International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Development of Android Application for Facial Age Group Classification Using TensorFlow Lite

Gilbert George*[1], Steve Adeshina[2], Moussa Mahamat Boukar [3]

Submitted: 05/05/2023 **Revised**: 11/07/2023 **Accepted**: 03/08/2023

Abstract: The concept behind the face age classification system is that every person has a distinct ethnicity and face. An individual's face has numerous distinctive structures and traits, much like their fingerprint. The task of determining facial age is difficult. Systems for facial recognition must function with extreme precision and accuracy, enable lightweight, portable devices, and be user-friendly. In comparison to a situation where just one photograph of each person is saved in the database, images captured while accounting for changing facial expressions or lighting circumstances allow the system to be more precise and accurate. The complete process of creating an Android mobile application for categorizing people as either adults, teenagers, or children is described and explained in full in this article, depending on the traits of their faces. Both the development tools and face classification techniques that have been employed in the creation of Android mobile applications are discussed and explained. The software solution explains the specifics of utilizing the OpenCV library and uses photos to display the actual outcomes of the mobile application.

Keywords: *Android, age group classification, convolution neural network, deep learning, still image*

1. Introduction

One of the most exciting and significant areas of research over the past ten years has been facial age detection systems. Several factors stem from the demand for automatic facial age detection which is important in areas, such as security and access control systems based on age detection, as well as human-computer interaction and disciplines, including medicine, computer vision, pattern recognition, image processing, and machine learning, are included in the studies. The incorporation of technology into numerous facets of our lives in the modern digital age has produced amazing developments. The usage of Android mobile applications for age prediction is one such fascinating use. This ground-breaking method uses the strength of deep learning, picture analysis, and the pervasiveness of smartphones to precisely estimate a person's age group based on facial traits. The exciting field of age prediction utilizing Android mobile applications is examined in this paper, along with its potential ramifications and advantages. The OpenCV library will be used in this study to investigate the feasibility of integrating a facial age classification system on a mobile device on the Android operating system.

The Mechanism of Age Prediction

Deep learning and computer vision are the foundations of age prediction [1], [2] utilizing Android mobile applications. These programs make use of deep learning algorithms that have been developed using enormous datasets of facial photos and age labels. The algorithms learn complex patterns and traits that alter as a person age during this training process.

1.1. Applications and Consequences

Entertainment and social media: Android age prediction apps are frequently marketed as amusing tools that provide users with a glimpse of who they might become in the future. These apps are increasingly popular on social media sites because they let users share their aged images with friends and followers, sparking trends and amusing conversations.

Industry of cosmetics and skincare: The technology underpinning age prediction apps has potential uses in the field of cosmetics and skincare. These resources can help consumers make decisions by helping them visualize the possible effects of various items and therapies.

Age prediction applications, while largely used for entertainment, could spark conversations about health and fitness. Users who receive forecasts that are noticeably older than their real age may think about altering their way of life to look younger.

Privacy and ethical issues: Privacy issues come up with any technology that uses personal data. Age prediction app creators must place a high priority on user data protection and guarantee open data usage.

Applications in Research and Medicine: Age prediction algorithms may one day be used in studies that are related to medicine. Analysing face changes over time may help

[1] *Department of Computer Science, Baze University, Abuja, Nigeria*
ORCID ID: 0000-0003-3080-5646
[2] *Department of Engineering, Nile University, Abuja, Nigeria.*
ORCID ID: 0000-0003-0405-5589
[3] *Department of Computer Science, Nile University, Abuja, Nigeria.*
ORCID ID: 0000-0001-8287-6257
** Corresponding Author Email: gilbertgeorge007@gmail.com*

with health research and offer insights into the ageing process.

2. Literature Review

With the use traditional machine learning techniques frequently include the manual definition of features, which might not always be the best representations of the challenges at hand. However, CNN and other deep neural networks for image identification can only make modest assumptions. Through a training process, the best network configuration is discovered using processed information. A unique kind of feed-forward network called a convolutional neural network (CNN or ConvNet) is employed mostly to assess visual imagery [4]. Similar to regular neural networks, convolutional neural networks are composed of neurons with biases and weights that can be learned. Because of their distinct methodology, ConvNets outperform the other deep neural network architectures. ConvNets connect several pixels rather than examining each pixel individually so they can comprehend a larger picture.

Open Computer Vision

A popular open-source package called OpenCV (Open-Source Computer Vision) [3], [4] offers a wide range of tools and functions for computer vision jobs. In modern technology's backbone, computer vision enables machines to analyse and comprehend visual data from the outside environment [5]–[8]. The sophisticated library OpenCV, created by Intel, provides several methods to make image and video analysis easier. In this essay, the contribution of OpenCV to the development of computer vision is examined. The following are some of the core features and functionality offered by OpenCV:

image manipulation, object detection, feature extraction, and machine learning integration. Because it supports numerous programming languages, including Python [8], C++, and Java, it is usable by a wide variety of developers. The extensive set of algorithms and tools provided by OpenCV serves as a base for creating reliable computer vision applications.

Image Preprocessing, Transformation, and Enhancement: OpenCV is a master at handling images. OpenCV [9], [10] makes difficult image manipulation jobs simpler, from elementary operations like resizing and cropping to sophisticated methods like image filtering, thresholding, and morphological operations.

Image stitching, panorama construction [11], and object tracking are all made possible by OpenCV's feature extraction and matching tools, which are available to academics and engineers. Scale-Invariant Feature Transform (SIFT) [12]and Oriented FAST and Rotated BRIEF (ORB) are two algorithms that improve the accuracy and resilience of feature-based applications.

OpenCV interfaces with popular machine learning frameworks like TensorFlow and PyTorch without a hitch, making it easier to create complex computer vision models [13], [14]. Modern deep learning models may now be implemented and used by practitioners for a variety of applications, including semantic segmentation and image classification.

Object detection and recognition: Support for object detection and recognition [5], [15], [16] is one of OpenCV's unique characteristics. The library offers pre-trained models for well-liked object detection algorithms, such as deep learning-based techniques and Haar cascades. Because of OpenCV's flexibility, programmers may include these models in a variety of applications, including those that recognize faces, pedestrians, and more.

Applications: Robotics, healthcare, autonomous vehicles, augmented reality, and surveillance are just a few of the industries where [17], [18] OpenCV is used. It is a cornerstone technology for the Fourth Industrial Revolution since it is crucial in enabling machines to receive and analyse visuals.

Related Works

This section examines some of the research on CNN-based automatic age and gender prediction based on facial photos.

This paper's authors [19] attempted to categorize human age and gender at both the coarser and finer levels. To classify data at a finer level, a 3-sigma control limit final classification is utilized instead of a feed-forward propagation neural network. Children, middle-aged adults, and elderly adults are included in the classification of age groups. The experiment's excellent efficiency is proved using benchmark database photos.

In this work [20], there are two types of testing: file-based testing and camera-based testing [6]. When testing is done using files, RGB, grayscale, or histogram equalized grayscale images are employed, and the RGB photos produce results that are more accurate than those of the other images. While RGB photos perform poorly in camera-based tests, CNN trained on grayscale images performs better.

In this research [2], a convolutional neural network (CNN)-based technique for predicting age and gender from facial photos is proposed. The IMDB-WIKI dataset, which contains information such as acquisition data, DOB, gender, and face detector face score, was employed in this study. Age estimation is aided by regression, while gender prediction is accomplished through categorization.

The convolutional neural network (CNN) technique was used in the research [21]as a platform for face detection and gender classification. The authors use an IMDB dataset. The TensorFlow Framework and Keras are used to implement the neural network successfully.

According to the paper [9], working with a small training set yields better results. This paper proposes the Feature Extraction based Face Recognition, Gender and Age Classification (FEBFRGAC) method. Based on facial image analysis, the Posteriori class probability classifier aids in determining a person's gender and the ANN model aids in determining their age.

In this work[22], the divide-and-rule strategy is utilized to forecast the age estimation using the ageing network (AgeNet) approach, which builds an age-estimated deep CNN based on classification and regression.

The algorithm used in the study [23] for video-based age and gender recognition with a specially trained variant of the Mobile Net convolution network with two outputs, the authors also compared existing neural-net models. For such data collectors as Kinect, IJB-A, Indian Movie, and EmotiW, the experimental findings are provided. The strategy enables the boost the age and gender recognition accuracy by 2-5% and 5-10%, respectively, in comparison to other standard methods.

In [24] this study, keystroke dynamics data from 50 people were collected on an Android smartphone using open-source data tools. From the raw data, 21 frequently utilized keystroke dynamics features were selected. While the remaining data was used for classification, the acquired data was used to train a Random Forest algorithm using four different training sample sizes. The method was then used to rank 21 various keystroke dynamics features in order of relevance. The findings demonstrated that each variable has a different level of significance in predicting age group and gender. Although research on predictive keystroke dynamics has been done using computer keyboards, literature on the same topic utilizing touchscreen smartphone virtual keyboards has also been published.

3. Methodology

We trained our model using Python and then converted it into a tensor flow lite format which can be used by Android devices.

We then created an application using Android, which reads images from the camera and then sends them to the model for classification. The suggested age detection workflow is depicted in Fig. 1. And Fig.2.

The phone camera begins to capture video images from the Android camera into the system's input is the first step. In the frame selection block, isolated frames are chosen at a set rate (about 10–20 times per second) from the video stream. The face region that is executed in the appropriate block must then be fixed, leaving nothing else.

The following step is reducing to a single scale all of the received images of a person on a single frame. Additionally, it is frequently utilized to subtract the mean image (image mean subtraction) from each image of a face [14]. The recognition of each frame using the CNN model comes in next. As a result, the SoftMax layer (1) is used to derive the estimations of posterior probabilities.

The functionality of the TensorFlow library is used to do recognition. A final recognition solution is implemented in favour of the corresponding class based on the data that is the output of the classifier fully connected block.

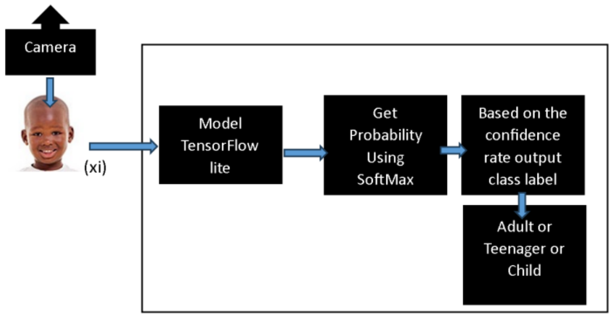

Fig.1. Shows our methodology for the application

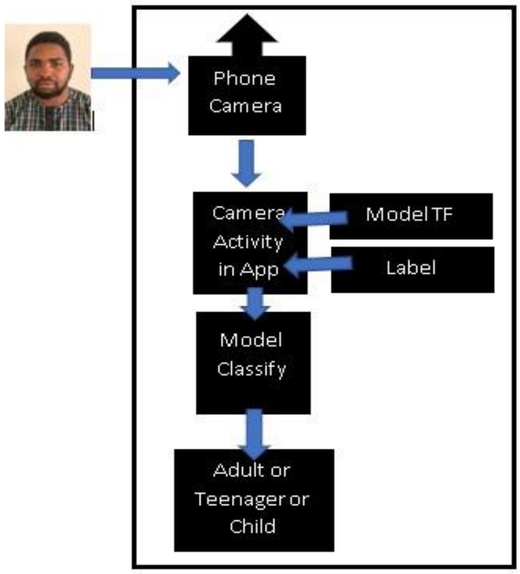

Fig. 2. Shows the model being loaded to the phone for use.

3.1. Dataset

CASIA African Facial Dataset

We made use of The CASIA African Facial database and the images in the database were taken in Kaduna, Nigeria, which is an African country. Approximately 1150 individuals took part in the practice of capture. The dataset has different ethnic Nigeria tribes, for this experiment, we made use of the Hausa ethnic group. The images in the database comprise a total of 38,546 images from 1,183 subjects. We made use of a total of 5392 facial images distributed to ages 10-20, table 1 shows the age distribution of the dataset.

Table 1 shows the dataset instances

Age	Amount
10-12	92
13-17	1805
18-20	3587
Total	5392

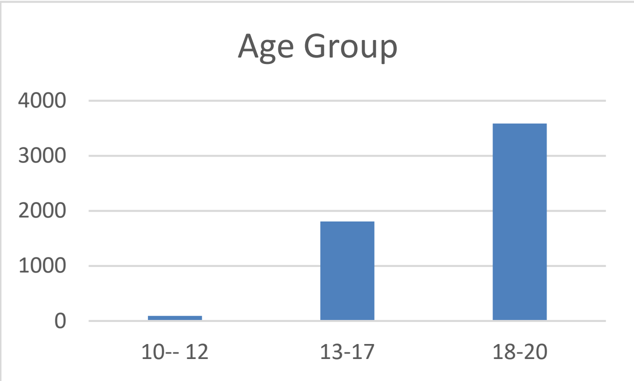

Fig.3. Shows the graphical illustration of the dataset used.

3.2. Experimental Setup

Using Jupyter Notebook, we put the suggested strategy into practice. Intel 12 gen I-core 7 with 10 cores has been used for training the networks. For the learning rate was set to 0.001 and the momentum to 0.9. We used SoftMax for the last layer for classification and accuracy as the performance metric.

We also choose the Android studio as our IDE for the development of the application.

3.3. Training

The training was carried out using an image size of 5392 images, divided into an 85:15 ratio, 85% was used for training and 15% was used for testing. We passed an input image size of 224*224 which will be passed to the convolutional layer for feature extraction and then the features extracted will be passed to the fully connected layers for classification. The last output had 3 nodes for the three class groups (adult, teenager and child).

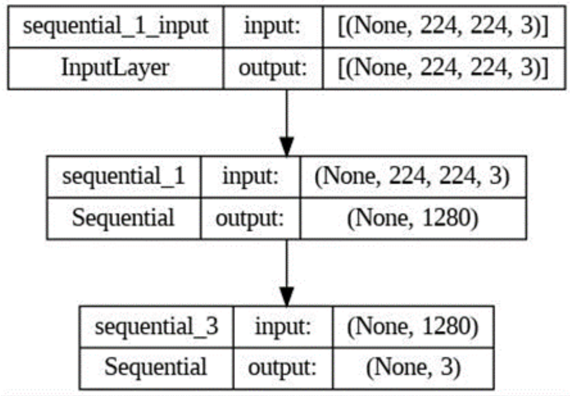

Fig.4. Shows the graphical illustration of our custom CNN model used.

3.4. Facial Age Group Classification

After training the model, the model is now converted from the keras h2 file to the TensorFlow Lite model and the label is set into a text file.The model.ts file and the label.txt file are then put into the asset folder in the Android application, the classifierqunatizedmobilenet class is responsible for loading the dataset from the asset folder using the codes in fig.3.

The image is then captured in real-time using the device's camera, and then passed into the image Shape array for the tensors feature and inference on the image is done using the recognize Image method, from which the class will be determined based on the outputted probabilistic score which is between 0-1 for each age group based on the image captured.

```
public ClassifierQuantizedMobileNet(Activity activity, Device device, int numThreads)
    throws IOException {
  super(activity, device, numThreads);
}

1 usage  rafi *
@Override
protected String getModelPath() {
  // you can download this file from
  // see build.gradle for where to obtain this file. It should be auto
  // downloaded into assets.
  return "model.tflite";
}

1 usage  rafi *
@Override
protected String getLabelPath() { return "labels.txt"; }
```

Fig.5. shows the code for loading the model to the application

We then declared a unique id for the class labels to be predicted (adult, teenager and child), and a name for the class, using confidence.

```java
// Reads type and shape of input and output tensors, respectively.
int imageTensorIndex = 0;
int[] imageShape = tflite.getInputTensor(imageTensorIndex).shape(); // {1, height, width, 3}
imageSizeY = imageShape[1];
imageSizeX = imageShape[2];
DataType imageDataType = tflite.getInputTensor(imageTensorIndex).dataType();
int probabilityTensorIndex = 0;
int[] probabilityShape =
    tflite.getOutputTensor(probabilityTensorIndex).shape(); // {1, NUM_CLASSES}
DataType probabilityDataType = tflite.getOutputTensor(probabilityTensorIndex).dataType();
```

Fig.6 shows the image transformation to tensors for the model classification

The code above reads the image size and shape and converts it into a tensor flow array for the classification process.

```java
/** Runs inference and returns the classification results. */
public List<Recognition> recognizeImage(final Bitmap bitmap, int sensorOrientation) {
  // Logs this method so that it can be analyzed with systrace.
  Trace.beginSection("recognizeImage");

  Trace.beginSection("loadImage");
  long startTimeForLoadImage = SystemClock.uptimeMillis();
  inputImageBuffer = loadImage(bitmap, sensorOrientation);
  long endTimeForLoadImage = SystemClock.uptimeMillis();
  Trace.endSection();
  LOGGER.v("Timecost to load the image: " + (endTimeForLoadImage - startTimeForLoadImage));

  // Runs the inference call.
  Trace.beginSection("runInference");
  long startTimeForReference = SystemClock.uptimeMillis();
  tflite.run(inputImageBuffer.getBuffer(), outputProbabilityBuffer.getBuffer().rewind());
  long endTimeForReference = SystemClock.uptimeMillis();
  Trace.endSection();
  LOGGER.v("Timecost to run model inference: " + (endTimeForReference - startTimeForReference));

  // Gets the map of label and probability.
  Map<String, Float> labeledProbability =
      new TensorLabel(labels, probabilityProcessor.process(outputProbabilityBuffer))
          .getMapWithFloatValue();
  Trace.endSection();

  // Gets top-k results.
  return getTopKProbability(labeledProbability);
}
```

Fig.7. shows the codes responsible for classification based on the confidence score

4. Results

4.1. Model Performance

In the training and validation process, the model performed very well with an accuracy of 98%, the confusion matrix has shown the model correctly classified the child age group and miscalculated 6 teenagers as adults and two adults as teenagers.

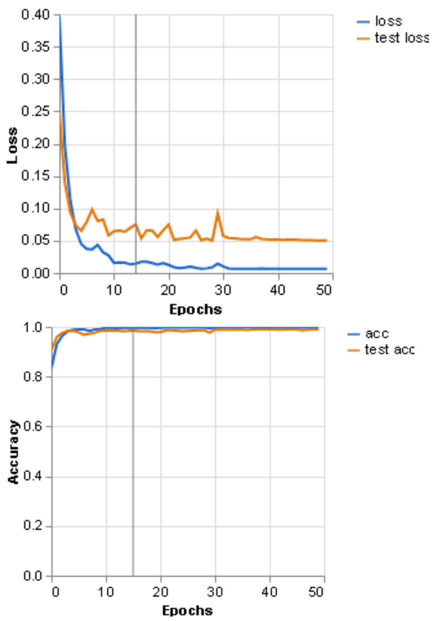

Fig.7. shows the training and loss per epoch on the training and testing

We can see the model performed very well in the training process we obtained an accuracy of 98%.

The fig.8. shows the confusion matrix and the classification per class label in the dataset

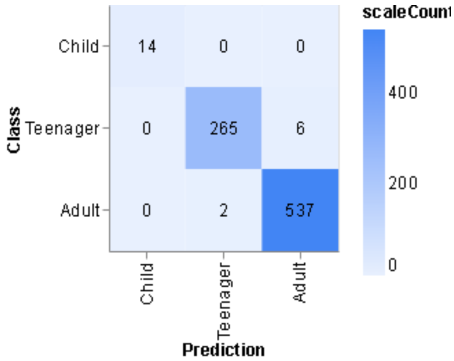

Fig.8. shows the confusion matrix

From the confusion matrix, we can see that the model predicted all the children correctly, it predicted 265 teenagers correctly and 6 as adults given the 'closeness' of teenagers to adults it is easy to mistake them even for humans. It classified the 537 adults correctly and two adults as teenagers.

Accuracy per class

CLASS	ACCURACY	# SAMPLES
Child	1.00	14
Teenager	0.98	271
Adult	1.00	539

Fig.9. shows the accuracy by class

From the above fig. we see the model had an accuracy of 98% for teenagers.

4.2. Testing on an Android device

We ran the application on an Android emulator using pixel XL and the results are presented below.

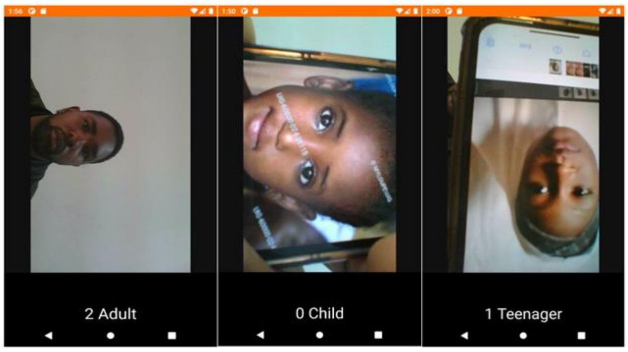

Fig.10. shows the application classifying in Realtime using emulator.

We also tested the application on a real android device using tecno f1 as seen in the figure .7. the application accurately predicted new test data.

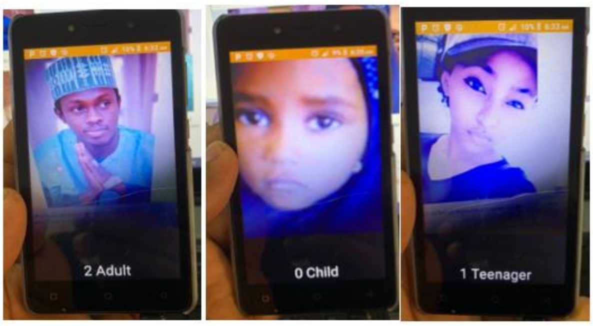

Fig.11. shows the application classifying in Realtime using emulator.

5. Conclusion

This study looked at how to classify a person's age group automatically from real-time face photos using a mobile device in real-time. Face age classification was performed using the TensorFlow lite model on an android phone for classifying age groups. The output layer for age group classification network displays three classes, namely "Adult", "Teenager" and "child" with the overall accuracy provided by the method was 98.89%. The use of Android mobile apps for age prediction exemplifies how technology and human curiosity have come together. These apps offer an interesting and approachable view into the future, despite the fact that their veracity and scientific foundation may differ. It will be fascinating to observe the continued advancement and potential applications of age prediction technology beyond its current entertainment value as the discipline of computer vision and machine learning continues to improve.

Future works, we would also like to encourage age prediction as a continuous value (age:21), emotion detection and sentiment analysis to this work.

References

[1] Singh, N. Rai, P. Sharma, P. Nagrath, and R. Jain, "Age, Gender Prediction and Emotion recognition using Convolutional Neural Network." [Online]. Available: https://ssrn.com/abstract=3833759

[2] K. Ito, H. Kawai, T. Okano, T. A.-2018 A.-P. S. and, and undefined 2018, "Age and gender prediction from face images using convolutional neural network," *ieeexplore.ieee.orgK Ito, H Kawai, T Okano, T Aoki2018 Asia-Pacific Signal and Information Processing Association, 2018•ieeexplore.ieee.org*, Accessed: Jul. 22, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8659655/

[3] S. Suwarno and K. Kevin, "Analysis of Face Recognition Algorithm: Dlib and OpenCV," *JOURNAL OF INFORMATICS AND TELECOMMUNICATION ENGINEERING*, vol. 4, no. 1, 2020, doi: 10.31289/jite.v4i1.3865.

[4] Salihbašić, T. O.-2019 42nd International, and undefined 2019, "Development of android application for gender, age and face recognition using opencv," *ieeexplore.ieee.org*, Accessed: Jun. 26, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8756700/

[5] Jabir, N. Falih, and K. Rahmani, "Accuracy and Efficiency Comparison of Object Detection Open-Source Models," *International journal of online and biomedical engineering*, vol. 17, no. 5, 2021, doi: 10.3991/ijoe.v17i05.21833.

[6] Malakar, A. Kumar, and S. Majumdar, "Detection of face mask in real-time using convolutional neural networks and open-CV," in *2021 2nd International Conference for Emerging Technology, INCET 2021*, 2021. doi: 10.1109/INCET51464.2021.9456415.

[7] Thakur, A. Prakash, A. K. Mishra, A. Goldar, and A. Sonkar, "Facial recognition with open Cv," in *Advances in Intelligent Systems and Computing*, 2020. doi: 10.1007/978-3-030-37218-7_24.

[8] S. More, S. Debbarma, N. Kandpal, and V. Singh, "Open CV Python Autonomous Car," *International Research Journal of Engineering and Technology*, 2019.

[9] E. Widodo, K. Adi, and R. Gernowo, "Medical image processing using python and open cv," in *Journal of Physics: Conference Series*, 2020. doi: 10.1088/1742-6596/1524/1/012003.

[10] S. Mohammed Shariff, R. Bhatia, R. Kuma, and S. Jha, "Vehicle Number Plate Detection Using Python and Open CV," in *2021 International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2021*, 2021. doi: 10.1109/ICACITE51222.2021.9404556.

[11] J. M. Lopez-Guede, A. Izquierdo, J. Estevez, and M. Graña, "Active learning for road lane landmark inventory with V-ELM in highly uncontrolled image capture conditions," *Neurocomputing*, vol. 438, 2021, doi: 10.1016/j.neucom.2020.07.151.

[12] S. Roopashree, J. Anitha, T. R. Mahesh, V. Vinoth Kumar, W. Viriyasitavat, and A. Kaur, "An IoT based authentication system for therapeutic herbs measured

by local descriptors using machine learning approach," *Measurement (Lond)*, vol. 200, 2022, doi: 10.1016/j.measurement.2022.111484.

[13] R. David *et al.*, "TensorFlow Lite Micro: Embedded Machine Learning for TinyML Systems," *Proceedings of Machine Learning and Systems*, vol. 3, pp. 800–811, Mar. 2021.

[14] "Real time face recognition with Android + TensorFlow Lite | by esteban uri | Medium." https://medium.com/@estebanuri/real-time-face-recognition-with-android-tensorflow-lite-14e9c6cc53a5 (accessed Jun. 26, 2023).

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", Accessed: Jun. 11, 2022. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/results

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5)", Accessed: Jun. 11, 2022. [Online]. Available: http://www.cs.berkeley.edu/˜rbg/rcnn.

[17] Saxena, P. S. Vamsi, C. V Eedala, and G. K. K, "Automated Object Identifing and Sorting Robot," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 11, 2022, doi: 10.22214/ijraset.2022.47594.

[18] A. Khan, A. A. Laghari, and S. A. Awan, "Machine Learning in Computer Vision: A Review," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 8, no. 32, 2021, doi: 10.4108/eai.21-4-2021.169418.

[19] P. Thukral, K. Mitra, and R. Chellappa, "A hierarchical approach for human age estimation," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 1529–1532, 2012, doi: 10.1109/ICASSP.2012.6288182.

[20] K. Jhang, … J. C. A. I. in I. and, and undefined 2019, "CNN training for face photo based gender and age group prediction with camera," *ieeexplore.ieee.orgK Jhang, J Cho2019 International Conference on Artificial Intelligence in, 2019•ieeexplore.ieee.org*, doi: 10.1109/ICAIIC.2019.8669039.

[21] J. Prajapati, A. Patel, and P. Raninga, "Facial Age Group Classification," *IOSR Journal of Electronics and Communication Engineering*, vol. 9, no. 1, pp. 33–39, 2014, doi: 10.9790/2834-09123339.

[22] H. Liao, Y. Yan, W. Dai, P. F.-M. P. in Engineering, and undefined 2018, "Age estimation of face images based on CNN and divide-and-rule strategy," *hindawi.comH Liao, Y Yan, W Dai, P FanMathematical Problems in Engineering, 2018•hindawi.com*, Accessed: Jul. 22, 2023. [Online]. Available: https://www.hindawi.com/journals/mpe/2018/1712686/

[23] S. Kharchevnikova and A. V. Savchenko, "Neural Networks in Video-Based Age and Gender Recognition on Mobile Platforms," *Optical Memory and Neural Networks (Information Optics)*, vol. 27, no. 4, 2018, doi: 10.3103/S1060992X18040021.

[24] O. Olasupo and A. O. Adesina, "Predicting Age Group and Gender of Smartphone Users Using Keystroke Biometrics," *Malaysian Journal of Science and Advanced Technology*, 2021, doi: 10.56532/mjsat.v1i4.24.

[25] Gummadi, A. ., & Rao, K. R. . (2023). EECLA: A Novel Clustering Model for Improvement of Localization and Energy Efficient Routing Protocols in Vehicle Tracking Using Wireless Sensor Networks. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2s), 188–197. https://doi.org/10.17762/ijritcc.v11i2s.6044

[26] Russo, L., Kamińska, K., Christensen, M., Martínez, L., & Costa, A. Machine Learning for Real-Time Decision Support in Engineering Operations. Kuwait Journal of Machine Learning, 1(2). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/117

[27] Veeraiah, V., Pankajam, A., Vashishtha, E., Dhabliya, D., Karthikeyan, P., & Chandan, R. R. (2022). Efficient COVID-19 identification using deep learning for IoT. Paper presented at the Proceedings of 5th International Conference on Contemporary Computing and Informatics, IC3I 2022, 128-133. doi:10.1109/IC3I56241.2022.10073443 Retrieved from www.scopus.com