Visualizing Songs

Let's make a video for your song!

CSE 590 - Computational Photography Project

NEHA BHATNAGAR (108750977)

RISHI JOSAN(108996773)

ABSTRACT

In this project, we develop a system which automatically generates a video composed of images related to the lyrics of the song provided by the user. Most of these images are extracted from ImageNet, restricting the words used to nouns. The video generation is thus fast as we are limiting the number of words. The system has two major components:

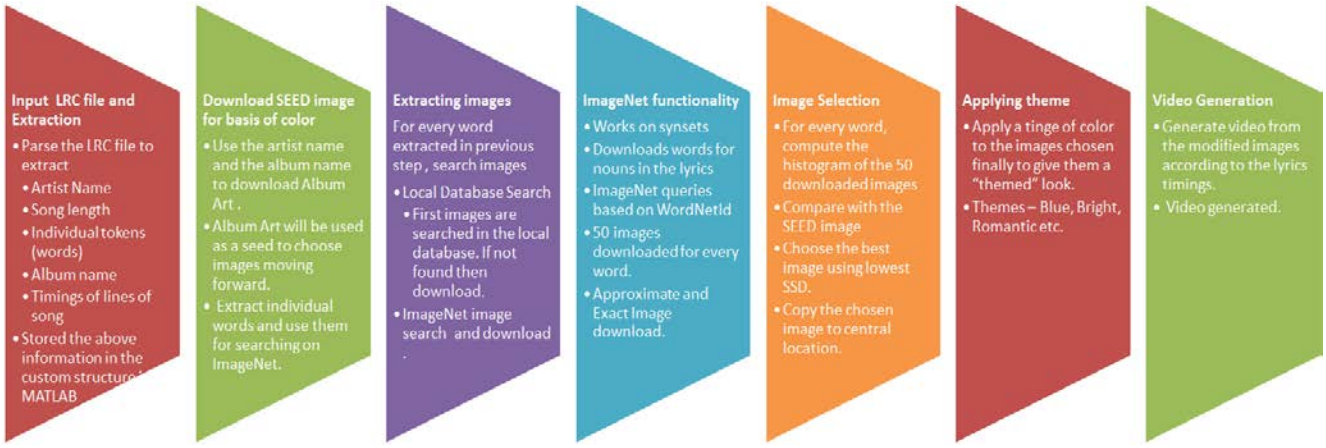
a. Extracting words and searching for images on ImageNet
b. Applying visual effects to the video to change the look and feel, using themes and color manipulation. To sync the video with the lyrics, timing is taken into consideration and we get a well synchronized video.

Keywords: Automated Music Video Generation, LRC, ImageNet, Google Image Search, Seed image, Themes, Color tinge

INTRODUCTION

The objective of “Visualizing Songs” is to pictorially represent the song with its lyrics. We intend to make the experience of listening to music much more interesting by adding a visual component, similar to a music video. We focus on the lyrics and search our big image database to find images related to the lyrics and then time their display according to the words being sung. We generate a video synched with the lyrics which combined with the audio presents a visualized song. Images are chosen from the large database according to the color and we provide an option to add a theme to the video which makes it consistent.

PROJECT OVERVIEW



1. Word Extraction – LRC Parsing

Retrieve LRC file for chosen song using software such as MiniLyrics.

Sample LRC File:

[ar: Bruno Mars]
[ti: Count On Me]
[id: gs_qwzksqsh]
[00:05.92]if you ever find yourself stuck in the middle of the sea
[00:11.06]i'll sail the world to find you
[00:16.67]if you ever find yourself lost in the dark and you can't see
[ar : Bruno Mars] - tells the artist's name
[ti : Count on Me] - tells song's name

Rest timings are listed for each line for the entire song.

The LRC file is parsed to extract metadata like the artist's name, the song name and album. The album art is downloaded using an interface for the Google image search API. This Seed is the image of the metadata (Artist/Album/Title) which acts as a reference for the images that will be chosen subsequently(based on Lyrics and colors).

If album art cannot be retrieved, we use one of the Theme images as the Seed. Additional details about themes is given later in the report.

Parsing the words and timings:
We create a custom structure in MATLAB for each line in the song. We extract the start time from the start of the line and store it. Next we store the end time of that particular line, which we approximate to be equal to the start time of the next sentence. Thus we are able to generate the time duration for a particular line of words.
Next we tokenize the complete sentence into words and then store them in cells. These are later passed to the image search function which searches for images related to the words passed.

Custom structure:

lyricsLine - line number

startTime - start time of the lyric line
endTime - end time of the lyric line
imageArr - images that have been found for the particular line
imPath - path of the images present above, to be used to locate them.

Example :
lyricsLine: [1x0 char]
startTime: {'0'}
endTime: {'5.92'}
imageArr: {[200x200x3 uint8]}
imPath: {'C:\builtdb\crazy_theme\16.jpg'}

Handling suffixes:
In addition to searching the word, special consideration is taken for words ending with suffixes like er/ed/ing/s .
We search for images for the original word, if no images are found we search again removing the suffixes er/ed/ing/s.
For example:
Word - Play.
For words such as player, playing, plays and played, the suffixes are stripped and the search is performed on “Play”.

2. Seed Image Selection

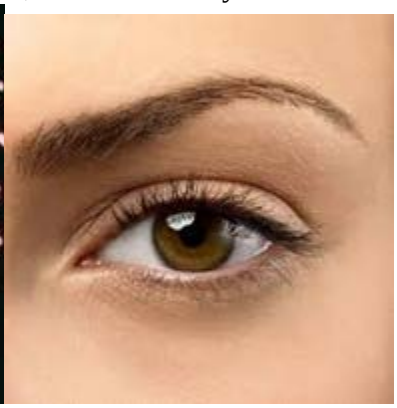
As mentioned earlier, the seed or the reference image has to be chosen to have a reference for the selection of subsequent images.

We first search for Album Art using an interface for the Google Image Search API. If that fails, we use a theme image.
For eg:
If we choose a yellow theme, we would have some yellow images present for that theme in our local database, we would find an image out of those and choose the images for the words that would come up in the song by calculating the color histogram in a similar fashion as mentioned above.

This shows the example as to how different kind of images for the word ‘eye’ are chosen when different seed images are chosen.

Case 1: Image retrieved for “Eye”, when the album art is chosen as the reference

Case 2: Image retrieved for “Eye”, when the “Party Theme” is chosen as reference

Case 3: Image retrieved for “Eye”, when the “Romantic Theme” is chosen as reference

3. Image Search and Selection

ImageNet

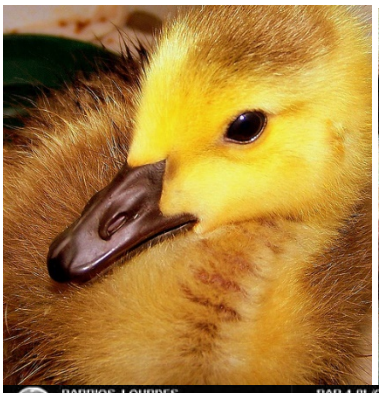
ImageNet provides an API to retrieve images for words (nouns only) based on the WordNet ID. There are two steps involved:

1. Every word from the lyrics file is first compared to a list of words in the WordNet database. If found, its WordNet ID is retrieved.
2. This WordNet ID is used to query a list of WordNet IDs for which images exist in ImageNet database.

There is one important concern which is addressed in the first step. We could search for either an exact or an approximate match for a word. For example, the exact match for Baby would correspond to the

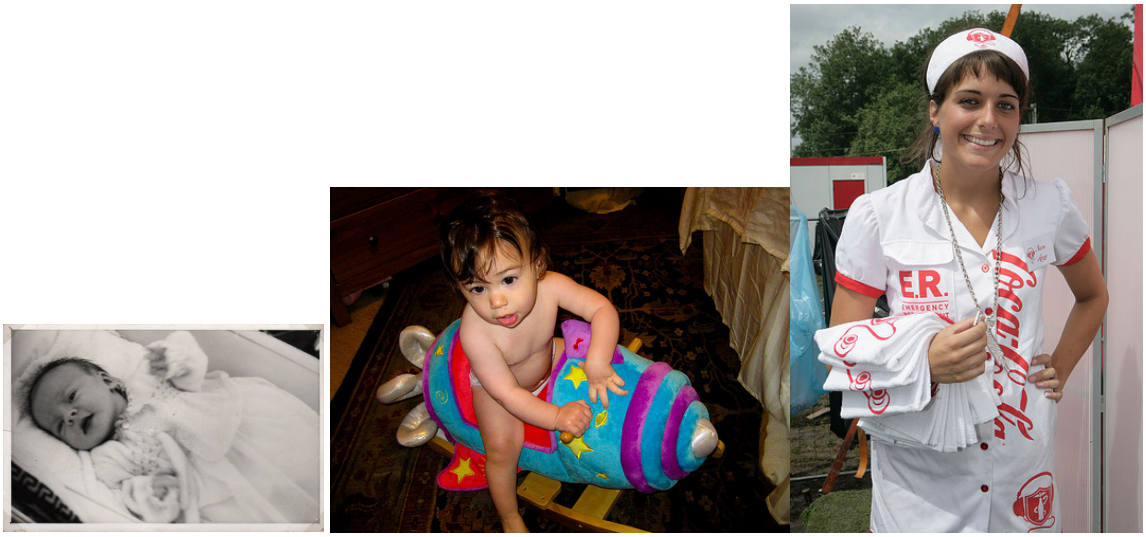
Synset - n01322221 – **Baby -** A very young mammal; "baby rabbits“
OR
Synset - n09828216 - **An unborn child; a human foetus**;

With the images retrieved as follows:



On the other hand, for an approximate match, the following Synsets and images are retrieved:

Synset - n09918554 – **Child, Baby** - An immature childish person
OR
Synset - n09827363 - **Baby, babe, sister -** (slang) sometimes used as a term of address for attractive young women



The choice of whether to have exact or approximate matches should be made song by song, depending on the preferences of the user. The resulting videos for both approaches have been showcased in the Results section

Note: We have not utilized the ImageNet API. Instead we used URLREAD and to retrieve image links from ImageNet's database and URLWRITE to download these images. The ImageNet API does not allow searches based on words. It need a WordnetID to retrieve images. It also doesn't give the developer to choose the number of images to download, and itself downloads all the images in a synset. The size of the archive downloaded is thus in the range of 50 – 100MB. Downloading the entire set is not feasible for our approach. Thus we use the approach as mentioned above instead of the API.

Google Search
We utilize an interface to the deprecated Image Search API by Google to download images for:

a. Metadata – Album/Artist/Title
b. Frequently Occurring Words

Here again we use URLREAD to retreive image links for a word and download them using URLWRITE.

As the API is deprecated, we can retrieve only 8 images per word. Misuse of the API or exceeding the quota limit will lead to the IP of the developed being banned. We thus exercise caution and use the method only for the two cases mentioned above.

Google image search provides the best image results. These are especially useful for retrieving the Metadata images.

Local Database
Images retrieved from the methods mentioned above are used to create a local database of images. 50 images are retrieved per ImageNet query and 8 images are retrieved per Google query. The objective is to have a set of images ready such that, our program first queries the local database for words before searching for them online.
This save time and bandwidth.

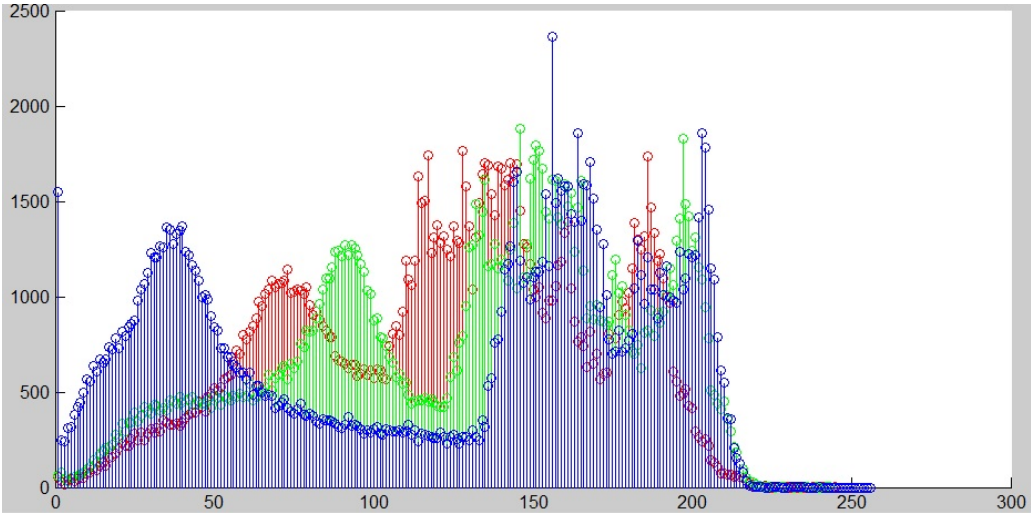
Image Selection:

To produce a video in a consistent way, we need to compare colors across images. We create a histogram for each individual color (R, G, B) for each image to have an idea of the relative amount of color in each image.

A sample of the histograms created is as follows:

Sample Image:

R, G, B Histograms:



To compare images, we compute an SSD of the histogram of each word Image (Typically every word has 50 corresponding images in the database) and the histogram of the seed image. The image with the least SSD value is selected and inserted into the video.

4. Image Enhancements and Video Generation

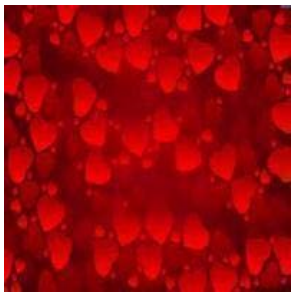
Themes:

We have a set of themes to add a recurring “Theme” to the song. The user has Wherever there are portions in the song without lyrics, images from the theme selected are inserted.

Themes Tried:

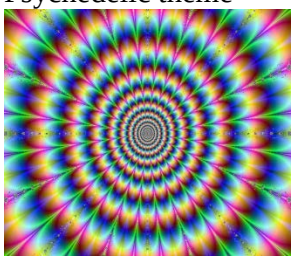
- Party theme

- Romantic Theme



- Chirpy theme

- Psychedelic theme



Color Tinge:

To give the all the images in the song a constant “look and feel” , we have played around with the color composition of the images.
We add a tinge to the colors depending on the dominant color in the seed image or the theme image chosen.

For instance, adding a red tinge to the ‘baby’ image results in:

Adding a yellow tinge according to the Chirpy theme, the ‘sleep’ image results in :

Video Generation:

Images chosen after computation need to be combined together synched up to the song. We generate a video using MATLAB with a rate of 30 fps and based on the timestamps stored in the custom structures.

The video thus generated is combined with the song.

5. RESULTS

Below are the results for different approaches taken :

i) Exact matches
ii) Approximate Matches

Example (exact match)

For the song, Katy Perry ‘s fireworks we have the following images which have been chosen :

The album art which is downloaded using the Google Image Search:

This image is chosen as the base image. All the images for the words in the lyrics will have their color histograms matched with this image.
A red tinge has been added considering the dominant color in the album art image is red.
Images chosen for the first sentence of the song:

[00:08.50]Do you ever feel like a plastic bag

[00:12.48]Drifting through the wind

[00:14.10]Wanting to start again

[00:15.78]

[00:16.40]Do you ever feel, feel so paper thin

[00:19.87]Like a house of cards

[00:21.87]One blow from caving in

[00:23.40]

Will be filled with the theme which we have chosen -party theme here, so images from the party theme will be displayed for this time duration.

[00:24.64]Do you ever feel already buried deep

[00:27.99]Six feet under scream

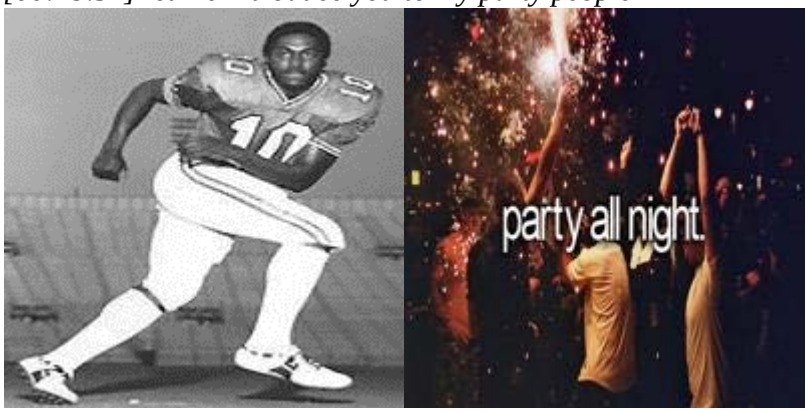
Example (Approximate match)
When ImageNet is queried for approximate matches rather than exact ones, the images selected do not exactly portray the meaning of the song.

Note: This produces interesting and funny results, which may be used to create an exaggerated interpretation of the song.

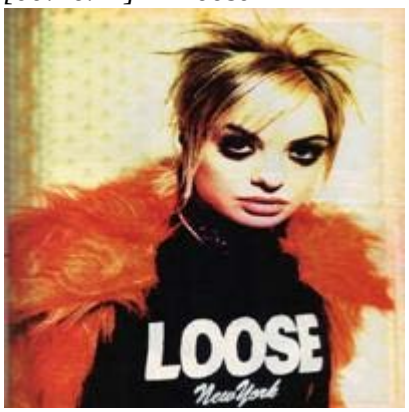
[00:05.71]J-LO!
No images found

[00:09.33]It's a new generation

[00:19.52]Let me introduce you to my party people

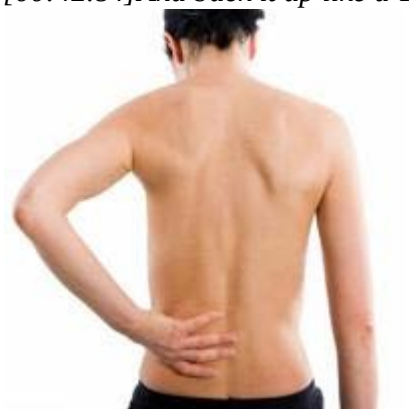


[00:26.27]I'm loose



[00:40.53]Now pump it up

[00:42.84]And back it up like a Tonka truck



In order to get relevant results it is better to develop a local database from ImageNet using the exact matches.

Sample Output Videos:

Exact word matching in the ImageNet database

- Slow song - Psychedelic theme + blue tinge + Theme image as the base image
  Pink Floyd – Another Brick in the Wall
  http://youtu.be/l8rLnGixj4g
- Country song - Green Theme + green tinge + Album art as base image
  John Denver – Country Roads
  http://youtu.be/Zw2qRWBEg5k
- Fast song - Party theme + red tinge + Album art as base image
  Katy Perry – Fireworks
  http://youtu.be/7hAjPnRYBQg
- Medium speed song - Chirpy yellow theme + yellow tinge + theme image as base image
  Bruno Mars – Count On Me
  http://youtu.be/9Sf901R_uYQ
  Yellow Tinge - http://youtu.be/_LFP2WV2_RU

Approximate Image search: ImageNet

- Fast song - romantic theme + Approximate Image search + theme image as base image
  Jennifer Lopez Ft. Pitbull – On the Floor (Approximate matches)
  http://youtu.be/3QxtrOc_xSk

Limitations:

- Since the LRC files contain just the start and the end time of the lyric, hence we cannot allot exact display time to each of the images chosen for that line, hence there might be delay in the individual images' display.
    - One resolution is finding the location of the word and calculating the position of occurrence of the word. This method gave a slight improvement but cannot be generalized to all songs.
- It works on jpeg images only and English songs.
- Since the source of images in ImageNet , the images may not be relevant sometimes according to the content.

Future Work:

- The sound analysis can be done for the songs and according to the frequency of the beats we can display the images, this should give better synched images.
- Can be extended to support multiple file types and languages.

References:

1. String Toolkit http://www.mathworks.com/matlabcentral/fileexchange/21710-string-toolkits
2. ImageNet - http://www.image-net.org/
3. Color Histograms: http://www.mathworks.com/matlabcentral/fileexchange/4875-color-image-histogram