Sounds of Music

Abirami R - 2018115003
Thupalli Hethana - 2018115120
Vibha Alaguraj - 2018115127

Problem statement:

▹For beginner music learners, understanding sheet music and learning to play notes will be tough task.

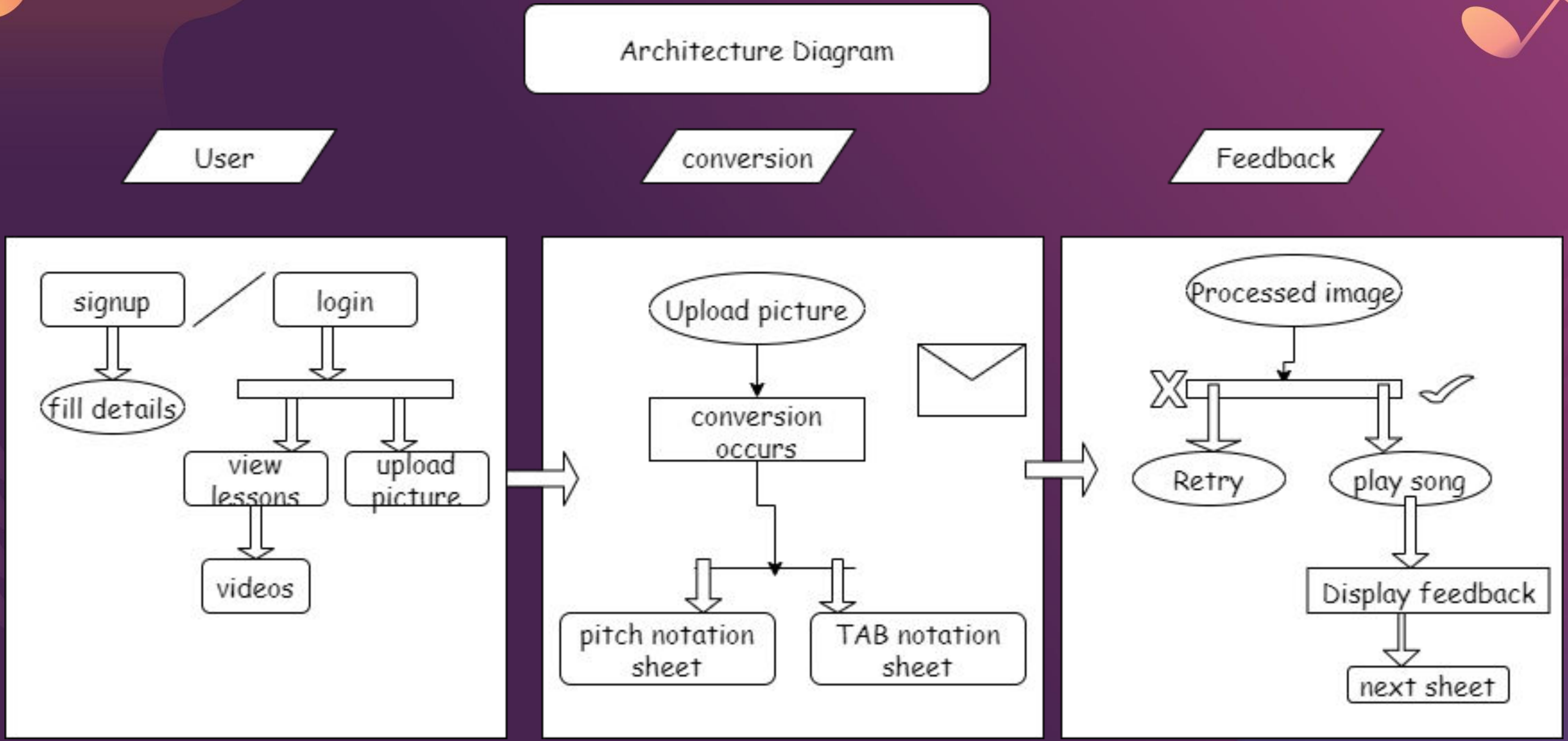
▹Distinguishing themselves what each note stands for and how to play that note would be a tough task that could discourage them from continuing to play the instrument.

▹If they are learning individually, it would also be hard to recognize if the notes they are playing are correct without any feedback to correct.

Existing Solution

- There are many music learning App like Music Theory learner, which makes the user to understand the theory concepts of music.
- Also explains the notes and concept but there is no opportunity of User to upload the music sheet available with them.
- Our Application Allows the user to upload their interested music sheet and understand the notes, pitch notation and hand notation to help them play the note.
- Thus it helps the beginners to understand each notes and play the music.
- Feedback of frequency is also there to provide real-time learning

Architecture



Architecture Diagram

User

conversion

Feedback

signup

login

fill details

view lessons

upload picture

videos

Upload picture

conversion occurs

pitch notation sheet

TAB notation sheet

Processed image

X

Retry

play song

Display feedback

next sheet

Creative and Innovative Component

- Here we are helping the students learn how to play an instrument without having any tutor.
- We will provide images for how to play the notes in different instruments.
- Students will be able to learn very easily based on the visual representations and correct their mistakes.
- This way of learning improves the performance of the student(it compares with the original note and gets the feedback and thus learn better).

Module 1:
Convert

Module 2:
Feedback

01

Convert Module

01

Convert Module

What it does:

❑ This Module is part of the back-end process.

❑ User will be able to upload an image of the music sheet and select which instrument its is for

❑ 5 steps are taken to process the image and return an output to the user

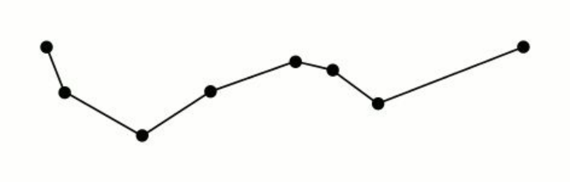
What it Returns:

❑ An image with Pitch Notation (Relative Frequency of each Note)

❑ An image with TABS Notation (Helps user learn the finger pattern for given note)

Step 1: Adjust Photo

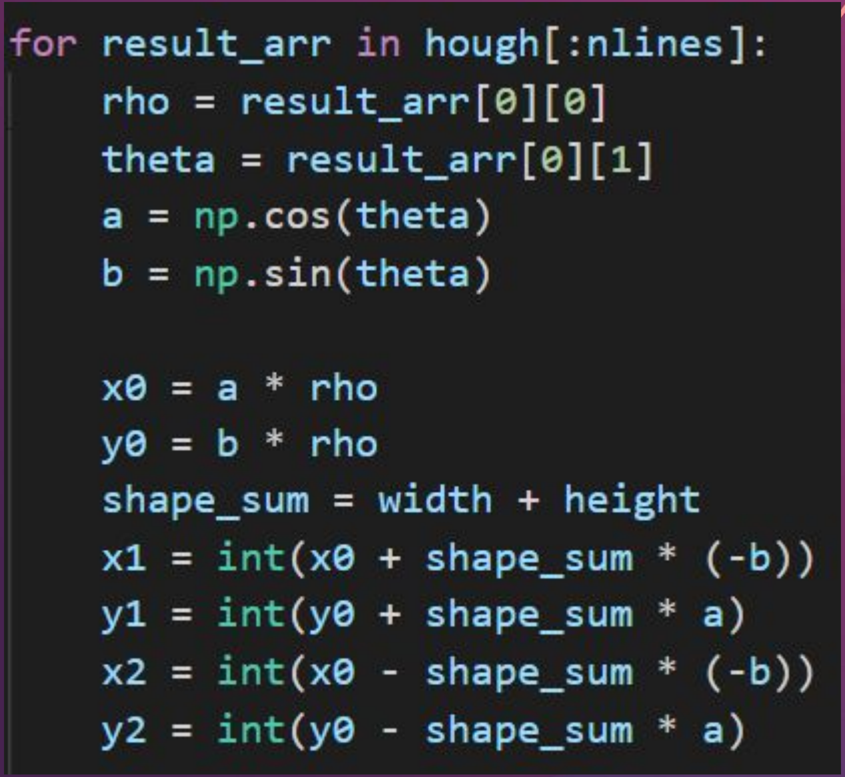
- ❑ Change the user uploaded image to grayscale (work with less colors)
- ❑ Blur with Gaussian Blur function. This will smooth picture and reduce detail (better for detecting edges
- ❑ Use the Canny method to then detect edges in the image
- ❑ To find the music sheet in image uploaded:
    - ❑ Douglas-Peucker Algorithm to reduce the number of points in the curve (to find the width and length lines of paper)
    - ❑ Calculate max width and length of sheet, find the 4 points and return the coordinates as rectangle
    - ❑ warpPerspective (given the size of the music sheet) transforms 2D image to only contain the sheet



Step 2: Detect Lines

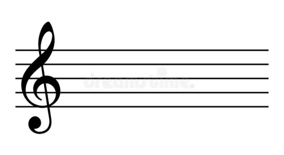
- Hough Transform Function
    - Used to detect straight lines
    - We use it to return a list of horizontal lines
    - Used to detect staff in later step

```
hough =
cv2.HoughLines(processed_image,
1, np.pi / 150, 200)
```



```
for result_arr in hough[:nlines]:
    rho = result_arr[0][0]
    theta = result_arr[0][1]
    a = np.cos(theta)
    b = np.sin(theta)

    x0 = a * rho
    y0 = b * rho
    shape_sum = width + height
    x1 = int(x0 + shape_sum * (-b))
    y1 = int(y0 + shape_sum * a)
    x2 = int(x0 - shape_sum * (-b))
    y2 = int(y0 - shape_sum * a)
```



Step 3: Group Staffs

- A Staff in music notation is considered to be 5 lines grouped together
- Calculate the approximate positions of the separate lines from the horizontal lines list we obtained in last step
  - If current line is far away from last detected line (Threshold given: 50) and lines detected is still greater than 5, consider it as start of new staff
  - If less than 5 lines, consider it has an anomaly and discard from list
  - Groups each 5 lines as a list of staffs
    - Staff hold the minimum and maximum range for the 5 lines
  - Draw over detected start and end lines of each staff in yellow

Step 4: Detect Blobs

- ❏ Remove horizontal lines, then Remove vertical lines
  - ❏ getStructuringElement(): pass in size and shape
    - ❏ Use one for detecting horizontal lines
    - ❏ Use other for detecting vertical lines
  - ❏ Passed to MorphologyEx()
    - ❏ erosion followed by dilation using MORPH_OPEN
    - ❏ To remove lines from image
- ❏ Use SimpleBlobDetector Algorithm to detect the blobs (notes in sheet)
  - ❏ Set up with default parameters
  - ❏ 4 steps: Thresholding, Grouping, Merging, Center & Radius Calculation
  - ❏ Enumerate blob and store as blobs list

Step 5: Extract/Map Note from Blob

- Blob list is sent along with which staff it is part of
- Each Blob is mapped as a Note Object
- Note Class represents single note and contains:
  - Position on Staff, Staff number, clef, center pt, pitch, hand
  - center pt: used for detecting position
  - Pitch, Hand: mapped from key based on position
- Detect_Position_on_Staff:
  - generating range of 3 upper lines and 3 lower lines for each beginning and end of staff (for notes that are outside of the traditional 5 lines)
  - Calculate distance of note's center point from nearby lines
    - Based on Threshold of 20

Step 5: Extract/Map Note from Blob

- ❑ If it is less than threshold, then consider note to be in between two staff lines
- ❑ Otherwise the note is considered to be placed on the staff line using the blob's center point
- ❑ Detect_pitch and Detect_hand:
    - ❑ Based on instrument selected, specific key will be selected where values range from -6 to 14 (these numbers based from the position calculated in previous step)
    - ❑ Based on the value calculated, the pitch notation and hand notation will be mapped to the note and stored in Note Class
- ❑ Make two images
    - ❑ Image with mapped Pitch Notation inserted (Blue)
    - ❑ Image with mapped Hand Notation inserted (Red)

02

Feedback module

02

Feedback module

- ❑ Using this module, it gets the audio input from the instrument (for example guitar)
- ❑ From this we find the note and its octave based on frequencies of the audio
- ❑ Every frequency of sound that is detected is then mapped to get the key and based on its strength it gives the octave.
- ❑ Through this module the user gets a better understanding on what he/she is playing as it gives the visual representations based on the key and its frequencies,the user will understand the proper pitch play the song.
- ❑ It also helps user to learn music with a more deeper understanding.
- ❑ The user can see the key notation while playing the song, so that they could correct their errors and learn properly.

Working:

- Here we are using MediaDevices.getUserMedia() API Navigator.getUserMedia() API.

- Navigator.getUserMedia() and MediaDevices.getUserMedia()prompts the user asking permission to use audio and video. Here we are using this as we are getting audio from user as the source for MediaStream.
- Checks if the user has given permission for the audio.
- If yes it goes to microphonePermissionGranted()

- Else it goes to microphonePermissionNotGranted() and displays error saying “no audio to work with”.

In microphonePermissionGranted()

- ❑ Performs Fast Fourier Transform (FFT) to get sampled frequency domain data and store it in frequency data.
- ❑ Then create bins as SVG so that they are scalable and append it to the mask so as to present it visually on the screen.

In draw()

To get the key:

- ❑ From the frequency data we calculate the key
- ❑ We first find out the highest frequency in the array and check if it's greater than the threshold(60)
- ❑ If it's greater than the threshold i.e 60 then it maps the key to the note and key to the color.
- ❑ Else we display its so silent.

Convert frequency to key:

Get the frequency and convert this to key using

12*log(freq/440)+49

Mapping key to the note:

Here we are using these 12 standard keys to get the note
['G#','A','A#','B','C','C#','D','D#','E','F','F#','G']
This is the order according to their frequencies and pitch

(key%12) gives the note and to find the octave we use key/12 this gives the pitch of the note.

Mapping key to color:

Based on the frequencies the key changes accordingly the color also changes accordingly.

Tools and Technology Used

❑ OpenCV

It is a huge open-source library platform used for computer vision and machine learning application. It is used in applications like image processing and object detection.

To detect the music sheet page for blob detection and detect the staff position.

❑ Numpy

NumPy is a python library that adds support for large, multi-dimensional arrays and matrices. It also has a large collection of high-level mathematical functions to operate on those arrays.

Tools and Technology Used

❑ Django

Django is a high-level open-source Python Web framework that is used for the creation of complex, database-driven websites. It emphasizes on re-usability of code components and quicker development of applications.

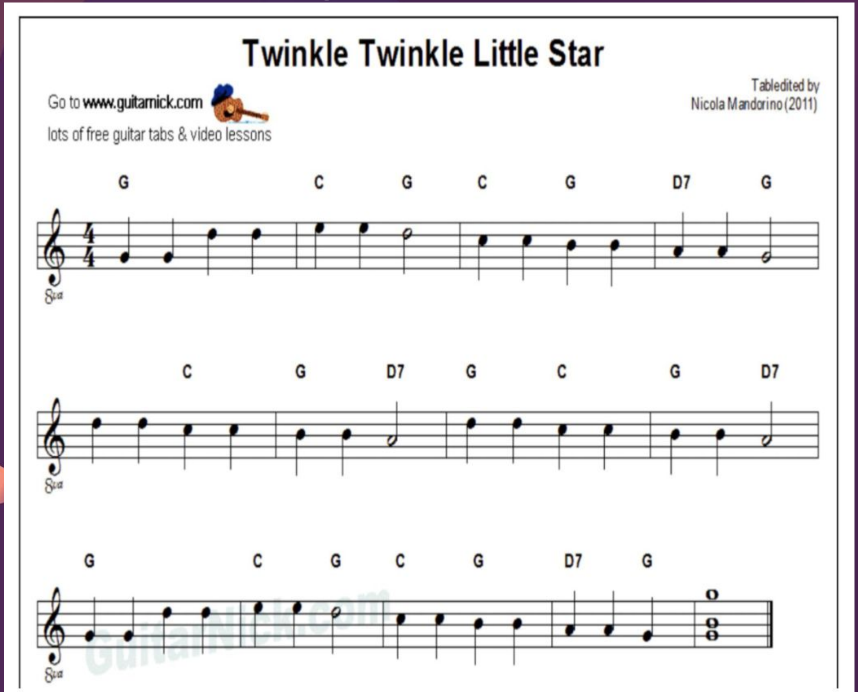
❑ SQLite3

SQLite3 is a relational database management system for SQL. It is not a client server database engine but is embedded into the end program creating databases that are faster to work with.

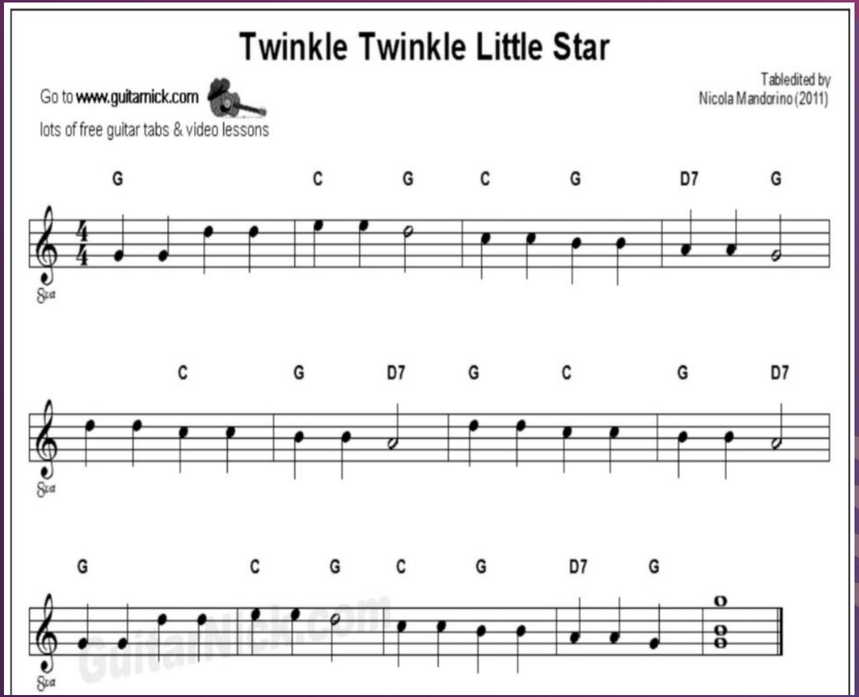
Results - 1:

Based on Guitar Music Sheet uploaded

Original/ Uploaded

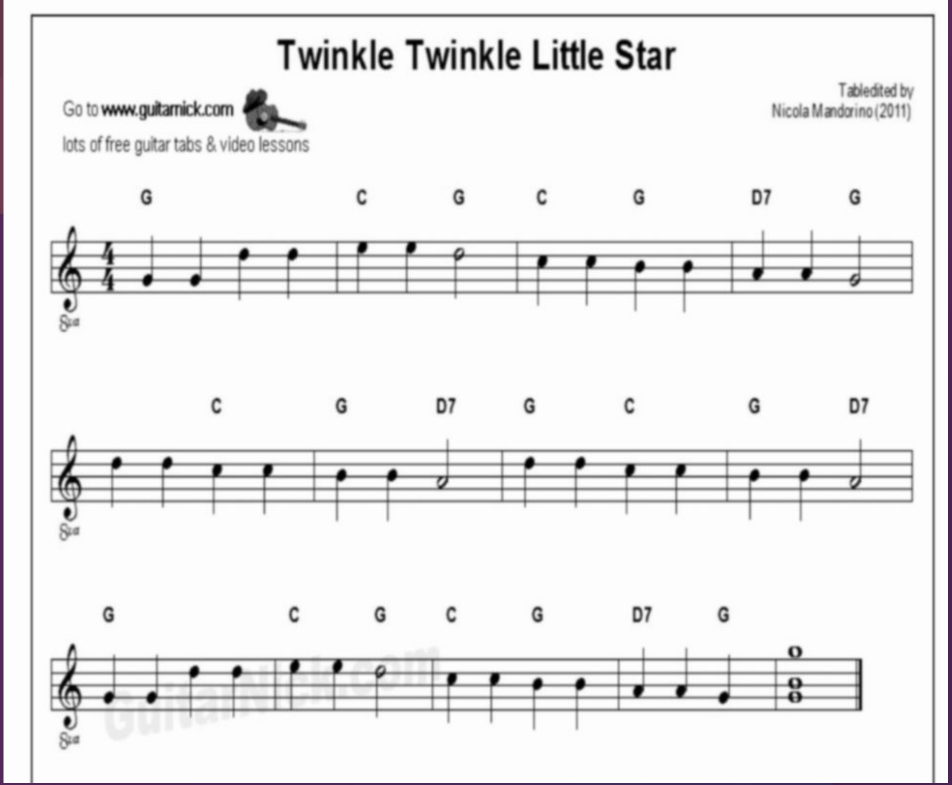


Grayscale

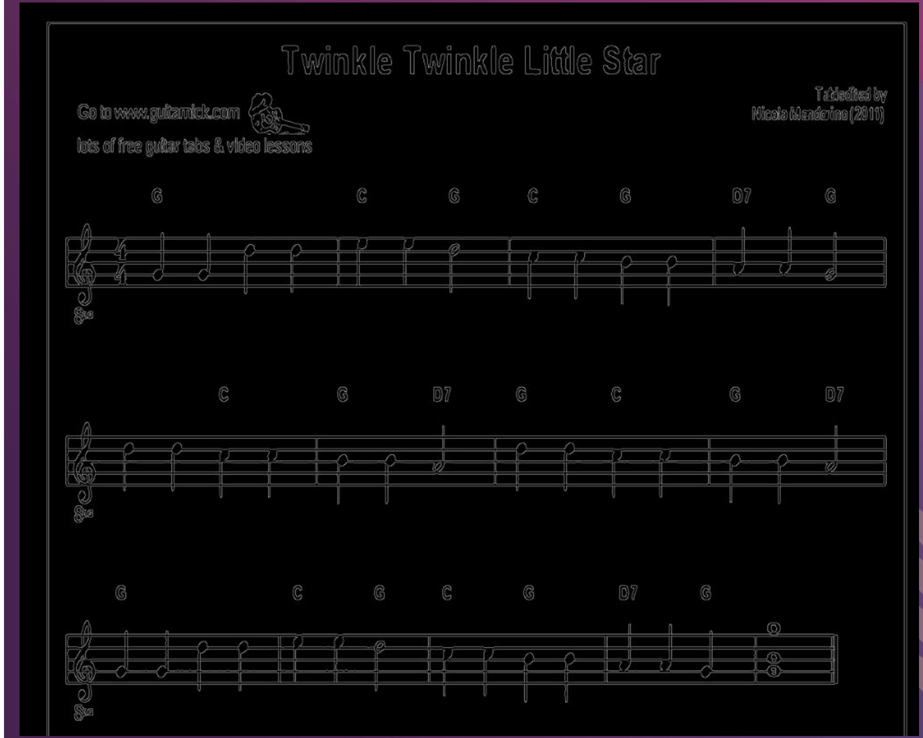


Results - 2:

Gaussian Blur

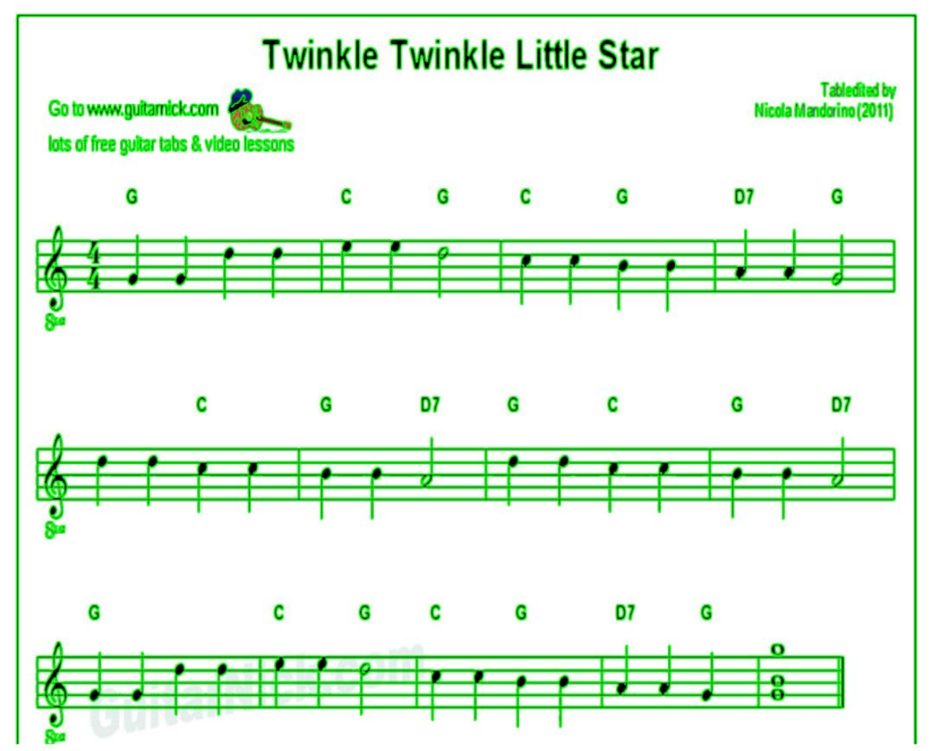


Canny

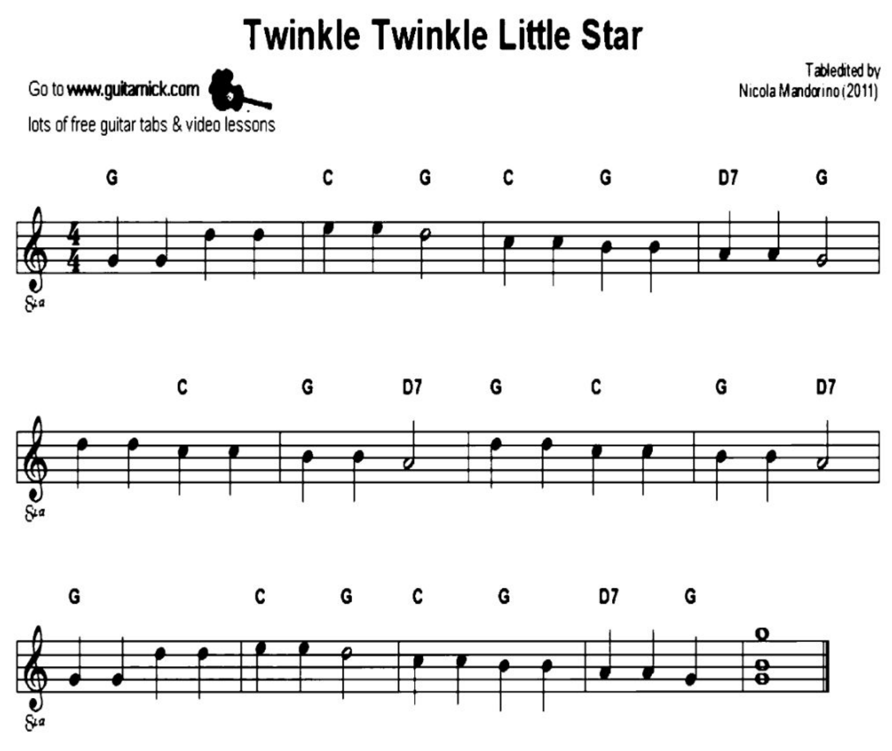


Results - 3:

Contour

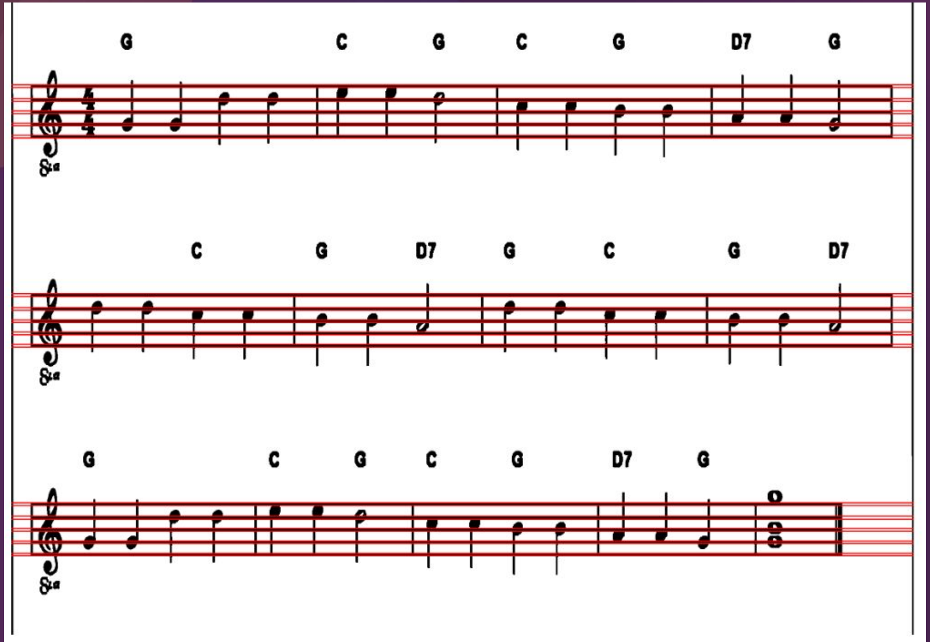


Adjusted

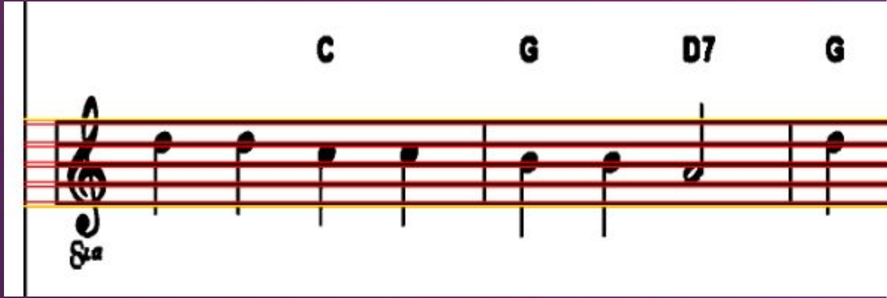


Results - 4:

Lined Detected (Red)

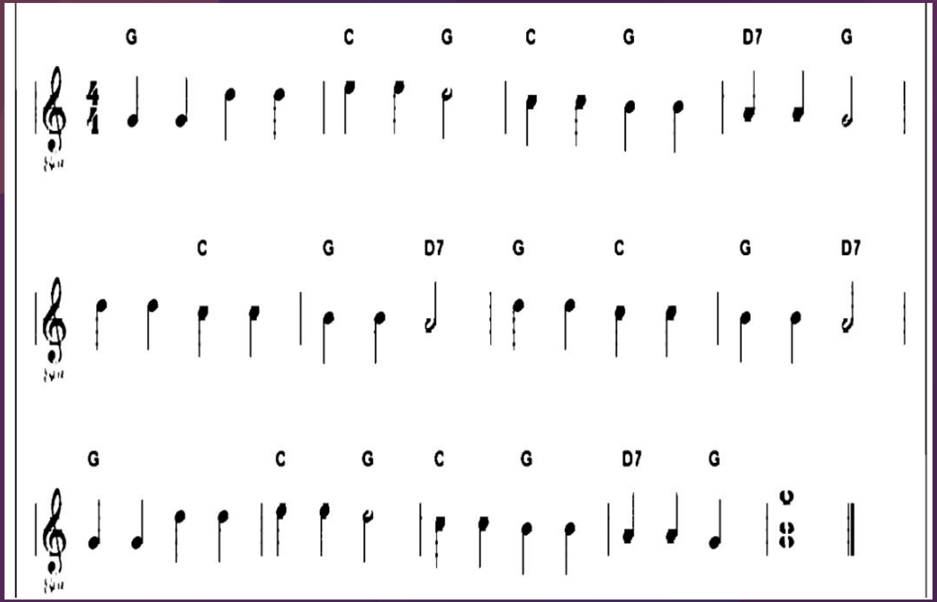


Staff Grouped (Yellow)

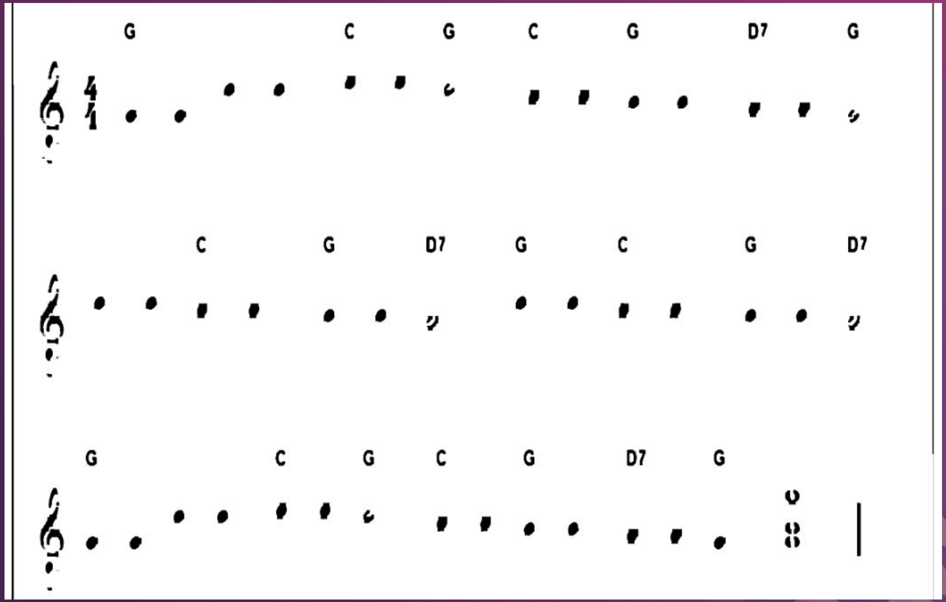


Results - 5:

Remove Horizontal Lines

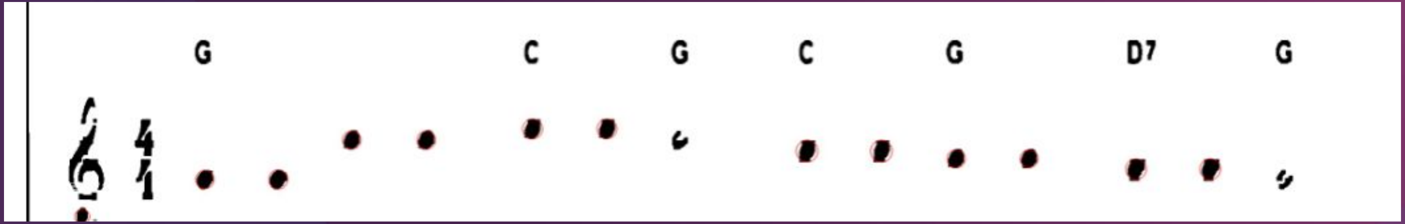


Remove Vertical Lines

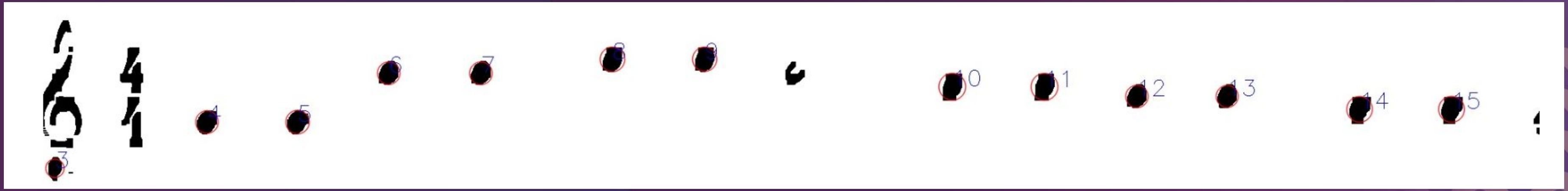


Results - 6:

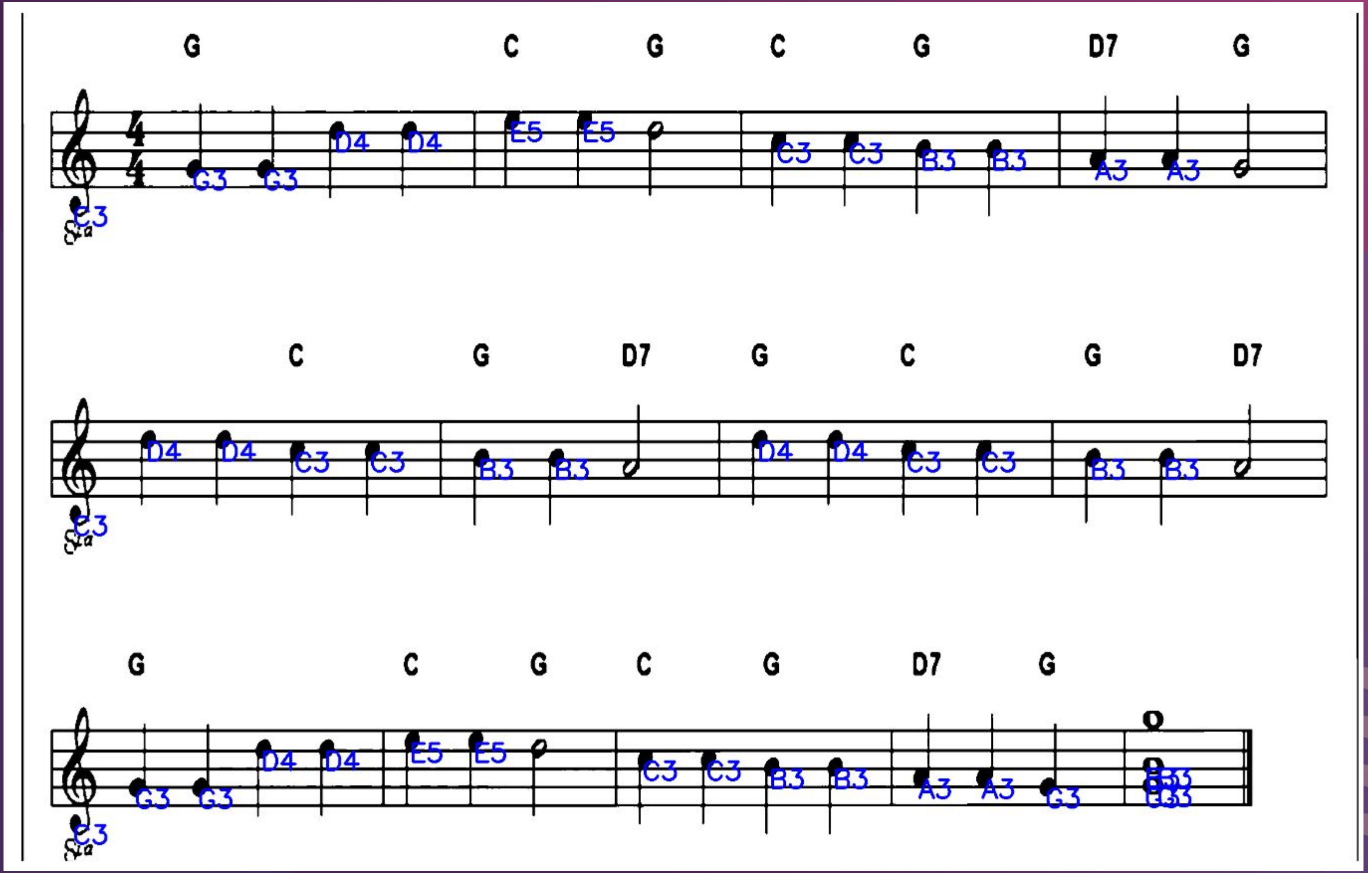
Blobs Detected



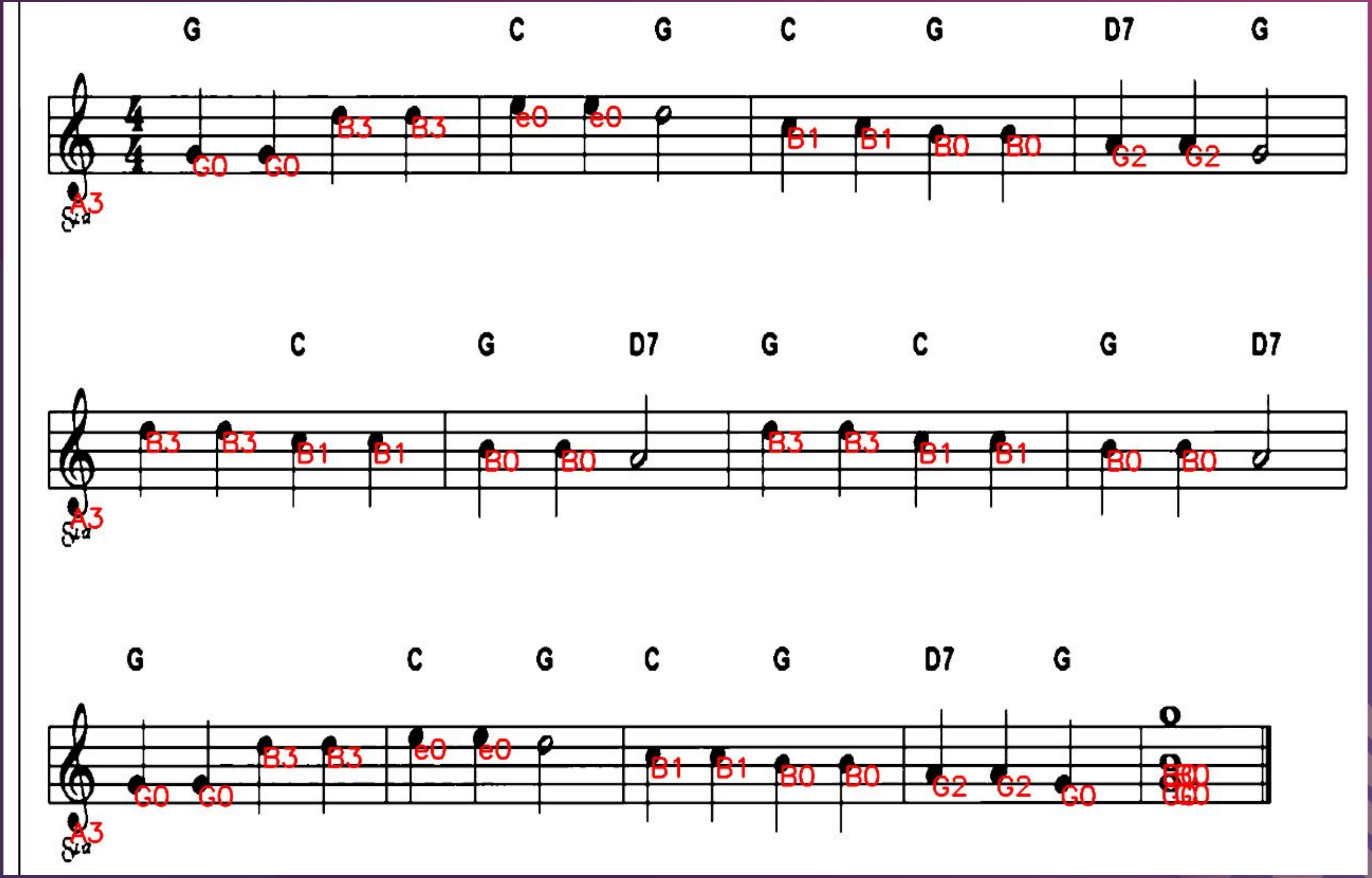
Blobs Numbered



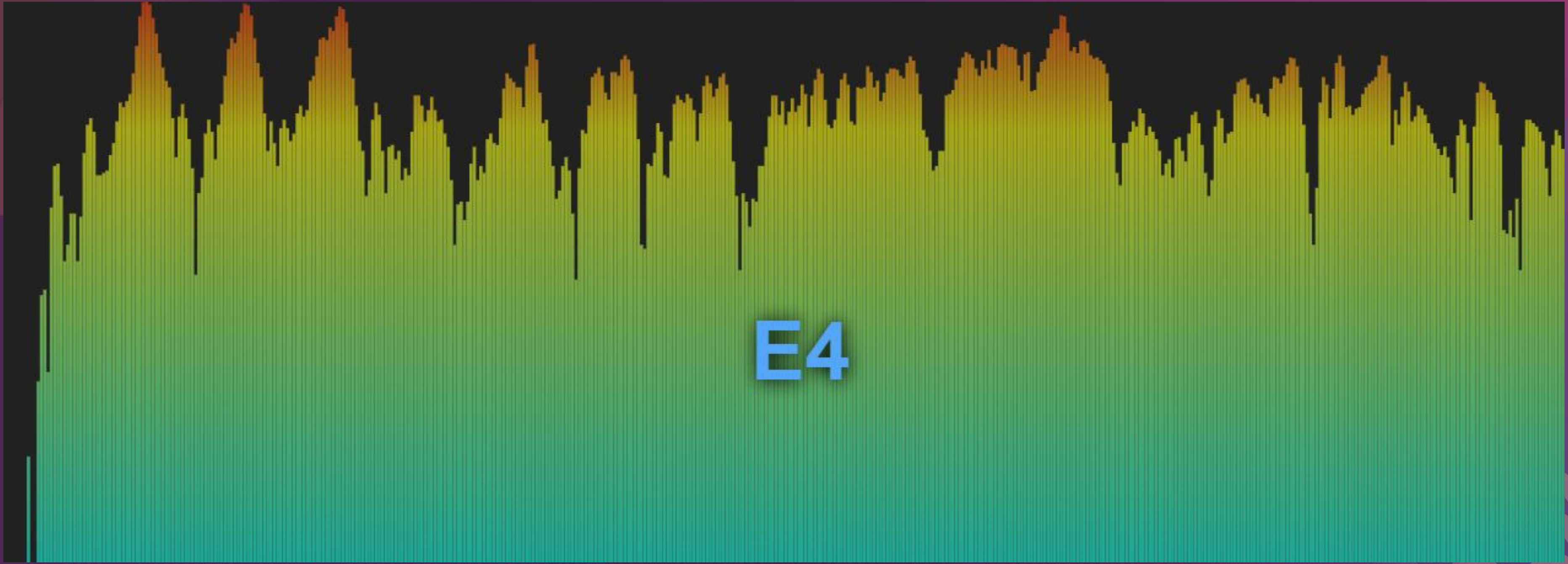
Final Result: Image with Pitch Notation


G C G C G D7 G
C3
G3 G3 D4 D4 E5 E5 C3 C3 B3 B3 A3 A3
C G D7 G C G D7
C3
D4 D4 C3 C3 B3 B3 D4 D4 C3 C3 B3 B3
G C G C G D7 G
C3
G3 G3 D4 D4 E5 E5 C3 C3 B3 B3 A3 A3 G3
B3 G3

Final Result: Image with Hand Notation


G C G C G D7 G
4/4
A3 G0 G0 B3 B3 e0 e0 B1 B1 B0 B0 G2 G2
C G D7 G C G D7
A3 B3 B3 B1 B1 B0 B0 B3 B3 B1 B1 B0 B0
G C G C G D7 G
A3 G0 G0 B3 B3 e0 e0 B1 B1 B0 B0 G2 G2 G0

Feedback



References

[1]Blob Detection using OpenCV. https://learnopencv.com/blob-detection-using-opencv-python-c/.

[2] Pitch Detection. https://teropa.info/blog/2016/08/10/frequency-and-pitch.html.

[3] Edges and Contour Basics. https://towardsdatascience.com/edges-and-contours-basics-with-opencv-66d3263fd6d1.

[4] C Major Scales for Guitar. https://www.cyberfret.com/guitar-scales/open-position-major-scales-for-guitar/.

[5] Run External Python Script in Django. https://www.youtube.com/watch?v=ZCV7atq7l8g&t=1400s.

[6] Image Processing Tutorial Using Python. https://www.youtube.com/watch?v=sfheWK72L74.

[7] How to Read Guitar Sheet Music. https://www.youtube.com/watch?v=vZ7TJKtUYbO

Thank You for Listening