# Machine Learning Team Project

***Team Ronaldo***

Minui Song / HYUNSONG LEE / HYEWON CHO

## 1. Project : Air-Guitar

<Air-Guitar> is a project aimed at creating a virtual musical instrument, specifically a guitar that can be played without a physical instrument by analyzing the player's hand shapes. The system infers which chord is being held by examining visual features of the player's left hand—such as the overall hand shape, relative positions of the joints, and the degree of finger bending—based on chord- specific images learned in advance. Instead of static images, the system processes real-time video to accurately classify chords from various angles and play the corresponding sound.

After the left hand forms a chord, the system detects the movement of the right hand to trigger the playback of specific audio. In addition, a virtual guitar appears near the user's torso on the screen, as if it were strapped around their neck. Its placement and scaling create a clear visual reference that enhances the sense of playing an actual instrument. Combined with real-time chord classification and gesture-based strumming, the system offers an experience closer to a "playable virtual instrument" rather than a simple gesture-recognition application.

## 2. Motivation & Goal

Many people have a desire to play at least one musical instrument in their lives, and the guitar is especially popular. However, guitars are expensive, difficult to maintain, and it's not uncommon for beginners to lose interest after purchasing one—only to let it collect dust and eventually sell it secondhand.

Our project began with a bold question: "What if people could practice guitar using only a webcam and a PC?" We aimed to build a program that allows users to play guitar using nothing but bare-hand movements, applying various computer vision techniques learned throughout the semester. Through this process, our goal was not only to create a functional system but also to deepen our academic understanding by engaging with real visual information processing methods.

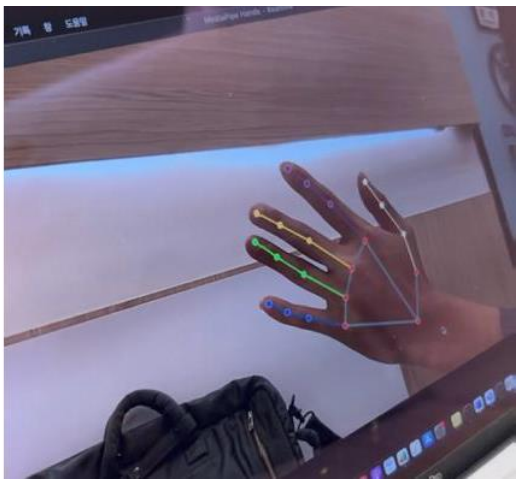
## 3. Technical Points

We utilized a variety of models and techniques throughout the project, including MediaPipe Hands and MediaPipe Pose..

### (1) Chord Recognition

The system extracts the left-hand region from the camera feed and classifies its shape into one of the three chords (G, C, D).

**- MediaPipe Hands**

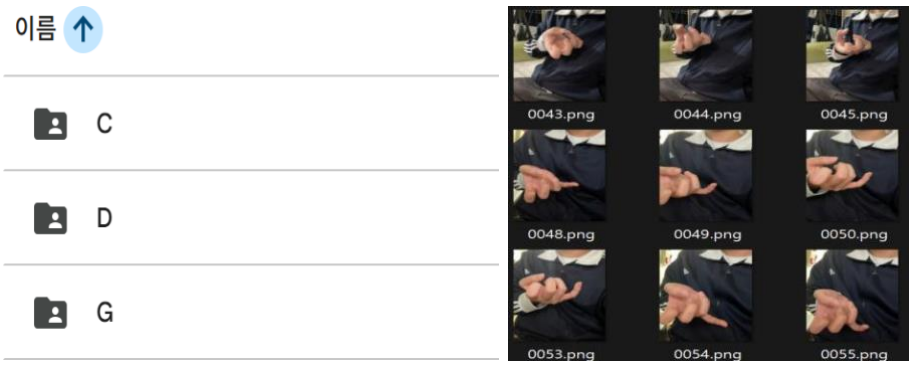


MediaPipe Hands is a real-time hand-tracking pipeline that infers skeletal structures and outputs key joint locations using only a standard RGB camera. Each landmark is returned as a 3D coordinate, which is collected frame-by-frame and vectorized for training.

Since the project relies heavily on visual information captured in different environments, this approach reduces the influence of variables such as skin color, lighting, and background, allowing the model to focus purely on geometric hand features. This ensures stable performance even during classroom demonstrations

Furthermore, guitar chords are defined by the relative positioning of finger joints, making a joint-coordinate-based representation well-suited for identifying repeated geometric patterns specific to each chord.

**- Chord Classification Model**




For each chord (G, C, D), team members recorded images of their left-hand finger positions from various angles and distances. After labeling the images by chord type, we extracted left-hand joint coordinates using MediaPipe Hands and vectorized them.

These vectors were used to train an MLP-based classifier. The model probabilistically determines which chord best matches the input hand configuration and outputs the chord with the highest confidence. Since this inference runs on every frame, the system continuously tracks chord changes in real time, not just in static images.

**(2) Virtual Guitar**


- Generating a Virtual Guitar in Front of the User : When the user assumes a playing posture in front of the camera, a virtual guitar appears near the upper body. The user can then place their hands as if holding a real guitar, with left-hand chord shapes and right-hand strumming visually aligned to the virtual instrument.

- Pose detection: MediaPipe Pose detects the user's shoulder positions in real time. The virtual guitar is positioned based on the midpoint between the shoulders and shifted slightly

downward to match a natural playing position. The guitar is also rotated according to the angle of the shoulder line, ensuring a realistic orientation..

- Guitar resizing: The guitar's size is dynamically set to three times the distance between the shoulders. As the user moves closer to or farther from the camera, the guitar scales accordingly, producing a natural visual effect.

### (3) Downstroke Recognition

Like playing a real guitar, the system plays the chord sound when the user swings the right hand. In our implementation, this motion is detected using MediaPipe Hands, focusing on the right hand's index fingertip and analyzing and analyzing its frame-by-frame movement on the image

- Velocity-based strum detection: For each frame, MediaPipe Hands provides 3D landmarks for both hands. From the right hand, the index fingertip(INDEX_FINGER_TIP) is selected as the representative point for strumming. Its normalized coordinates are converted into pixel coordinates using the current frame width and height. The system stores the fingertip's vertical position(y coordinate) from the previous frame and compares it to the current frame's position. A positive difference between the current and previous values indicates a downward movement. If this vertical displacement exceeds a predefined sensitivity threshold(which can be manually set by the user), the system interprets it as a deliberate downstroke.

- Sensitivity control and debouncing: To accommodate different users and camera conditions, the downstroke threshold is exposed as an adjustable "Sensitivity" trackbar in the UI. A lower threshold makes the system react to smaller motion, while a higher threshold requires a more pronounced swing to register as a proper strum motion. Additionally, a cooldown timer is applied after each detected downstroke. Once a stroke is recognized, the system waits for a short period(0.2 sec) before accepting another stroke. This debouncing logic prevents one physical strumming motion from being counted multiple times due to small vibrations or even frame noise.

When a valid downstroke is detected, the system triggers the chord sound corresponding to the currently recognized left-hand chord. Because chord classification and downstroke detection run at the very same frame, the user can change chord shapes with the left hand and immediately hear the expected chord sound whenever a new downstroke is performed.

## 4. Implementation

1 ) Dataset Collection & Labeling

- Each team member recorded their own left-hand images while holding G/C/D chords.
- We captured diverse data by varying distance, angle, and hand orientation.
- All images were manually labeled and organized into chord-specific folders.

2 ) Feature Extraction & MLP Training

- MediaPipe Hands was used to extract 3D landmarks from each image.
- These (x, y, z) joint coordinates were vectorized and used as MLP inputs.
- The classifier was trained via supervised learning to distinguish G/C/D chords.
- Testing showed 91% accuracy, with some confusion between C and D due to structural similarity.

The team agreed to clearly differentiate the ring finger position during chord demonstration.

3 ) Real-Time Chord Recognition

- Webcam frames are processed with MediaPipe Hands to extract live joint coordinates.
- These coordinates are fed into the trained MLP to predict the current chord.
- This repeats every frame, continuously updating the “active chord.

4 ) Virtual Guitar Rendering

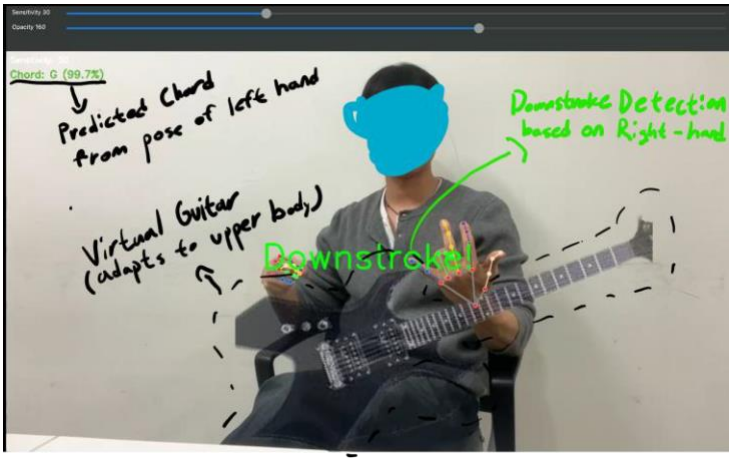
- MediaPipe Pose extracts shoulder coordinates, and a virtual guitar is placed around the natural holding position.
- The guitar rotates to match the shoulder angle and scales according to shoulder width for realism.

4) Right-Hand Strumming Detection

- A strumming zone is generated in alignment with the virtual guitar body.
- MediaPipe Pose is used to analyze right arm motion and compute frame-by-frame movement speed.
- If the right hand moves quickly and is inside the strumming zone, the system recognizes a strum and plays the sound of the active chord

## 5. Outcome

Demo video

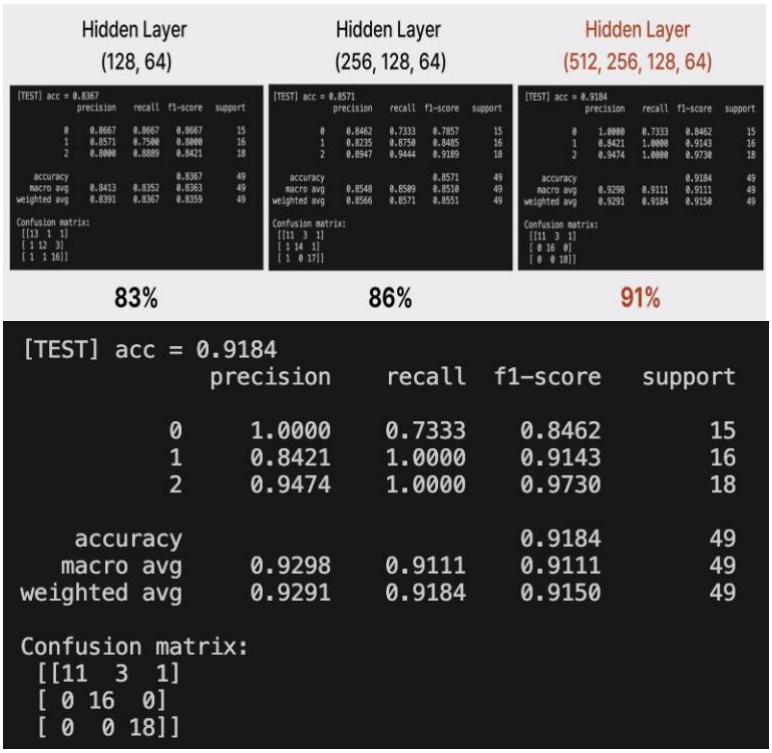



Key screenshot from Demo video(attached)

- We Created a demo video showcasing the program performing music composed of G/C/D chords. Video file is attached with report.

- Experienced ensemble playing by performing together with a real guitar.

### Model performance




The model achieved 91% accuracy in testing. Certain chords achieved a perfect recall of 1.0, but some misclassification patterns appeared—particularly between the C and D chords, which

share visually similar structures. We concluded that ensuring clearer positioning of the ring finger during performance would mitigate this issue..

## 7. Challenge & Future works

- Challenge

Since no dataset existed in the first place, we had to create it ourselves. It's a bit disappointing because with a wider variety of chords and a larger amount of data, we could have built a much richer and more robust program.

- Future Works

We would like to enable upstroke detection as well.

We also want to implement support for recognizing multiple users simultaneously so that more than one person can play the guitar together.

I would like to extend this to various instruments and create a real virtual band.

## 8. Reflection

Working on Air-Guitar project was a valuable learning experience, especially as part of an introductory Computer Vision course. Since it was a team project, We had the opportunity to collaborate closely with teammates, share ideas, and combine our strengths to solve problems together. Throughout the process, We gained hands-on experience with OpenCV, which helped me understand how computer vision techniques are applied in real-world applications.

Collecting our own dataset, preprocessing the coordinates, and implementing the model from scratch allowed us to appreciate the importance of data quality and the challenges involved in building a working system without pre-existing resources. Although there were difficulties, such as handling variations in hand positions and tuning the model for accuracy, overcoming these challenges was both rewarding and educational.

Overall, this project not only strengthened my understanding of computer vision fundamentals but also sparked my interest in exploring more advanced topics and real-time interaction systems. It was a meaningful learning experience that helped us grow both technically and collaboratively.