

# 기타 TAB 악보를 위한 CNN을 활용한 손 탐지 모델

20190048 강현준

## Hand detection for Guitar TAB using CNN

20190048 Hyonjun Kang

### 요 약

Convolution layer를 활용한 구글의 mediapipe의 오픈소스 API를 활용한 손 탐지 모델과 Opencv의 고전적인 컴퓨터 비전 기법, Morphology, Sobel Edge detection, Hough line Transform를 활용하여 기타 TAB 악보를 비전 기반으로 제작해본다.

### Abstract

In the report, we implemented google's mediapipe open API hand detection model using convolutional layers and traditional computer vision techniques from opencv such as Morphology, Sobel Edge detection, Hough line Transform to implement a TAB generator using computer vision.

### Key words

mediapipe, opencv, CNN, morphology, sobel edge detection, hough line transform, hand detection, fret detection

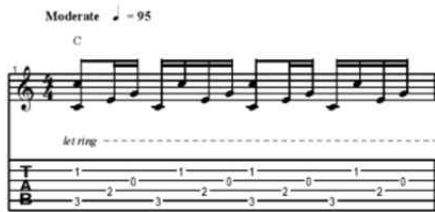
## 1. 서 론

악보는 악기를 연주, 음악을 기록하는 데에 있어 필수적인 요소다. 오선지 악보는 음의 높낮이와 박자만을 나타낸 악보로, 모든 악기에 공통으로 존재한다. 하지만 현악기의 경우, 왼손의 운지법과 음의 작용선을 나타낸 TAB(Tablature) 악보라는 것이 존재한다. 여기서 운지법이란 왼손이 현악기의 현을 잡고 있는 프렛(현악기의 넥의 주변보다 높은 부분)을 나타낸다. 운지법을 제대로 잡는다면 현악기의 프렛을 넘나들며 연주할 수 있지만, 운지법을 제대로 잡지 않는다면 손가락 움직임이 불편해져 연주에 지장을 준다. 여기서 초보 TAB 악보의 경우 프렛의 번호와 더불어 손가락의 위치를 덤으로 알려

주는 경우가 많다. 하지만 일정 수준 이상의 TAB 악보는 이를 생략하고 제작하는 경우가 다반수이다. 이는 초보자, 혹은 곡을 처음 연습하는 기타 플레이어들에게 특정 곡, 혹은 기타 자체에 진입장벽을 만드는 것으로, 현악기의 악보 보는 실력을 나타내는 지표로도 활용될 수 있다.

이 중요한 운지법을 우리는 사실 악보만을 보고는 판단하기 어렵다. 사람마다 성향이 다르고, 초보자의 경우에는 감조차 오지 않는 경우가 다반수일 것이다. 기타의 경우 TAB 악보의 공유와 제작이 하나의 문화로 자리 잡아 기타 연주영상, 혹은 음악의 악보만을 가지고 TAB 악보를 만드는 것이 대중적이다. 따라서 본 보고서에서는 기타의 연주영상을 가지고 손가락의 위치와 기타의 넥의 위치를 알아

그림 1. TAB 악보



### 1) 손가락 탐지

## 그림 2. VGG16

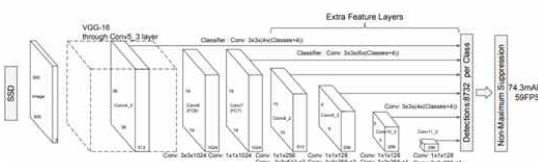


Figure 1 illustrates the architecture of the proposed multi-stage pose estimation network. The network consists of multiple stages, each processing an input image to produce a pose estimation result.

**(a) Stage 1:** The input image  $x$  is processed by a Convolutional Pose Machine (CPM) with pooling (P) and convolution (C) layers to produce a feature map  $b_1$ .

**(b) Stage  $\geq 2$ :** The feature map  $b_1$  is combined with a heatmap  $x'$  via element-wise multiplication ( $\psi_2$ ) to produce  $b_2$ . This process repeats for subsequent stages, where  $b_{t-1}$  is combined with  $x'$  via  $\psi_T$  to produce  $b_T$ .

**(c) Stage 1:** Detailed view of the CPM architecture. The input image  $x$  is processed by a sequence of pooling (P) and convolution (C) layers to produce the feature map  $b_1$ .

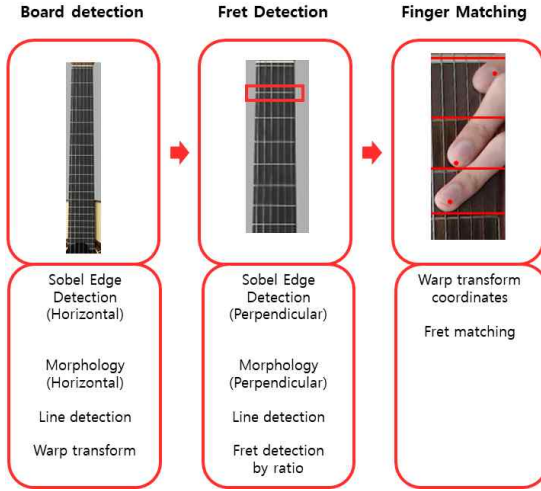
**(d) Stage  $\geq 2$ :** Detailed view of the multi-stage process. The feature map  $b_{t-1}$  is combined with the heatmap  $x'$  via element-wise multiplication ( $\psi_T$ ) to produce the final output  $b_T$ .

2) 기타 탐지

두 번째 방법은 과제로 진행한 특징점 매칭이다. 이 경우에는 가장 간단하게 진행한 뒤 한계점에 맞추어 진행하지 않았다. 기타의 넥과 플랫폼의 경우 마땅히 특징점이라고 할 부분이 존재하지 않아 정확도가 보장되지 않았다.

기타 탐지 모델은 크게 3부분으로 진행됐다. 기타의 넥(neck, board) 탐지, 각 fret 탐지, finger matching이다. 각 과정에서 고전적 기법 2~3가지가 사용되었다. sobel edge detection, morphological methods, Probabilistic hough line transform으로 이루어지는 이 과정[2]은 수직, 수평 성분의 프렛, 현을 탐지한다.

그림 4. 기타 탐지 모델



a) board detection

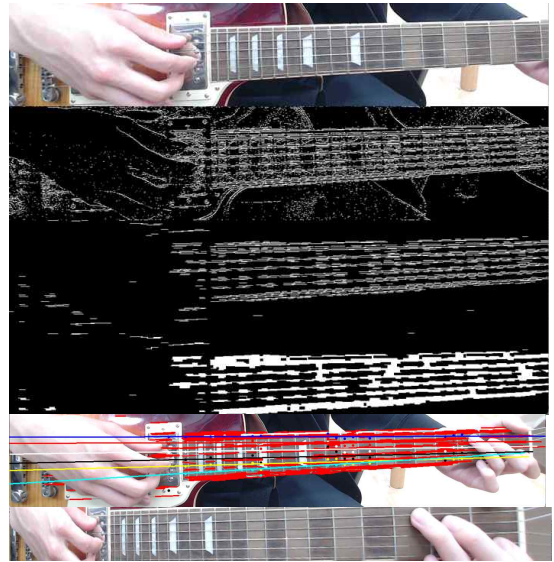
기타의 넥을 탐지하기 위해서는 우선 mediapipe의 손 감지 모델이 양손을 감지할때까지 기다린다. 이후 양손의 위치에 대한 1초의 buffer를 만든 뒤, 위치의 median을 활용하여 기타 넥의 위치를 가져온다. 이때 median을 활용한 이유는 자잘한 손의 움직임을 상쇄하기 위해서다. 또한 기타 넥의 위치를 특징하는 이유는 주변 배경에서의 쓰레기 데이터를 차단하기 위해서다.

이후 기타의 현의 특징을 활용해 넥의 상하 제한을 특징한다. 기타의 현의 간격은 일정하기 때문에 넥 이미지에서 세로로 간격이 일정한 점들을 잡아 현으로 단정하고 이를 활용해 넥을 특징한다. 이를 이해 우선 2<sup>nd</sup> order sobel edge detection을 활용하여 수평 성분의 엣지를 검출한다. 그 뒤 결과에서 수평 방향의 엣지들 중 길이가 긴 엣지를 남기기 위해 좌우로 긴 엣지를 남기기 위해 (1, 11) 커널을 활용한 MORPH\_OPEN[3]을 실행한다. 이 경우 좌우로 긴 엣지들만 남아 현을 검출할 가능성이 커진다. 여기서 추가할 점은 기타의 현은 선이 아닌 면이므로 현의 위, 아래에 엣지가 검출된다. 따라서 (5, 5) 커널을 활용하여 MORPH\_CLOSE를 활용해 위, 아래의 엣지를 합쳐준다. 여기까지 진행 한 뒤 Probabilistic Hough line transform[4]으로 선을 검출하면, 수평 방향의 line들이 더 잘 검출된다.

그 뒤로는 이미지의 세로 간격이 같은 6개의 점을 특징하여 현의 각도와 위치를 특징한다. 마지막

으로 프렛의 위치를 특정하기 위해 현의 연장선으로 이미지의 양 끝을 자른다. 이 이미지 또한 buffer를 사용해 자잘한 움직임을 상쇄한다. 끝으로 갈수록 가늘어지는 기타 넥의 특성상 직사각형으로의 transformation matrix가 특징된다. 최종적으로 특정된 이 matrix와 직사각형의 넥 이미지를 반환한다.

그림 5. 넥 탐지 과정



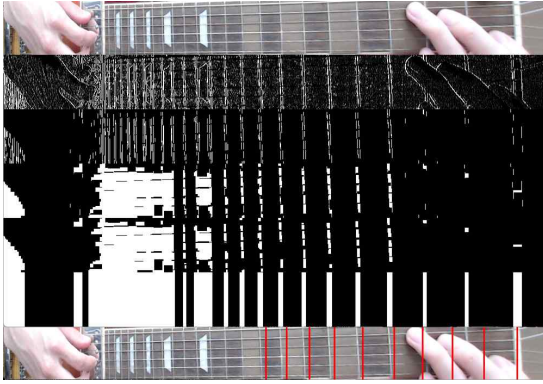
b) fret detection

기타의 넥을 특정 한 뒤는 운지법을 위한 fret을 탐지한다. 이 부분은 넥 탐지보다 morphology가 더욱 사용된다. 프렛의 경우 현보다 많은 수와 넓은 두께의 면이다. 따라서 엣지 검출 후 엣지를 가공하는 과정이 길다.

넥 탐지와 동일하게 2<sup>nd</sup> order sobel edge detection으로 수직 성분의 엣지를 검출한다. 그 뒤 (7, 1) 커널의 MORPH\_OPEN으로 수직 성분의 엣지를 강조한다. 그 뒤 (1, 11) 커널의 MORPH\_CLOSE로 거리가 있는 프렛의 엣지를 합친다. 이 과정에서 간격이 좁은 프렛의 경우는 지워지게 된다. 따라서 10개의 프렛을 검출하는 것이 한계다. 그 뒤 (1, 3)의 MORPH\_OPEN으로 얇은 성분을 지우고 남은 성분들을 프렛으로 가정한다. 위치를 특정한 뒤에는 흑여나 탐지하지 못한 프렛을 피타고라스 음계에 따라 채워 넣는다. 결과적으로 세로 방향의 10개의 프렛을 검출 할 수 있다.

### III. 결 론

그림 6. 프렛 탐지 과정

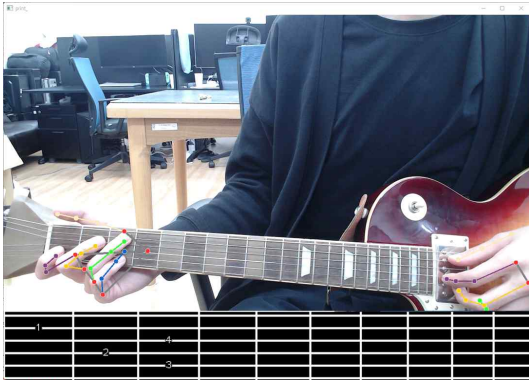


#### c) finger matching

손가락의 경우 mediapipe를 통해 평면 이미지에서 손가락의 위치는 이미 특정되었다. 따라서 우리가 넥 이미지를 구하기 위해 사용한 transform matrix를 활용해 좌표를 transform한 다음 프렛의 위치와 비교하여 손가락의 현재 현 번호와 프렛 번호를 구한다.

#### d) final result

그림 7. 최종 화면



위 화면은 손 위치/넥 위치 버퍼가 가득 차게 되면 넥과 프렛을 검출하고 위와 같이 손가락이 검출됨을 확인함과 동시에 화면 밑에 운지법이 출력된다. 실시간으로 진행되며 검출은 실시간에 가까운 fps 20 맞춰 진행된다.

본 보고서에서는 손가락 탐지를 위한 오픈소스 모델과 기타 탐지를 위한 고전적인 컴퓨터 비전 기법들을 활용해왔다. 결과적으로 특정 프레임(사진)에서 검출이 잘 되는 것을 확인했다. 하지만 그렇지 못한 프레임이 90%가 넘는 것 또한 확인했다. 그 이유로 첫 번째, 손 탐지 모델이 정확하지 못하다. mediapipe의 손 탐지 모델은 손의 모습이 온전한 경우는 정확하게 탐지한다. 그 예로 오른손의 손가락의 위치에 흔들림은 거의 없었다. 하지만 손의 절반이 가려진 왼손의 경우에는 흔들림이 강했다. 이로 인해 좌표가 불안정하여 운지법이 제대로 잡히지 않았다.

두 번째는 morphology 기법의 일정함이다. 커널에 사용된 변수들이 일정하여, 기타를 일정 거리, 일정 각도에 맞출 경우 넥과 프렛의 검출율이 높아졌지만 이 외에는 검출율이 그렇게 높지 않았다. 이는 고전적인 기법의 한계로 예상된다. 이를 해결하기 위해 커널의 크기와 연관된 다른 변수를 찾고 이를 계속 이어나가는 게 필요할 것으로 보인다.

세 번째는 기타의 정형화되지 않은 모양이다. 기타의 경우 넥에는 운지법에 도움이 되도록 하는 가이드 역할의 자개가 보통 박혀있거나, 기타의 헤드 부분이 장식된 경우가 많다. 이는 정형화된 현재 알고리즘상 방해가 되는 부분으로, 본 보고서에서는 마스킹을 통해 일정 부분 조절하였다. 이것이 제거된 경우에는 프렛이 아예 인식되지 않는 경우가 더 많아진다.

위 세 가지 이유로 현재 프로젝트는 부족한 점이 많다. 이에 대한 개선 방안으로 두가지를 제시한다. 첫 번째는 기타 탐지 모델을 model tracking으로 바꾸는 것이다. 두 번째는 기타와 손가락 탐지 모델의 기법을 바꿔 진행하는 것이다. 이 경우 손가락은 general한 모델이 아닌 기타에 의해 손이 반정도 가려진 상황에서의 모델을 만드는 것이 중요하겠다. 기타 탐지는 기타의 기울어짐 정도는 파악하는 것이 이 개선점이다. 위치를 프로젝트 몇 번 진행하다보면 더욱 완벽한 모델을 만들 수 있을 것으로 보인다.

## IV. 부 록

GitHub 주소

[https://github.com/fender2758/CAU\\_CV\\_projects/tree/main/project2/src](https://github.com/fender2758/CAU_CV_projects/tree/main/project2/src)

## 참 고 문 헌

- [1] <https://google.gitleab.io/mediapipe>
- [2] Joseph Scarr, Richard Green, Retrieval of Guitarist Fingering Information using Computer Vision (n.p.: 2010 25th International Conference of Image and Vision Computing New Zealand, n.d.), 1-2.
- [3] <https://webnautes.tistory.com/1257>
- [4] [https://opencv24-python-tutorials.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_houghlines/py\\_houghlines.html](https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html)
- [5] [https://en.wikipedia.org/wiki/Pythagorean\\_tuning](https://en.wikipedia.org/wiki/Pythagorean_tuning)