

Araştırma Makalesi

RASPBERRY PI TABANLI GİTAR AKORT CİHAZI

Ömer Faruk ACAR[†], Serhan YARKAN^{††}[†] İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, Türkiye^{††} İstanbul Ticaret Üniversitesi, Mühendislik Fakültesi, İstanbul, Türkiye

omerfarukacar@outlook.com, syarkan@ieee.org



0000-0001-8471-2619, 0000-0001-6430-3009

Atıf/Citation: ACAR, Ö., F., YARKAN, S., (2022). Raspberry Pi Tabanlı Gitar Akort Cihazı, Journal of Technology and Applied Sciences 5(2), s.1-10, DOI: 10.56809/icujtas.1127936

ÖZET

Akort, müzik aletlerinin çalındığında istenilen notayı vermesi için müzik aletlerinin ayarlanması işlemidir. Doğru notayı çalabilmek istenilen müziği elde edebilmek için önemli bir adımdır. Müzik aletlerinin akort edilebilmesi için çeşitli yöntemler bulunmaktadır. Bu çalışmada; gitar akort sisteminin oluşturulması, sanatçıların gitar ses kayıtları ile çalınan müziğin kıyaslanması ve gerçekleştirilmesi ele alınmıştır. Donanım için Raspberry Pi, yazılım için Python programlama dili kullanılmıştır. Gerçekleştirilen sistemde, Raspberry Pi mikrofon ile kullanıcılardan alınan gitar seslerini okur ve verilen girdi yazılıma gönderilir. Uygulama ile kullanıcıdan alınan sesler yazı formatına dönüştürülerek sanatçılardan alınan gitar sesleri ile kıyaslanabilir hale gelmesi sağlanmaktadır. Yapılan kıyaslama ile kullanıcılar ve sanatçılar arasındaki nota farklılıkları gözlemlenebilir hale gelmiştir.

Anahtar Kelimeler: Akort, Gitar, Python, Raspberry Pi.

RASPBERRY PI BASED GUITAR TUNER

ABSTRACT

Tuning is the process of musical instruments to give the desired note when played. Playing the right note is an important step in achieving the desired music. There are various methods for tuning musical instruments. In this study; creation of the guitar tuning system, comparison and implementation of the guitar sound recordings of the artists and the music played are discussed. Raspberry Pi is used for hardware and Python programming language is used for software. In the implemented system, Raspberry Pi reads the guitar sounds received from the users with a microphone and the given input is sent to the software. With this application, the sounds received from the user are converted into text format, making them comparable to the guitar sounds received from the artists. With the comparison, note differences between users and artists became observable.

Keywords: Guitar, Python, Raspberry Pi, Tuning.

Geliş/Received	:	14.06.2022
Gözden Geçirme/Revised	:	07.07.2022
Kabul/Accepted	:	12.07.2022

1. GİRİŞ

Teknolojik ilerlemeler ile çeşitli müzik aletleri ortaya çıkmıştır. Günümüz teknolojisinde doğru müzik notalarını ortaya çıkartmak için çeşitli yöntemler oluşturulmuştur. Bu yöntemlerin başında dijital akort cihazları gelmektedir.

Bu çalışmadaki amaç, Raspberry Pi kullanarak açık kaynak kodlu gitar akort sistemini oluşturmaktır. Raspberry Pi kullanılmasının sebebi Linux işletim sistemine sahip olup birçok programlama dilini desteklemesi ve donanımsal olarak giriş çıkış birimlerinin olmasıdır. Bu doğrultuda öncelikle programlama dili olarak Python programlama dili seçilmiş ve ucuz bir mikrofon kullanılmıştır. Gerçekleştirilen sistemde, Raspberry Pi mikrofon ile kullanıcılardan alınan gitar seslerini okur ve verilen girdi yazılıma gönderilir. Uygulama ile kullanıcıdan alınan sesler yazı formatına dönüştürülerek sanatçılardan alınan gitar sesleri ile kıyaslanabilir hale gelmesi sağlanmaktadır. Yapılan kıyaslama ile kullanıcılar ve sanatçılar arasındaki nota farklılıkları gözlemlenebilir hale gelmiş ve bu farklılık ile gitarın akort işlemi yapılabilir hale gelmiştir.

Bu çalışmada gitar notalarının bulunması, dijital akort işlemleri ve performans arttırmak için literatür araştırılması yapılmıştır. Ardından donanımsal gereklilikler, yazılımsal tasarım konuları anlatılmış ve uygulama bölümüne geçilmiştir.

2. LİTERATÜR ARAŞTIRMASI

Bu bölümde müzik aletlerinin dijital akortlarının yapılması ve dijital gitar akort yöntemleri üzerine yapılan çalışmalar incelenmiştir. Bir müzisyenin güzel müzik çalabilmesi için mükemmel akortlu cihazlara sahip olması gerekmektedir. Acemi bir müzisyenin bu akort ayarını yapabilmesi eğitimsiz bir kulak için oldukça zor olabilir. İnsan kulağı 20Hz ile 20KHz arasındaki sesleri iştebilmektedir. Belirli bir ses oktav seviyesi için, notanın frekansı gerekli temel frekansından daha yüksekse, buna keskin nota denir. Benzer şekilde, gerekli temel frekansının altına inerse, düz nota olarak adlandırılır (Thakuria ve ark., 2019).

Müzisyenler müzik aletlerini akort etmek için ortalama 10 dakika harcamaktadırlar. Aynı zamanda okullardaki öğrencilerin müzik aletlerinin akortları öğretmenleri tarafından yapılmakta ve ciddi bir zaman kaybına yol açmaktadır. Geleneksel yöntemlerle ayarlanan müzik aletlerinde her bir tel elle gerginliği ayarlanarak doğru frekansı yakalamak üzerine olmuştur (Bocanegra, 2005).

Her bir notanın farklı frekansı bulunmaktadır. Müzik aletlerinin çıkardıkları frekanslar farklılık göstermektedir. Kaminskyj ve Materka müzik aletlerinin kaynağını otomatik olarak tanımlayan bir sistem gerçekleştirmiştir (Kaminskyj ve Materka, 2015). Bu çalışmada 12 dizili gitar kullanılmıştır. 12 dizili gitarın minimum frekansı 440Hz olup maksimum frekansı 880Hz'dir. (Sturman, 2012). Her bir notaya karşılık gelen frekansın tablosu Tablo 1'de gösterilmiştir. Bu frekans değerleri doğru notayı bulmak için yazılımda kullanılmıştır.

Tablo 1. 12 Dizili Notaların Frekans Değer Tablosu.

Nota Sırası	Nota	Frekans Değeri
0	A	440
1	A#	466.16
2	B	493.88
3	C	523.25
4	C#	554.37
5	D	587.33
6	D#	622.25
7	E	659.26
8	F	698.46
9	F#	739.99
10	G	783.99
11	G#	830.61
12	A'	880

Arduino mikro denetleyici kartı kullanılarak Kumar ve arkadaşları tarafından otomatik gitar akort cihazı tasarlanmıştır. Tasarımda gelen sinyaller MATLAB çalıştırılan bir sistemi besler. Oluşan frekans farkı hesaplanarak PWM (Pulse Width Modulation) sinyalleri motor sürücüsünü besler ve motor ile gitar tellerinin gerilimi ayarlanarak gitarın istenilen akort seviyesine getirilmesi sağlanmıştır (Kumar ve ark., 2017). Benzer şekilde MATLAB programlama dili kullanılarak Regeena ve Saji tarafından grafik ara yüzü olan gitar akort sistemi geliştirilmiştir (Regeena ve Saji, 2009). Notaların belirlenmesi konusunda diğer bir çalışmada da MATLAB programı ile müzik notalarının belirlenmesi konusunda grafik arayüz programı oluşturulmuştur (Mamur ve ark., 2018).

3. DONANIMSAL TASARIM

3.1. Raspberry Pi ve Özellikleri

Raspberry Pi içinde bir bilgisayar için gerekli olan ram, işlemci, giriş ve çıkışlar gibi tüm birimleri tek bir devre kartı üzerinde toplayan sistemdir. Şekil 1'de görüldüğü gibi küçük boyutundan ve işlevsel olmasından dolayı akıllı evlerde, gömülü sistemlerde, robotik projelerde kullanılmaktadır. Arduino gibi kartların gücünün yeterli olmadığı yerlerde tercih edilmektedir. Linux üzerinde çalışabilecek nerdeyse tüm programlama dillerinde yazılım yazmasına imkan vermektedir. Raspberry Pi 4B Broadcom 2711 Dört çekirdekli Cortex-A72 64-bit destekli @1.5GHz işlemciye sahiptir. 2.4GHz/5.0GHz 802.11.b/g/n/ac destekli kablosuz ağ desteklemektedir. 2x USB 2.0 + 2x USB 3.0 portuna sahip olup, video için 2 x mikro-HDMI, 4K 60FPS destekli sistem kullanmaktadır. 4 kutuplu 3.5mm bağlantı üzerinden ses çıkışına sahiptir. Ses için bir girişe sahip olmayan Raspberry Pi harici 3.5mm USB Ses Kartı ile mikrofon bağlantısı sağlanmıştır.



Şekil 1. Raspberry Pi 4 4GB.

3.2. USB Ses Kartı

Raspberry Pi modelleri içerisinde dahili olarak ses giriş modülü bulunmamaktadır. Bu yüzden harici olarak ses kartı kullanılması gerekmektedir. Şekil 2'de mikrofon ile Raspberry Pi cihazının bağlantısını sağlamak için kullanılan ses kartı gösterilmiştir.



Şekil 2. USB Ses Kartı.

3.3. Stereo Mikrofon

Sesin doğru şekilde alınabilmesi için projede Şekil 3'te gösterildiği gibi stereo mikrofon girişi kullanılması gerekmektedir. Projede Şekil 4'te görülen ucuz mikrofon seçilmiştir.



Şekil 3. USB Stereo Mikrofon Girişi.



Şekil 4. Mikrofon.

3.4. Bağlantı Şeması

USB ses kartı, Raspberry Pi ve stereo mikrofon arasındaki bağlantı Şekil 5'te gösterildiği gibidir.



Şekil 5. Bağlantı Şeması.

4. İŞARET VE SİSTEM MODELLEME

Akustik işaret üretildikten sonra mikrofondan Raspberry Pi sistemine ulaşmasındaki süreç, kablosuz haberleşmede çok sık kullanılan doğrusal, zamanla değişen kanal yaklaşımı olarak modellenenmektedir (Sait ve ark., 2020). Karmaşık temel bant denklemi ile zamanla değişen, doğrusal kanalın birim darbe tepkisi t anında Doppler etkisinin gözlemlenebildiği verici ve alıcı arasındaki zamanın frekansını temsilen f_c altındaki:

$$h(t; \tau) = \sum_{l=0}^{L-1} g_l(t) e^{-j2\pi f_c \tau_l(t)} \delta(\tau - \tau_l(t)) \quad (1)$$

şeklinde ifade edilmektedir. Formül için $j = \sqrt{-1}$ olmak üzere L azami yankı sayısı g_l yankıya ait genliği τ_l gecikmeyi ifade etmektedir. Temel bantta $m(t)$ konuşma işareti ve $w(t)$ ortam gürültüsünü ifade ettiğinde, alıcıya ulaşan işaret:

$$r(t) = \int_{-\infty}^{\infty} m(t - \tau) h(t; \tau) d\tau + w(t) \quad (2)$$

şeklinde ifade edilmektedir. $r(t)$, ideal bir mikrofona yakın olduğu $L = 1$ (yankısız ve doğrudan) bağlı hareket içermeyen alıcı-vericinin zamanı arasında hiçbir kayma olmadığı:

$$r(t) = h \cdot m(t) + w(t) \quad (3)$$

şeklinde ifade edilmektedir. Bu durumda $h = g_0 e^{-j2\pi f_c \tau_0(t)}$ alıcının gelen işaretin yalnızca fazını ve genliğini değiştiren bir sabit olmaktadır. Akustik işaretlerde gelen sabit faz kullanıcının deneyimini etkilememektedir. (Toole, 2000) (Dang ve Lee, 2018).

Ortam gürültüsü için beyaz Gauss gürültüsü sıkça kullanılmaktadır. $w(t)$ rasgele süreci için olasılık dağılımı:

$$fW(w) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(w-\mu w)^2}{2\sigma^2}} \quad (4)$$

şeklinde ifade edilmektedir. $w(t)$ genelleştirilmiş Gauss biçimli olsa da $W \sim \mathcal{N}(0, \sigma)$ şeklinde ifade edilebilmektedir (Doire ve ark., 2017). Gürültüden bağımsız varsayılan h , çoğunlukla durağan rasgele süreç olarak değerlendirilip akustik enerjinin yoğun olan yönden az yoğun olan alıcıya doğru ulaştığı senaryoda, K yoğun enerjinin geldiği yöndeki gücün ortalamasının diğer yönden gelen gücün oranını göstermesi ile ve Ω ise toplam gücü göstermekte olup:

$$fH(h) = \frac{2h(K+1)}{\Omega} e^{-K - \frac{h^2(K+1)}{\Omega}} I_0 \left(2h \sqrt{\frac{K(K+1)}{\Omega}} \right) \quad (5)$$

şeklinde Rice dağılımı ile tanımlanabilmektedir. Mikrofona çok yakın olan konuşmacı için $K \rightarrow \infty$ olacağından (5)'in (4)'e yakınsadığı varsayılabilir. $m(t)$ süreci için olasılık dağılımı:

$$fM(m) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{|m|}{2\sigma^2}} \quad (6)$$

şeklinde ifade edilmektedir. Gürültülü konuşma işareti için olasılık dağılımları (4) ve (6) şeklinde olmalı ve sıfır etrafındaki karakterlerinin benzer şekilde olması, verinin gürültüden ayrılabilmesi için işaretlerin zamandaki uzunluklarının da dikkate alınması gerektiğini belirtir.

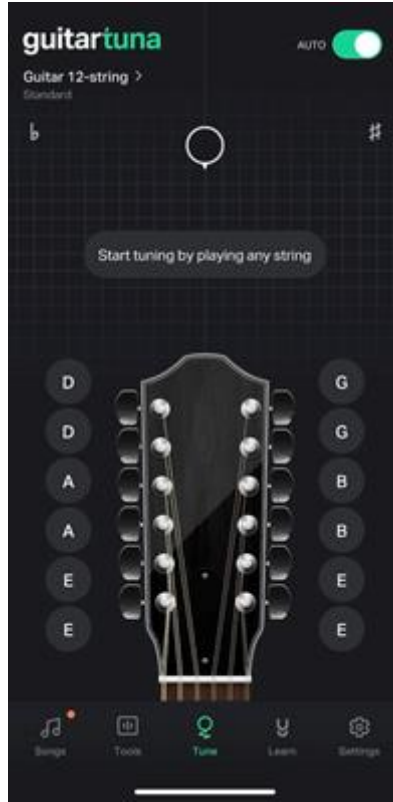
5. YAZILIMSAL TASARIM

Bu çalışmada Raspberry Pi 4 programlama kartına Rasbian işletim sistemi kurulmuş ve programlama dili olarak Python seçilmiştir. Kullanılan kodlar ve çıktıları <https://github.com/omerfarukacar/GuitarTuner.git> adresinden erişilebilir haldedir. İndirmek için `git clone https://github.com/omerfarukacar/GuitarTuner.git` komutu çalıştırılması gerekmektedir. Bu projede toplamda 4 farklı adımdan oluşmaktadır. İzlenen yol aşağıdaki gibidir.

- 1-Referans kaynaktan alınan seslerin .wav(Waveform Audio File) formatında kaydedilmesi.
- 2-Kayıtların kıyaslanabilmesi için dizi formatına dönüştürülmesi.
- 3-Kullanıcıdan alınan gitar seslerinin canlı olarak gösterilmesi ve kaydedilmesi.
- 4-Sonuçların dizi olarak kıyaslanması ve doğruluk oranının bulunması.

5.1. Referans Kaynaktan Alınan Seslerin .wav Formatında Kaydedilmesi

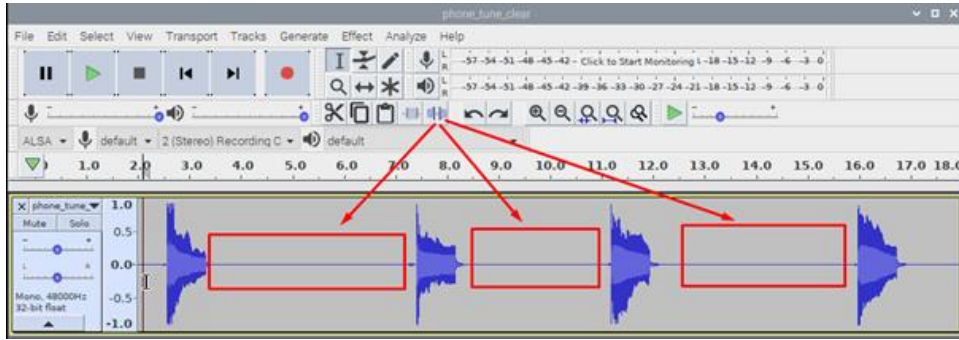
Referans kaynak olarak kullanılmak amacı ile Şekil 6'da gösterilen Appstore'dan indirilen GuitarTuna uygulaması seçilmiş ve 12 dizi standart ses oluşturulmuştur. Kayıtlar IOS işletim sistemli mobil cihazın (Iphone XR) mikrofonu ile Appstore'dan indirilen Kaydedici Pro uygulaması ile kaydedilmiştir. Şekil 7'de gösterilen kayıtlar uygulama içerisinde .wav formatı kaydedilebilmektedir. Kayıt sonrasında sesler arasındaki parazitler Audacity programı ile temizlenmiş ve temiz referans kaynak olarak kullanılmıştır. Temizleme işlemi Şekil 8'de gösterilen Silence Audio Selection butonu ile yapılmıştır. Çıktı `Phone_tune_clear.wav` ismi ile kaydedilmiştir.



Şekil 6. 12 Dizili Ses Oluşturmak İçin GuitarTuna Uygulaması.



Şekil 7. Kaydedici Pro Uygulaması ve .wav formatındaki ses kaydı.



Şekil 8. .wav formatındaki sesin temiz kaynak olması için parazitlerinin silinmesi.

5.2. Kayıtların Kıyaslanabilmesi İçin Dizi Formatına Dönüştürülmesi

Gitar sesinin yazı formatına dönüştürülmesi ve yazı formatında saklanması sanatçılar ile kullanıcılar arasında kıyaslama yapabilmeyi kolaylaştırmıştır. Gitar notalarının yazıya dönüştürülmesi konusunda farklı yazılım dillerinde farklı açık kaynak kodları mevcuttur (Vonsegger 2021). Program açık kaynak kodlu olup 12 dizili gitar frekansları açık kaynak koda eklenmiştir. Programın Çalışması İçin Gerekli Kurulumlar

```
pip install numpy
pip install scipy
sudo apt-get install python3-scipy
pip install matplotlib
pip install python-Levenshtein==0.12.0
```

Temiz referans olarak kullanılan .wav formatındaki kaydın dizi formatına dönüştürülmesi için kod <https://github.com/omerfarukacar/GuitarTuner/tree/master/wavToText> adresinden indirilmelidir. Kaynak dosyasının wav_to_text.py programı kullanarak yazıya dönüştürülmüş hali aşağıdaki gibidir. Program gerekli kurulumlar yapıldıktan sonra GuitarTuner/wavToText klasörünün içinden; python wav_to_text.py phone_tune_clear.wav komutu ile çalıştırılmaktadır.

```
['G3', 'D4', 'B3', 'E4', 'D4', 'D4', 'E4', 'A3', 'E4', 'B3', '']
```

5.3. Kullanıcıdan Alınan Gitar Seslerinin Canlı Olarak Gösterilmesi ve Kaydedilmesi

12 dizili gitar notalarının anlık olarak ekranda gösterilmesi için açık kaynaklı kod referans alınmıştır (Andrey ve Zurstraßen, 2021). Gösterilmesi anında dizi olarak dedect_out.csv dosyasına kaydedilmektedir. Kayıt yapılmasının amacı kıyaslama yapmak için kullanılmasıdır. Canlı olarak seslerin gösterilmesini ve kaydedilmesini sağlayan kod <https://github.com/omerfarukacar/GuitarTuner/tree/master/NoteDedect> adresinden indirilmelidir.

Programın çalıştırılabilmesi için gerekli olan kurulumlar aşağıda belirtildiği gibidir.

Gerekli Python Kütüphane Sürümleri

```
Python 2.7.16
matplotlib 3.5.1
numpy 1.22.3
pydub 0.25.1
python-Levenshtein 0.12.0
sounddevice 0.4.4
scipy 1.8.0
```

Programın Çalışması İçin Gerekli Kurulumlar

```
pip install numpy
pip install scipy
pip install --upgrade numpy
pip install sounddevice
sudo apt-get install python-dev libatlas-base-dev
sudo apt-get install libportaudio2
```

Program gerekli kurulumlar yapıldıktan sonra GuitarTuner/NoteDedect klasörünün içinden noteDedect_12String.py dosyası; python3 noteDedect_12String.py komutu ile çalıştırılmaktadır. Canlı olarak kaydedilen gitar notaların çıktısı:

G4,G4,G4,G4,G3,G3,G3,G3,B3,B3,B3,B3,E4,E4,D4,D4,D4,D4,A3,A3,A3,A3,A3,E4,E4,E4,B3,B3
şeklindedir. Tekrar eden notalar teke indirilip kıyaslama yapılmıştır.

5.4. Sonuçların Dizi Olarak Kıyaslanması Ve Doğruluk Oranının Bulunması

Referans kaynaktan alınan sesin dizi olarak kaydedilmesi ve gerçek zamanlı olarak dinlenen seslerin dizi olarak kaydedilmesi kıyaslamayı kolaylaştırmıştır. Kıyaslama yapmak amacı ile python programlama dili kullanılıp CompareStrins.py adlı program yazılmıştır. Programın çalıştırılması için gerekli kod <https://github.com/omerfarukacar/GuitarTuner/tree/master/compareStrings> adresinden indirilmelidir. Program GuitarTuner/compareStrings klasörünün içinden; python3 CompareStrings.py komutu ile çalıştırılmaktadır. Program çıktısı aşağıdaki gibidir.

```
['G4', 'D4', 'B3', 'E4', 'D4', 'D4', 'E4', 'A3', 'E4', 'B3', '']
```

```
['G4', 'G3', 'B3', 'E4', 'D4', 'D4', 'A3', 'A3', 'E4', 'B3', '']
```

```
['true', 'false', 'true', 'true', 'true', 'true', 'false', 'true', 'true', 'true']
```

```
%80,0 true
```


6. SONUÇLAR

Bu uygulamada Raspberry Pi cihazı kullanılarak gitar notalarının hem ses kayıtları içinden hem de canlı olarak dinlenme sırasında bulunması başarılı bir şekilde gerçekleştirilmiştir. Raspberry Pi sıfırdan kurulum aşamaları ve gerekli programların kurulumu detaylı bir şekilde açıklanmıştır.

Yazılan python programlama dili ile kullanıcılardan alınan kayıtlı ve canlı gitar notaları string formatına dönüştürülüp kıyaslanabilir hale getirilmiştir. Kıyaslama sonucunda python programlama dili ile canlı notalar ve kayıtlı notalar arasındaki fark oransal olarak gösterilmesi sağlanmıştır.

Bu çalışmanın bir sonraki adımı olarak görsel ara yüz oluşturulması, canlı olarak hangi notalarda hata yapıldığının kullanıcılara belirtilmesi ve canlı okunan tek notanın birden fazla aynı notayı tek nota halinde vermesi hedeflenmektedir.

KAYNAKLAR

- Acar, Ö. F., (2022). Guitar Tuner, <https://github.com/omerfarukacar/GuitarTuner> adresinden 14 Nisan 2022 tarihinde alınmıştır.
- Andrey, M. ve Zurstraßen, N. (2021). Guitar Tuner, https://github.com/not-cheiken/guitar_tuner adresinden 14 Nisan 2022 tarihinde alınmıştır.
- Bocanegra, A. (2005). Automatic Acoustic Guitar Tuner, Massachusetts Institute of Technology.
- Cankurtaran H.S., Boyacı A., Yarkan S. (2020). Genişletilmiş DTK Yardımı ile Akustik İşaret Tabanlı, Dilden Bağımsız Bir Ses–Dudak , Sekli Eşleştirme Yöntemi ve Gerçeklemesi.
- Doire, C., Brookes, M., ve ark. (2017). Single-Channel Online Enhancement of Speech Corrupted by Reverberation and Noise,” IEEE Trans Audio Speech Lang Process.
- Dong, H. Y., Lee, C., M. (2018). Speech Intelligibility Improvement in Noisy Reverberant Environments Based on Speech Enhancement and Inverse Filtering.
- Kaminskyj, I., Materka, A. (1995). Automatic source identification of monophonic musical instrument sounds. IEEE International Conference on Nov 1995.
- Kumar, A., Srivastava, S., Chandra, M., Sahoo, G. (2017). Guitar tuner using cepstral analysis and fuzzy controller on arduino board, Microsyst Technol.
- Mamur, H., Aktaş, A., Kuzey, S. (2018). Müzik Notalarının MATLAB Yazılımıyla Belirlenmesi, Bilge International Journal of Science and Technology Research.
- Regeena, M. ve Saji, A. K. (2009). A Digital Guitar Tuner, (IJCSIS) International Journal of Computer Science and Information Security.
- Sturman, R. (2012). A dynamical systems approach to musical tuning, Dynamical Systems An International Journal.
- Thakuria, U., Narayanan, S., ve Kumbhare, R. (2019). Musical Instrument Tuner, Asian Journal of Convergence in Technology.
- Toole, F. E. (2000). The Acoustics and Psychoacoustics of Loudspeakers and Rooms-The Stereo Past and the Multichannel Future.
- Vonseggern, I. (2021). Note Prediction, <https://github.com/ianvonseggern1/note-prediction> adresinden 14 Nisan 2022 tarihinde alınmıştır.

TEŞEKKÜR ve BEYANLAR / ACKNOWLEDGEMENT and DECLARATIONS

Yazarların makaleye olan katkıları eşit orandadır. Yazarlar arasında herhangi bir çıkar çatışması bulunmamaktadır. Yapılan çalışmada araştırma ve yayın etiğine uyulmuştur

Not: Bu makale, İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik Haberleşme Mühendisliği Tezli Yüksek Lisans Programı'nda, Doç. Dr. Öğretim Üyesi Serhan YARKAN danışmanlığında, Ömer Faruk ACAR tarafından yürütülecek olan, “Raspberry Pi Tabanlı Gitar Akort Cihazı” başlıklı yüksek lisans tezinin ön çalışmalarından yararlanılarak hazırlanmıştır.