

Album Covers Project Idea

MS

LastFM API

API Methods

Album

[Album.addTags](#)
[Album.getInfo](#)
[Album.getTags](#)
[Album.getTopTags](#)
[Album.removeTag](#)
[Album.search](#)

Artist

[Artist.addTags](#)
[Artist.getCorrection](#)
[Artist.getInfo](#)
[Artist.getSimilar](#)
[Artist.getTags](#)
[Artist.getTopAlbums](#)
[Artist.getTopTags](#)
[Artist.getTopTracks](#)
[Artist.removeTag](#)
[Artist.search](#)

Auth

[Auth.getMobileSession](#)
[Auth.getSession](#)
[Auth.getToken](#)

Chart

Geo

[Geo.getTopArtists](#)
[Geo.getTopTracks](#)

Library

[Library.getArtists](#)

Tag

[Tag getInfo](#)
[Tag.getSimilar](#)
[Tag.getTopAlbums](#)
[Tag.getTopArtists](#)
[Tag.getTopTags](#)
[Tag.getTopTracks](#)
[Tag.getWeeklyChartList](#)

Track

[Track.addTags](#)
[Track.getCorrection](#)
[Track.getInfo](#)
[Track.getSimilar](#)
[Track.getTags](#)
[Track.getTopTags](#)
[Track.love](#)
[Track.removeTag](#)
[Track.scrobble](#)
[Track.search](#)
[Track.unlove](#)
[Track.updateNowPlaying](#)

User

[User.getArtistTracks](#)
[User.getFriends](#)
[User.getInfo](#)
[User.getLovedTracks](#)
[User.getPersonalTags](#)
[User.getRecentTracks](#)
[User.getTopAlbums](#)
[User.getTopArtists](#)
[User.getTopTags](#)
[User.getTopTracks](#)
[User.getWeeklyAlbumChart](#)

Current Work (go to code)

```
haar_face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_alt.xml')
sift = cv2.xfeatures2d.SIFT_create()

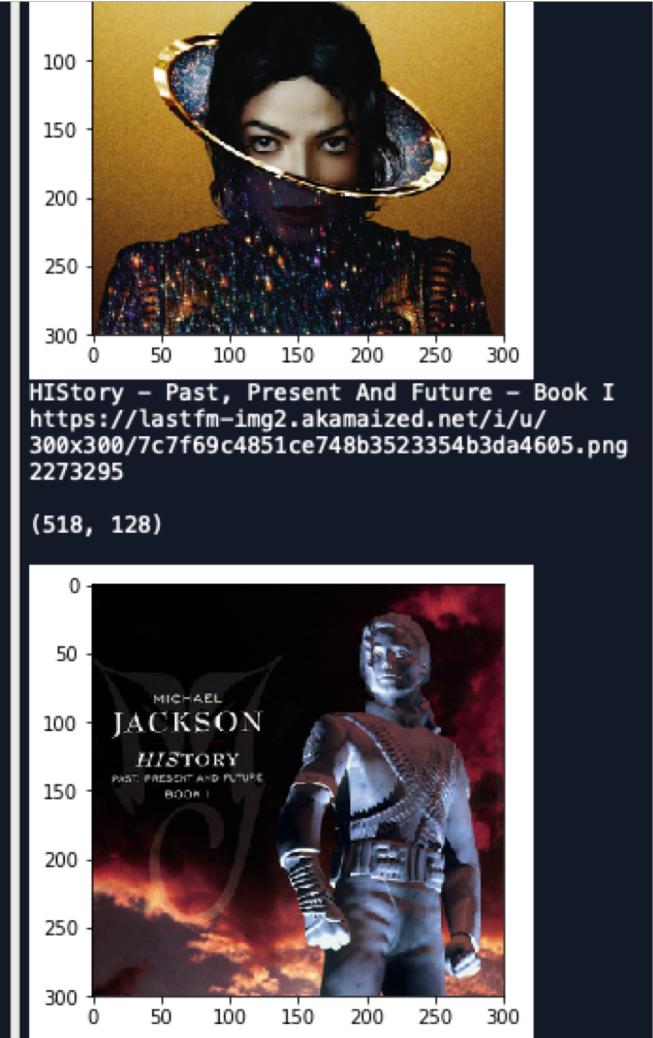
def write_cover_png(address,filename,show=False):
    pic = urllib.request.urlopen(address).read()
    with open(filename,'wb') as f:
        f.write(pic)
    if show:
        picture = plt.imread(filename)
        test1 = cv2.imread(filename)
    #    gray_img = cv2.cvtColor(test1, cv2.COLOR_BGR2GRAY)
    #    (kps, desc) = sift.detectAndCompute(test1,None)
    #    #FACE STUFF
    #    faces = haar_face_cascade.detectMultiScale(gray_img,1.1,1) #initially 1.1,1
    #    print('Num Faces: '+str(len(faces)))
    #    print(descs.shape)
    #    plt.figure()
    #    plt.imshow(picture)
    #    plt.show()

#address = 'https://lastfm-img2.akamaized.net/i/u/300x300/94e53cb2ea84a37cc5b2ba6acc6d6dd8.png'
#pic = urllib.request.urlopen(address).read()
#with open('test.png','wb') as f:
#    f.write(pic)
#https://lastfm-img2.akamaized.net/i/u/300x300/94e53cb2ea84a37cc5b2ba6acc6d6dd8.png

#write_cover_png(address,'test.png')

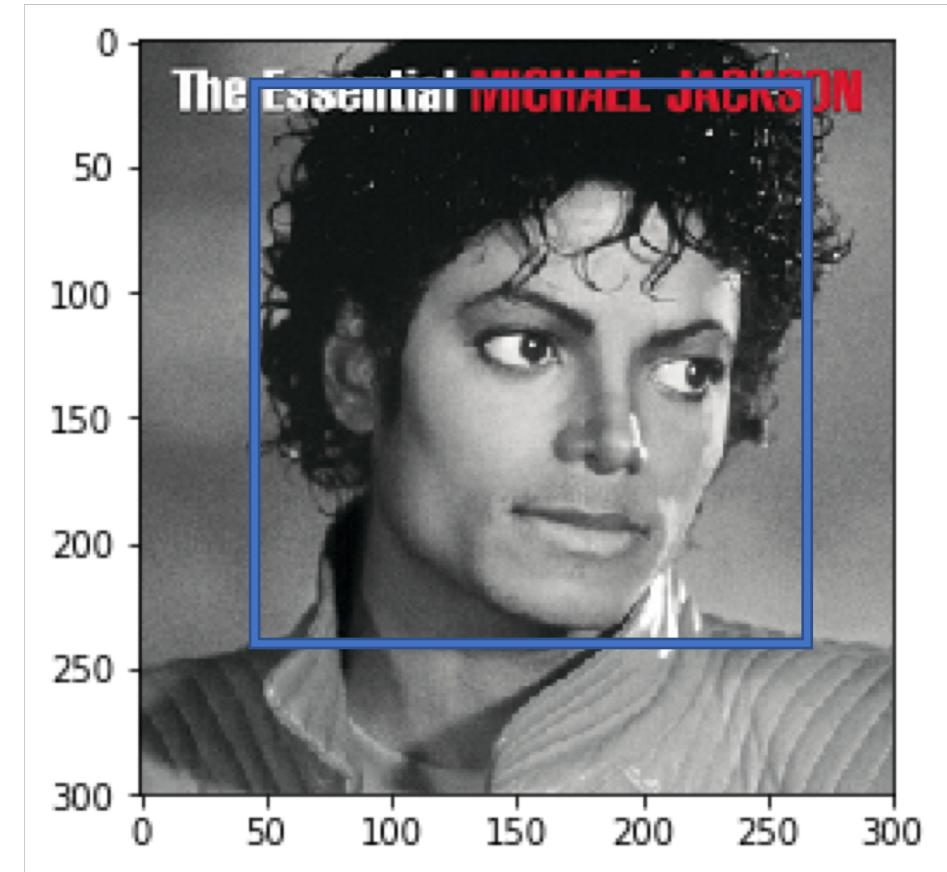
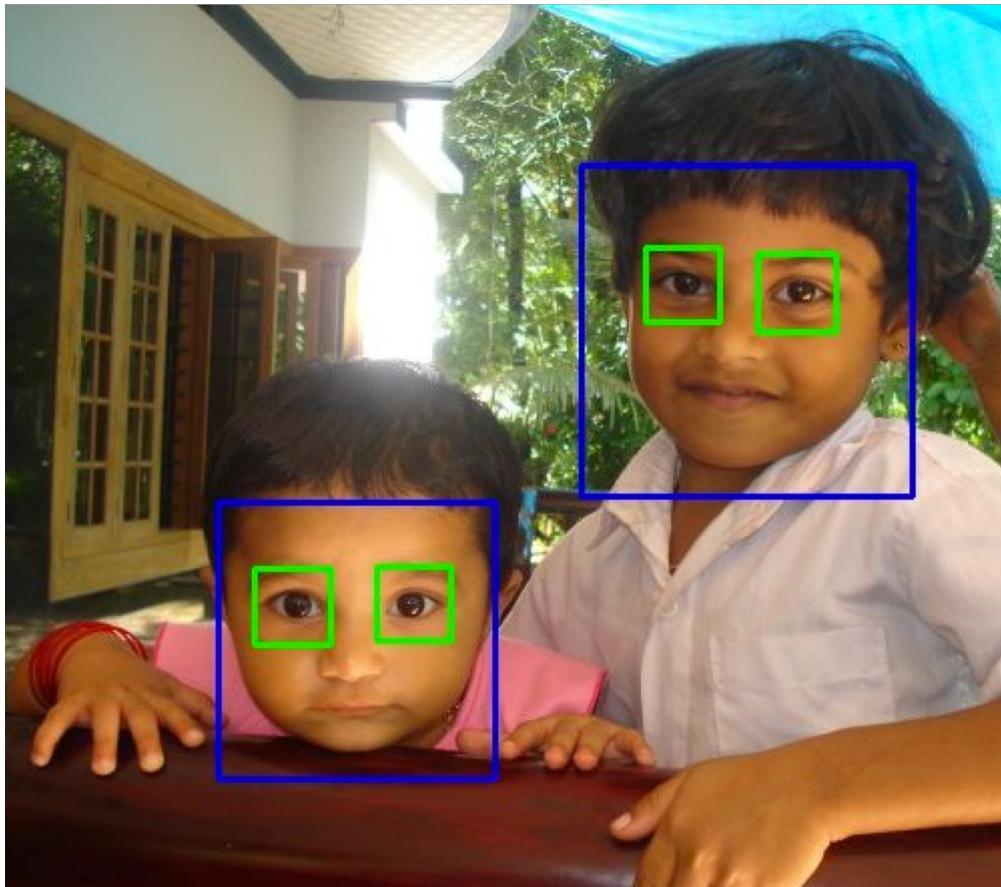
classic_rock_artists = open('classic_rock_artists.txt', 'r').read().split('\n')

#"http://ws.audioscrobbler.com/2.0/?method=artist.getTopAlbums&artist=Spock%27s+Beard&api_key="+my_api_key
artist = 'Michael Jackson'
address = "http://ws.audioscrobbler.com/2.0/?method=artist.getTopAlbums&artist="+\ artist+"&api_key="+my_api_key+"&format=json"
#data = urllib.request.urlopen(address).read()
data = json.loads(urllib.request.urlopen(address).read())
```

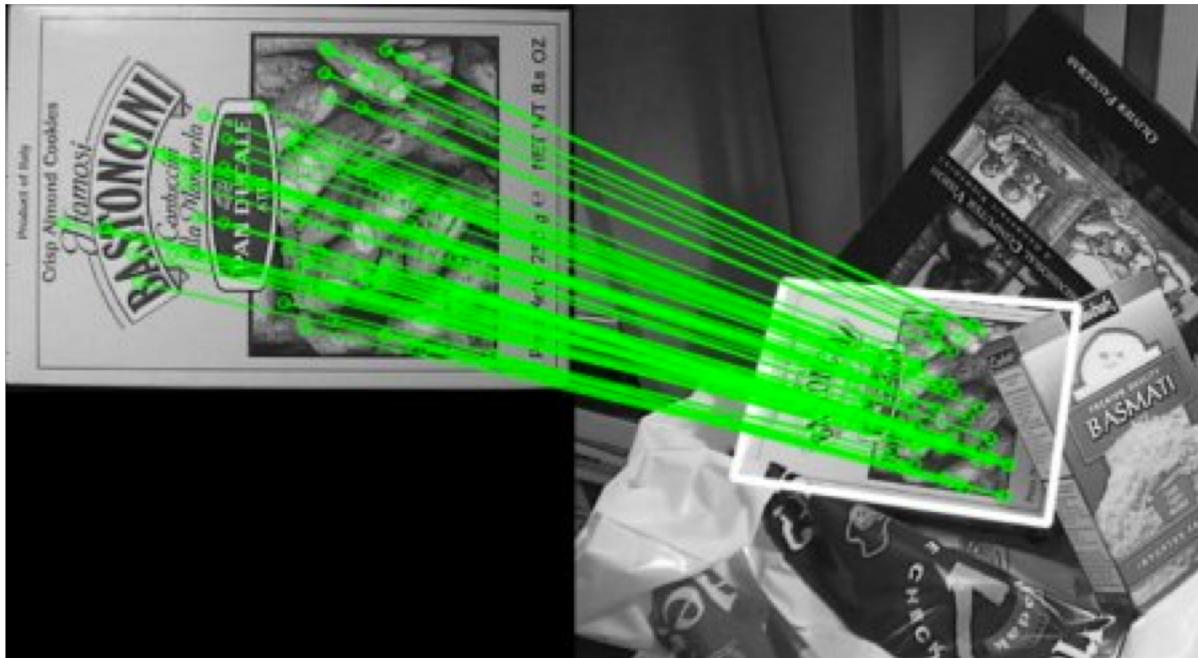


Haar Face Detection

- Find # of faces in album cover, save this info



RANSAC Capabilities

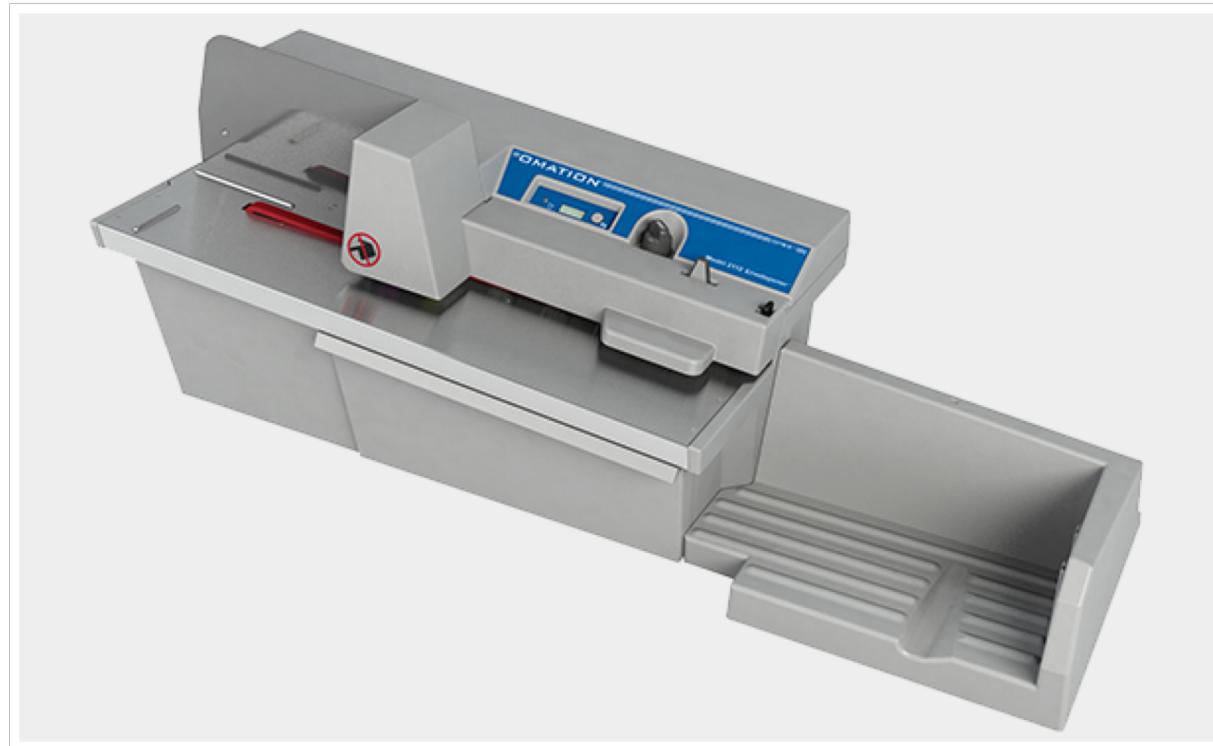


Identification of Harris Corners
Matching points of interest



Application to Music Company Identification

- This may be able to be connected to APIs referencing pricing based on album condition. Utilizing this could greatly speed up how albums are priced in music record stores.



Analysis of Genre/Popularity by Cover

You Can Judge an Artist by an Album Cover: Using Images for Music Annotation

Jānis Lībekš
Swarthmore College
Swarthmore, PA 19081
Email: jlibeks1@cs.swarthmore.edu

Douglas Turnbull
Department of Computer Science
Ithaca College
Ithaca, NY 14850
Email: dturnbull@ithaca.edu

Abstract—While the perception of music tends to focus on our acoustic listening experience, the *image* of an artist can play a role in how we categorize (and thus judge) the artistic work. Based on a user study, we show that both album cover artwork and promotional photographs encode valuable information that helps place an artist into a musical context. We also describe a simple computer vision system that can predict music genre tags based on content-based image analysis. This suggests that we can automatically learn some notion of artist similarity based on visual appearance alone. Such visual information may be helpful for improving music discovery in terms of the quality of recommendations, the efficiency of the search process, and the aesthetics of the multimedia experience.

I. INTRODUCTION

Imagine that you are at a large summer music festival. You walk over to one of the side stages and observe a band which is about to begin their sound check. Each member of the band has long unkempt hair and is wearing black t-shirts,

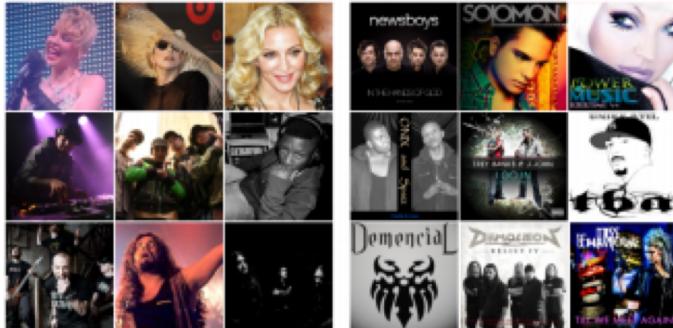


Fig. 1. Illustrative promotional photos (left) and album covers (right) of artists with the tags *pop* (1st row), *hip hop* (2nd row), *metal* (3rd row). See Section VI for attribution.

appearance. Such a measure is useful, for example, because it allows us to develop a novel music retrieval paradigm in which a user can discover new artists by specifying a query image. Second, images of artists also represent an unexplored

Other analysis could also be done based on data that could be pulled from LastFM along with album cover.

GAN Album Cover Gen

- Utilizing a GAN to create album cover for different kinds of architectures
- Someone might actually use it?

