Intellisense Media Player

Anshuman Pandey
Meerut Institute of Engineering and Technology
N.H.58, Delhi-Roorkee Highway, Baghpat Bypass Road Crossing, Meerut, Uttar Pradesh 250005

Hardik Goel
Meerut Institute of Engineering and Technology
N.H.58, Delhi-Roorkee Highway, Baghpat Bypass Road Crossing, Meerut, Uttar Pradesh 250005

ABSTRACT
Intellisense Media Player is an advance Android Video Player which is based on VideoView API. This API is custom API built on the top of ExoPLayer API. This API features video controlling (Pause, Play) by detecting the user's face through mobile camera. If user is watching the media, then it must play and if not so it can then be paused automatically. Additionally this custom made android Video view API supports PIP (Picture in Picture) mode that facilitates multitasking. By saying multitasking it means that a video can still be played without keeping application opened. A small alert window on the top of all application running currently will be pinned acting as a video player. VideoView API also equipped with tilt detection based seeking (example left tilting leading to seek the video 'x' seconds backward, if possible). This Feature requires stabilization if available, if not so it can be turned off. This custom built Video View also supports basic features like double tap to forward/backward, scroll horizontal to seek media, scroll vertical to alter volume/brightness, video locking etc..

Additionally, Intellisense Media Player also includes a Tag Based Searching algorithm featuring "how the video must be recommended to watch", "how video storing and fetching would be performed" built upon Google Firebase real time database. This algorithm is not a part of VideoView API (refer2.3).

Keywords
Intellisense Media, look based player, intelligent media player, Face Detection, PIP, orientation, media player, Video View, Tag Based Searching.

1. INTRODUCTION
Normally when you are watching a video or a movie and someone calls you, you have to look somewhere else or go away from your system for some time, so you may miss some portion of the video/movie. Later on you must rewind or seek the video from where you left off. Well the solution to this problem is solved using this technique. An AI featured media player that stops playing the media when face is not detected in front camera of the mobile phone. The media players starts playing the media file or resume it as soon as the face is detected or anyone looks at it. This is achieved using the mobile front camera. Media Player keeps on playing the media file while face is detected from front camera. The Media Player pauses when user's face is partially detected or not detected at all. This Application is equipped with other essential media functions like volume up, or volume down, seeking forward or backward by detecting the orientation.

1.1 Problem with existing system
Current Applications don't have knowledge whether user is interacting or not. Screens in most of the organization, Malls, and Metro Streets etc. keep playing the content even if there is no user interaction, that's wastage of power. No possibility for multitasking.

Mostly existing systems do not use any kind of intelligence recognition. Because of which accuracy is low or inadequate. Face Detection and motion sensing are neither implemented together nor individually. Due to lack of these features and services in the system user sometime misses the important part of the video or sometime waste his time in rewinding and forwarding the video. Existing systems have no capability to detect motion or tilt.

1.2 Aims & goals of proposed system
The main aim of this project is to develop an intelligent and advance media player based on facial recognition and motion sensing.

This research has the following main goals for this media player to achieve the objective:

1. To make multimedia applications capable of **sensing human activity.**

2. Accuracy of the outcome from media player must be high or acceptable.

3. To **Reduce Power Consumption**.

4. To provide users various features for **Multitasking**.

5. The Media Player pauses when user's face is partially detected or not detected at all.

6. Ease of control of media

7. Interface of the media player must be user friendly and efficient.

8. Appropriate stability is expected in tilt based media control.

9. Recommended media on the basis of tag must be appropriate and of user interest.

10. Storing of media from Admin Panel should take minimal complexity.

11. A media must be fetched in a constant time given an ID.

For this project motion sensing and face recognition is used for controlling the media player. Face recognition feature is used for stopping and resuming the media which media player is playing. Various motion sensing techniques are being used for controlling other functionalities of media player like volume up, or volume down, seeking forward or backward.

P I P (Picture – in Picture Mode) mode is used for providing multitasking feature, so as to make user independent of single - tasking.

2. DESIGN AND IMPLEMENTATION OF PROPOSED METHODOLOGY

2.1 Face detection

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces in a visual scene. Face detection can be regarded as a specific case of object-class detection. Face-detection algorithms focus on the detection of frontal human faces. It is analogous to image detection in which the image of a person is matched bit by bit.

Our project uses face detection technique provided in standard Google`s Firebase Machine Learning Kit (ML Kit). It uses runtime detection of faces to take media playback decisions like when to pause and when to play the video file. This service is although stoppable by user end and terminates automatically in PIP mode.

2.2 Picture in picture (pip) mode

PICTURE IN PICTURE (Abbreviated as PIP) window is a particular type of overriding window (Screen) mainly used for activities that need to be active on screen but should not take up the whole screen space. Mainly PIP mode is used for activities like watching videos, video calls, navigation, etc doing other activities meanwhile. PIP Window allows the user to keep watching the video in a pinned small window on the screen at any corner as decided by the user while browsing content on the screen. By default Window would be pinned at the bottom right corner of the screen as depicted in the image below.

PIP mode has been made available for the android users having minimum API Level 26 (Android 8.0 Oreo or above). This mode has been made available for the above mentioned android users in android SDK tools 26 or above. Aim is to design a similar picture in picture mode for all API levels supporting wide range of android versions.

Since PIP Window is an Overriding window it means it will cover some portion of the applications on which window is placed. The PIP window will always appear in the top-most layer of the screen that means all other applications running below the PIP Window will be below to it and are invisible.

PIP window can be moved or relocated to another location using some special toggles provided within the window itself. When the window is tapped at the center following two options appear generally:

1. At the center of the PIP window: A button to enter into a full screen model

2. At Upper Right Corner of the PIP window: A button to exit or close the window (Appear as “X”).

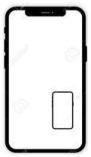

Fig 1: PIP mode example

2.3 Motion sensing

It is concerned with the actions to be performed through the change in angle of the mobile phone. In this project rotation vector sensor is used.

The rotational vector sensor used in the motion sensing is particularly adaptable means that it can be used for multiple purposes and in multiple applications. Example it can be used for motion related tasks for wide ranges like:

a. Monitoring of comparative orientation changes.

b. Detecting and sensing the gestures.

c. Monitoring Angular changes.

For example: For developing a game the rotational vector sensor (abbreviated as RVS) is absolute, Other than Developing game RVS can be supremely used in an augmented reality application (AR Application), or a 2-D or 3-D compass, RVS can be used for applications that involve stabilization of cameras like Document Scanners.

In most of the above cases, using RVS sensors is a better choice over Accelerometer and GMS (Geomagnetic Field Sensor) or the Orientation sensor.

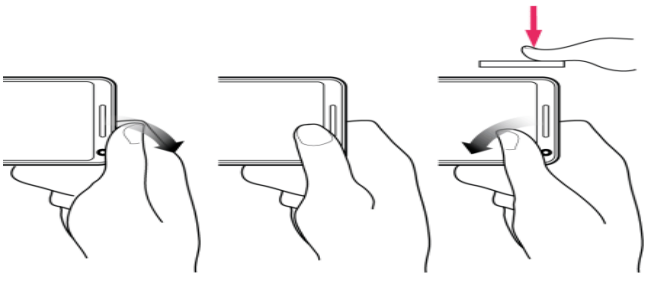

Fig 2: Motion (Orientation) Sensing Example

Tilt toward right - leads to forward the video for particular seconds.

Tilt toward left - leads to rewind the video for particular seconds.

2.4 Tagging (Tag based searching)

Tagging is tag based searching for videos or related content in real-time database (Firebase Database).

This algorithm involves various steps for searching:

1. Get all the tags for which data is to be retrieved

2. For each of the tag do the following:

2.1 Check if each character in the tag must be a valid character and in lower case.

2.2 Take each character and enter into the corresponding node within the real-time database.

When all the characters have been traversed all the items that have been appeared after fully accessing the last node in the database must be added to the database.

For Example:
If tag is “Devotion” then firstly it determines each character must be valid & in lower case which is true. Then it takes each character from the tag & go to that node (Here ‘D’) then go to node ’e’ then ‘v’ and so on as shown below.

D-->e-->v-->o-->t-->i-->o-->n

All nodes appearing after node ‘n’ will be added to the list. Incase no node exists then last node after which no node has been found must be traversed fully.

For Example:
If the tag is **"Devotional"** but **'a'**and **'l'** does not exist then last node considered will be **'n'.**

D-->e-->v-->o-->t-->i-->o-->n-->a-->l

3. Repeat (step 2) for each of input tags and adds the items in a set (s) for each tag.

4. Now sort the items in the set based on:
4.1 Number of hits/Number of times accessed.
4.2 Uploaded date and time.
4.3 First tag items will be at first.
4.4 Already accessed by the users preferably at last.

3. EXPERIMENTAL RESULTS

After developing the proposed media player, it has been tested rigorously and two of it major features facial detection based media controlling, facilitating multitasking through PIP (Picture in Picture) mode has been depicted in the upcoming illustrations as experimental results.

Example:

Media playing gets paused or interrupted as the user is not watching the same.
Media resumes its state from same point of interruption as soon as user starts interacting or watching the media.

Below is the experimental analysis of face detection based media player controlling that is used in Intellisense Media Player. First column represents number of faces detected from the front camera of the device. Second column represents Time elapsed from start of the module (0ms represents experimental results at the start of this module). This module triggers after every 100ms.

A Boolean Array (third column) is a variable sized array (size =5). Each bit in this array represents a decision whether to play or pause a media. Each time module triggers if no faces are detected 0 in added at the end of this array and in other case 1 is added to the array. In case size of array is reached start from index 0. Here values are updated without taking into consideration the value initially placed at that index.

Column 4 (Majority) represents majority of values in the Boolean array of that trigger. Based on that majority a final decision is to be carried out, that is if majority is 0 it represents to pause the player and if majority comes out to be 1 play or start the player.

Column 6 (Ideal) represents what must be the decision by considering only column 1, that is the number of faces. If 1 or more faces are detected then play or start the player in other cases pause the player.

Since majority of previous 'x' number of triggers is taken into consideration therefore it provides a better and stable decision by considering not only the decision of that particular trigger but also taking into account the previous x-1 decisions. Here 'x' represents size of the Boolean array.

Table 1: Face detection module analysis

#Faces	Time Relative	Boolean array[5]	Majority	Decision	Ideal
1	0ms	[1,0,0,0,0]	0	Pause	Play
1	100ms	[1,1,0,0,0]	0	Pause	Play
1	200ms	[1,1,1,0,0]	1	Play	Play
0	300ms	[0,0,0,0,0]	0	Pause	Pause
1	400ms	[1,1,1,1,1]	1	Play	Play
0	500ms	[0,0,1,0,0]	0	Pause	Pause
2	600ms	[1,1,1,1,1]	1	Play	Play

This is an illustration of PIP mode running on an android device with video playing in a small pinned window on the top of all windows.

Device Specifications:

Android Version: Lollipop (L)

Android API Level: 22

Device Brand: Samsung Galaxy J3

Fig 3: Pip Mode Device Testing

Picture in Picture mode is developed for almost all android API level and below is the experimental testing result in various API levels starting from API level 19.

Column 1 and 2 represents the Android Name and its corresponding API levels. While Column 3 represents market share in percentage of that API level.

Example:

There are 11.2% devices of total running android devices in the word that works on API level 23.
There are 31.3% devices of total running android devices in the world that works on API level 28.

Note: API level 19 to 29, covers almost 99% of total running android devices in the world.

Table 2: Picture in Picture module analysis

Android Name	API Level	Market share	Result
Kit Kat	19	4%	Supported
Kit Kat Watch	20	4%	Supported
Lollipop	21	1.8%	Supported
Lollipop	22	7.4%	Supported
Marshmallow	23	11.2%	Supported
Nougat	24	7.5%	Supported
Nougat	25	5.4%	Supported
Oreo	26	7.3%	Supported
Oreo	27	14%	Supported
Pie	28	31.3%	Supported
Q	29	8.2%	Supported

4. CONCLUSIONS

The main aim of this project is to make user experience a great media player with some intelligence and sensing features. Automating of the media player has enabled us to achieve this goal to a wide extent. Automating is done with techniques, technologies and features like Google Camera2 Android API for Face Detection, PIP Mode (Achieved through services, window managers and window parameters), Motion Sensing and tag based search algorithm.

For Multitasking Picture in Picture mode (P I P mode) is used. This will help user to go through multiple task and feel free of relying on thing.

5. REFERENCES

[1] Sashabrava; Android for Professionals; Android VideoView and optimized VideoView: pp number 180-191.

[2] Hamed GH; Android for Professionals; ExoPLayer API and tracker rendering with PIP Implementation in a VideoView: pp number 862- 864.

[3] Herbert Schildt; Java Complete Reference; Tenth Edition; Java SE 9 (JDK 9) Java Networking ;pp number 749-766

[4] Herbert Schildt; Java Complete Reference; Tenth Edition; Java SE 9 (JDK 9) J Java Collections Framework ;pp number 749-766

[5] Chintan Soni; Omar Aflak Android for Professionals; Android Firebase: Firebase Cloud Messaging (FCM), Firebase Storage, and Firebase Real-time Database: pp number 1065- 1092.

[6] Android App Development; Google Android; PIP Mode support (Picture in Picture mode) Ref: https://developer.android.com/guide/topics/ui/picture-in-picture#:~:text=Android%208.0%20(API%20level%2026,content%20on%20the%20main%20screen.

[7] Android App Development; Google Android; Face Detection with ML kit in Android ref: https://developers.google.com/ml-kit/vision/face-detection/android

[8] Google Camera2 API for Android: Google Face Detection kit ref: https://developer.android.com/reference/android/hardware/camera2/package-summary

[9] Motion Sensing Event Handling: Android Official Developers page ref: https://developer.android.com/guide/topics/sensors/sensors_motion.