American International Journal of Sciences and Engineering Research
Vol. 4, No. 1; 2021
ISSN 2641-0303   E- ISSN 2641-0311
Published by American Center of Science and Education, USA

MACHINE LEARNING BASED CLOUD MUSIC APPLICATION WITH FACIAL RECOGNITION USING ANDROID STUDIO (MUSYNC)

Maheen Zofishan
M.Phil
Computer Sciences Department
Institute of Southern Punjab Multan, Pakistan
E-mail: maheenzofishan@gmail.com

K. M. Anwarul Islam
Associate Professor
Department of Business Administration
The Millennium University, Dhaka, Bangladesh
E-mail: anwarul@themillenniumuniversity.edu.bd

Farisa Ghazal
M.Phil
Department of Economics & Management Sciences
Minhaj University, Lahore, Punjab, Pakistan
E-mail: farisa.ghazal@gmail.com

ABSTRACT

This paper output is a music player application but when it comes to its features it will be way more than a simple music player. It is developed on Android Studio and other tools like: Firebase is used as database, Android phone camera, Music library of Android Phone are used in the development of application. When user changes his phone or reset his phone then all of his data is lost or user has to put all the data in his computer and then back to his mobile phone except data that is backed up online. Message data, photos and contacts are that things that users backed up online. But music files normally don't get backed up and user troubles in re downloading the files or moving files in computer and back to phone. In this purposed work the targeted problem is resolved as MUSYNC application is be able to automatically backup all the mp3 data from the phone and user will get all of his data by just signing in the application in his new phone. The purposed application has a feature of sync music. Users can sync music with another one and that person will able to listen to same music instantly. Application also provides a unique feature of mood detection using digital image processing DIP. This feature is able to check your face emotion and play music according to it. User just has to take a picture and that is it, this music player plays the music according to your mood. This feature is useful when user having tough time what to listen.

Keywords: Android Studio, Sync Music, MUSYNC, MP3, DIP, Digital Image Processing, Music Player.

INTRODUCTION

In other music player applications, as it takes a lot of time to remember name of each and every song when you lost the music data, Chiu and Ko (2017). This application back up the music files that you want to back up on cloud and you don't need to re-download them. This application is able to solve a problem of backing up and restoring your data automatically, that you do manually, Nathan and Kannan (2017). When it comes to music and friends, another problem also rises i.e., when two persons want to listen to same music at the same time on hands-frees. Normally you have to share one hand-free or buy small hardware that connect two headphones with a device and it's a drawback, Shanthamallu et al. (2017).

With the following scope I have finalized the idea and the functionalities of my proposed work that is going to make it unique and different from the other existing systems and will remove the deficiencies already present in the available and current systems, Marr (2019).

OBJECTIVE OF THE STUDY

In smart phones, the music player application wasn't able to back up the music. It means whenever you change your device, the user has to re-download the music and it takes a lot of time and you don't even remember each and every song. It can also be a problem whenever you reset your phone, your data is deleted and your whole data is gone including your music files, so you don't have backup for those files. There can also be a probability that you lost your phone or it is stolen then you lost all of your files, few of them are backed up, like your messages, your photos, WhatsApp data, Contacts and many more but there's no music player that provides you this service of cloud computing and backups your MP3 files, Xu et al. (2015).

So, gathering all of the music files back that you loved is almost impossible.

- In order to solve these problems, objective of the proposed application is to provide the facilities that can deal with all these problems and will provides a smooth experience at the user-end.

LITERATURE REVIEW

Development of an application which is an android Music Player, Anusooya et al. (2020) and Gopalakrishnan and Venkateswarlu (2018). that is mainly designed and build to overcome the problem that is stated as, no music player is able to back up your music files on a cloud, Janssen et al. (2012), that can be restored to your device whenever you lost your phone, you lost your data or you have changed you phone. Another major problem is if more than one person is willing to listen to same music or any MP3 file, Van Den Broek et al. (2012), there are two possibilities, if they are sitting together, they need to share their earphones which will affect your music quality or they have to buy a special hardware to attach more than one earphone to one device, Lima et al. (2017). Other possibility is that the users are in some different places they want to listen to same music, Xu et al. (2015), but there's no Music player application that runs same music file on more than one android device, Chen et al. (2016). This application will back up the music files that you want to back up on cloud and you don't need to re-download them, as it takes a lot of time to remember name of each and every song when you lost the music data, Ahmed (2020). So, this application solves a problem of backing up and restoring your data automatically, that you do manually. This application also helps the user to sync to the other friend and they can also listen to same music on the same time. The purposed application will also help the user to sync even if the friends are at distant place (Akmandor et al., 2020).

Related Applications

Table 1. Related Application analysis with proposed project

Application Name	Weakness	Proposed Project Solution
Amp me	It only provides one functionality i.e., sync music with another friend. No backup	• Music backup
Google drive	Does not automatically backup music. It is a general application for storing your data online.	• Automatically backup • Music player functionalities
Default music player of phone	Does not provide music backup No sync option	• Music Backup • Sync with friend

Convolutional Neural Network

In deep learning, CNN (convolutional neural network) is a class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual imagery, Traore et al. (2018). CNNs use a variation of multilayer perceptron's designed to require minimal preprocessing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics, Ashqar et al. (2019).

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex, Zeiler and Fergus (2014). Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field, Takabi et al. (2016).

Models (Transferring)

The model is generated on a desktop machine because of high processing power needed to train the model and model is converted to Tensor Flow lite format for using on android using Linux TOCO converter which is not currently working for Windows, Guo et al. (2019).

Analysis from Literature Review

By using those apps there were few problems in every project application that are discussed above, Masri and Erdal (2019). And I have created an application which was solution to every proposed problem. For example, from amp.me we got to know that Music Application wasn't able to backup and restore music. According to the above given concepts my project uses a trained model that is trained on a dataset of 35000+ images. The data set is taken from kaggle which was used for the facial expression recognition competition, Kaggle.com, (2018). The dataset is divided in to 2 files training and testing made up of 48*48 gray scale images. Then the model is changed and deployed to android using tensor flow lite. The Application then uses the model to extract emotion from a given image that is taken live from an android phone and then transferred to the model to get the emotion and based on that emotion the user can use select to listen to the songs present in his own device or to Stream music over the internet, Sommerville (2006).

- I have an accuracy of 60-70% to further improve it. I am getting feedback of the user if the emotion detected is not correct in which the image of the user is converted to byte array and uploaded to GCS (Google Cloud Service) where the image is saved and a collection is made in order to further train the model manually.
- The accuracy is observed to be lesser than usual because human expressions are more likely to differ from person to person, Ni et al. (2020).
- Also, the model is observed to give better results on European faces relative to Asian. The application takes user feedback with the image that is used for collection of data and model can be trained further for variety of people thus making the model more efficient and accurate, Ofni Systems (2018).

Current System

A brief description of an existing system.

Table 2. Related System Analysis, proposed project solution, Chuttergoon and Nagowah (2020)

The related system information for proposed system is as follows:

Application Name	**Weakness**	**Targeted Project Solution**
The name of related application(s).	Weaknesses may include limited features, low quality functionality and processes.	The way the proposed project mitigates the weaknesses.
Emotion Detection from Text	It uses text to categorize emotion which means user has to spend time in typing.	The text will be replaced with image of the user.
Xam Cognitive	It just tells the percentage of all emotions by using image of the user	The app will be able to suggest music based on emotions.
FACE-e App	It computes the expressions and emotions of tiny toddlers who can't express it all through their words.	The emotion detection can also be applied to adults.
Amp me	It only provides one functionality i.e., sync music with another friend. No backup.	Music Backup
Google Drive	Does not automatically backup music. It is a general application for storing your data online.	Automatically backup Music Player functionalities
Default Music Player of Phone	Does not provide music backup No sync option	Music Backup Sync with friend

RESEARCH METHODOLOGY

Incremental model is a kind of model in which the project scope is divided into multiple segments and modules and each module will be developed separately in the form of small increments. The reason to use incremental model for the proposed system is that system consist of small modules or functionalities. The modules requirements might change during the development phase so it provides freedom and options more than rigidly developing using sequential way.

Designed Methodology

The designed methodology that I am going to use is MVC which is very clear as it is an android mobile application. VC Pattern stands for Model-View-Controller Pattern. This pattern is used to separate application's concerns.

- **Model**- Model represents an object or JAVA POJO carrying data. It can also have logic to update controller if its data changes.
- **View**- View represents the visualization of the data that model contains.
- **Controller**- Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate (Giannakos et al., 2020).

Methodology and Software Lifecycle for This Project

The software process model used for the proposed system is agile model. Incremental model is a kind of model in which the project scope is divided into multiple segments and modules and each module is developed separately in the form of small increments. The reason to use incremental model for the proposed system is that system consist of small modules or functionalities. The modules requirements might change during the development phase so it provides freedom and options more than rigidly developing using sequential way, García et al. (2017).

Rationale behind Selected Methodology

The reason to use incremental model for the proposed system is that system consist of small modules or functionalities. The modules requirements might change during the development phase so it provides freedom and options more than rigidly developing using sequential way. So, it can independently work on parts of modules and then combine them together. Giving us a bit of freedom and this helped solve many integration problems, Martikainen and Said (2018).

System Architecture

The architectural style used is object-oriented architecture. The data is passed between different components through message passing. There are mainly three subsystems which are as follows:

- **Face detection:** The face of the user is detected and a rectangular box is drawn for the detected face.
- **Emotion detection:** This subsystem is responsible for detecting emotion from the camera using CNN based trained model.
- **Media player:** The audio controls like playing song, moving to other songs etc.
- **Sync Music:** The multiple android devices will be connected, and music will be played on all the connected devices, Huang et al. (2010).

- **Backup and Restore:** The song will be backed up and will be restored when needed.

Algorithm

Media player

Getting and displaying songs from the phone

- Get path of the audio files in phone.
- Declare a cursor object for moving through all files.
- For (Cursor points to the first file and cursor is not null and move cursor to next)
- Get column for title of track
- Get column for id of the track
- Get column for artist of the song
- Get column for URL of the song
- Get columns string
- Get cover art of the track in bytes
- Store data in Class Song End For
- Display songs in custom list view, Indolia et al. (2018).

Detect face

- Open camera from the app
- Capture image of the user
- Convert image in gray scale.
- Draw a rectangle around the face.
- Declare path for storing the image.
- Store image on the given path, Thakker and Kapadia (2015)

Play song

- Declare media player object.
- Get intent from the previous activity having song to be played
- Set seek bar for playing song and moving it
- Implement methods for change on seek bar.
- Stop playing music on user pause click
- Change pause button to play button
- Play music from the same position on user play click, Mertes et al. (2013).

Forward song

- When user clicks forward button.
- Move song position and seek bar position to current position+5000

Emotion detection

- Train model from the dataset using CNN.
- Input image to the trained model
- Calculate each emotion’s score
- Display user emotion.

Sync Music

- Friends are added and a connection is built.

- When connection is built both devices share a same node of Real Time Database.
- One device sends the information, and simultaneously it is sent to the receiver phone.
- Music on both devices run on the same time, Julin et al. (2020).

Backup Music

- Music files from the memory are shown in an adapter view.
- There's a button on adapter view
- By clicking that button, the name of file is saved in cloud, and file is stored in Firebase Storage, Julin et al. (2020).

Restore Music

- By clicking on download button, the names of all the songs are fetched from cloud storage.
- Names are split by using tokenizer, and each name is sent simultaneously and every song is fetched from the Firebase Storage, Parekh et al. (2020).

External APIs

Describe the APIs used in the following table.

API	Description	Purpose	Functions
Media Player	This class is the primary API for playing sound and video.	The purpose of API is to play music from the list and to start and stop the player with built in functions.	• Start() • Stop() • Reset() • Release() • On Prepared() • Prepare Async() • Get Current Position() • Get Duration() • Is Playing()
Firebase	The google engine is used to store files from user as backup of their music files, also helped me in Syncing.	The user can save his fav music files and can get in any phone anywhere he likes. and also, can sync phones and play music simultaneously.	• On Success Listener() • On Failure Listener() • On Progress Listener()

Karthikeyan (2018).

DATA TESTING AND EVALUATION

Manual Testing

1. Allow Permissions

To ensure that user gives access to all required permission in the device.

No.	Test case/Test script	Attribute and value	Expected result	Result
1.	Allow Permissions	The User Is Prompted For Permissions and the user clicks ALLOW.	Show Home Page	PASS

2. View Playlist

The user can view playlist from any of the six given playlists.

No.	Test case/Test script	Attribute and value	Expected result	Result
2.	View Playlist	The user clicks the desired playlist	Open the desired playlist	PASS

3. Play a Song

The user can view the playlist generated and play any song.

No.	Test case/Test script	Attribute and value	Expected result	Result
3.	Play Song from The Playlist	The User Selects the desired song from the playlist.	Play Song	PASS

4. Pause Song

The user can view the playing song and can pause any playing song.

No.	Test case/Test script	Attribute and value	Expected result	Result
5.	Resume Song	The User Clicks the play button.	Music plays and icon is changed	PASS
	6. Play Next Song The user can view the playing song and can select next song to play.			
No.	**Test case/Test script**	**Attribute and value**	**Expected result**	**Result**
6.	Play Next Song	The User Clicks The Next Button	The Song is changed to the next song in List.	PASS

No.	Test case/Test script	Attribute and value	Expected result	Result
4a.	Pause Song	The User Clicks the Pause button.	Music paused and icon is changed	PASS
4b.	Pause Music For External Actions	A Calls Comes During Music is Playing.	The Song is paused and speaker control is released.	PASS

5. Resume Song

The user can resume song from the notification or from the player.

7. Play Previous Song

The user can view the playing song and can select previous song to play.

No.	Test case/Test script	Attribute and value	Expected result	Result
7.	Previous Song	The user clicks the previous button	The Song is change to the Previous song in the list.	PASS

8. Forward a Song

The user can forward a song which is playing.

No.	Test case/Test script	Attribute and value	Expected result	Result
8	Forward Song From Seek bar.	The User moves the Seek bar forward in any Desired position.	Songs Plays from the Desired position.	PASS

9. Reverse a Song

The user can reverse a song which is playing.

No.	Test case/Test script	Attribute and value	Expected result	Result
9.	Reverse Song From Seek bar.	The User moves the Seek bar Reverse in any Desired position.	Songs Plays from the Desired position.	PASS

10. Open Camera

The user can shuffle songs to listen to random songs in the playlist.

No.	Test case/Test script	Attribute and value	Expected result	Result
10.	Open Camera for emotion detection	The User Clicks on the camera button	The Camera Opens Up To Capture The Image	PASS

11. Detect Face

The user can open camera and face is detected.

No.	Test case/Test script	Attribute and value	Expected result	Result
11a.	Detect Face	User capture image of a face	Picture is taken and face is detected	PASS
11b	Detect Face	User captures image of a non-face object.	Picture is taken and face is not detected and user is shown a message	FAIL

12. Show emotion result

The user is shown the detected emotion.

No.	Test case/Test script	Attribute and value	Expected result	Result
12a.	Show Emotion detected	User captures the face image	The user is shown the emotion detected by the system and playlists to choose from.	PASS
12b.	Show emotion detected	User captures a non-face image	No emotion is detected	FAIL

13. Choosing Play Mode

The user is requested to choose from the playlists of offline or online.

No.	Test case/Test script	Attribute and value	Expected result	Result
13.	Choose Play Mode	User chooses “OFFLINE” mode.	The custom playlist is opened for the respective emotion.	PASS

14. Add Songs to the playlist

The user can add songs to the playlist that are to be generated by user and are custom generated playlists.

No.	Test case/Test script	Attribute and value	Expected result	Result
14.	Add songs to the playlist.	User clicks on “ADD SONGS” button	The system opens the device songs list to choose a song.	PASS

15. BACKUP MUSIC

The user can add songs to backup folder.

No.	Test case/Test script	Attribute and value	Expected result	Result
15.	Add songs to backup	User clicks on “backup” button	A notification will appear and songs will be uploaded to Firebase Storage.	PASS

16. Restore Music

The user can restore the uploaded songs into backup.

No.	Test case/Test script	Attribute and value	Expected result	Result
16.	Get songs back from Firebase to local memory.	User clicks on “restore” button	A notification will appear and songs will be downloaded to the local memory.	PASS

17. Sync Music

The user can sync music with friends.

No.	Test case/Test script	Attribute and value	Expected result	Result
17	Sync Music with your friend.	User clicks on “Join Party” button.	It should take you to a media player, where the song you play will be played on your friends phone as well.	PASS

18. Search for people.

The user can search for users using his username.

No.	Test case/Test script	Attribute and value	Expected result	Result
18	Search for people.	User can search for people by putting their username in the search box.	A notification will appear and songs will be uploaded to Firebase Storage.	PASS

19. Add Friends

The user can add friends.

No.	Test case/Test script	Attribute and value	Expected result	Result
19	Add people as your friends.	User clicks on “send friend request”	A notification will appear on the receiver’s application.	PASS

20. Accept Friend Request.

The user can accept friend’s request.

No.	Test case/Test script	Attribute and value	Expected result	Result
20.	Accepting friend’s requests.	User clicks on “accept request” button and the sender will be added as your friend.	The user will be able to see the profile and will be able to perform various function.	PASS

21. View Friends Timeline

The user can accept friend’s request.

No.	Test case/Test script	Attribute and value	Expected result	Result
20.	A user will be able to see other’s profile.	User searched and clicked of a person’s profile.	The user will be able to see the profile of his friend.	PASS

22. Image Upload.

The user can accept friend’s request.

No.	Test case/Test script	Attribute and value	Expected result	Result
20.	Upload display picture.	User clicks on “upload image” button and can browse a picture and can upload the picture.	The user will be able to upload a picture for his profile.	PASS

23. Update Bio.

The user can accept friend's request.

No.	Test case/Test script	Attribute and value	Expected result	Result
20.	Update bio.	User clicks on "update bio" button	The user will able to change his information.	PASS

24. View Friends.

The user can accept friend's request.

No.	Test case/Test script	Attribute and value	Expected result	Result
20.	View Friends.	User clicks on "view friends" button.	The user will be able to see the friends he have added.	PASS

25. Remove Friend.

The user can accept friend's request.

No.	Test case/Test script	Attribute and value	Expected result	Result
20.	Remove people from friend list	User goes to someone's Profile and will click on "Remove Friends" button.	The user will be able to see the profile and will be able to remove him from his friend list.	PASS

26. Uploaded Songs.

The user can accept friend's request.

No.	Test case/Test script	Attribute and value	Expected result	Result
20.	See uploaded songs.	User will click on "View Uploaded Songs"	The user will be able to see the lists of songs he have already uploaded.	PASS

Wang and Li (2021).

Unit Testing

We have tested each module and the functionality thoroughly for bugs and errors. We have identified most of them. We have removed almost all the bugs found and still working to find more bugs to solve.

Acceptance Testing

For all the functionality flow of the system, the final acceptance test is done by the supervisor.

Functional Testing

The functional testing will take place after the unit testing. In this functional testing, the functionality of each of the module is tested. This is to ensure that the system produced meets the specifications and requirements (Giannakos et al., 2020).

CONCLUSION

MUSYNC is a great android-based application and has excessive scope. This proposed cloud music player application plays music according to user moods with the help of machine learning and facial recognition which is based on different mood also stated by, Kumar (2018), and Jain, et al. (2017), states of the stimuli that is being clicked by android phone camera and also it has a variety of other features too. Like, create account, backup and restore option, update user bio, send friend request, add friends, search friends via user id or by authorize Gmail account, update user profile, Glover (2018), provide local storage, music library of phone including audio and video songs, which is not provided by any other applications. Also, there's a Party Mode, which sync your phone with someone else's phone and you can play music together at the same time within the same place. For all the cloud storages by means of keeping the track and record of everything, Firebase is used as database that is provided by Google platform, Kundinger (2021). This is quite an expensive application and people pay thousands of rupees for its purchases.

FUTURE WORK

For future we have thought on three different features, and we'll surely work on it as well. Like, sharing music, from which users can share music using this application. Secondly, we'll also thinking about making it a kind of social application. By which people will be able to tell their views about some great music and will able to post statuses of their timelines. Final work that can be done is giving this syncing module an offline mode. So that people can also syncing their phones without internet. Using Wi-Fi-Direct of some other mean. That'll make it a huge progress and we believe that it'll get a lot of attention from general public.

REFERENCES

Ahmed, H. (2020). Facial Recognition Access Control System Research Manual.

Akmandor, A. O., Dai, X., & Jha, N. K. (2020). YSUY: Your Smartphone Understands You--Using Machine Learning to Address Fundamental Human Needs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Anusooya, S., Anitha, R., Aamira, S. J., & Sowmyaa, G. (2020). Music Therapy by Analyzing EEG Signals. *Bulletin of Scientific Research*, 9-16.

Ashqar, B. A., & Abu-Naser, S. S. (2019). Identifying images of invasive hydrangea using pre-trained deep convolutional neural networks. *International Journal of Academic Engineering Research (IJAER), 3*(3), 28-36.

Chen, Y. (2016). Developing a music player mobile application with cloud server.

Chiu, M. C., & Ko, L. W. (2017). Develop a personalized intelligent music selection system based on heart rate variability and machine learning. *Multimedia Tools and Applications, 76*(14), 15607-15639.

Chuttergoon, A., & Nagowah, L. (2020, November). Eye Guide: An Assistive Mobile Technology for Visually Impaired Persons. In *2020 3rd International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering (ELECOM)* (pp. 198-203). IEEE.

García, I. C. A., Salmón, E. R. L., Riega, R. V., & Padilla, A. B. (2017, December). Implementation and customization of a smart mirror through a facial recognition authentication and a personalized news recommendation algorithm. In *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* (pp. 35-39). IEEE.

Giannakos, M., Voulgari, I., Papavlasopoulou, S., Papamitsiou, Z., & Yannakakis, G. (2020). Games for Artificial Intelligence and Machine Learning Education: Review and Perspectives. *Non-Formal and Informal Science Learning in the ICT Era*, 117-133.

Giannakos, M., Voulgari, I., Papavlasopoulou, S., Papamitsiou, Z., & Yannakakis, G. (2020). Games for Artificial Intelligence and Machine Learning Education: Review and Perspectives. *Non-Formal and Informal Science Learning in the ICT Era*, 117-133.

Glover, J. (2018). *Unity 2018 augmented reality projects: build four immersive and fun AR applications using ARKit, ARCore, and Vuforia*. Packt Publishing Ltd.

Gopalakrishnan, R., & Venkateswarlu, A. (2018). *Machine Learning for Mobile: Practical guide to building intelligent mobile applications powered by machine learning*. Packt Publishing Ltd.

Guo, Q., Chen, S., Xie, X., Ma, L., Hu, Q., Liu, H., ... & Li, X. (2019, November). An empirical study towards characterizing deep learning development and deployment across different frameworks and platforms. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)* (pp. 810-822). IEEE.

Huang, M., Liu, H., Xiao, Y., & Cai, Q. (2010, October). Research on digital images' color-difference by altering lightness and chroma: Analysis and evaluation of color-difference formulae. In *2010 3rd International Congress on Image and Signal Processing* (Vol. 5, pp. 2347-2350). IEEE.

Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual understanding of convolutional neural network-a deep learning approach. *Procedia computer science*, *132*, 679-688.

Jain, M., Kaushik, N., & Jayavel, K. (2017, February). Building automation and energy control using IoT-Smart campus. In *2017 2nd International Conference on Computing and Communications Technologies (ICCCT)* (pp. 353-359). IEEE.

Janssen, J. H., Van Den Broek, E. L., & Westerink, J. H. (2012). Tune in to your emotions: a robust personalized affective music player. *User Modeling and User-Adapted Interaction*, *22*(3), 255-279.

Julin, A., Kurkela, M., Rantanen, T., Virtanen, J. P., Maksimainen, M., Kukko, A., ... & Hyyppä, H. (2020). Evaluating the quality of TLS point cloud colorization. *Remote Sensing*, *12*(17), 2748.

Karthikeyan, N. G. (2018). *Machine Learning Projects for Mobile Applications: Build Android and IOS Applications Using TensorFlow Lite and Core ML*. Packt Publishing Ltd.

Kumar, A. (2018). *Mastering Firebase for Android Development: Build real-time, scalable, and cloud-enabled Android apps with Firebase*. Packt Publishing Ltd.

Kundinger, T., Riener, A., & Bhat, R. (2021, September). Performance and Acceptance Evaluation of a Driver Drowsiness Detection System based on Smart Wearables. In *13th International Conference on Automotive User Interfaces and Interactive Vehicular Applications* (pp. 49-58).

Lima, D., Sousa, M., Ara, R., Hannum, J., Rocha, A. D., & Barbosa, A. T. (2017). Software with biofeedback to assist people with Down syndrome. *International Journal of Computer Applications*, *158*(5).

Marr, B. (2019). *Artificial intelligence in practice: how 50 successful companies used AI and machine learning to solve problems*. John Wiley & Sons.

Martikainen, K., & Said, K. (2018). A facial recognition application for elderly care: Caregiver verification and identification.

Masri, İ., & Erdal, E. (2019, September). An Intelligent Vision System to Navigate a Robot Without On-Board Sensors. In *2019 4th International Conference on Computer Science and Engineering (UBMK)* (pp. 124-128). IEEE.

Mertes, J. G., Marranghello, N., & Pereira, A. S. (2013). Real-time module for digital image processing developed on a FPGA. *IFAC Proceedings Volumes*, *46*(28), 405-410.

Nathan, K. S., Arun, M., & Kannan, M. S. (2017, December). EMOSIC—An emotion based music player for Android. In *2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)* (pp. 371-276). IEEE.

Ni, T., Zhou, R., Gu, C., & Yang, Y. (2020). Measurement of concrete crack feature with android smartphone APP based on digital image processing techniques. *Measurement*, *150*, 107093.

Ofni Systems. (2018). *Functional Requirements (Functional Requirement Specifications,*

Parekh, P., Patel, S., Patel, N., & Shah, M. (2020). Systematic review and meta-analysis of augmented reality in medicine, retail, and games. *Visual computing for industry, biomedicine, and art*, *3*(1), 1-20.

Shanthamallu, U. S., Spanias, A., Tepedelenlioglu, C., & Stanley, M. (2017, August). A brief survey of machine learning methods and their sensor and IoT applications. In *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)* (pp. 1-8). IEEE.

Sommerville, I. (2006). Software Engineering. Old Tappan: Pearson Education UK.

Takabi, H., Bhalotiya, A., & Alohaly, M. (2016, November). Brain computer interface (BCI) applications: Privacy threats and countermeasures. In *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)* (pp. 102-111). IEEE.

Thakker, S., & Kapadia, H. (2015, September). Image processing on embedded platform Android. In *2015 International Conference on Computer, Communication and Control (IC4)* (pp. 1-6). IEEE.

Traore, B. B., Kamsu-Foguem, B., & Tangara, F. (2018). Deep convolution neural network for image recognition. *Ecological Informatics*, *48*, 257-268.

Wang, X., & Li, C. (2021). Android malware detection through machine learning on kernel task structures. *Nauru computing*, *435*, 126-150.

Xu, J. (2015, June). Design and Implementation of the Music Player based on Android. In *2015 International Conference on Management, Education, Information and Control* (pp. 1746-1753). Atlantis Press.

Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.

Copyrights