**Music Generation using Deep Learning**

# Project Report

**SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE**

**OF**

## BACHELOR OF TECHNOLOGY

**IN**

## COMPUTER SCIENCE AND ENGINEERING

**SUBMITTED BY**

## N Vivek Kumar Varma (15103043)

## Karlapalepu Jithamanyu (15103050)

## Nunnaguppala Jayanth Sai Surya (15103089)

**UNDER THE SUPERVISION OF**

|  |  |
| --- | --- |
|  |  |
|  |  |

# 

# Dr. Samayveer Singh Assistant Professor

### DEPARTMENT OF COMPUTER SCIENCE ENGINEERING DR. B. R. AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY

**JALANDHAR – 144011, PUNJAB (INDIA)**

**June, 2019**

**DR. B. R. AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY, JALANDHAR**

**CANDIDATES’ DECLARATION**

We herewith certify that the work that is being conferred within the project report entitled “Music Generation using Deep Learning” in partial fulfilment of necessities for the award of degree of B.Tech. (Information Technology) submitted to the Department of Information Technology of Dr. B R Ambedkar National Institute of Technology, Jalandhar, is an authentic record of our own work carried out during a period from July, 2018 to May, 2019 under the supervision of Dr. Samayveer Singh (Assistant Professor). The matter presented in this dissertation has not been submitted by me in any other University/Institute for the award of any degree.

N Vivek Kumar Varma (15103043)

Karlapalepu Jithamanyu (15103050)

Nunnaguppala Jayanth Sai Surya (15103089)

This is to certify that the above statement made by the candidates is correct and true to the best of my knowledge.

Mr. D. K. Gupta Associate Professor (Dept. of CSE)

**??**

**ACKNOWLEGDEMENT**

Foremost, we'd prefer to categorical our feeling to my supervisor Mr. D. K. Gupta, professor, for the helpful comments, remarks and engagement through the training method of this Project. We have a tendency to cannot impart him enough for his tremendous support and facilitate. He actuated and inspired North American nation throughout this work. Without his encouragement and guidance this project would not have materialized. We consider ourselves extremely fortunate to have a chance to work under his supervision. In spite of his busy schedule, he was always approachable and took his time off to guide us and gave appropriate advice.

We also wish to thank whole heartily all the faculty members of the Department Information Technology for the invaluable knowledge they have imparted on us and for teaching the principles in most exciting and enjoyable way. We also extend our thanks to the technical and administrative staff of the department for maintaining an excellent working facility.

We would like to thank our families for their continuous support and blessings throughout the entire process, both by keeping us harmonious and helping us putting the pieces together. We also like to thank all our batch mates for the useful discussions, constant support and encouragement during whole period of the work.

Last but not the least, we would like to thank almighty GOD for giving us enough strength and lifting us uphill this phase of life.

**Bhushan Mandal Deepak Kalsi Shalini Garg Deepanshu Setia**

# ABSTRACT

Generating music with non repeating long-term structure is one of the main challenges in the field of automatic composition. The use of deep learning to solve such problems in literary arts has been a recent trend that has gained a lot of attention and automated generation of music has been an active area. A model of music needs to have the ability to recall past details and have a clear, coherent understanding of musical structure. This project deals with the generation of music using raw audio files in the MIDI format relying on LSTM architecture. The work is focuses on using information about musical structure (notes and chords) of sample audio files to aid learning. The music generated from the model is then evaluated based on human judgement.

**TABLE OF CONTENTS**

#### CANDIDATES’ DECLARATION i

**ACKNOWLEDGEMENT ii**

[**ABSTRACT iii**](#_heading=h.gjdgxs)

[**LIST OF FIGURES vi**](#_heading=h.30j0zll)

[**LIST OF TABLES vii**](#_heading=h.1fob9te)

[**LIST OF ABBREVIATIONS viii**](#_heading=h.3znysh7)

**CHAPTER 1: INTRODUCTION**

* 1. [Historical Background 2](#_heading=h.2et92p0)
  2. [Introduction to Biometrics 3](#_heading=h.44sinio)
  3. [Comparison Between Various Biometrics 4](#_heading=h.2jxsxqh)
  4. [Motivation 4](#_heading=h.tyjcwt)

**CHAPTER 2: LITERATURE SURVEY**

* 1. Ear Detection in 2D 6
     1. Appearance Based Techniques 6
     2. Force Field Transformation Based Techniques 7
     3. Geometric Features Based Techniques 7
  2. Ear Detection in 3D 7

2.2 Related Work and Contribution 7

**CHAPTER 3: PROBLEM STATEMENT AND OBJECTIVES**

* 1. [Problem Statement 9](#_heading=h.3dy6vkm)
  2. Objectives 9

**CHAPTER 4: METHODOLOGY**

* 1. [Image Acquisition 10](#_heading=h.1t3h5sf)
     1. Experimental Setup 10
     2. Environment Description 11
  2. [ROI Detection 12](#_heading=h.4d34og8)
  3. [Image Registration 13](#_heading=h.2s8eyo1)
  4. [Feature Extraction 14](#_heading=h.17dp8vu)
     1. First Feature Vector (FV1) 15
     2. Second Feature Vector (FV2) 15
     3. Classification 15
     4. Advantages of two-stage classification 16
  5. [Matching 16](#_heading=h.3rdcrjn)
     1. Model Description: Faster R-CNN 16
     2. Model Training 18

**CHAPTER 5: RESULTS**

* 1. [Result 25](#_heading=h.26in1rg)
  2. [Applications of Ear Recognition System 29](#_heading=h.lnxbz9)
     1. Application of Android-based Ear Biometrics Identification 29
     2. Development of Multimodal Identification System using Face and Ear 29

**CHAPTER 6: CONCLUSION**

* 1. [Conclusion 30](#_heading=h.35nkun2)
  2. [Future Scope 30](#_heading=h.1ksv4uv)

**REFERENCES**

# ????

# LIST OF FIGURES

**Figure 1.1** Block Diagram of Ear Recognition System 1

**Figure 1.2** Different Biometric traits 3

**Figure 1.3** Structure of Ear 5

**Figure 4.1** Directory Structure 10

**Figure 4.2** Different Variations in Capturing Images for dataset 12

**Figure 4.3** ROI in the Side Face Image 13

**Figure 4.4** Ear Alignment 13

**Figure 4.5** Gray Scale Image of ear intersected by Normal Lines 14

**Figure 4.6** Angles θ1, θ2, θ3,…., θn as FV1 15

**Figure 4.7** Internal ear intersected by Normal Lines 15

**Figure 4.8** Flow Chart of RPN 17

**Figure 4.9** Input to Pre-trained Model and Getting IO1 with collection of bounding boxes 18

**Figure 4.10** Giving IO1 to new pre-trained ImageNet 19

**Figure 4.11** Getting IO2 19

**Figure 4.12** Giving IO2 as Input 20

**Figure 4.13** Final Output of the Model 20

**Figure 5.1** Loss Graph With Classification Loss and Localization Loss 25

**Figure 5.2** Losses After 100 and 4 Lakh Iterations 25

**Figure 5.3** Loss Graph 26

**Figure 5.4** Detected Ear With % of Accuracy Written on it 27

**Figure 5.5** Comparison between Less Trained Model and Highly Trained Model 28

**Figure 5.6** Resulted Image with Bounding Box and % of Accuracy 28

# ????

# LIST OF TABLES

**Table 1.1** Comparison Among Biometric Traits 4

**Table 5.1** Results 26

# ????

# LIST OF ABBREVIATIONS

### Abbreviation Description

IoU Intersection over union

FV First Vector

RPN Region Proposed Network

IO Intermediate Output

ROI Region of Interest

#### INTRODUCTION

Music Generation, as the name indicates, means generation of music. This can be achieved either by man creating it or by a computer using various techniques. Machine Learning can be used to generate music. Generative music is a term popularized by Brian Eno to describe music that is ever-different and changing, and that is created by a system.

**Historical Background**

Much of the work on computer music has drawn on the relationship between music and mathematics, a relationship which has been noted since the Ancient Greeks described the "harmony of the spheres". The world's first computer to play music was the CSIR Mark 1 (later named CSIRAC), which was designed and built by Trevor Pearcey and Maston Beard from the late 1940s. Early computer-music programs typically did not run in real time, although the first experiments on CSIRAC and the Ferranti Mark 1 did operate in real time. From the late 1950s, with increasingly sophisticated programming, programs would run for hours or days, on multimillion-dollar computers, to generate a few minutes of music. Interesting sounds must have a fluidity and changeability that allows them to remain fresh to the ear. In computer music this subtle ingredient is bought at a high computational cost, both in terms of the number of items requiring detail in a score and in the amount of interpretive work the instruments must produce to realize this detail in sound.

In 1960, Russian researcher R.Kh.Zaripov published worldwide first paper on algorithmic music composing using the "Ural-1" computer. In 1965, inventor Ray Kurzweil premiered a piano piece created by a computer that was capable of pattern recognition in various compositions. The computer was then able to analyze and use these patterns to create novel melodies. The computer was debuted on Steve Allen's I've Got a Secret program, and stumped the hosts until film star Henry Morgan guessed Ray's secret.

**Computer-generated Music**

Computer-generated music is music composed by, or with the extensive aid of, a computer. We can distinguish two groups of computer-generated music: music in which a computer generated the score, which could be performed by humans, and music which is both composed and performed by computers. There is a large genre of music that is organized, synthesized, and created on computers.

Machine improvisation uses computer algorithms to create improvisation on existing music materials. This is usually done by sophisticated recombination of musical phrases extracted from existing music, either live or pre-recorded. In order to achieve credible improvisation in particular style, machine improvisation uses machine learning and pattern matching algorithms to analyze existing musical examples. The resulting patterns are then used to create new variations "in the style" of the original music, developing a notion of stylistic reinjection. This is different from other improvisation methods with computers that use algorithmic composition to generate new music without performing analysis of existing music examples.

**Machine Learning**

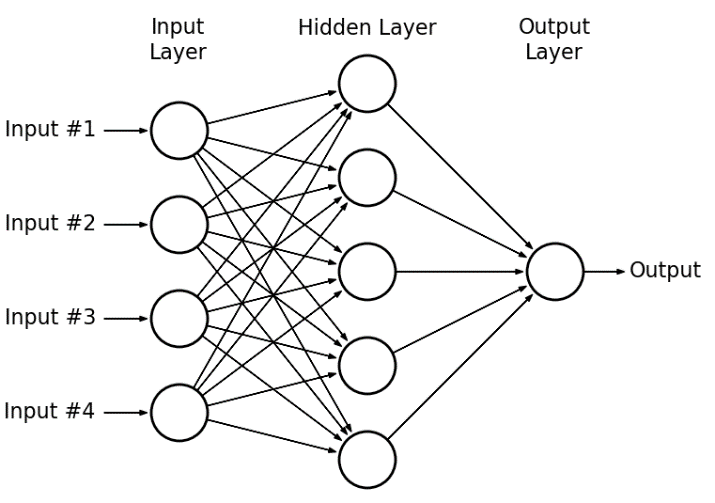
Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering, computer vision and Music Generation, where it is infeasible to develop an algorithm of specific instructions for performing the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers.

Machine learning tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the training data for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback. Semi-supervised learning algorithms develop mathematical models from incomplete training data, where a portion of the sample input doesn't have labels. In unsupervised learning, the algorithm builds a mathematical model from a set of data which contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or clustering of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in feature learning.

**Artificial neural networks**

Artificial neural networks (ANN) or connectionist systems are computing systems indirectly inspired by the [biological neural networks](https://en.wikipedia.org/wiki/Biological_neural_network) and [astrocytes](https://en.wikipedia.org/wiki/Astrocytes) that constitute animal [brains](https://en.wikipedia.org/wiki/Brain). Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules. For example, in [image recognition](https://en.wikipedia.org/wiki/Image_recognition), they might learn to identify images that contain cats by analyzing example images that have been manually [labeled](https://en.wikipedia.org/wiki/Labeled_data) as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge about cats, for example, that they have fur, tails, whiskers and cat-like faces. Instead, they automatically generate identifying characteristics from the learning material that they process.

An ANN is based on a collection of connected units or nodes called [artificial neurons](https://en.wikipedia.org/wiki/Artificial_neuron), which loosely model the [neurons](https://en.wikipedia.org/wiki/Neuron) in a biological brain. Each connection, like the [synapses](https://en.wikipedia.org/wiki/Synapse) in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it.

****

**Figure : A Neural Network**

**Deep Learning**

Deep learning is a class of [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [algorithms](https://en.wikipedia.org/wiki/Algorithm) that use multiple layers to progressively extract higher level features from raw input. For example, in image processing, lower layers may identify edges, while higher layer may identify human-meaningful items such as digits/letters or faces. In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a [matrix](https://en.wikipedia.org/wiki/Matrix_(mathematics)) of pixels; the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn which features to optimally place in which level on its own. The term Deep Learning was introduced to the machine learning community by [Rina Dechter](https://en.wikipedia.org/wiki/Rina_Dechter) in 1986 and to [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_Neural_Networks) by Igor Aizenberg and colleagues in 2000, in the context of Boolean threshold neurons. The first general, working learning algorithm for supervised, deep, feedforward, multilayer [perceptrons](https://en.wikipedia.org/wiki/Perceptron) was published by [Alexey Ivakhnenko](https://en.wikipedia.org/wiki/Alexey_Grigorevich_Ivakhnenko) and Lapa in 1965. A 1971 paper described a deep network with 8 layers trained by the [group method of data handling](https://en.wikipedia.org/wiki/Group_method_of_data_handling) algorithm.

Deep learning architectures such as [deep neural networks](https://en.wikipedia.org/wiki/Deep_learning#Deep_neural_networks), [deep belief networks](https://en.wikipedia.org/wiki/Deep_belief_network), [recurrent neural networks](https://en.wikipedia.org/wiki/Recurrent_neural_networks) and convolutional neural networks have been applied to fields including [computer vision](https://en.wikipedia.org/wiki/Computer_vision), speech recognition, [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), audio recognition, social network filtering, [machine translation](https://en.wikipedia.org/wiki/Machine_translation), [bioinformatics](https://en.wikipedia.org/wiki/Bioinformatics), [drug design](https://en.wikipedia.org/wiki/Drug_design), medical image analysis, material inspection and [board game](https://en.wikipedia.org/wiki/Board_game) programs, where they have produced results comparable to and in some cases superior to human experts.

Deep Learning is currently the most used field in the realization of Artificial Intelligence. Research in artificial intelligence (AI) is known to have impacted medical diagnosis, stock trading, robot control, and several other fields. Perhaps less popular is the contribution of AI in the field of music. Several music software programs have been developed that use AI to produce music. Like its applications in other fields, the A.I. in this case also simulates mental task. A prominent feature is the capability of the A.I. algorithm to learn based on information obtained such as the computer accompaniment technology, which is capable of listening to and following a human performer so it can perform in synchrony.

There are many models of neural networks that can be used to create programs which can successfully generate music. Some models are RNN, GAN, and LSTM.

In this project we made use of LSTM model to generate music.

**LSTM**

Long short-term memory (LSTM) is an artificial [recurrent neural network](https://en.wikipedia.org/wiki/Recurrent_neural_network) (RNN) architecture used in the field of [deep learning](https://en.wikipedia.org/wiki/Deep_learning). Unlike standard [feedforward neural networks](https://en.wikipedia.org/wiki/Feedforward_neural_network), LSTM has feedback connections that make it a "general purpose computer" (that is, it can compute anything that a [Turing machine](https://en.wikipedia.org/wiki/Turing_machine) can). It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected [handwriting recognition](https://en.wikipedia.org/wiki/Handwriting_recognition) or [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition). A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

LSTM networks are well-suited to [classifying](https://en.wikipedia.org/wiki/Classification_in_machine_learning), [processing](https://en.wikipedia.org/wiki/Computer_data_processing) and [making predictions](https://en.wikipedia.org/wiki/Predict) based on [time series](https://en.wikipedia.org/wiki/Time_series) data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the exploding and [vanishing](https://en.wikipedia.org/wiki/Vanishing_gradient_problem) gradient problems that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, [hidden Markov models](https://en.wikipedia.org/wiki/Hidden_Markov_models) and other sequence learning methods in numerous applications.

In theory, classic (or "vanilla") [RNNs](https://en.wikipedia.org/wiki/Recurrent_neural_network) can keep track of arbitrary long-term dependencies in the input sequences. The problem of vanilla RNNs is computational (or practical) in nature: when training a vanilla RNN using [back-propagation](https://en.wikipedia.org/wiki/Back-propagation), the gradients which are back-propagated can ["vanish" (that is, they can tend to zero) or "explode" (that is, they can tend to infinity)](https://en.wikipedia.org/wiki/Vanishing_gradient_problem), because of the computations involved in the process, which use [finite-precision numbers](https://en.wikipedia.org/wiki/Round-off_error). RNNs using LSTM units partially solve the [vanishing gradient problem](https://en.wikipedia.org/wiki/Vanishing_gradient_problem), because LSTM units allow gradients to also flow unchanged. However, LSTM networks can still suffer from the exploding gradient problem.

Music derives meaning through repetition. Modeling musical coherence created by internal repetition and reference to earlier material in a piece is a persistent issue in computational music generation, and presents a challenge especially for machine learning methods.

**Data Representation:**

Before using the LSTM model, it is necessary to have a suitable dataset which consists of the music we wish to allow the model to learn from. In order to do that first we need to represent the music in a format that a computer can recognize.

Any music consists of notes and chords. They can be defined as follows:

**Note:** A note is your basic building block of music. A note is one key on a piano.

**Chord:** A chord is 3 or more notes played together at the same time. For example, a C Major chord consists of 3 notes, C E and G. It can be any group of 3 different notes, some just wouldn't sound good.

Music constitutes of different permutations of notes and chords combined in a certain pattern.

**Representation of Music:** Music can be represented using various formats like sheet music, ABC Notation, MIDI format etc. Our key task is to represent music as a sequence of events. As we will be using LSTM which takes sequence as an input.

**Sheet Music:** Here, music is represented by a sequence of musical notes. Each musical note is separated by a space. This can be used to represent both single instrument and multi instrument music.

**ABC notation of Music:** There are two parts in ABC-notation.

Part-1 represents meta data. Lines in the Part-1 of the tune notation, beginning with a letter followed by a colon, indicate various aspects of the tune such as the index, when there are more than one tune in a file (X:), the title (T:), the time signature (M:), the default note length (L:), the type of tune (R:) and the key (K:).

Part-2 represents the tune, which is a sequence of characters where each character represents some musical note.

**MIDI (Musical Instrument Digital Interface):** MIDI is a technical standard that describes a communications protocol, digital interface, and electrical connectors that connect a wide variety of electronic musical instruments, computers, and related audio devices for playing, editing and recording music.

MIDI files are structured into *chunks*.

Each chunk consists of:

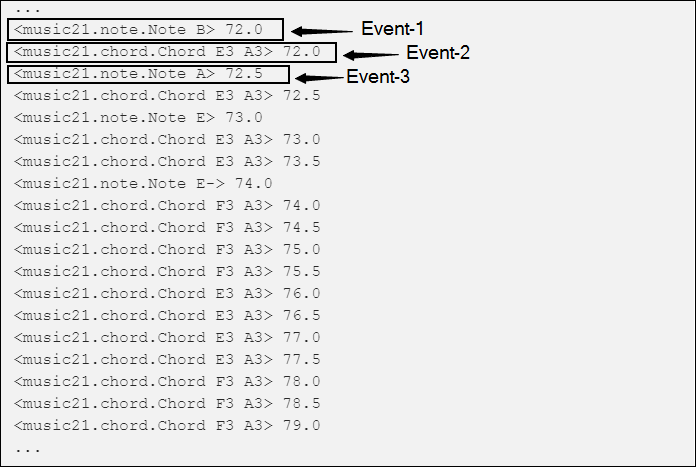
A 4-byte *chunk type* (ascii)

A 4-byte *length* (32 bits, msb first)

*length* bytes of data

There are two types of chunks. Namely, Header Chunks(which have a chunk type of "MThd") and Track Chunks(which have a chunk type of "MTrk").

A MIDI file consists of a single header chunk followed by one or more track chunks. MIDI itself does not make sound, it is just a series of messages like “note on,” “note off,” “note/pitch,” “pitch-bend,” and many more. These messages are interpreted by a MIDI instrument to produce sound.



Above images shows the representation of music generated using Music21 which is a python library which generates MIDI format music. Here, it shows Event-1 which is basically Note B. Then Event-2 represents Chord E3 A3 then Event-3 represents Note A and so on.

In this project we made use of MIDI format to represent the audio files.

**Motivation**

We have been acquainted with the concept of machine learning during our time spent on developing our knowledge and found that field to be quite interesting. We started exploring different parts of the field and it further developed our interest in it. It fuelled our desire use to apply this concept to solve some real life problems and we decided to do the same as our final year project. Machine Learning is already being used in various fields for further advancements.

As technology develops, it has always been our goal to digitalise all tasks and make thing easier. But one field which we have not been fully successfully been able to apply machine learning to perfection is arts. Music is one part of arts in which researches are currently being conducted. There is always a constant need for new songs which have to be composed again and again to maintain variety. Using machine learning to solve this problem can be beneficial.

Our mentors, Dr.Samayveer Singh and Dr. Paramvir Singh, advised us to explore all the present developments in this field and further plan to choose the best plan to complete our project.

1. **LITERATURE SURVEY**

2.1 SentiMozart

Facial expressions are one of the best and the most intuitive way to determine a persons emotions. They most naturally express how a person is feeling currently. The aim of the proposed framework is to generate music corresponding to the emotion of the person predicted by our model. The proposed framework is divided into two models, the Image Classification Model and the Music Generation Model. The music would be generated by the latter model which is essentially a Doubly Stacked LSTM architecture. This is to be done after classification and identification of the facial expression into one of the seven major sentiment categories: Angry, Disgust, Fear, Happy, Sad, Surprise and Neutral, which would be done by using Convolutional Neural Networks (CNN). Finally, the performance of the proposed framework is evaluated using the emotional Mean Opinion Score (MOS) which is a popular evaluation metric for audio-visual data.

2.2 Wavenet

A deep neural network for generating raw audio waveforms. The model is fully probabilistic and autoregressive, with the predictive distribution for each audio sample conditioned on all previous ones; nonetheless this paper shows that it can be efficiently trained on data with tens of thousands of samples per second of audio. When applied to text-to-speech, it yields state-of-the-art performance, with human listeners rating it as significantly more natural sounding than the best parametric and concatenative systems for both English and Mandarin. A single WaveNet can capture the characteristics of many different speakers with equal fidelity, and can switch between them by conditioning on the speaker identity. When trained to model music, we find that it generates novel and often highly realistic musical fragments. It is also shown that it can be employed as a discriminative model, returning promising results for phoneme recognition.

2.3 MuseGAN

Generating music has a few notable differences from generating images and videos. First, music is an art of time, necessitating a temporal model. Second, music is usually composed of multiple instruments/tracks with their own temporal dynamics, but collectively they unfold over time interdependently. Lastly, musical notes are often grouped into chords, arpeggios or melodies in polyphonic music, and thereby introducing a chronological ordering of notes is not naturally suitable. In this paper, three models for symbolic multi-track music generation under the framework of generative adversarial networks (GANs) are proposed. The three models, which differ in the underlying assumptions and accordingly the network architectures, are referred to as the jamming model, the composer model and the hybrid model. The proposed models are trained on a dataset of over one hundred thousand bars of rock music and then applied to generate piano-rolls of five tracks: bass, drums, guitar, piano and strings. A few intra-track and inter-track objective metrics are also proposed to evaluate the generative results, in addition to a subjective user study. This shows that the models can generate coherent music of four bars right from scratch (i.e. without human inputs). The models can also be extended to human-AI cooperative music generation: given a specific track composed by human, we can generate four additional tracks to accompany it.

2.4 A First Look at Music Composition using LSTM Recurrent Neural Networks

In general music composed by recurrent neural networks (RNNs) su®ers from a lack of global structure. Though networks can learn note-by-note transition probabilities and even reproduce phrases, attempts at learning an entire musical form and using that knowledge to guide composition have been unsuccessful. The reason for this failure seems to be that RNNs cannot keep track of temporally distant events that indicate global music structure. Long Short-Term Memory (LSTM) has succeeded in similar domains where other RNNs have failed, such as timing & counting and CSL learning. In the current study we show that LSTM is also a good mechanism for learning to compose music. We compare this approach to previous attempts, with particular focus on issues of data representation. We present experimental results showing that LSTM successfully learns a form of blues music and is able to compose novel (and we believe pleasing) melodies in that style. Remarkably, once the network has found the relevant structure it does not drift from it: LSTM is able to play the blues with good timing and proper structure as long as one is willing to listen.

#### PROBLEM STATEMENT AND OBJECTIVES

#### 

#### Problem Statement

There is constant demand for new musical content for a multitude of uses, ranging from artistic expression, to jingles for new TV shows, to music in games, to elevator music.

#### Objective

“To generate original music content using deep learning.”

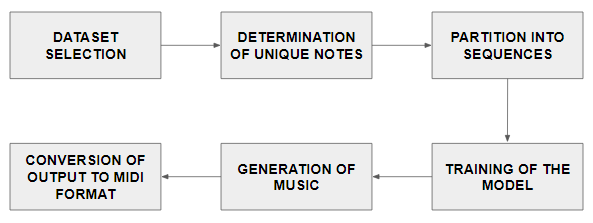
The model takes in a part of a music file as input and produces its continuation.

#### METHODOLOGY

Music generation can be seen as creation of new music content using pre-existing datasets.

Basic overview of the project involves six steps:

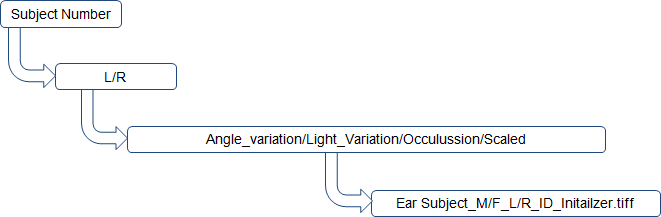
1. Dataset Selection: The initial step involves selection of songs which are similar in nature. Meaning which, they make use of same instruments and belong to the same genre. Selecting dissimilar songs results in the generation of unpleasant music.
2. Determination of unique notes: Music consists of different set of notes, it is essential for us to know what notes are present in the dataset being used. So all the music files in the dataset are gone through to gather all the unique notes.
3. Partition into sequences: The dataset is then divided into sequences of equal length which are further used for training the model.
4. Training the model: The project makes use of an LSTM model, which is trained using the sequences obtained in the above mentioned step.
5. Generation of music: Now, a single music file is preprocessed and sent as an input to the trained model which then predicts the further notes until a predefined length.
6. Conversion of output to MIDI format: The output of notes and chords produced is concatenated and converted into the MIDI format.



#### Figure: 1.1 (Block Diagram)

#### 

#### Image Acquisition

The side face images of around 510 subjects (both male and female) have been captured under different lighting conditions and with other variations in angle, scaling, and occlusion. For this purpose, we designed a GUI which we call as PhotoBooth App that does video streaming and extract the frames out of it. All images were taken with a distance of 15-20 cms between the ear and camera. The image size varies between 1-2 MB. The App also maintains a directory structure so that each image can be uniquely stored and identified in the database. The directory structure is shown the figure below:

#### Figure: 4.1(Directory Structure)

* + 1. Experimental Setup Tool Description:

We developed a tool for capturing the live images of subjects taken at various angle positions.

The tool which run on Linux/Windows OS had input parameters as Unique Id (unique for every subject – student roll number), Subject Number (to keep track of number of subjects), Image Initialization Number (ith image of subject) and various options (only one can be selected) like Left/Right and Occlusion/Angle Variation/Scale/Light Variation. Along with this there is a capture button and close button to quit the tool.

* + 1. Environment Description
       - Location - We chose the closed labs for the conduction of photo shoots. The labs were well lit with not so bright or dull light.
       - Members’ role - One of the member guided the students about the process in which they have to move their head in various angles, other member operated the tool and captured the pictures of subjects. Rest members assisted in the process of taking consent and in ensuring the stability of the setup.
       - Subjects’ role - The subjects were asked to sit on the chair (supported rotation) and rotate their head at various angles in in-plane and out plane fashion so as to get their pictures of right and left ear clicked.
       - Camera Used - The camera used was of Logitech with Autofocus supported camera with 5MP resolution and no flash support.
       - Click Positions – There were 10 positions in each in-plane and out-plane with varying angles from (-30, -15, 0, +15, +30) where ‘-’ signifies clockwise direction and ‘+’ signifies anti-clockwise direction. The same process involved both right and left ear of subject.
       - Subject Information – The subjects were in the age ranging from 18-22 years including both the genders (male/female). Total subjects covered is equal to 10,000 with 20 images of each subject.
       - Images and Head Rotation Information - Total 20 images were captured, i.e., with left ear we clicked 5 in-plane and 5 out-plane and same with right ear. The head is first set to any of the extreme positions, say -30 (in-plane) and then slightly rotated to next positions in the same plane, say -15 (in-plane) till it reach the other extreme position, say +30 (in- plane).
       - Sample Image Data (Size, Quality) - The image dimensions was fixed to 1280 960 pixels with the size of roughly 1 Mb in tiff format (raw format). All the images are of medium quality.

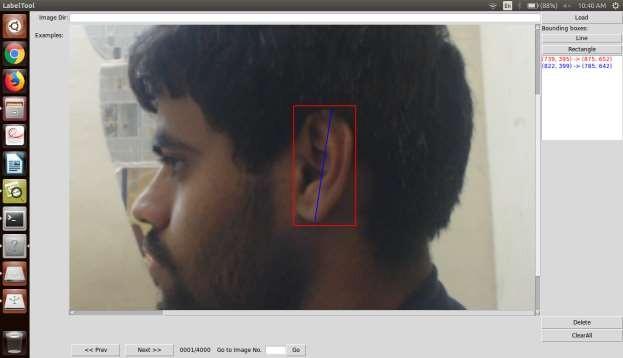
The image should be carefully taken such that outer ear shape is preserved. The less erroneous the outer shape is the more accurate the results are. The dataset will be open-sourced soon after completing some formalities. Some example pictures from the dataset are demonstrated as follows.



#### Figure: 4.2(Different Variations in Capturing Images for dataset)

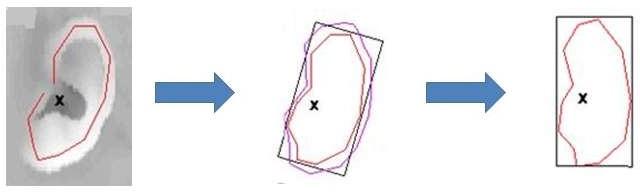
#### ROI Detection

The following stage is to extricate the ear i.e. Area of Interest (ROI) from the caught side face picture. To facilitate this step, we designed another tool named Labelling Tool to manually draw the boundaries around and close to the ear. As we do this, the coordinates of the rectangle are stored in a text file. After labelling all the images in this manner, we get our ROI.



#### Figure: 4.3(ROI in the Side Face Image)

#### Image Registration

Image registration is that the method of adjusting digital pictures so the results area unit a lot of appropriate for show or additional image analysis. Let’s say, you'll take away noise, sharpen, or brighten a picture, creating it easier to spot key options. The tools used for image registration embrace many alternative types of software package reminiscent of filters, image editors and alternative tools for dynamic numerous properties of a complete image or elements of a picture.

**Figure: 4.4(Ear Alignment)**

#### Feature Extraction

Biometric feature extraction is that the method by that key options of the sample area unit elect or increased. Typically, the method of feature extraction depends on a group of algorithms; the tactic varies counting on the kind of identification used. Additional refined forms of image improvement tools will apply changes additional specifically to bound elements of a picture. skilled packages like those offered by Adobe permit styleers to try to to a additional specialized or skilled reasonably image improvement or to pursue results for graphic design comes wherever the particular image is modified into a stylized or otherwise embellished version of itself. Additional advanced forms of image improvement tools conjointly embody options like Wiener filters for actual de-blurring of pictures and different advanced resources for reestablishing or informational pictures which will be in poor condition, on account of imperfect picture catch conditions, maturing or diverse causes.

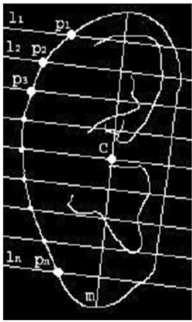
For ear, we have a tendency to use two-stage classification. Here options area unit extracted at angles. Options area unit divided into 2 vectors. 1st feature is found victimization the outer form of the ear. Second feature vector is found victimization all different edges. To seek out the angels, the terms max-line and traditional line area unit used.

* Max Line

It is the longest line that may be drawn with each its endpoints on the sides of the ear. The length of a line is measured in terms of geometer distance. If there area unit over one line, options comparable to every max-line area unit to be extracted. The max-line m, traditional line l1, l2, l3,….., ln named from high to bottom. Center of the max-line is c. P1, P2, P3,……, Pn area unit the points wherever the fringes and also the traditional lines see.

* Normal Line

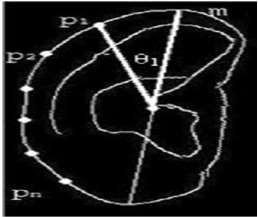
Lines that area unit perpendicular to the max-line and that divide the max-line into (n+1) equal elements, wherever n could be a positive number.



#### Figure: 4.5(Gray Scale Image of ear intersected by Normal Lines)

* + 1. First feature vector (FV1)

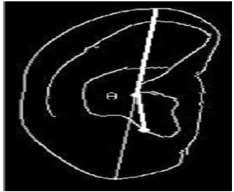
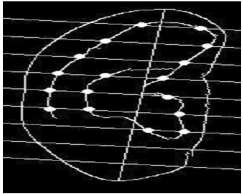
It can be defined by FV1 = [θ1, θ2, θ3,…., θn].



#### Figure: 4.6(Angles θ1, θ2, θ3,…., θn as FV1)

* + 1. Second feature vector (FV2)

All the points wherever the perimeters of the ear and traditional line ran into except the outer edge.



#### Figure: 4.7(Internal ear intersected by Normal Lines)

* + 1. Classification

Classification is that the task of finding a match for a given question image. Here classification is performed in 2 stages. In initial stage the primary feature vector is employed whereas in second stage second feature vector is employed. Let FV1 = [θ1, θ2, θ3,…., θn], FV2= [α1, α2, α3… αn] be the primary feature vectors of the 2 pictures that ar to be matched. exploitation these vectors 2 major distinction (d1) and therefore the variety of purpose matched (w1) are to be calculated



These two points are said to be matched if their corresponding angles are same or below the threshold value



Where Xi=1 if threshold |θi-αi| is less than some threshold value else Xi=0. The two images are said to be matched with respect to the first feature vector if d1 and w1 are less than some threshold values.

In the second stage two focuses are to be coordinated if their points are around same and furthermore they compare to a similar typical line. Give w2 a chance to be the quantity of the focuses that are coordinated however as the extent of the second component vector isn't settled the level of the coordinated focuses is computed by the formula given below



Where V1, V2 are the sizes of the second feature vector of the two images. Two images are to be matched finally if they are matched with respect to the first feature vector and the point Pt is greater than some threshold value.

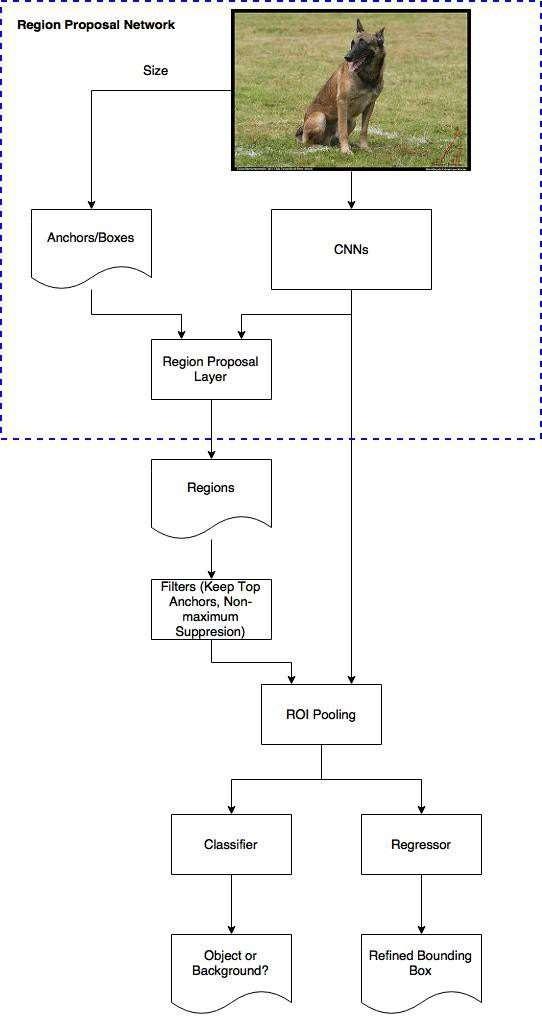
* + 1. Advantage of two stage classification

1. A given question image is initial tested against all the photographs within the info mistreatment initial feature vector.
2. Solely the photographs square measure matched within the initial stage square measure thought of for second stage of classification.
3. As the size of the FV1 is less, that is n (number of normal line) so only n comparison is needed for the first stage classification.
4. In the second stage characterization m\*n examination are required, expecting m focuses for every typical line.
5. If the classification is single stage, than total comparison required are I\*((n) + (m\*n)), where is the number of images in the database.
6. If the order is separated into two phase the correlation would be I\*n+I1\*(m\*n) where I1 is the quantity of picture that are coordinated as for the principal highlight vector.
7. Saved calculation is (I – I1)\*(m\*n).

#### Matching

* + 1. Model Description: Faster R-CNN

Faster R-CNN has 2 networks: region proposal network (REGION PROPOSED NETWORK) for generating region proposals and a network mistreatment these proposals to sight objects. the most completely different here with Fast R-CNN is that the later uses selective search to get region proposals. The time price of generating region proposals is far smaller in REGION PROPOSED NETWORK than selective search, once REGION PROPOSED NETWORK shares the foremost computation with the item detection network. Briefly, REGION PROPOSED NETWORK ranks region boxes (called anchors) and proposes those presumably containing objects. The design is as follows:



#### Figure: 4.8(Flow Chart of REGION PROPOSED NETWORK)

* Anchors:

Anchors play a very important role in Faster R-CNN. Associate degree anchor may be a box. Within the default configuration of Faster R-CNN, there square measure nine anchors at an edge of a picture.

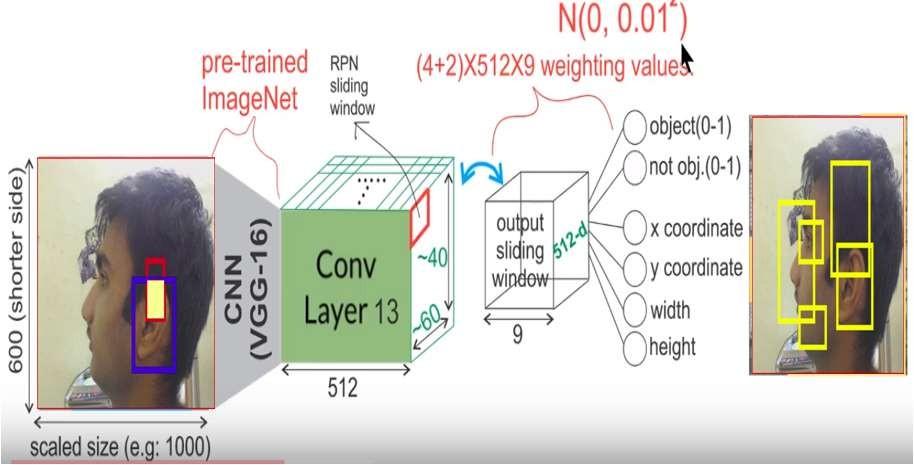
* Region Proposal Network:

The output of an area proposal network (REGION PROPOSED NETWORK) may be a bunch of boxes/proposals that may be examined by a classifier and regressor to eventually check the incidence of objects. To be a lot of precise, REGION PROPOSED NETWORK predicts the likelihood of associate degree anchor being background or foreground, and refine the anchor.

* + 1. Model Training

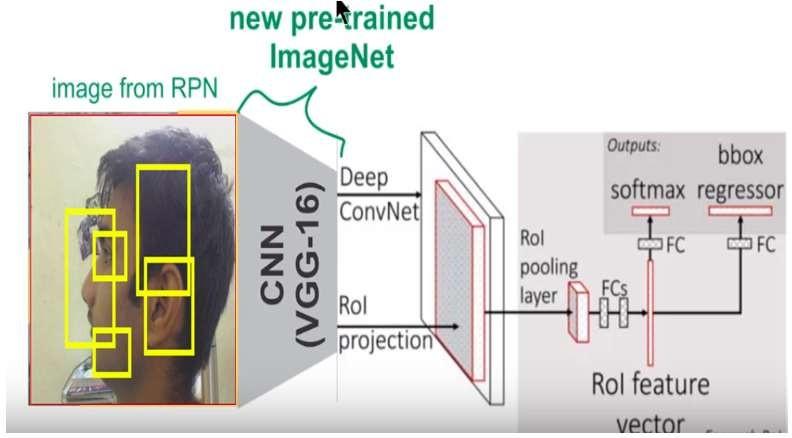
Training a model is done in the following steps:

1. Train Region Proposed Network , initialized with ImageNet pre-trained model (Proposer).



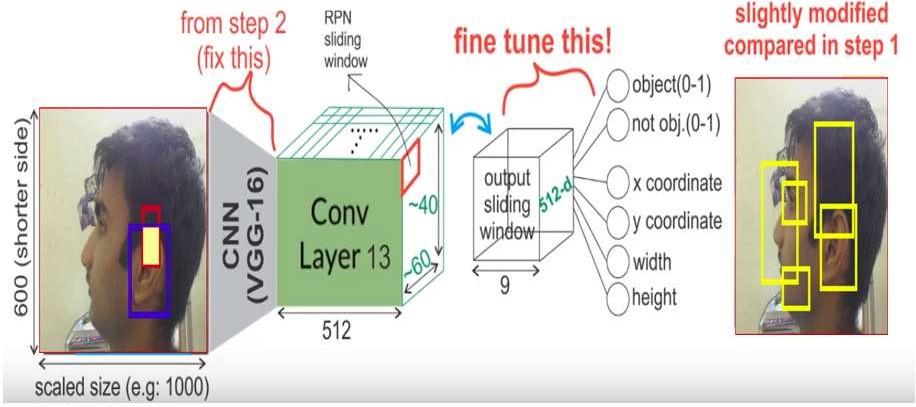
#### Figure: 4.9(Input to Pre-trained Model and Getting IO1 with collection of bounding boxes)

1. Train a separate detection network by Fast R-CNN using proposals generated by Step1 Region Proposed Network (Detector)

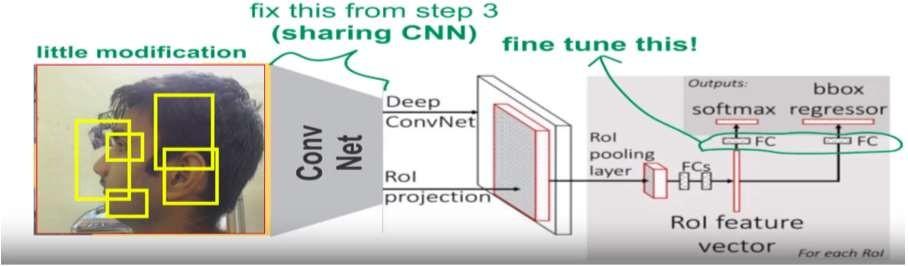


#### Figure: 4.10(Giving IO1 to new pre-trained ImageNet)

1. Fix convolution layer , fine-tune distinctive layers to Region proposed network , initialized by detector network in Step two (i.e. Fine Tuned Proposer)

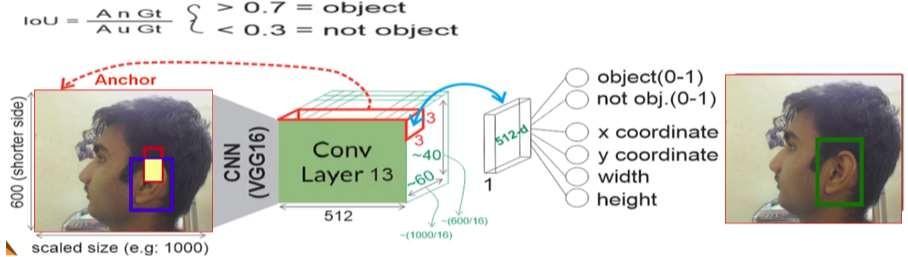


#### Figure: 4.11(Getting IO2)

1. Fix convolution layer, fine-tune fully connected layers of Fast R-CNN (Fine Tune Detector))

#### Figure: 4.12(Giving IO2 as Input)

* + 1. Region Proposed Network as Proposor



#### Figure: 4.13(Final Output of the Model) Training, target and loss functions

The REGION PROPOSED NETWORK completes 2 entirely unexpected sort of expectations: the paired grouping and also the bounding box regression adjustment.

For coaching, we tend to take all the anchors and place them into 2 totally different classes. The individuals who cover a ground-truth question with AN Intersection over Union (IoU) bigger than

0.5 zone unit thought of "closer view" and individuals that don't cover any ground truth protest or have underneath 0.1 note of hand with ground-truth objects area unit thought of “background”.

Then, we tend to indiscriminately sample those anchors to create a mini batch of size 256 — making an attempt to take care of a balanced magnitude relation between foreground and background anchors.

The REGION PROPOSED NETWORK uses all the anchors elite for the mini batch to calculate the classification loss victimization binary cross entropy. At that point, it utilizes exclusively those minibatch stays set apart as frontal area to ascertain the relapse misfortune. For shrewd the targets for the regression, we tend to use the foreground anchor and also the nearest ground truth object and calculate the proper \DeltaΔ required to rework the anchor into the thing.

Instead of employing a straightforward L1 or L2 loss for the regression error, the paper suggests victimization sleek L1 loss. Sleek L1 is essentially L1, however once the L1 error is little enough, outlined by an exact sigma, the error is taken into account nearly correct and also the loss diminishes at a quicker rate.

Using dynamic batches is difficult for variety of reasons. Even if we tend to try and maintain a balanced magnitude relation between anchors that area unit thought of background and people that area unit thought of foreground, that's not invariably doable. Betting on the bottom truth objects within the image and also the size and ratios of the anchors, it's doable to finish up with zero foreground anchors. In those cases, we tend to communicate victimization the anchors with the most important note of hand to the bottom truth boxes. This can be far away from ideal, however sensible within the sense that we tend to invariably have foreground samples and targets to find out from.

#### Post processing

**Non-maximum suppression:** Since anchors sometimes overlap, proposals find yourself additionally overlapping over an equivalent object. To resolve the problem of duplicate proposals we tend to use an easy recursive approach known as Non-Maximum Suppression (NMS). NMS takes the rundown of proposition arranged by score and repeats over the arranged rundown, disposing of those recommendations that have a note of hand bigger than some predefined threshold with a proposal that features a higher score.

While this appearance straightforward, it's important to take care with the note of hand threshold. Too low and you will find yourself missing proposals for objects; too high and you'll find yourself with too several proposals for an equivalent object. A value generally utilized is 0.6.

**Proposal selection:** while applying NMS, we tend to keep the most astounding N recommendations arranged by score. Within the paper N=2000N=2000 is employed, however it's doable to lower that range to as very little as fifty and still get quite smart results.

#### Standalone application

The REGION PROPOSED NETWORK is utilized independent from anyone else while not having the second stage demonstrate. In issues wherever there's solely one category of objects, the objectness chance is used because the final category chance. This can be as a result of for this case, “foreground” = “single class” and “background” = “no single class”.

Some samples of machine learning issues which will take pleasure in a standalone usage of the REGION PROPOSED NETWORK area unit the popular (but still challenging) face detection and text detection.

One of the benefits of victimization solely the REGION PROPOSED NETWORK is that the gain in speed each in coaching and prediction. Since the REGION PROPOSED NETWORK could be a terribly straightforward network that solely uses convolutional layers, the prediction time is quicker than victimization the classification base network.

#### Training and targets

Targets for R-CNN area unit calculated in nearly an equivalent means because the REGION PROPOSED NETWORK targets, however taking under consideration the various doable categories. We tend to take the proposals and also the ground-truth boxes, and calculate the note of hand between them.

Those proposition that have a note of hand greater than 0.5 with any ground truth box get distributed to it ground truth. The individuals who have in the vicinity of 0.1 and 0.5 get labeled as background. Contrary to what we tend to did whereas collecting targets for the REGION PROPOSED NETWORK, we tend to ignore proposals with none intersection. This can be as a result of at this stage we tend to area unit presumptuous that we've smart proposals and that we area unit additional inquisitive about resolution the tougher cases. Of course, of these values area unit hyper parameters which will be tuned to raise match the kind of objects that you just try to seek out.

The targets for the bounding box regression area unit calculated because the offset between the proposal and its corresponding ground-truth box, just for those proposals that are allotted a category supported the note of hand threshold.

We indiscriminately sample a balanced mini batch of size sixty four within which we've up to twenty fifth foreground proposals (with class) and seventy fifth background.

Following an equivalent path as we tend to did for the REGION PROPOSED NETWORKs losses, the classification loss is currently a multiclass cross entropy loss, victimization all the chosen proposals and also the sleek L1 loss for the twenty fifth proposals that area unit matched to a ground truth box. we've to take care once obtaining that loss since the output of the R-CNN absolutely connected network for bounding box regressions has one prediction for every of the

categories. Once shrewd the loss, we tend to solely need to take under consideration the one for the proper category.

#### Post processing

Similar to the REGION PROPOSED NETWORK, we tend to find yourself with a bunch of objects with categories allotted which require any process before returning them.

In order to use the bounding box changes we've to require under consideration that is that the category with the very best chance for that proposal. We tend to even have to ignore those proposals that have the background category because the one with the very best chance.

After obtaining the ultimate objects and ignoring those foreseen as background, we tend to apply class-based NMS. This can be done by grouping the objects by category, sorting them by chance so applying NMS to every freelance cluster before change of integrity them once more.

For our final list of objects, we tend to can also set a chance threshold and a limit on the quantity of objects for every category.

#### Training

In the original paper, quicker R-CNN was trained employing a multi-step approach, coaching elements severally and merging the trained weights before a final full coaching approach. Since then, it's been found that doing end-to-end, joint coaching results in higher results.

After golf stroke the entire model along we tend to find yourself with four totally different losses, 2 for the REGION PROPOSED NETWORK and 2 for R-CNN. We’ve the trainable layers in REGION PROPOSED NETWORK and R-CNN, and that we even have the bottom network that we will train (fine-tune) or not.

The decision to coach the bottom network depends on the character of the objects we wish to find out and also the computing power accessible. If we wish to find objects that area unit just like those who were on the first dataset on that the bottom network was trained on, then there's no real would like aside from making an attempt to squeeze all the doable performance we will get. On the opposite hand, coaching the bottom network is dearly-won each in time and on the required hardware, to be able to match the entire gradients.

The four very surprising misfortunes territory unit joined utilizing a weighted aggregate. This can be as a result of we tend to might want to grant classification losses additional weight relative to regression ones, or even provide R-CNN losses additional power over the REGION PROPOSED NETWORKs’.

Aside from the customary misfortunes, we have a tendency to try and have the regularization misfortunes that we have a tendency to skipped for quickness anyway is delineated each in REGION PROPOSED NETWORK and in R-CNN. We tend to use L2 regularization for a few of

the layers and betting on that base network being employed and if it’s trained, it's going to even have regularization.

We train victimization random Gradient Descent with momentum, setting the momentum worth to

0.9. You'll essentially prepare Faster R-CNN with the other streamlining agent while not catching any enormous disadvantage.

The learning rate begins at 0.0010.001 so abatements to 0.00010.0001 when 50K stages. This can be one in all the hyper parameters that typically matters the foremost. Once coaching with Luminoth, we tend to sometimes begin with the defaults and tune it from then on.

#### Evaluation

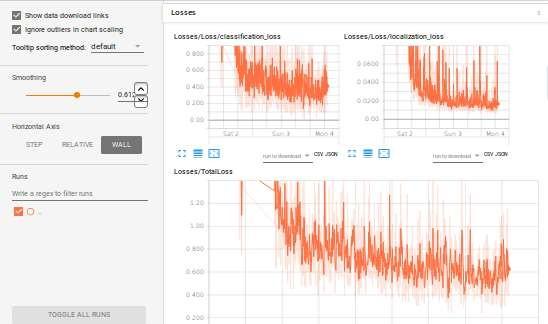
The analysis is finished victimization the quality Mean Average preciseness (mAP) at some specific note of hand threshold (e.g. mAP@0.5). mAP could be a metric that comes from data retrieval, and is often used for shrewd the error in ranking issues and for evaluating object detection issues. We won’t enter details since these kind of metrics merit a blogpost of their own, however the necessary takeaway is that mAP penalizes you after you miss a box that you just ought to have detected, additionally as after you find one {thing} that doesn't exist or find an equivalent thing multiple times.

#### RESULTS

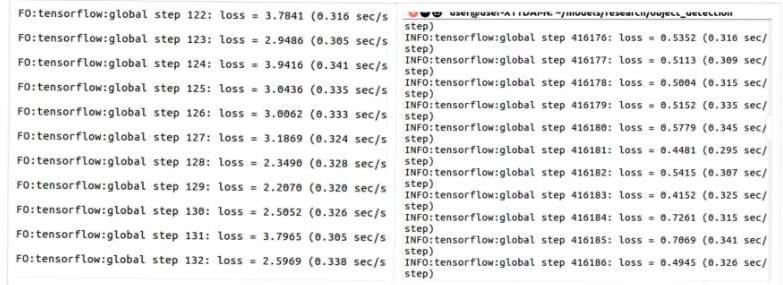
#### Result

The results are as follows:

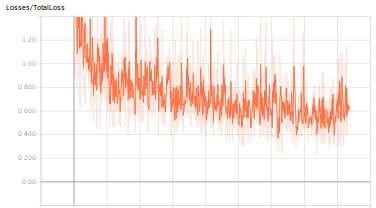
* + 1. A bounding box enclosing ear.
    2. A label assigned to bounding box.
    3. A probability for each label and bounding box.



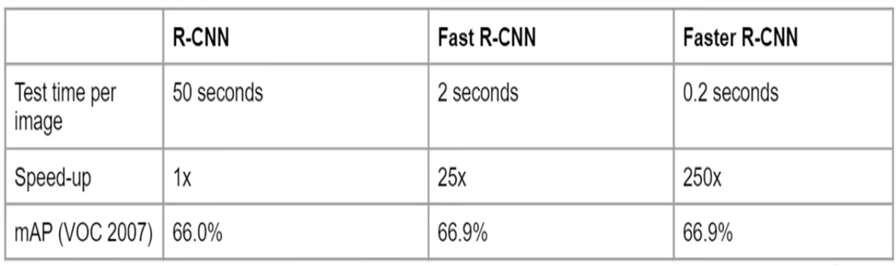
#### Figure: 5.1(Loss Graph With Classification Loss and Localization Loss)

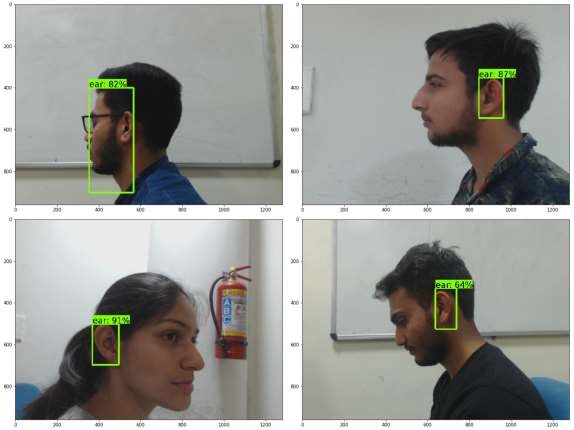
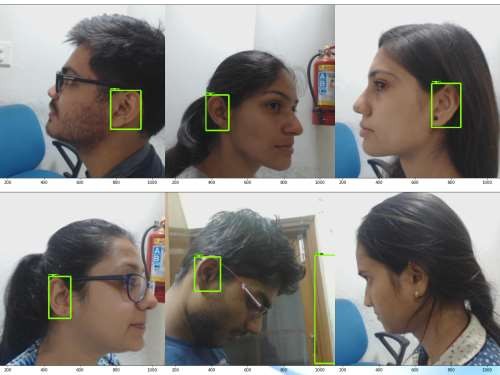


**Figure: 5.2(Losses After 100 and 4 Lakh Iterations)**

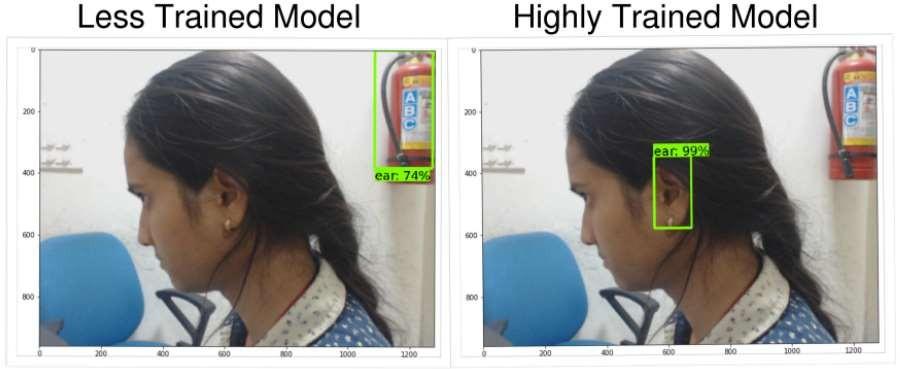


**Figure: 5.3(Loss Graph)**

**Table: 5.1(Results)**



**Figure: 5.4(Detected Ear With % of Accuracy Written on it)**



**Figure: 5.5(Comparison between Less Trained Model and Highly Trained Model)**



**Figure: 5.6(Resulted Image with Bounding Box and % of Accuracy)**

#### Applications of Ear Recognition System

Ear biometrics can be very acknowledged biometrics by clients in conceivable access control applications and government security, for example, visa/international ID programs. As per clients, the ear biometrics is less rushed than fingerprinting. In addition, clients conceded that they would feel less great while participating in confront picture acknowledgment since individuals tend to mind what they look like on photos.

Moreover, in the ear biometrics frameworks there is no compelling reason to contact any gadgets and in this manner there are no issues with cleanliness.

* + 1. Application of Android-based Ear Biometrics Identification

Smartphone was selected as application technology because they are easy to use and can be used anywhere and anytime. This is new arrangement and has never been given by past research, this application titled "Use of Android-based Ear Biometrics Identification". This application utilizes three strategies in the extraction procedure of ear includes that is utilizing Canny Edge Detection, Hough Transform, and Oriented FAST and Rotated BRIEF (ORB). These three techniques will be utilized to search for ear highlights. Vigilant Edge Detection is utilized to discover the edges or ear lines to get ear shape, subsequent to getting the line will be looked ear roundabout lines utilizing Hough Transform, lastly Oriented FAST and Rotated BRIEF (ORB) is utilized to locate the key point from the ear.

This application anticipated that would have the capacity to perceive the ear picture as per the proprietor's information exactly and precisely.

* + 1. Development of Multimodal Identification System using Face and Ear

Ear pictures will function supplements for different biometric modalities in automatic recognition systems and supply identity cues once different data is unreliable or maybe inaccessible. In police work applications, to Illustrate, wherever face recognition technology could struggle with profile faces, the ear will function a supply of data on the identity of individuals within the police work footage. The system provides a brand new approach to the identification system by mistreatment biometry face and ear of human mechanically. This approach consists of 3 stages adore pre- processing, feature extraction with 3 approaches, with combined Principal element Analysis (PCA), Linear Discriminant Analysis (LDA), and form Dimensions, matching, and decision-making with regulated threshold result shows that the face and ear match square measure combined as biometry.

#### CONCLUSION AND FUTURE SCOPE

#### Conclusion

The project was completed successfully to develop a system on Ear Recognition System. This framework can be utilized for validation purposes in blend with other biometric frameworks. It adds to an additional level of security. Additionally its rich and stable structure that is safeguarded since birth and is very one of a kind in people gives favorable circumstances over other biometric frameworks.

#### Future Scope

Although many algorithms for ear detection and recognition are planned within the literature, however there are not any industrial biometric recognition systems at now that overtly use options of the ear for human recognition. However the performances of ear recognition algorithms are tested on some customary ear datasets and experiments counsel that the ear pictures may result in smart recognition accuracy on ideal image. However, the performance of ear recognition ways on non-ideal pictures obtained below varied illumination and occlusion conditions is however to be established.

The main challenge is how to deal with occlusions by jewelry, hair, eyeglasses and clothing. If the ear is completely occluded, there is no possibility for identification by using ear recognition. It is vital to judge the strength of ear recognition system in terms of the degree of occlusion acceptable to the system. a technique to handle such occlusion is by capturing the thermogram in conjunction with the actinic ray image. In a thermogram, the hair can be easily detected as its temperature is usually lower than that of the skin. Essentially to confront acknowledgment, impediment because of posture varieties is another test in ear acknowledgment framework. Several seeing edges must be thought about, from profile to frontal since in applications, similar to observation, subjects may show up in any view. Calculations should conjointly outfit for tilt pictures.

Ear recognition continues to be a vigorous field of analysis. Though the ear life science is made in characteristics however there ar still some issues that require to be worked on to create automatic ear recognition system simpler and economical in world applications. Additionally to variation in illumination, alternative open analysis issues embody occlusion because of hair, ear symmetry, ear print, ear classification, and ear individuality. It’s doable to use the infrared pictures of ears to beat the matter of occlusion of the ear by hair. For enhancing the execution of ear biometrics, we can consolidate the ear qualities with face and side countenances. This is known as multi-biometrics or multimodal biometrics. Likewise we can enhance the execution of the ear acknowledgment framework by utilizing combination systems. In this fusion technique, features set are extracted by using different methods for a same image and then the matching scores are generated by all features set all fused for final decision. The future research work should be extended upon 3D images so detection rate as well recognition accuracy can be improved.

#### REFERENCES

1. Yi Zhang and Zhichun Mu: ‘Ear Detection under Uncontrolled Conditions with Multiple Scale Faster Region‐ Based Convolutional Neural Networks’, published in Symmetry 2017, 9, 53
2. Emersic, Z., Stepec, D., Struc, V., Peer, P., George, A., Ahmad, A., Omar, E., Boult, T. E., Safdari, R., Zhou, Y., Zafeiriou, S., Yaman, D., Eyiokur, F. I., Ekenel, H. K.: ’The unconstrained ear recognition challenge’, International Joint Conference on Biometrics (IJCB), 2017
3. Fevziye Irem Eyiokur , Dogucan Yaman , Hazım Kemal Ekenel: ‘Domain Adaptation for Ear Recognition Using Deep Convolutional Neural Networks’, arXiv:1803.07801v1 [cs.CV] 21 Mar 2018
4. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ’ImageNet classification with deep convolutional neural networks’, Advances in Neural Information Processing Systems (NIPS), 2012, pp. 1097- 1105
5. Simonyan, K., Zisserman, A.: ’Very deep convolutional networks for large-scale image recognition’, International Conference on Learning Representations (ICLR), 2015
6. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: ’Going deeper with convolutions’, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1-9
7. Yan, P.; Bowyer, K. Empirical evaluation of advanced ear biometrics. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; p. 41.
8. Yan, P.; Bowyer, K.W. Biometric recognition using 3D ear shape. IEEE Trans. Pattern Anal. Mach. Intell.2010, 29, 1297–1308.
9. Deepak, R.; Nayak, A.V.; Manikantan, K. Ear detection using active contour model. In Proceedings of the International Conference on Emerging Trends in Engineering, Technology and Science, Pudukkottai, India, 24–26 February 2016; pp. 1–7.
10. Chen H.;Bhanu B. Contour matching for 3D ear recognition. In Proceedings of the IEEE Workshop on Applications of Computer Vision and Motion and Video Computing, Breckenridge, CO, USA, 5–7 January 2005; pp. 123–128.
11. Chen, H.; Bhanu, B. Shape model‐ based 3D ear detection from side face range images. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition‐ Workshops, San Diego, CA, USA, 21–23 September 2005; p. 122.
12. Chen, H.; Bhanu, B. Human ear recognition in 3D. IEEE Trans. Pattern Anal. Mach. Intell.2007, 29, 718–737.
13. Ganesh, M.R.; Krishna, R.; Manikantan, K.; Ramachandran, S. Entropy based binary particle swarm optimization and classification for ear detection. Eng. Appl. Artif. Intell.2014, 27, 115–128
14. Sana, A.; Gupta, P.; Purkai, R. Ear biometrics: A new approach. In Advances in Pattern Recognition; Pal, P., Ed.; World Scientific Publishing: Singapore, 2007; pp. 46–50.
15. Prakash, S.; Jayaraman, U.; Gupta, P. A skin‐ color and template based technique for automatic ear detection. In Proceedings of the International Conference on Advances in Pattern Recognition (ICAPR 2009), Kolkata, India, 4–6 February 2009; pp. 213–216.
16. Attarchi, S.; Faez, K.; Rafiei, A. A new segmentation approach for ear recognition. Lect. Notes Comput. Sci.2008, 5259, 1030–1037.
17. Halawani, A.; Li, H. Human ear localization: A template‐ based approach. In Proceedings of the International Workshop on Pattern Recognition (ICOPR 2015), Dubai, UAE, 4–5 May 2015.
18. Joshi, K.V. Oval shape detection and support vector machine based approach for human ear detection from 2D profile face image. Int. J. Hybrid Inf. Technol.2014, 7, 113–120.
19. Islam, S.M.S.; Bennamoun, M.; Davies, R. Fast and fully automatic ear detection using cascaded AdaBoost. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Copper Mountain, CO, USA,7–9 January 2008; pp. 1–6.
20. Abaza, A.; Hebert, C.; Harrison, M.A.F. Fast learning ear detection for real‐ time surveillance. In Proceedings of the Fourth IEEE International Conference on Biometrics: Theory Applications and Systems, Washington, DC, USA, 27–29 September 2010; pp. 1–6.
21. Shih, H.C.; Ho, C.C.; Chang, H.T.; Wu, C.S. Ear detection based on arc‐ masking extraction and AdaBoost polling verification. In Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH‐ MSP 2009), Kyoto, Japan, 12–14 September 2009; pp. 669–672.
22. Yuan, L.; Mu, Z. Ear recognition based on Gabor features and KFDA. Sci.WorldJ.2014, 2014, doi: org/10.1155/2014/702076
23. Biometric Ear Recognition System by Neha Kuduk, Akshada Hinge, Kirti Kshirsagar published in International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 03 | Mar -2017
24. B. S. El-Desouky , M. El-Kady , M. Z. Rashad , Mahmoud M. Eid: ‘Ear Recognition And Occlusion’, International Journal of Computer Science & Information Technology (IJCSIT) Vol 4,

No 6, December 2012

1. G. Giacinto and F. Roli. Design of effective neural network ensembles for image classification processes. Image Vision and Computing Journal, 19(9–10):699–707, 2001.
2. Li Yuan and Zhichun Mu: ‘Ear Recognition based on 2D Images’, IEEE Publications