[5][6]. However, image quality often faces challenges due to noise, blurriness, or inadequate sharpness, which can hinder further interpretation and analysis[7][8]. These limitations necessitate the development of effective image enhancement and feature extraction techniques[9][4].

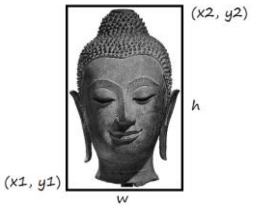

Figure 1. Digital Image of Buddha

Mathematically, a digital image is represented as a function of light intensity in a two-dimensional plane. The light intensity at a given coordinate (x, y) is denoted by f(x, y), where (x, y) are the coordinates of a pixel in the image[10] [11]. A digital image can be defined as an N x M matrix, where N is the height and M is the width of the image, and each element in the matrix represents a pixel[12]. An N x M image consists of NM pixels[13]. In digital image processing, color is typically represented by a combination of three primary colors: red, green, and blue (RGB)[14][15]. Each color component has an intensity range from 0 to 255. For example, yellow is a combination of red and green, with RGB values of R=255, G=255, and B=0. Each pixel in a color image requires 3 bytes to store its color information[15][16].

$$= \begin{bmatrix} x_{1,11} & \cdots & x_{1,1n} & \cdots & x_{1,m1} & \cdots & x_{1,mn} \\ x_{2,11} & \cdots & x_{2,1n} & \cdots & x_{2,m1} & \cdots & x_{2,mn} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{k,11} & \cdots & x_{k,1n} & \cdots & x_{k,m1} & \cdots & x_{k,mn} \end{bmatrix}$$

Figure 2. illustration of the image of the Buddha matrix

Recent advances in Convolutional Neural Networks (CNNs) have had a significant impact on image processing, particularly in the domain of facial recognition[17]. CNNs, deep learning architectures that utilize convolution operations, have demonstrated superior performance in tasks such as image smoothing, sharpening, Gaussian blurring, edge detection, and object recognition[18][17]. For example, facial expression recognition using multi-branch CNNs has achieved remarkable improvements over traditional methods, highlighting the power of convolution operations in extracting and interpreting image features[18][19].

Despite CNNs' success in processing facial images, their application to historical or cultural objects—such as Buddha statues—remains unexplored. The distinctive facial features of Buddha statues, including texture variations, erosion, or inconsistent lighting, pose challenges to accurate recognition and analysis[20]. Most previous studies have focused on conventional facial images and failed to address the issues posed by cultural heritage artifacts, leaving a gap in the current literature.

The application of convolutional neural networks (CNNs) in image processing has revolutionized tasks such as facial expression recognition and artifact analysis. However, the adaptation of CNNs to address the unique challenges of processing the faces of Buddha statues represents a novel approach in cultural heritage preservation. The results of this study can have significant implications for the academic community and practical applications in the preservation and recognition of historical artifacts[21][20].

This research aims to bridge this gap by applying convolution techniques combined with bounding box methods to process facial images of Buddha statues. The bounding box approach helps isolate facial regions, improving the efficiency and accuracy of subsequent convolution operations. By integrating these methods, this study seeks to improve image quality, extract relevant features, and facilitate more effective facial analysis of Buddha statues.

The novelty of this study lies in the application of convolution techniques to historical artifacts, specifically Buddha statues, which overcomes challenges such as noise, low contrast, and non-uniform lighting. Furthermore, this study explores the impact of kernel size and stride parameters on the performance of convolution operations, an area that has not been thoroughly investigated in the context of cultural heritage image processing.

This study contributes to the preservation of cultural heritage. The findings can also improve automatic artifact recognition systems, benefiting museums, researchers, and cultural preservation efforts. The application of convolutional neural networks (CNNs) in image processing has revolutionized tasks such as facial expression recognition and artifact analysis. However, the adaptation of CNNs to address the unique challenges of processing the faces of Buddha statues represents a novel approach in cultural heritage preservation. The results of this study can have significant implications for the academic community and practical applications in the preservation and recognition of historical artifacts.

II. Previous Research Methods

Convolutional Neural Network (CNN) is a Multilayer Perceptron (MLP) development designed to process two-dimensional data. CNN is included in the Deep Neural Network type because the network depth is high and is widely applied to image data. In the case of image classification, MLP is unsuitable because it does not store spatial information from image data and considers each pixel to be an independent feature, thus producing poor results and designing a program to process images using the convolution method.

Convolution is an essential mathematical operator for many image-processing operations. It provides a way to combine two arrays, usually of different sizes but with the exact array dimensions, resulting in a third array with the exact dimensions[22]. In image processing, convolution can be used to apply operators with pixel output values derived from a linear combination of certain input pixel values[15].

A technique for smoothing or clarifying an image is replacing the pixel value with several pixel values that match or are close to the original pixel[23]. However, with convolution, the size of the image remains the same and does not change. Convolution has two functions, f(x) and g(x), which are defined as follows:

$$T : M^{n\times n} \to R^{n^2}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \to \begin{bmatrix} a_{11} \\ \vdots \\ a_{1n} \\ a_{21} \\ \vdots \\ a_{2n} \\ \vdots \\ a_{n1} \\ \vdots \\ a_{nn} \end{bmatrix} \qquad (1)$$

$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(a) \cdot g(x-a)da \qquad (2)$$

$h(x)$ = Convolution product
$f(x)$ = Input function (digital image matrix)
$g(x)$ = Kernel functions

$$h(x,y) = f(x) * g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(a,b) \cdot g(x-a, y-b)dadb \qquad (3)$$

$h(x,y)$ = Convolution product
$f(x)$ = Input function (digital image matrix)
$g(x,y)$ = Kernel functions

$$h(x) = f(x) * g(x) = \sum_{-\infty}^{\infty} f(a) \cdot g(x-a) \qquad (4)$$

$h(x)$ = Convolution product
$f(x)$ = Input function (digital image matrix)
$g(x)$ = Kernel functions

$$h(x,y) = f(x) * g(x,y) = \sum_{-\infty}^{\infty}\sum_{-\infty}^{\infty} f(a) \cdot g(x-a, y-b) \qquad (5)$$

$h(x,y)$ = Convolution product
$f(x)$ = Input function (digital image matrix)
$g(x,y)$ = Kernel functions

$$f(i,j) = Ap_1 + Bp_2 + Cp_3 + Dp_4 + Ep_5 + Fp_6 + Gp_7 + Hp_8 + Ip_9 \qquad (6)$$

$f(i,j)$ = The result of the convolved image
$A - I$ = Kernel matrix
$P_1 P_9$ = Kernel functions

The output of the previous layer is subjected to convolution operations by the Convolution Layer. This layer is the main process underlying a CNN[24]. Applying a function repeatedly to the output of another function is known as convolution in mathematics. In image processing, convolution means applying a kernel (yellow box) to the image at all possible offsets as shown in Fig. 4. The green box as a whole is the image that will be convolved. From the upper left corner to the lower right, the kernel travels. Thus, the image on the right displays the convolution results of this image. The purpose of convolution on image data is to extract features from the input image. Depending on the spatial information in the data, convolution will result in a linear transformation of the input data. The weights in this layer specify the convolution kernel used so that the convolution kernel can be trained based on the input to the CNN[25][26].

Subsampling is the process of reducing the size of image data. In image processing, subsampling also aims to increase the position invariance of features[27]. In most CNNs, the subsampling method used is max pooling. Max pooling divides the output from the convolution layer into several small grids and then takes the maximum value from each grid to construct a reduced image matrix[21] as shown in Figure 4. The red, green, yellow, and blue grids are the grid groups for which the maximum value will be selected. So the results of this process can be seen in the grid collection to the right. This process ensures that the features obtained will be the same even if the image object experiences translation. The use of pooling layers in CNN only aims to reduce image size so that it can be easily replaced with a convolution layer with the same stride as the pooling layer in question.

Fully Connected Layer This layer is a layer that is usually used in MLP applications and aims to carry out transformations on data dimensions so that data can be classified linearly[28][15]. Each neuron in the convolution layer needs to be transformed into one-dimensional data first before it can be entered into a fully connected layer. Because this causes the data to lose its spatial information and is not reversible, a fully connected layer can only be implemented at the end of the network. it is explained that a convolution layer with a kernel size of 1 x 1 performs the same function as a fully connected layer but still maintains the spatial character of the data[25][17]. This means that the use of the fully connected layer in CNN is now not widely used.

III. PROPOSED METHODE

The convolution process between the image and kernel can be described as follows:

1. Start by placing the kernel in the upper left corner of the image, then calculate the pixel value at position (0,0) from the kernel.
2. Shift the kernel one pixel to the right, and recalculate the pixel value at position (0,0) of the kernel.
3. Once again, move the kernel one pixel to the right, and calculate the pixel value at position (0,0) of the kernel.
4. Shift the kernel one pixel down, and re-initialize the convolution process from the left side of the image. Each convolution iteration shifts the kernel or pixel to the right.
5. If the convolution result produces a negative pixel value, the value is set to 0. Conversely, if the convolution result produces a pixel value greater than the maximum gray value (255), the value is trimmed to the maximum gray value.
6. Note that problems can arise when the convolved pixels are at the edges of the image because some convolution coefficients cannot be placed on the image pixels.
7. During the convolution operation, the convolution kernel is shifted pixel by pixel, producing output pixel values f(i,j), which are then stored in a new matrix.
8. The entire convolution process is useful for carrying out filtering operations on images, allowing the identification of certain patterns or features in the image.

Figure 3, the image processing process begins with the original image input step as the initial stage. Next, image dimensions are calculated to determine the size or dimensions of the image to be processed. The next step involves input of the kernel matrix, where the kernel matrix becomes an integral part of the convolution process. The image matrix convolution process (f(x, y), g(x, y)) is then executed, where the original image undergoes convolution with the kernel matrix to produce a convolution image.

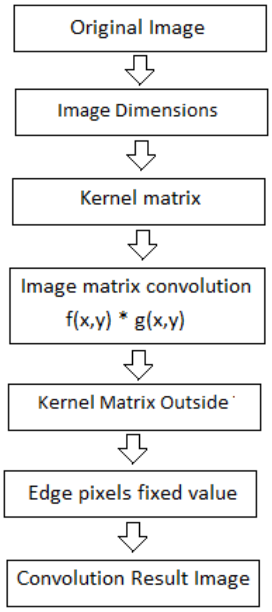

Figure 3. Convolution method flow

If the kernel matrix is placed outside the image boundaries and the pixel values at the edges remain unchanged, this step ensures that the convolution process occurs correctly. After the convolution stage, the next step is to display and replace the checked pixels with new pixel values. This refers to replacing the pixel values in the convolution image according to the values produced during the convolution process. It is important to note that this process maintains pixel values at the edges of the image according to predefined rules. Overall, this process details the steps from the initial input image to produce a convolutional image by paying attention to kernel placement, dimension calculations, and replacing pixel values according to specific rules.

IV. RESULTS AND DISCUSSITIONS

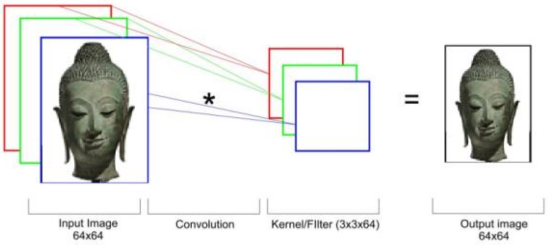

Figure 4. Convolution Process

The process of combining two series of numbers to produce a third series of numbers. If implemented on numbers, we get a convolution as an array matrix. The image has three channels, or what is commonly referred to as RGB, and its pixel size at input is 64x64x3, which indicates that its pixel height and width are 64. Each pixel channel has a different matrix value. The input will be convo with the specified filter value. Filter another block or cube with a smaller height and width but the same depth as the original image. Filters determine what patterns will be detected, which are then convolved or multiplied by the values in the input matrix. The values in each column and row in the matrix depend on the pattern type to be detected. The number of filters in this convolution is 64 pixels with a kernel size (3x3), and the resulting image is 64 feature maps with matrix samples in the input image. Because the input image has a pixel size 64x64, the researcher only took a portion of the matrix values as samples in the convolution process.

The resulting image is sometimes flawed; there are often glitches such as sudden changes in grayscale intensity, excessive brightness or darkness, lack of sharpness, blurriness, and noise. Images are not always free from such interference. The following illustrates an input image measuring 5x5, which will be processed using convolution techniques with a kernel measuring 3x3, producing an output image measuring 3x3. The stages of the image improvement process include:

a. Refining the image of Buddha's face

$$\begin{bmatrix} 128 & 118 & 99 & 99 & 113 \\ 122 & 152 & 192 & 121 & 100 \\ 201 & 160 & 203 & 105 & 143 \\ 222 & 176 & 118 & 124 & 163 \\ 205 & 160 & 148 & 159 & 129 \end{bmatrix} * \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 15 & 14 & 12 \\ 14 & 18 & 22 \\ 68 & 18 & 23 \end{pmatrix}$$

The final values of image smoothing are as follows:

128	118	99	99	113
122	152	192	121	100
201	160	203	105	143
222	176	118	124	163
205	160	148	159	129

(a)

15	14	12	99	113
14	18	22	121	100
68	18	23	105	143
222	176	118	124	163
205	160	148	159	129

(b)

Figure 5. (a) Pixel values before smoothing, (b) pixel values after smoothing.

From Figure 5 above, you can see the change in the intensity value of the middle pixel (7th pixel) which was originally valued at (128, 118, 99, 122, 152, 192, 201, 160, 203) after the smoothing process, the quality of the facial image changed to (15, 14, 12, 14, 18, 22, 68, 18, 23).

b. Image smoothing and sharpening are important processes in improving the quality of facial images. One way to do this is to multiply the center value of the 5x5 image by the 3x3 kernel, as shown below:

$$\begin{bmatrix} 128 & 118 & 99 & 99 & 113 \\ 122 & \mathbf{152} & \mathbf{192} & \mathbf{121} & 100 \\ 201 & \mathbf{160} & \mathbf{203} & \mathbf{105} & 143 \\ 222 & \mathbf{176} & \mathbf{118} & \mathbf{124} & 163 \\ 205 & 160 & 148 & 159 & 129 \end{bmatrix} * \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -192 & 0 \\ 160 & 812 & -105 \\ 0 & -118 & 0 \end{pmatrix}$$

Figure 6 Illustration of Convolution Process Calculation

The final values of image smoothing are as follows:

128	118	99	99	113
122	**152**	**192**	**121**	100
201	160	203	105	143
222	**176**	**118**	**124**	163
205	160	148	159	129

(a)

128	118	99	99	113
122	**0**	**-192**	**0**	100
201	**-160**	**812**	**-105**	143
222	**0**	**-118**	**0**	163
205	160	148	159	129

(b)

Figure 7. (a) Pixel value before sharpening, (b) pixel value after sharpening.

From Figure 6 above, you can see the change in the intensity value of the middle pixel (7th pixel) which was originally valued at (152, 192, 121, 160, 203, 105, 176, 118, 124) after the image side smoothing process changed to (0, -192, 0, -160, 812, -105, 0, -118, 0).

Figure 7 shows the convolution process using a kernel size of 3x3, using a stride of 1. Stride here means that the number of kernel shifts in the input matrix is one. convolution process where a kernel of size 3x3 starts on the left side. This process is called a sliding window. However, in this research, a padding value of 1 is given, namely adding the value 0 around the input matrix value so that the input and output have the same matrix value, so as not to reduce the information in the image. This process is carried out from the top left corner to the bottom left corner. The dot product calculation can be seen as follows:

The Fast Fourier Transform method was applied for quality improvement, using image input and convolution processes to smooth and sharpen the image. It succeeded in improving the quality of the face of the Buddha statue, which was experiencing noise.

The result of applying computational convolution operations to a local area is that the computation for a pixel in the output image involves neighboring pixels in the input image. Can be generated as follows:

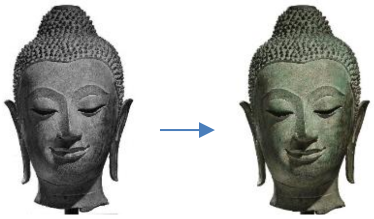

Figure 8. Convolution results on the Buddha face image matrix

The input image will be processed through two pre-processing stages: wrapping and cropping. At the wrapping stage, the edges of the main objects in the picture are checked. This edge is then identified as the maximum edge, ensuring that the results of cropping the object in the image remain intact. The training stage begins by converting the image into vector form, and the first process flow looks like in Figure 8.

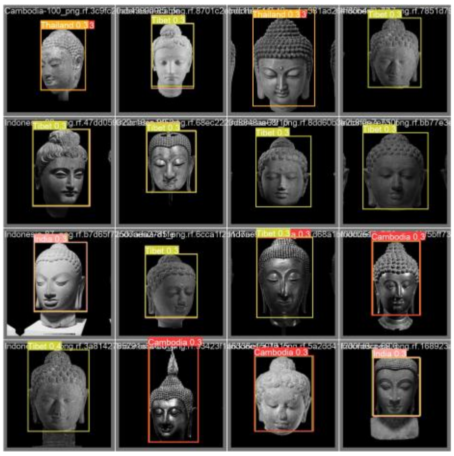

Figure 9. The Buddha face image with CNN and bounding box

Figure 9. It is the result The bounding box results using the Convolutional Neural Network (CNN) method in this image show the model's ability to detect and classify Buddha statue faces from various origins, such as Cambodia, Tibet, and India. Each detected face is marked with a colored bounding box and a label and probability score to indicate the model's confidence level. Yellow bounding boxes generally indicate detections

with medium to high confidence levels, while red indicates a lower score, indicating the model's uncertainty in classifying faces. Although some detections are accurate and precisely surround the face area, there are also bounding less precise boxes, especially on faces with similar features or poor lighting conditions.

Overall, these results reflect the challenges in detecting Buddha statue faces with a high level of similarity, such as textures, expressions, or facial structures that are almost similar. CNN can provide initial classification based on the extracted features, but the varying probability scores indicate a need for increased accuracy, especially in conditions of images with noise or artifacts. Further research can focus on using hybrid feature extraction methods or a combination of other algorithms, such as DCT and SIFT, to improve detection accuracy and better distinguish facial features.

The image data processing process starts with an arbitrary-sized image, which is then resized to 416. An image that only has one color scale, namely gray. The aim of differentiating color images to obtain greyscale photos is to reduce the information needed to process each image element. This is because gray is one color in the red, green, and blue color components, which has the same intensity, so it is only necessary to determine one intensity value for each image element that is needed to determine each image element in a color image.

V. Conclusion

The convolution kernel used is a $3 \times 3\ dan\ 2 \times 2$ matrix, so the image processing carried out has a negligible effect. However, the difference between the original and processed images is still visible. The initial process for image convolution is to convert it into an image matrix with degrees of gray (0 – 255), where each point has a value and is then multiplied by the kernel matrix. After the kernel matrix has Stride to shift the filter, the Stride process is complete, followed by the Padding process, namely increasing the pixel size with a specific value. The result or output of a convolution is a Feature Map. The convolution matrix with the smallest size is produced by a 3 × 3 kernel with Stride 2, while the largest is made by a 2 × 2 kernel Stride 1. An enormous Stride with the same kernel size produces a smaller convolution matrix size. A smaller convolution matrix will be produced if the kernel size is larger and has the same Stride. The application of convolution techniques in processing facial images of Buddha statues includes various methods, including the SIFT method, ORB, Otsu method, and convolutional neural networks. These techniques play an essential role in overcoming challenges related to face recognition, age estimation, the impact of social status on face perception, and complex image-processing tasks. This reference synthesis underscores the importance of convolution techniques in advancing image analysis and processing, particularly in the context of cultural artifacts and social perception.

References

[1] I. G. I. Sudipa, P. W. Aditama, and C. P. Yanti, "Developing Augmented Reality Lontar Prasi Bali as an E-learning Material to Preserve Balinese Culture," *J. Wirel. Mob. Networks, Ubiquitous Comput. Dependable Appl.*, vol. 13, no. 4, pp. 169–181, 2022, doi: 10.58346/JOWUA.2022.I4.011.

[2] W. M. W. Isa, N. A. M. Zin, F. Rosdi, and H. M. Sarim, "Digital preservation of intangible cultural heritage," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 3, pp. 1373–1379, 2018, doi: 10.11591/ijeecs.v12.i3.pp1373-1379.

[3] H. Suyuti and A. Setyanto, "The Use of Augmented Reality Technology in Preserving Cultural Heritage : A Case Study of Old Jami Mosque of Palopo," vol. 2, no. 1, 2023.

[4] E. Shcherbina and A. Salmo, "Exploring Impact of Historical and Cultural Heritage on the Sustainability of Urban and Rural Settlements," *E3S Web Conf.*, vol. 457, pp. 1–8, 2023, doi: 10.1051/e3sconf/202345703001.

[5] V. N. Kristanto, I. Riadi, and Y. Prayudi, "Forensic Analysis of Faces on Low-Quality Images using Detection and Recognition Methods," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 2, pp. 218–225, 2023, doi: 10.29207/resti.v7i2.4630.

[6] I. N. G. A. Astawa, M. L. Radhitya, I. W. R. Ardana, and F. A. Dwiyanto, "Face Images Classification using VGG-CNN," *Knowl. Eng. Data Sci.*, vol. 4, no. 1, p. 49, 2021, doi: 10.17977/um018v4i12021p49-54.

[7] S. Shivaprakash and S. V. Rajashekararadhya, "A Smart Face Recognition and Verification using Optimal Spatial and Spectral Feature Selection with Adaptive Multiscale Mobilenet," *Adv. Artif. Intell. Mach. Learn.*, vol. 3, no. 3, pp. 1407–1443, 2023, doi: 10.54364/aaiml.2023.1183.

[8] R. R. Damanik, D. Sitanggang, H. Pasaribu, H. Siagian, and F. Gulo, "An application of viola jones method for face recognition for absence process efficiency," *J. Phys. Conf. Ser.*, vol. 1007, no. 1, 2018, doi: 10.1088/1742-6596/1007/1/012013.

[9] R. A. Hapsari and A. Purwinarko, "Implementation of Convolutional Neural Network Algorithm Using Vgg-16 Architecture for Image Classification in Facial Images," *Recursive J. Informatics*, vol. 1, no. 2, pp. 83–92, 2023, doi: 10.15294/rji.v1i2.68059.

[10] A. Said, H. Hasbullah, and B. Baharudin, "IMAGE-BASED MODELING : A REVIEW," 2009.

[11] S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 606–619, 2014, doi: 10.1109/TPAMI.2013.147.

[12] S. Wanner, C. Straehle, and B. Goldluecke, "Globally consistent multi-label assignment on the ray space of 4D light fields," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1011–1018, 2013, doi: 10.1109/CVPR.2013.135.

[13] R. A. Farrugia, C. Galea, and C. Guillemot, "Super Resolution of Light Field Images Using Linear Subspace Projection of Patch-Volumes," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 7, pp. 1058–1071, 2017, doi: 10.1109/JSTSP.2017.2747127.

[14] E. Park, "24 . Histogram-based colour image analysis on tourism photography," 2011.

[15] K. Liu, K. Deng, and Y. Jiang, "Dunhuang Decorative Pattern Digital Intelligent Enhancement Algorithm," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 435, no. 1, 2018, doi: 10.1088/1757-899X/435/1/012029.

[16] M. Awais *et al.*, "Novel Framework: Face Feature Selection Algorithm for Neonatal Facial and Related Attributes Recognition," *IEEE Access*, vol. 8, pp. 59100–59113, 2020, doi: 10.1109/ACCESS.2020.2982865.

[17] Y. Yang and F. Fan, "Ancient thangka Buddha face recognition based on the Dlib machine learning library and comparison with secular aesthetics," *Herit. Sci.*, vol. 11, no. 1, pp. 1–16, 2023, doi: 10.1186/s40494-023-00983-8.

[18] N. Soni, E. K. Sharma, and A. Kapoor, "Hybrid meta-heuristic algorithm based deep neural network for face recognition," *J. Comput. Sci.*, vol. 51, no. March, p. 101352, 2021, doi: 10.1016/j.jocs.2021.101352.

[19] W. xia, Y. Lu, S. Wang, Z. Wang, P. Xia, and T. Zhou, "LFMamba: Light Field Image Super-Resolution with State Space Model," pp. 1–13, 2024, [Online]. Available: http://arxiv.org/abs/2406.12463

[20] B. Zhong, C. Qiao, D. Yoo, D. Gong, and Y. Gong, "Analysis of the materials and processes of hanging sculptures in Guanyin Hall," *Herit. Sci.*, vol. 12, no. 1, pp. 1–21, 2024, doi: 10.1186/s40494-023-01112-1.

[21] A. Basu *et al.*, "Digital Restoration of Cultural Heritage With Data-Driven Computing: A Survey," *IEEE Access*, vol. 11, no. June, pp. 53939–53977, 2023, doi: 10.1109/ACCESS.2023.3280639.

[22] H. Knutsson and C.-F. Westin, "Normalized and differential convolution," no. x, pp. 515–523, 2002, doi: 10.1109/cvpr.1993.341081.

[23] K. Schindler, “An overview and comparison of smooth labeling methods for land-cover classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11 PART1, pp. 4534–4545, 2012, doi: 10.1109/TGRS.2012.2192741.

[24] M. Jogin, Mohana, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva, “Feature extraction using convolution neural networks (CNN) and deep learning,” *2018 3rd IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. RTEICT 2018 - Proc.*, pp. 2319–2323, 2018, doi: 10.1109/RTEICT42901.2018.9012507.

[25] Y. Ma, Y. Liu, Q. Xie, S. Xiong, L. Bai, and A. Hu, “A Tibetan Thangka data set and relative tasks,” *Image Vis. Comput.*, vol. 108, p. 104125, 2021, doi: 10.1016/j.imavis.2021.104125.

[26] Y. Qi and F. Zhao, “Saliency-Aware Automatic Buddhas Statue Recognition,” pp. 1–14, 2024, [Online]. Available: http://arxiv.org/abs/2402.16980

[27] L. Marlinda, S. Rustad, R. S. Basuki, F. Budiman, and M. Fatchan, “Matching Images on the Face of A Buddha Statue Using the Scale Invariant Feature Transform (SIFT) Method,” *7th Int. Conf. Inf. Technol. Comput. Electr. Eng. ICITACEE 2020 - Proc.*, pp. 169–172, 2020, doi: 10.1109/ICITACEE50144.2020.9239221.

[28] Y. Qian, C. B. El Vaigh, Y. Nakashima, B. Renoust, H. Nagahara, and Y. Fujioka, “Built Year Prediction from Buddha Face with Heterogeneous Labels,” *SUMAC 2021 - Proc. 3rd Work. Struct. Underst. Multimed. Herit. Contents, co-located with ACM MM 2021*, pp. 5–12, 2021, doi: 10.1145/3475720.3484441.