MyHistory: Automatic Photo Album Creation

Yifei Oo
School of Computer Science and Engineering
Nanyang Technological University
Singapore, Singapore
yoo001@e.ntu.edu.sg

Owen Noel Newton Fernando
School of Computer Science and Engineering
Nanyang Technological University
Singapore, Singapore
ofernando@ntu.edu.sg

Deepu Rajan
School of Computer Science and Engineering
Nanyang Technological University
Singapore, Singapore
asdrajan@ntu.edu

***Abstract*—This paper proposes a method to create customized albums in six different categories, namely buildings, food, drinks, people, pets, and scenery and recommendations for users to remove near duplicate photos. The method comprises two main components, the recommendation system and the classification model. The recommendation system employs criteria such as blur, face, open-eye and smile detection to determine the best photo. The classification model is built with transfer learning by fine-tuning pre-trained networks, namely VGG19 and MobileNetV2. The results showed that both fine-tuned models achieved an accuracy of approximately 97% on the test dataset. This classification system is to help users create their personal albums automatically. The album creation process is streamlined through the classification model, which is then complemented by the recommendation system based on the photo's category. The output of this method is a crafted album within the specified category, enriched by the removal of near-duplicate photos, offering users a personalized and refined album creation experience.**

Keywords*—*Photo Gallery Organization, Photo Album, Blur Detection, Face Detection, Smile Detection, Open-Eye Detection, Gaze Detection

I. Introduction

In today's smartphone-dominated era, capturing moments through photographs has become incredibly convenient. With high-resolution cameras embedded in smartphones, taking pictures has become effortless. As a result, our phone galleries are often flooded with an abundance of photos. While this plethora of old images may evoke nostalgia, organizing and retrieving these memories poses a significant challenge. Despite the automatic sorting system based on date and time built in smartphones' photo galleries, locating specific memories amidst the cluttered gallery remains a time-consuming task. Hence, this paper proposes a method to address the issue of efficient photo organization through an automated classification system.

The proposed method involves classifying images into six distinct categories: buildings, food, drinks, people, pets and scenery. In addition to that, the system employs a recommendation process that identifies similar photos and recommends the best ones to keep while deleting redundant near duplicates. To determine the quality of an image, various criteria are considered, including blur detection, identification of eye status (open or closed), detection of smiling faces and gaze direction, which determine if the person is looking at the camera. Fig. 1 [1] demonstrates the desired output based on some of the criteria. In Fig. 1 (a), (c), (e), three sets of similar photos are presented while in Fig. 1 (b), (d), (f), the recommended photos are showed. The rationales behind the recommendation of these photos are explained in Table 1.

Fig. 1(a). Set 1

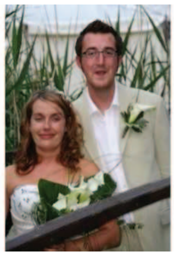

Fig. 1(b). Recommended Photo for Set 1

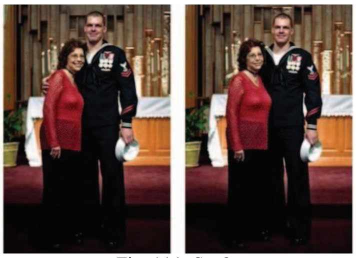

Fig. 1(c). Set 2

Fig. 1(d). Recommended Photo for Set 2

Fig. 1(e). Set 3

Fig. 1(f). Recommended Photo for Set 3

By combining the classification system with the metadata of photos, users can automatically create customizable albums through simple queries. For example, the system could automatically create an album that contains all food images captured during the user's trip in Thailand within a certain date. This would allow users to effortlessly revisit those delightful moments and share them with friends or family. Fig. 2 illustrate the overall flow for generating the distinct albums.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00176

Firstly, the classification model undergoes training. Subsequently, it is deployed to classify images in a smartphone's photo gallery. After the images are classified into the category queried by the user, they undergo further analysis through the recommendation system. Within the recommendation system, images in all the categories undergo blur detection. Images belonging to the "people" category will undergo additional assessment, including smile detection, gaze detection and eye status identification. The primary objective of the recommendation system is to suggest which photos to retain when multiple similar ones exist. The bottom layer of Fig. 2 represents the images allocated to their respective output albums.

TABLE I. RECOMMENDED PHOTOS JUSTIFICATIONS

Set	Justifications
Set 1, Fig. 1 (a)	In the left photo, the girl appears to have her eyes nearly closed, hence the photo on the right is recommended.
Set 2, Fig. 1 (c)	The individuals in the left photo exhibit more brighter smiles compared to those in the right photo, hence the photo on the left is recommended.
Set 4, Fig. 1 (e)	In the left photo, the man is not looking at the camera, hence the photo on the right is recommended.

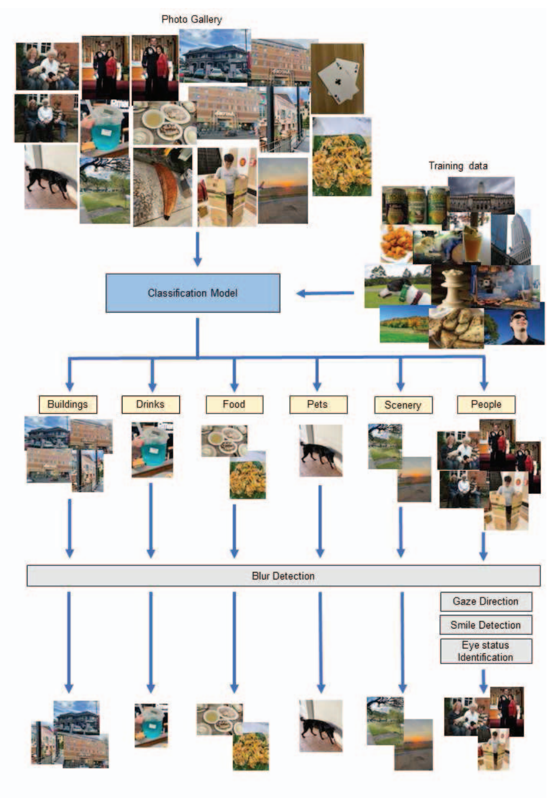

Fig. 2. Overall Flow of Proposed Method

This paper will first delve into the details of the classification model, including the dataset used for training, the models used for fine-tuning and the achieved results. Next, we describe the criteria for evaluating image quality, which are the small components of the recommendation system.

II. RELATED WORK

Over the past few years, researchers have proposed various methods for automating the organization of photo galleries. One such approach is the generation of albums using computer vision [2]. In this proposed method, the authors evaluated the quality of photos based on criteria such as light exposure, blurriness, and the presence of objects. Another method of photo gallery organization involves classifying images into important and unimportant categories [3]. The researchers analyzed random images taken by smartphones and concluded that majority of the images could be classified into five categories, namely camera images, screenshots or scanned documents, quotes, quotes by famous people and computer screenshots. Furthermore, the detection and removal of near-duplicate photos was implemented to assist in photo galleries organization.

In 2020, Savchenko proposed an event recognition method based on sequential grouping of confidence scores and neural attention [4]. The recognition process hinged on the utilization of Convolution Neural Network (CNN) such as MobileNet [5], Inception [6], ResNet [7] and EfficientNet [8]. The results obtained from this research show the feasibility of CNN in classifying real-life photos into their respective albums. Concurrently, [9] proposed a smart album management system with an emphasis on face recognition. This paper addresses the common practice of storing a substantial number of facial images on mobile phones, indicating the need for a system that offers automation in organizing images containing people.

Existing real-world application, such as Google Photos, sort photos based on predefined categories like screenshots, selfies, videos, 360° photos and videos, photo scan and motion photos. These categories lack the consideration of contextual information of the images. On the other hand, Apple Photos offers similar categories as Google Photos, but provide users the ability to search for photos using a wide range of keywords such as food, drink, beach, vehicle, and tree. Additionally, Apple Photos categorizes photos based on their location.

In line with the objective of this research, which is to enable users to reminisce about cherished moments, the focus of this paper will be on classifying images that evoke memorable experiences, such as those captured during trips. Takano et al. [10] proposed a recommendation system that suggests sightseeing places for travelers based on five genres: scenery, nature, activity, architecture, and food. In addition, [11] developed a framework capable of identifying online destination images in six main categories, which were food, accommodation, traffic, sightseeing, shopping, and recreation, each with their respective sub-categories. These categories proposed by the authors serve as a reference for the image categories used in this paper.

Building upon previous research, this paper aims to utilize different approaches to organise photo galleries, that is combining automatic album creation and image quality assessment. The classification model will allow users to create albums automatically based on the contextual information of the photos. Subsequently, a recommendation system will be established to access the quality of images and delete redundant ones. Ultimately, users will be able to create albums based on both the metadata and contextual relevance, facilitating easy access to their photos in the future.

III. Proposed Method

The proposed method consists of two components, the classification model followed by the recommendation system. The classification model allow users to automatically select photos and create albums, while the recommendation system ensures high-quality photos are chosen. In the subsequent sections, an overview of each component will be explained.

A. *Classification Model*

1) Data Collection

To train the classification model, a dataset comprising images from six distinct categories was collected. The six categories are buildings, food, drinks, people, pets, and scenery, which are photos that are most found in one's photo gallery. Most of the images were obtained using the Flickr API with the code in this GitHub repository [12], while some of the images in the people and scenery categories were sourced from ImageNet [13]. To account for images that do not belong to any of the predefined categories, additional random images were downloaded from Kaggle [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29] and classified as "others". The dataset consisted of a total of 13,101 images, with the following distribution across each category: buildings (1,877), food (1,884), drinks (1,888), people (1,878), pets (1,882), scenery (1,842), and others (1,867). The dataset was then split into training, validation, and test sets, with a ratio of 7:2:1 respectively.

2) Data Cleaning

Images downloaded from Flickr required relabeling as some images did not belong to their respective categories. Images with multiple categories, such as those featuring people with food or pets, or people in front of buildings, were removed. Table 2 provides details on the specific images removed from each category.

3) Transfer Learning and Fine-Tuning

After preparing the data for training, transfer learning was employed using two pre-trained networks, VGG19 [30] and MobileNetV2 [31]. Both models had been previously trained on the ImageNet dataset [32]. TensorFlow [33] was used as a framework to implement the code. The optimizer and loss function used during training were Adam and Sparse Categorical Accuracy respectively.

TABLE II. REMOVED IMAGES OF EACH CATEGORY

Category	Removed Imaged
Buildings	-Images taken from the inside of the building -Construction -Close up images of buildings -People's houses
Food	-Fruits -Stalls selling food -Food in packages -Raw ingredients
Drinks	No special cases
Pets	No special cases
People	No special cases
Scenery	-City view -Sea view with boats -Close up image of flowers

4) Results

Both the models achieved an accuracy of 0.97 on the test set. The loss during transfer learning is shown in Fig. 3 and Fig. 4. The blue line represents the training set, there is a significant decrease in the loss. The orange line represents the validation set, where the loss mostly fluctuates. MobileNetV2 is the preferred network due to its comparable accuracy to VGG19 but with a significantly smaller size (size of MobileNetV2 and VGG19 are 14MB and 549MB respectively) [25]. To reduce false positives, a confidence threshold of 0.99 was established. This means that an image will be classified to its respective category only if the model predicts with a confidence level of 0.99 or higher. Fig. 5 shows the graph of precision against recall for different thresholds. Table 3 shows the precision for each category achieved by MobileNetV2 after applying the confidence threshold.

TABLE III. PRECISION FOR EACH CATEGORY ACHIEVED BY MOBILENETV2

Category	Precision
Buildings	0.998
Food	0.997
Drinks	0.997
Pets	0.999
People	0.999
Scenery	0.998

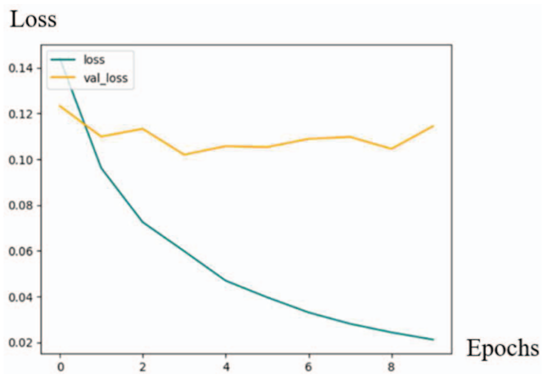

Fig. 3. Loss against Epochs for MobileNetV2

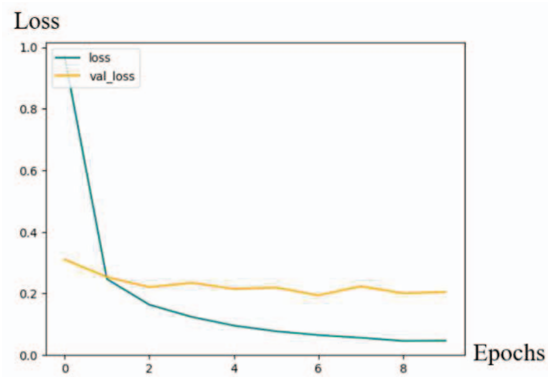

Fig. 4. Loss against Epochs for VGG19

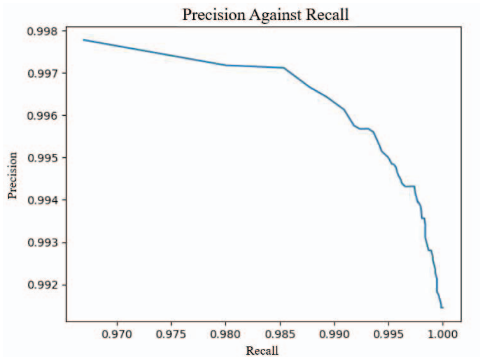

Fig. 5. Graph of Precision Against Recall

B. *Recommendation System*

1) Blur Detection

To detect blur images, the Laplacian operator implemented by the OpenCV [34] library was used. The variance of Laplacian can be used to measure blurriness, where low variance indicates increased blurriness and high variance indicates clearer images [35]. Hence, by comparing the variance of Laplacian, the blurrier images could be determined. However, this method does not consider artistic images that are intentionally blurred as exemplified in Fig. 6.

When tested on dataset [36], the results of this method demonstrated a 0.91 chance of producing a higher Laplacian variance for images that are in fact clearer.

2) Face Detection

To detect faces in images, YOLOv8 from Ultralytics [37] was used. These detected faces are then processed using the dlib library [38] to identify landmarks on the face, facilitating access to the eyes and mouth for subsequent detections.

3) Smile Detection

For smile detection, the approach described in [39] was adopted. This method analyses the geometric properties of facial landmarks and calculates the smile ratio based on (1) to determine the presence of a smile.

$$\text{smile ratio} = \text{lip's width} / \text{jaw's width} \tag{1}$$

The smile ratio is higher when a person smiles, and lower when a person does not smile. Therefore, a threshold is applied to the smile ratio to detect the presence of a smile. While [39] originally set the threshold at 0.32, it was tuned to 0.41 based on the GENKI-4K Subset [40] to achieve higher precision.


Fig. 6. An example of an artistic image intentionally captured blurred [38]

4) Eye Status Idenficifation

To determine if the eyes are open or closed, a metrics called the eye aspect ratio (EAR) [41] was used. EAR is calculated based on the distance between specific facial landmarks as shown in (2).

$$\text{EAR} = (|P2 - P6| + |P3 - P5|) / 2\times|P1 - P4| \tag{2}$$

|P – P'| denotes the distance between the two points P and P', and the landmark locations are illustrated in Fig. 7.

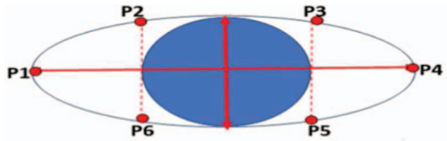

Fig. 7. Landmark locations of an eye [41]

Lower EAR values indicate closed eyes, while higher values indicate open eyes. According to [41], the EAR threshold is set typically set between 0.2 and 0.3. To reduce false positives (i.e., predicting closed eyes when they are open), a thresh-old of 0.2 is set in this study.

5) Gaze Direction

To determine if a person is looking at the camera, an assumption is made that the person's face is oriented towards the camera. With the landmarks located by dlib, the eye positions can be identified. The gaze direction is determined by calculating the area of the white region in the eyes [42]. The steps are as follows: first, the eye is enclosed within a bounding box, and the image within the bounding box is converted to grayscale. Next, a threshold of 70 is applied to change the pixel color values: pixels with a value of 70 or below are set to black (0), while those above 70 are set to white (255). Afterwards, the bounding box is divided into left and right sides, as shown in Fig. 8, and the number of white pixels on each side is computed. Lastly, the gaze ratio is calculated using (3), white_L and white_R denotes the number of white pixels in left area and right area respectively.

$$\text{gaze ratio} = \text{white}_L / \text{white}_R \tag{3}$$

The gaze ratio is calculated for both eyes to obtain the average gaze ratio. The threshold is set such that if the average gaze ratio falls within the range of 0.831 to 1.177, it is inferred that the person is looking at the camera. Based on this threshold, this method achieved a precision of 0.79 on part of the Columbia Gaze Data Set [43].

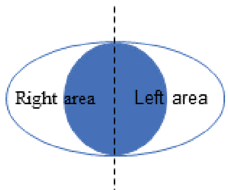

Fig. 8. Left and Right side of an eye

One of the limitations of this method is that the person's face is assumed to be facing the camera. In addition, the feasibility of this method requires further research as eyes may appear small in photos, potentially leading to insufficient white pixels for accurate gaze detection.

6) Evaluation

The assessment of blur detection was conducted using the Blur Dataset [36], comprising 350 sets of similar images. Each set consisted of a clear image, a motion-blurred image, and a defocused-blurred image. In 91% of cases, the clearer image exhibited a higher variance of Laplacian in comparison to the other two blurred image.

The face detection was evaluated on 710 faces. The model performed 695 true positive detections, alongside 15 false positives and 18 false negatives. Given the preference for quality predictions on true values, the precision was adopted as the evaluative metric, yielding a precision score of 0.98.

The smile detection was evaluated based on the GENKI-4K Subset [40], featuring 600 smiling and non-smiling faces each. The methodology employed achieved an precision of 0.82 in distinguishing smiling and non-smiling individuals.

The eye status identification was accessed with the CEW dataset [44], comprising 1192 images of closed eyes and 1231 images of open eyes. Given the priority of correctly predicting closed eyes, precision was used as the evaluation metric, resulting in a precision score of 0.93.

Gaze direction evaluation was conducted on a subset of the Columbia Gaze Data Set. The subset consisted of 315 images with individuals looking straight, left, and right (105 images each). The adopted methodology achieved a precision of 0.79 for the classification of whether the person's gaze was directed straight ahead or towards the sides.

An overview of the evaluation of all the criteria within the recommendation system are shown in Table 4.

TABLE IV. EVALUATION OF CRITERIA OF THE RECOMMENDATION SYSTEM

Criteria	Precision
Blur Detection	0.91
Face Detection	0.98
Smile Detection	0.86
Eye Status Identification	0.93
Gaze Direction	0.71

IV. CONCLUSION

The proposed solution in this paper is to aid users in organizing photo galleries using computer vision. To overcome the challenge of having too many photos, a classification model and a recommendation system is built to automatically sort images according to user's requirement, and then deleting redundant ones. Although tested on few data sets, the feasibility of the recommendation system requires further research on real-world photos. In addition, more criteria could be added to access the quality of an image such as straightness and symmetricity of the image and the presence of hand gestures of people to further improve the system's capabilities. Other than the six categories mentioned, more categories or sub-categories could be added to provide users with more options. In addition to that, photos belonging to multiple categories are not addressed. Face recognition systems could also be integrated for users to create albums based on people. In the future, this proposed method could be developed into real-world applications that can minimize the necessity for manual sorting and save time when organizing photo galleries. It would allow users to interact with their photos and cherish memories like never before.

REFERENCES

[1] A. Gallagher, T. Chen, "Understanding Groups of Images of People," IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[2] T. Souza, J. P. Lima, V. Teichrieb, C. Nascimento, F. Q. da Silva, A. L. Santos, and H. Pinho, "Generating an album with the best media using computer vision," in Design, User Experience, and Usability: Designing Interactions. DUXU 2018, Lecture Notes in Computer Science(), vol 10919, pp. 338-352, 2018. https://doi.org/10.1007/978-3-319-91803-7_25

[3] A. Sachdeva, R. Kapoor, A. Sharma and A. Mishra, "Categorical classification and deletion of spam images on smartphones using image processing and machine learning," 2017 International Conference on Machine Learning and Data Science (MLDS), Noida, India, pp. 23-30, 2017. 10.1109/MLDS.2017.10.

[4] A. Savchenko, "Event recognition with automatic album detection based on sequential grouping of confidence scores and neural attention'" in 2020 International Joint Conference On Neural Networks (IJCNN) 2020 Jul 19 (pp. 1-8). IEEE.

[5] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017. arXiv preprint arXiv:1704.04861.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9. 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[8] M. Tan, and Q. Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In International conference on machine learning, pp. 6105-6114. PMLR, 2019.

[9] Z. Feng, W. Liu, Y. Yu, "Smart Album Management System Based on SE-ResNeXt," Journal of ICT Standardization. 2022 Dec 2:563-82.

[10] K. Takano, S. Hussaya, I.O. Nanyakorn, E. Shibamoto, and X. Chen, "Photo sharing service for recommending sightseeing places," Information Modelling and Knowledge Bases XXIX, vol 301, pp. 409-419, 2018. 10.3233/978-1-61499-834-1-409

[11] R. Wang, J. Luo, S. (Sam) Huang, "Developing an artificial intelligence framework for online destination image photos identification," Journal of Destination Marketing & Management, vol 18, 2020, 100512, ISSN 2212-571X. https://doi.org/10.1016/j.jdmm.2020.100512.

[12] J. Heaton, "pyimgdata", GitHub Repository, 2020; https://github.com/jeffheaton/pyimgdata

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE conference on computer vision and pattern recognition (CVPR), pp. 248–255, 2009.

[14] A. Jain, "Bike Dataset," [Data set], Kaggle, n.d., retrieved May 20, 2023 from https://www.kaggle.com/datasets/ayushjain001/bike-dataset?select=bike

[15] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[16] A. Mehta, "Chess pieces detection images dataset, version 31," [Data set], Kaggle, n.d., retrieved May 20, 2023 from https://www.kaggle.com/datasets/anshulmehtakaggl/chess-pieces-detection-images-dataset

[17] A. Nagaraj, "ASL Alphabet," [Data set], Kaggle, 2018, retrieved May 20, 2023, https://doi.org/10.34740/KAGGLE/DSV/29550

[18] A. W. Harley, A. Ufkes, K. G. Derpanis, "Evaluation of Deep Convolutional Nets for Document Image Classification and Retrieval," in ICDAR, 2015

[19] Cchang, "Garbage classification," [Data set], Kaggle, 2018, retrieved May 20, 2023, https://doi.org/10.34740/KAGGLE/DS/81794

[20] C. Dieckhaus, "QR codes, version 1," [Data set], Kaggle, n.d., retrieved May 20, 2023 from https://www.kaggle.com/datasets/coledie/qr-codes

[21] DeepNets, "Shoes Classification Dataset | 13k Images |, version 2," [Data set], Kaggle, n.d., retrieved May 20, 2023 from https://www.kaggle.com/datasets/utkarshsaxenadn/shoes-classification-dataset-13k-images

[22] Gerry, "Headgear 20 classes-image classification, version 1," [Data set], Kaggle, n.d., retrieved May 20, 2023 from https://www.kaggle.com/datasets/gpiosenka/headgear-image-classification

[23] J. Krause, M. Stark, J. Deng, F.-F. Li, "3D object representations for fine-grained categorization," in 4th IEEE Workshop on 3D Representation and Recognition, at ICCV 2013 (3dRR-13), Sydney, Australia, Dec. 8, 2013.

[24] K. Kannan, "5 flower types classification dataset," [Data set], Kaggle, n.d., retrieved May 20, 2023 from https://www.kaggle.com/datasets/kausthubkannan/5-flower-types-classification-dataset

[25] Makra, "1k random shapes with random operations, version 1," [Data set], Kaggle, n.d., retrieved May 20, 2023 from https://www.kaggle.com/datasets/makra2077/1000-random-shapes-with-random-operations

[26] N. Chakrabarty, "Brain MRI Images for Brain Tumor Detection, version 1," [Data set], Kaggle, n.d., retrieved May 20, 2023 from https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection

[27] Makra, "1k random shapes with random operations, version 1," [Data set], Kaggle, n.d., retrieved May 20, 2023 from https://www.kaggle.com/datasets/makra2077/1000-random-shapes-with-random-operations

[28] S. Sakharovskiy, "U.S. coins," [Data set], Kaggle, 2023, retrieved May 20, 2023. https://doi.org/10.34740/KAGGLE/DSV/5343400

[29] V. Lanz, "Insect village synthetic dataset," [Data set], Kaggle, 2022, retrieved May 20, 2023, https://doi.org/10.34740/KAGGLE/DSV/4111904

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014. https://doi.org/10.48550/arXiv.1409.1556

[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510-4520, 2018. https://doi.org/10.48550/arXiv.1801.04381

[32] Keras, "Keras Applications," n.d.; https://keras.io/api/applications/

[33] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems", 2015. Software available from tensorflow.org.

[34] G. Bradski, "The OpenCV Library", Dr. Dobb's Journal of Software Tools, 2000

[35] R. Bansal, G. Raj and T. Choudhury, "Blur image detection using Laplacian operator and Open-CV," 2016 International Conference System Modeling & Advancement in Research Trends (SMART), Moradabad, India, pp. 63-67, 2016. 10.1109/SYSMART.2016.7894491.

[36] A. Alekseev, "Blur dataset, version 2," [Data set], Kaggle, n.d., retrieved June 12, 2023 from https://www.kaggle.com/datasets/kwentar/blur-dataset

[37] G. Jocher, A. Chaurasia and J. Qiu, "Ultralytics YOLOv8, version 8.0.0," 2023; https://github.com/ultralytics/ultralytics

[38] D. E. King, "Dlib-ml: A Machine Learning Toolkit", Journal of Machine Learning Research 10, pp. 1755-1758, 2009

[39] F. Cunha, "Basic Smile Detection Using OpenCV and DLIB," in Towards Data Science, Aug 10, 2021. https://towardsdatascience.com/basic-smile-detection-using-opencv-and-dlib-aeb22afb9e67

[40] http://mplab.ucsd.edu, "The MPLab GENKI Database, GENKI-4K Subset."

[41] C. Dewi, R. Chen, X. Jiang, H. Yu, "Adjusting eye aspect ratio for strong eye blink detection based on facial landmarks," PeerJ Computer Science 8:e943, 2022, https://doi.org/10.7717/peerj-cs.943

[42] S. Canu, "Eye blinking detection – gaze controlled keyboard with Python and Opencv p.2," in Pysource, January 2019; https://pysource.com/2019/01/10/eye-blinking-detection-gaze-controlled-keyboard-with-python-and-opencv-p-2/

[43] B.A. Smith, Q. Yin, S.K. Feiner, S.K. Nayar "Gaze Locking: Passive Eye Contact Detection for Human?Object Interaction," ACM Symposium on User Interface Software and Technology (UIST), pp. 271-280, Oct, 2013.

[44] F.Song, X.Tan, X.Liu and S.Chen, "Eyes Closeness Detection from Still Images with Multi-scale Histograms of Principal Oriented Gradients," Pattern Recognition, 2014.