Major Technical Project (MTP) - 2017

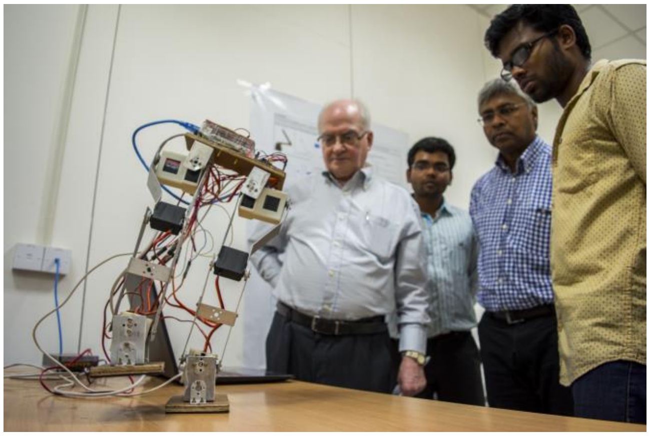
School of Computing & Electrical Engineering

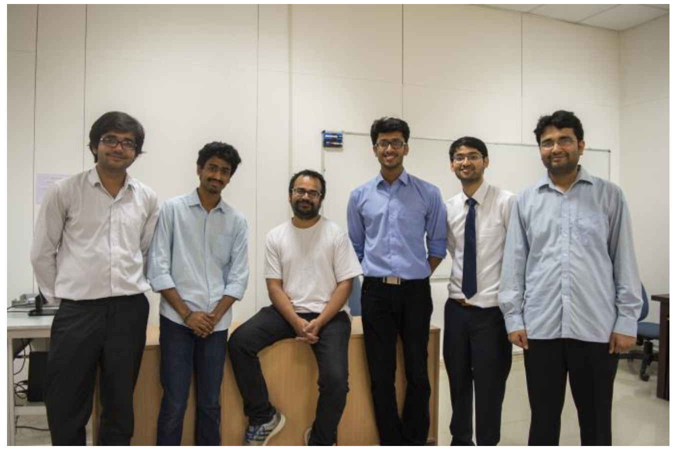
IIT Mandi Kamand Himachal Pradesh

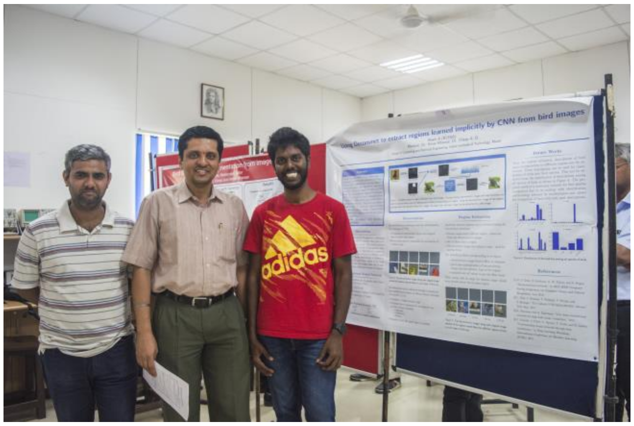
MTP Coordinators

Dr. Arnav Bhavsar (arnav@iitmandi.ac.in)

Dr. Samar Agnihotri (samar@iitmandi.ac.in)






adidas

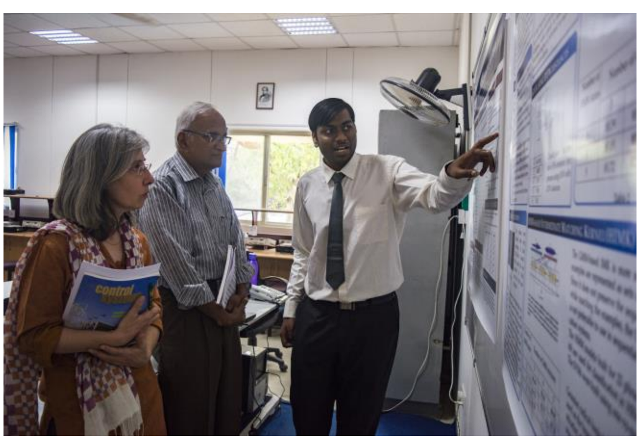




Table of Content

12	Opening Lead Recommendation in Contract Bridge
13	Line Fault Detection in Tennis and Video Summarization
14	Modeling Human Decisions with Deep and Conventional Reinforcement Learning Algorithms in Games
15	Power Quality Analysis in Power Distribution System
16	Quantum Maxflow Mincut Problem
17	Setting up of WRF-GFS Model for Regional Climate Prediction
18	Streaming Based Genome Mapping on Apache Flink
19	Wireless Power Transfer 2.0

Algorithm implementation for Computation of Network Coding Capacity Bounds

Toyaz Sai Madhav, Thejas Babu
Dr.Satyajit Thakor

IIT MANDI

Introduction

- Network Coding is a networking technique in which transmitted data is encoded and decoded to increase the network throughput, reduce delays and make the network more robust.
- It increases the throughput(capacity) of a network and it is useful to know the bound on the throughput that can be achieved using this technique.
- There are many bounds existing in theory, such as geometrical bound, graphical bound which require high computational effort as mentioned in [1].
- In [1], [2], theory and algorithms which reduces the computational effort of computing graphical bounds are developed.The algorithms which reduce the computational effort to find the capacity bounds developed in above mentioned papers are to be implemented effectively.

Objective

- The algorithms AllMaxSetsC in [1], is implemented along with Procedure B and Procedure C in [1] as part of internship by few of my colleagues.
- Our objective is to start from this point and implement other algorithms which are AllIrrSets(computation of all irreducible sets), ReducedLPEq(in [2]) and ReducedLPIneq(in [2]) in an efficient way and design new algorithm which is efficient than previous if possible.
- The output of the AllMaxSetsC and AllIrrSets is given as input to the other two algorithms in [2] to get the final result.

Network Coding Capacity Bound

Polymatroid Function:
A function h:$2^X \rightarrow$ R is *polymatroidal* if it satisfies the following axioms for all disjoint A, B $\subseteq X$

$$h(A \mid X \setminus \{A\}) \geq 0, A \in X \quad (1a)$$

$$I(A;B \mid C) \geq 0, A \neq B; C \subseteq X \setminus \{A,B\} \quad (1b)$$

- The above are the polymatroidal inequalities along with constraints C1, C2, C3, C4 in [1] characterise the network coding capacity region.
- Some of the inequalities 1a, 1b may become redundant when above contraints are imposed.
- Hence these redundant inequalities removal reduces number of inequalities and dimensions.

Butterfly Network

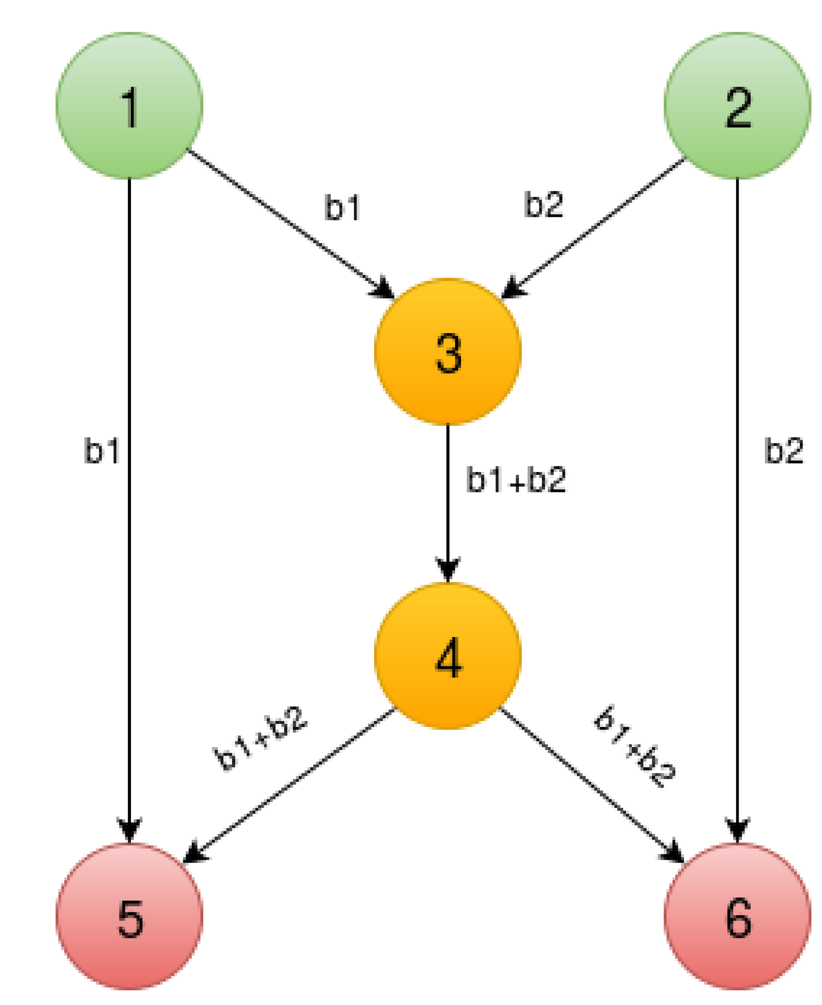


Figure 1: Butterfly Network

Above shown is the butterfly network along with which network coding working principle is depicted.

Algorithm Flow

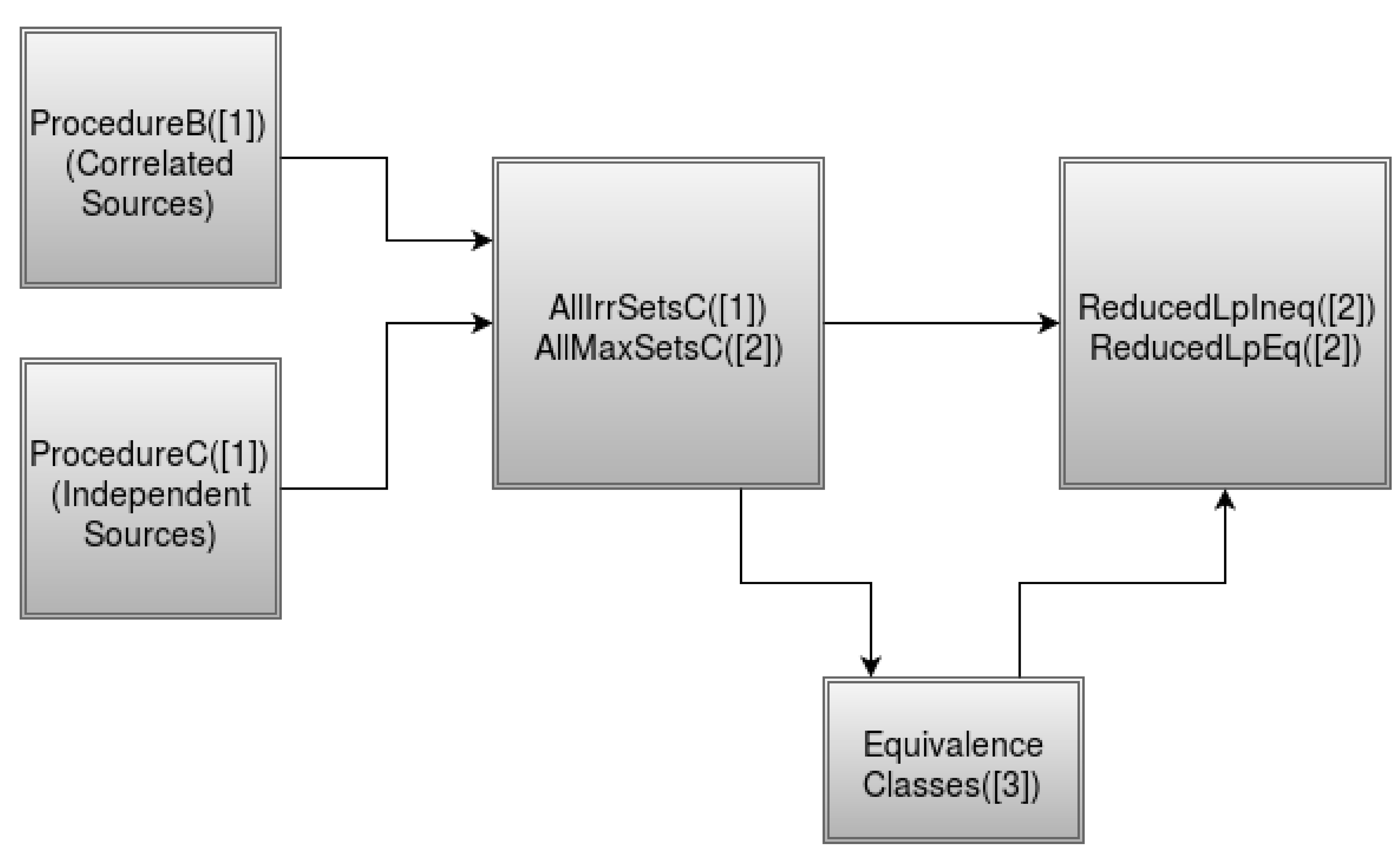


Figure 2: Algortihms flowchart

Results

Algortihm Used	Inequalities	Dimensions
	Reduction(%)	Reduction(%)
-	4617(0%)	512(0%)
ReducedLpIneq([2])	760(83.54%)	54(92.38%)
Equivalence class	329(92.87%)	40(92.19%)

Table 1: Butterfly network Results(Correlated Sources)

Algortihm Used	Inequalities	Dimensions
	Reduction(%)	Reduction(%)
-	4617(0%)	512(0%)
ReducedLpIneq([2])	449(90.28%)	39(95.31%)
Equivalence class	115(97.51%)	19(96.29%)

Table 2: Butterfly network Results(Independent Sources)

- Some random graphs are generated by using approach used in [4] and more simulation results are generated by processing these networks
- These simulation networks FDG's nodes vary from 5 to 15.

References

[1] S. Thakor, A. Grant, and T. Chan. *Cut-Set Bounds on Network Information Flow.* IEEE TRANSACTIONS ON INFORMATION THEORY, Vol. 62, No. 4, April 2016.

[2] S. Thakor, A. Grant, and T. Chan. *On complexity reduction of the LPbound computation and related problems* in Proc. Int. Symp. Netw. Coding, Jul. 2011, pp. 1 - 6.

[3] S. Thakor, A. Grant, and T. Chan. A Minimal Set of Shannon-type Inequalities for Functional Dependence Structures, accepted in International Symposium on Information Theory, 2017.

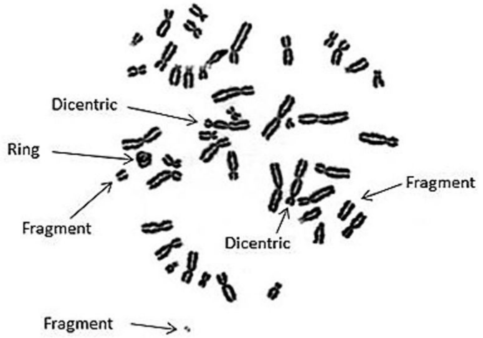
[4] S. Thakor, Xiaoli Xu and Y. L. Guan. *Reduced functional dependence graphs.* IET Networks,12th June 2014.

Automated Discrimination of Dicentric and Monocentric Chromosomes

Onkar Singh, Swapnil Sharma and Dr. A. K. Sao

OBJECTIVE

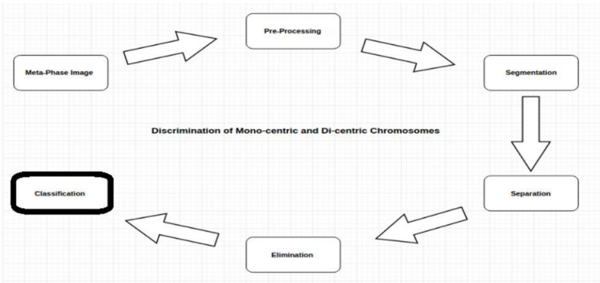
- Evaluate cell slides efficiency for further computation.
- To segment chromosomes from a given cell slide.
- To identify dicentric chromosomes present in a cell slide.

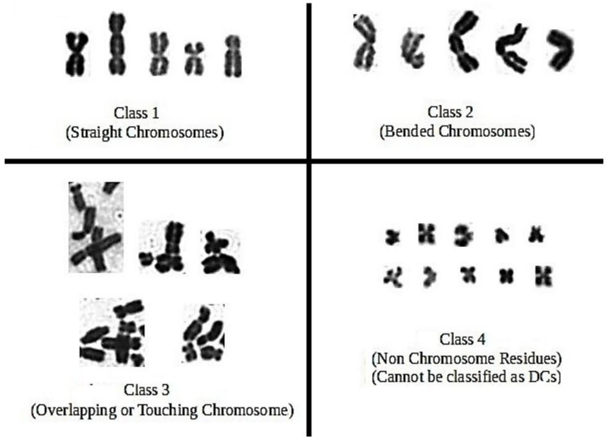


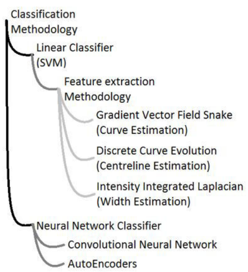
MOTIVATION

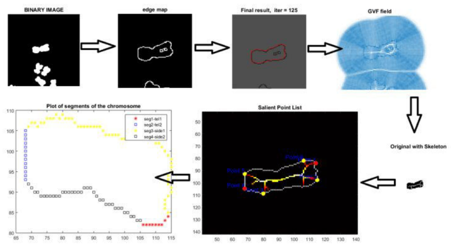
- Radiation exposure is harmful.
- Medication of radiation depends on accurate assessment of absorbed radiation dose.
- The only validated method to estimate radiation dosage is dicentric chromosomal assay.
- An individual can complete one slide of metaphases manually in 1-2 days working 7-8 hours/day.

APPROACH

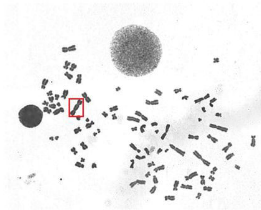




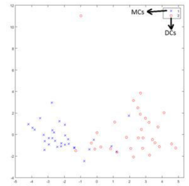




RESULTS

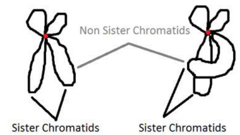


Segmented DCs



Feature Plot for DCs and MCs

PROBLEMS



Sister Chromatid Separation

Poor metaphase slides and poor quality images

FUTURE WORK

- Statistical evaluation using huge corpus
- Feature selection can be further enhanced to incorporate better distinction.
- DNN framework can be explored for detection and segmentation

REFERENCES

- Y. Li, J. H. Knoll, R. C. Wilkins, F. N. Flegal, and P. K. Rogan, “Automated discrimination of dicentric and monocentric chromosomes by machine learning-based image processing,” Microscopy research and technique, 2016.
- R. Uttamatanin, P. Yuvapoositanon, A. Intarapanich, S. Kaewkamnerd,
R. Phuksaritanon, A. Assawamakin, and S. Tongsima, “Metasel: a metaphase selection tool using a gaussian-based classification technique,” BMC bioinformatics, Vol. 14, no. 16, p. S13, 2013.

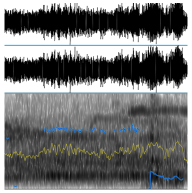
All of metaphase dataset Provided by DRDO

Ball Tracking, Event Detection, Video Summarization of Football

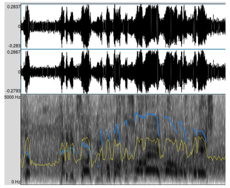
Vishwa Mangal

Mentor: Dr. Aditya Nigam

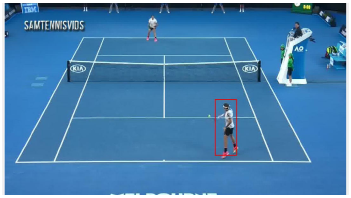
- Ball Tracking done using STRUCT (structured output for kernel Tracking).
- Ground truth for the first few frames are given and then it tracks the ball in subsequent frames.
- Stores the features from the ground truth and searches in radius of "R" which best match the feature.

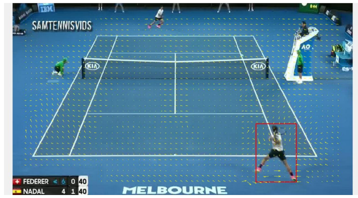


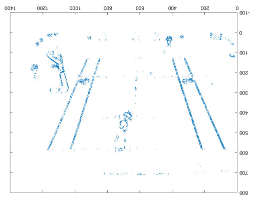
- Audio summarization using PRAAT language.
- High Pitch implies the important moment of match.
- The threshold was found at 60% of the mean pitch.



Raw Data collection of Forehand, Backhand, and Service → Prediction of player using YOLO in each frame. → optical flow between the frames → Using MBH to remove the noise. → Train using 3D - CNN







Data Specification

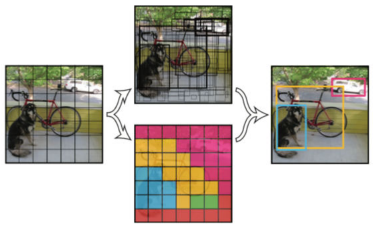
- Australian open final 2017.
- video to frame at 15FPS.
- Manually classified frames into forehand, backhand, service.
- 30 instances each of forehand, backhand, and service.

Performance

- Train/Test split at 50/10 for 2 class and 80/10 for three class.
- On testing with two class i.e forehand and backhand, we got the 100% accuracy.
- On testing with three class we got the accuracy of 70%.

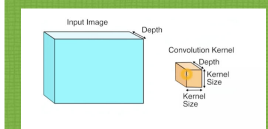
Future Work

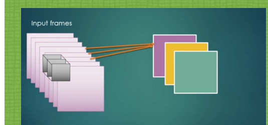
- Classify the strokes in untrimmed videos using T - CNN.



Working Of YOLO
YOU ONLY LOOK ONCE

3D - CNN






```
('Accuracy:', 0.80000001)
('Epoch', 500, 'completed out of', 500,
Prediction shape: (?, 2)
y shape: <unknown>
('Accuracy:', 0.80000001)
Done. Finishing accuracy:
('Accuracy:', 1.0)
('fitment percent:', 1)
```

70% accuracy with 3 class .

100% accuracy with 2 class.

```
('Accuracy:', 0.40000001)
('Epoch', 500, 'completed out of', 500, 'loss:'
Prediction shape: (?, 3)
y shape: <unknown>
('Accuracy:', 0.5)
Done. Finishing accuracy:
('Accuracy:', 0.70000005)
('fitment percent:', 1)
```

Using Deconvnet to extract regions learned implicitly by CNN from bird images

Aswin A (B13305)
Mentors: Dr. Arnav Bhavsar, Dr. Dileep A. D.
School of Computing and Electrical Engineering, Indian Institute of Technology, Mandi

Abstract

Identification of regions of the object plays an important role in the task of fine-grained classification. For classifying birds, various regions of birds such as head, body, leg, tail, etc. have to be identified. Filters of higher level layers of CNN trained to classify different species of birds implicitly learn regions of the birds mentioned above. Identification of these filters can be achieved with the help of deconvnet[1]. Using the deconvolution image outputted by deconvnet, regions of birds are extracted from bird images. Caltech-UCSD Birds-200-2011[2] dataset is used for the evaluation of our method.

Keywords: *fine-grained, region identification, deconvolution*

Introduction

Fine-grained recognition of objects is a challenging problem in the field of computer vision and is becoming increasingly popular. The overall structure of objects is almost similar in fine-grained classification.This problem is addressed by humans by looking at the minute variation in different parts of the object. Similar methodology is followed in this work and as a first step, regions of the birds are extracted from bird images which will be used to identify minute differences between bird species.

Convolutional Neural Network

An image is classified into its species using Convolutional Neural Network (CNN).

- Architecture - VGG19 [3]
- Weights - fine tuned for Caltech-UCSD dataset [2]
- Features from $pool5$ and $fc7$ layers
- Linear SVM

Input Features	Accuracy (%)
$pool5$	76.7
$fc7$	77.9

Table 1: *Performance on species classification.*

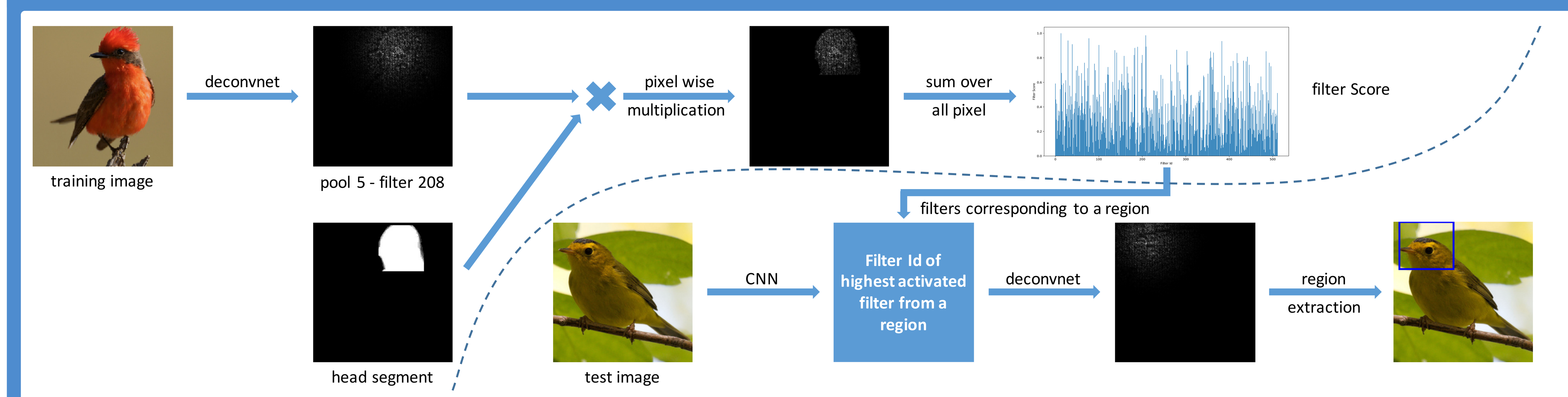


Figure 1: Complete process of region extraction from bird images. During training, deconvolution images and segmentations are used to calculate filter scores for selecting set of filters corresponding to a region. Using the deconvolution image of the highest activated filter from the set of filters for a region, the region from the input image can be extracted for a test image.

Deconvolution

Deconvolution is a technique used for understanding the learnings of CNN.

- Activations of intermediate layers are mapped back to input pixel space
- Shows input patters responsible for activations of a feature map

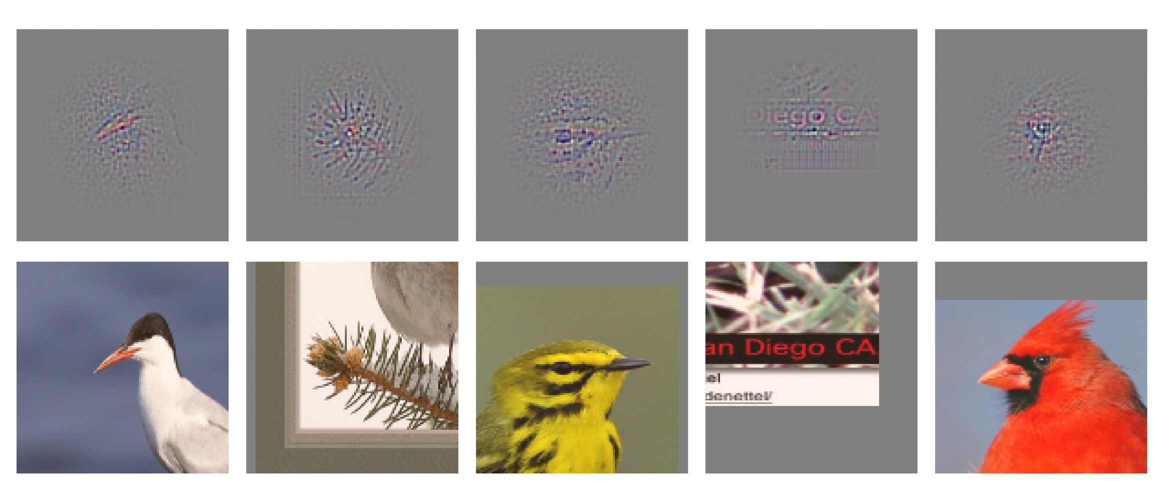


Figure 2: Top deconvolution images along with original image patches for randomly selected filters in $pool4$ layer of VGG net.

Deconvolution can be achieved through a neural network architecture known as Deconvolution Network (deconvnet) [1] [4].

- Attached to target feature map
- Performs three operations to reconstruct the activity of layer beneath:
  - unpooling
  - rectification
  - filtering
- Repeated until the input layer is reached

Region Extraction

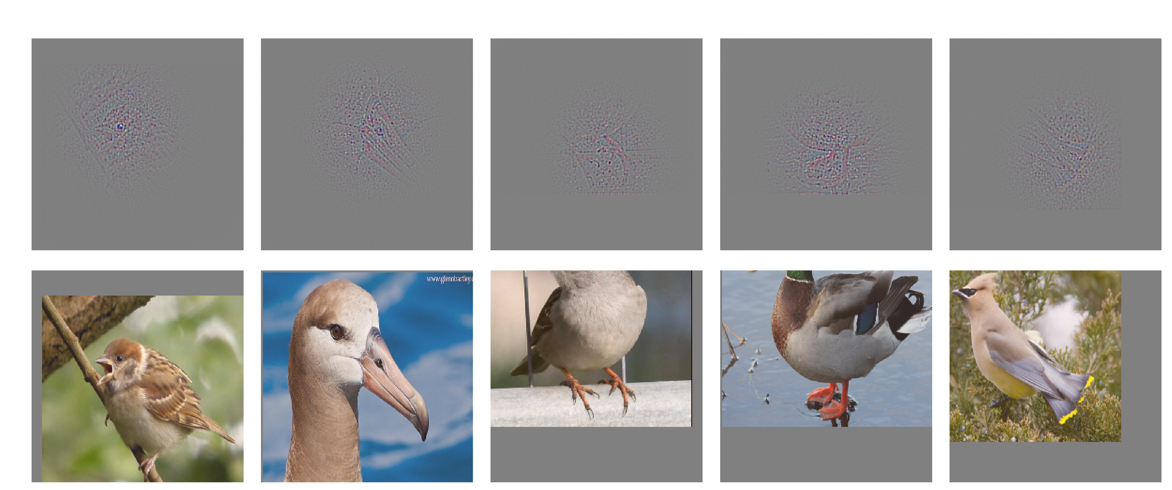
Implicitly learned regions can be extracted using deconvolution technique.

- Feature maps learn different region - observed from top deconvolution images
- Set of filters learn a region
- Deconvolution image and saliency map - used for scoring filters

For identifying filters corresponding to a region,

- deconvolution image from a filter is obtained
- pixel wise multiplication of deconvolution image and the saliency map of the region
- summed over all values to get the filter score

Filter scores will be large for feature maps which learn details about the focused region.



$(a)Head$ $(b)Head$ $(c)Leg$ $(d)Leg$ $(e)Tail$

Figure 3: Top deconvolution images along with original image patches of the highest scored filters for different regions of bird in $pool5$ layer of VGG net.

Future Works

From the regions extracted, descriptions of bird parts corresponding to different regions can be extracted. These descriptions can be used for classification of birds into their species. This can be observed from the distribution of descriptions among all species of birds as shown in Figure 4. The proposed method is extremely suitable for bird species recognition due to its analogy with identification methodology followed by ornithologists and allows for recognition of birds with only part descriptions.

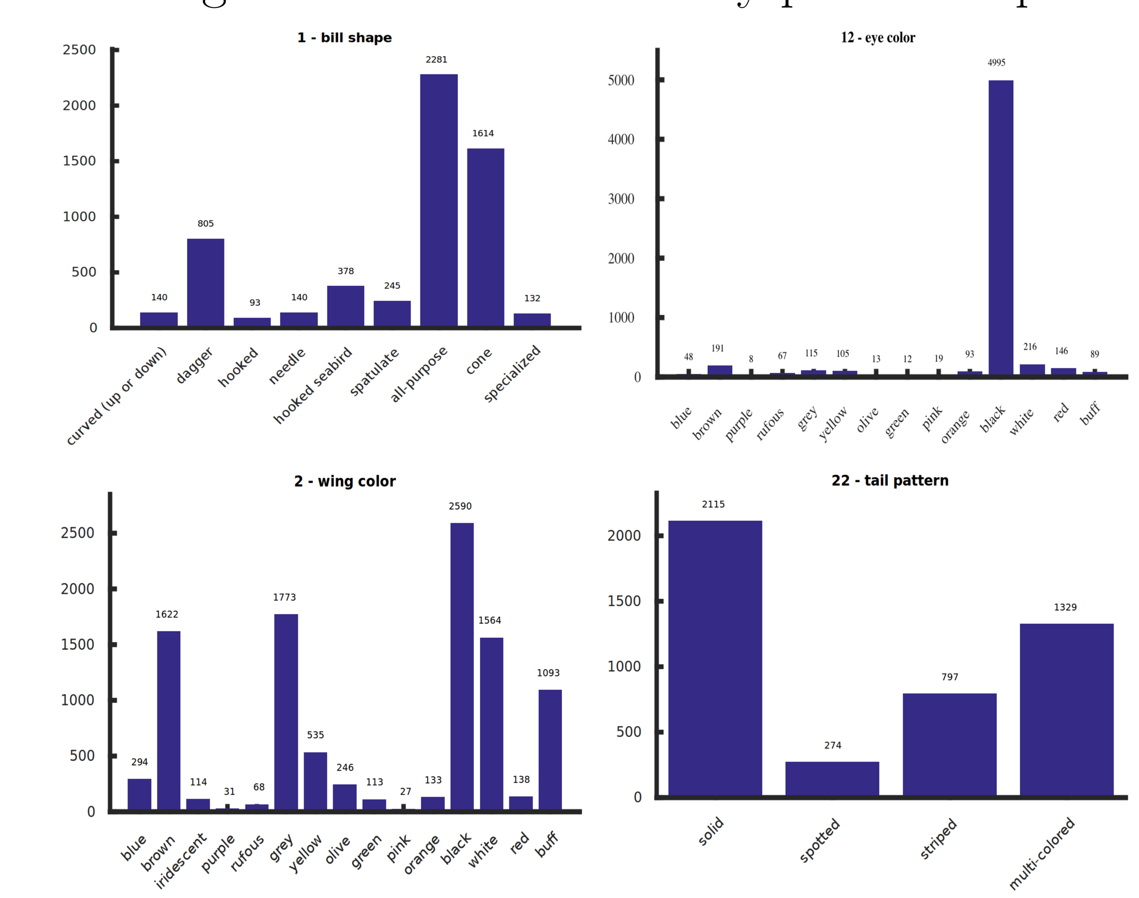


Figure 4: Distribution of descriptions among all species of birds.

References

[1] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010.

[2] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep., 2011.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[4] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," in *Deep Learning Workshop, International Conference on Machine Learning (ICML)*, 2015.

Bird Phrase Classification Using Audio Signals

Hitesh Tarani (B13139)

Guide: Dr. Padmanabhan Rajan & Dr. Dileep A.D.

School of Computing and Electrical Engineering, IIT Mandi

Introduction

Acoustic communication in birds is quite rich and is one of the primary ways in which birds make their presence known to each other, as well as this is the most direct way for humans to detect them, often at times when they are difficult to see. Therefore there is a great need for increased use and further development of automated analysis of avian sounds.

Bird phrases

Bird Songs typically comprise a sequence of smaller units, termed ***phrases***, which are separated from each other by longer pauses(silences).

- Need to preserve **sequence information** associated in the production of the bird songs.
- Automatic phrase recognition system challenging due to noisy environments and limited training data.
- Sparse-Representation(SR) and Support Vector Machines (SVM) have been previously used.
- Dynamic kernel based SVMs have performed considerably well in classifying bird species. Also HMM-based dynamic kernels have been used for speaker recognition tasks.
  This motivates us to use them for bird phrase classification as well.

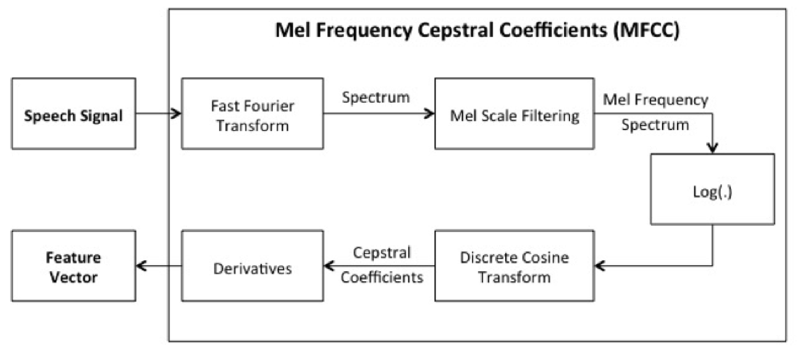
Dataset description

Cassin's Vireo

Bird phrases were obtained from song recordings of male CAVIs(*Vireo cassinii*). Each sound file was recorded in 16-bit, mono, 44.1 kHz sampling rate.

There are two collections of bird phrase data, viz. cassins_1, with 45 phrase classes and cassins_2, with 97 phrase classes. The number of clips per phrase class vary from 2 to 42 in cassins_1, and from 1 to 160 in cassins_2. A total of *36 phrase classes* and later a set of *22 common phrase classes* in common from cassins_1 and cassins_2 were considered for classification. For our classification purposes, cassins_1 is being used for training, and cassins_2 for testing. MFCC features were extracted from each file with frame size=*20ms* and frame shift=*10ms*.

MFCC Feature Extraction



In Mel Frequency Cepstral Coefficients(MFCC) feature extraction, each recording is divided into small windows, where speech is assumed stationary. Short-time analysis is performed on each frame as shown in above figure.

KNN using DTW distances

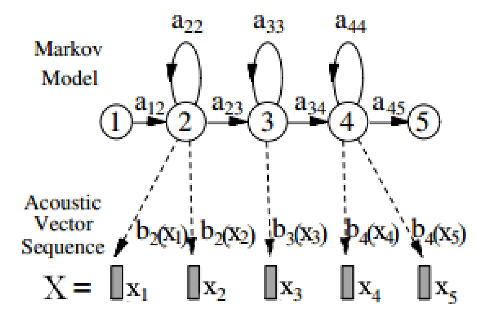
```
int DTWDistance(s: array [1..n], t: array [1..m]) {
   DTW := array [0..n, 0..m]

   for i := 1 to n
       DTW[i, 0] := infinity
   for i := 1 to m
       DTW[0, i] := infinity
   DTW[0, 0] := 0

   for i := 1 to n
       for j := 1 to m
           cost := d(s[i], t[j])
           DTW[i, j] := cost + minimum(DTW[i-1, j  ],    // insertion
                                       DTW[i  , j-1],    // deletion
                                       DTW[i-1, j-1])    // match

   return DTW[n, m]
}
```

k	Accuracy
2	**60.85**
4	53.97
5	59.08
6	58.11

Classification accuracies of 36 common bird phrases using kNN with DTW distances.

Hidden Markov Models(HMM)

A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. Continuous-density Hidden Markov Models(CDHMMs) using Gaussian mixture emissions were implemented using *hmmlearn*.



Probabilistic parameters of an Hidden Markov Model

GMM-HMMs were then built to classify bird phrases of 36 classes (common phrases from cassins_1 and cassins_2) by training from cassins_1 and testing from cassins_2. The data could be classified with an accuracy of *58-63%* as shown in Table 1.

Number of HMM states	Number of GMM mixtures		
	3	4	5
3	59.79	62.87	62.1
4	62.34	60.85	58.73
5	**63.23**	58.29	59.24

Table 1

Number of HMM states	Number of GMM mixtures		
	3	4	5
3	85.59	85.06	84.54
4	85.85	**85.98**	82.7
5	85.72	82.57	83.36

Table 2

There is no support of **Left-to-Right HMMs** in *hmmlearn*. However, for HIMK to preserve sequence of feature vectors, we need to use Left-to-Right HMMs only. So Hidden Markov Model Toolkit (HTK) were used for building them. Left-to-Right HMMs were then built for 22 phrase classes, having at least 10 examples per class. The maximum accuracy obtained was around *86%*.

HMM-based Intermediate Matching Kernel(HIMK)

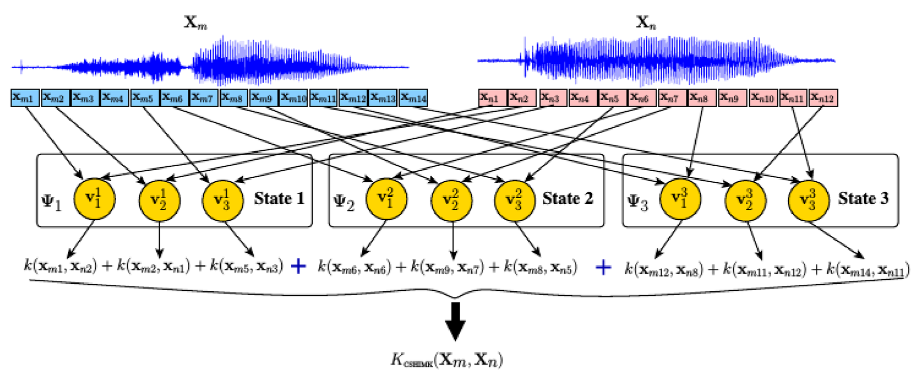


Fig. 6.2: Illustration of construction of CSHIMK between a pair of sequences of feature vectors $\mathbf{X}_m$ and $\mathbf{X}_n$. A 3-state left-to-right HMM with a 3-component GMM for each state is considered in the illustration. The Ψ_i denotes the GMM for state i and $\mathbf{v}_q^i$ denotes the qth component of Ψ_i. The feature vectors for matching are selected using the values of $R_{iq}(\mathbf{x}_t|\mathbf{X},\lambda)$.

Number of HMM states	Number of GMM mixtures		
	3	4	5
3	86.64	86.64	**86.91**
4	86.52	82.72	81.28
5	86.78	—	—

The GMM-based IMK is more suitable where the examples are represented as sets of feature vectors. Since it does not preserve the sequence information while matching the examples, the GMM-based IMK is not preferable to use in sequential pattern classification systems.
The HMM models built for 22 phrase classes using *htk* are used for classification in HIMK-based SVM. The SVMs were built using one-vs-rest in LIBSVM.

Gaussian radical basis function (RBF) kernel will be used as the base kernel. For multi-class pattern classification, the HIMK-based SVM models are built in one-against-the-rest class manner. An input test example will be passed into every SVM and will be classified by the *winner-takes-all* strategy which has the maximum discriminant score.

Conclusion

- kNN using Dynamic Time Warping(DTW) distances perform slightly lower to HMMs.
- Limiting phrase classes with at least 10 examples greatly improved the performance.
- HIMK-based SVMs perform similar to HMMs.
- Results obtained are very close to state-of-the-art techniques, viz. Sparse Representation(SR) and Support Vector Machines(SVM), which have an accuracy of 85-90%.

References

[1] L. N. Tan, K. Kaewtip, M. L. Cody, C. E. Taylor, and A. Alwan, "Evaluation of a sparse representation-based classifier for bird phrase classification under limited data conditions," 2012.

[2] A. D. Dileep, "Matching based dynamic kernels for classification of varying length patterns of speech," PhD Thesis, Department of Computer Science and Engineering, IIT Madras, Chennai, July 2013.

Bird Species Classification Using Audio Signals

Ayush Garg (B13111)
Guide:- Dr. Padmanabhan Rajan & Dr. A.D. Dileep
School Of Computing And Electrical Engineering, IIT Mandi

INTRODUCTION

Communication through **speech** in case of birds is quite elaborate and important just like human beings. This is also an important because this helps bird know the location of other birds and communicate with them, e.g. giving out a distress call. Within the Avian subclass there are about **10,000** bird species and the automatic recognition of these bird species is called **Bird Classification**.

There are a total of **209 species** found in and around the **Great Himalayan National Park(GHNP)** out of these we are considering **26** prominent bird species for our project.

SCOPE OF WORK

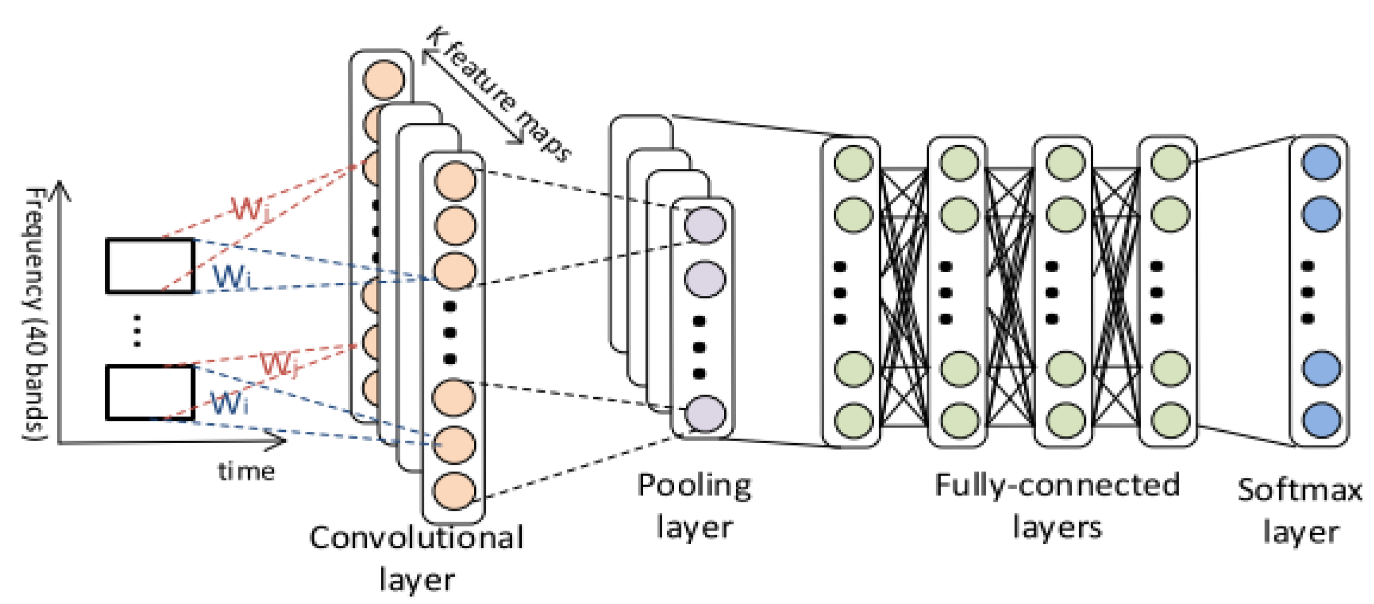
The main focus of our work is to explore state-of-the-art techniques applied to speech and **audio signals** for representing bird calls and for their classification.

The techniques we have used for the bird species classification:

- **CNN** with Mel filter-bank Energy Coefficients(**MFEC**) features.
- **CNN** with **raw data**(samples itself).

We compare our results with **DNNs**. We also try to visualize the filters learned by the CNN for both the approaches.

CNN ARCHITECTURE USED



The network is composed of **filter stage** followed by a **classification stage**. The filter stage consists of one or more **Convolution layers** with variable number of filters used and one or more **Pooling layers**. The classification stage consists of one or more **fully connected layers** followed by a **Softmax layer**.

REFERENCES

[1] T. S. Brandes, "Automated sound recording and analysis techniques for bird surveys and conservation," Bird Conservation international, vol. 18, pp. I63–S173, 2008.

[2] Deep Chakraborty, Paawan Mukker, Padmanabhan Rajan and A.D.Dileep, "Bird call identification using dynamic kernel based support vector machines and deep neural networks," 2016.

[3] Palaz, Dimitri Magimai.-Doss, Mathew Collobert, Ronan, "Convolutional neural networks-based continuous speech recognition using raw speech signal," 2015.

BIRD SPECIES

Table 1: Database of bird calls. The total training data for each species is approximately 14 minutes long.

Bird Species	Number of samples for training	Number of samples for test
Lesser Cuckoo	7	3
Black Throated Tit	8	12
Black and Yellow Grosbeak	10	12
Blackcrested Tit	5	9
Chestnut-crowned Laughingthrush	8	11
Eurasian Treecreeper	9	5
Golden Bushrobin	10	14
Great Barbet	10	20
Grey Bellied Cuckoo	10	7
Grey Bushchat	8	9
Greyhooded Warbler	7	3
Greywinged Blackbird	4	6
Himalayan Monal	11	25
Large-billed Crow	7	4
Orange-flanked Bushrobin	8	10
Oriental Cuckoo	9	7
Pale-rumped Warbler	6	6
Rock Bunting	6	7
Rufous-gorgetted Flycatcher	9	8
Rufous-bellied Niltava	9	9
Russet-backed Sparrow	14	38
Spotted Nutcracker	19	31
Streaked Laughingthrush	9	4
Western Tragopan	8	5
White-cheeked Nuthatch	10	52
Yellow-bellied Fantail	11	12

MFEC FEATURES

Extracting Mel Filter Bank Energy Coefficients (MFEC) is quite similar to the process of extracting MFCC features that are conventionally used. The MFCC features when used with CNN present a major problem because the discrete cosine transform projects the spectral energies into a new basis that may not maintain locality. Hence, we use log-energy computed Mel Filter Bank Energy Coefficients (MFEC) without the use of Discrete Cosine Transform (DCT).

MFEC features are commonly derived as follows:

- Take the Fourier transform of a signal(windowed).
- Map the powers of the spectrum obtained from above onto the mel scale, using triangular overlapping windows.
- Take the logs of the powers at each of the mel frequencies.

CNN WITH MFEC FEATURES

Representation of Data

- The avian sounds are each 44.1 KHz data.
- These audio signals are split into frames of 20ms each with a shift of 10ms.
- The context windows consisting of 15 such sequential frames taken at once.
- MFEC features are extracted from each frame and concatenated to form the context windows.

Architecture Used with MFEC features

- 1 convolution layer with window size of 5X15 and number of filters as 160.
- This is followed by one max pooling layer.
- The fully connected layer has 256 nodes.

Architecture Used with Raw data

- First convolution layer with window size 5X15 and 64 filters.
- Second convolution layer with window size 5X1 and 192 filters.
- Each convolution layer is followed by one max pooling layer.
- Then the fully connected layer again has 256 nodes.

Results

Table 2: Accuracy obtained using CNN for bird species classification with log MFEC features and raw data.

Input	32-MFEC	40-MFEC	48-MFEC	Raw Data
Accuracy	**98.48**	97.87	**98.48**	95.74

COMPARISON WITH DNN RESULTS

The DNN has been used conventionally for bird species classification. The following are the results from the Deep's paper using DNN on the bird species data.

Table 3: Accuracy obtained by using DNN for bird species classification with MFCC and log MFEC features.

Hidden Layers	Hidden Nodes	MFCC	log MFEC
2	256	95.44	97.17
	512	96.05	98.18
	1024	95.74	98.18
3	256	96.35	97.42
	512	**96.65**	**98.48**
	1024	96.35	98.18

CONCLUSION

We observed that CNN is a good approach for bird species classification giving up to 98.48% accuracy using MFEC features and also giving 95.74% accuracy with raw data. Therefore, we observe that CNN performs quite well even with the raw data itself which motivates us to think that CNNs can be applied on to spectrogram images extracted from bird audio signals. Also, CNNs and DNNs may be applied to a noisy dataset to realize which technique is better for noisy datasets.

Benchmarking Database systems for developing efficient spatial and non-spatial querying framework for astronomical databases

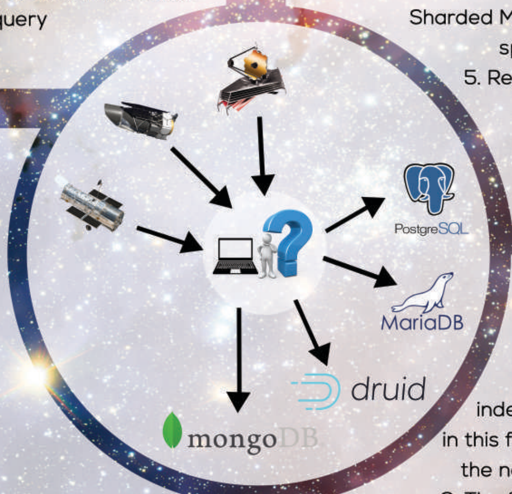
By: Ayush Yadav
Mentors: Dr. Sriram Kailasam (IIT Mandi), Dr. Ivelina Momcheva (STScI)

Objective

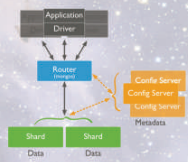
1. Compile a comprehensive list of spatial, range and search queries to benchmark based on the recent use by the scientific community
2. Evaluate performance for spatial, range and search queries on different databases and benchmark the results.
3. Identify specifications of the future frameworks for astronomical query processing.

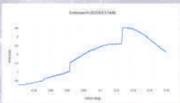
Work Done

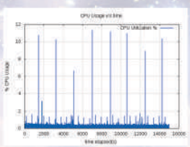
1. Evaluate the indexing techniques and performance on a single node implementation of PostgreSQL, Druid and MongoDb for spatial, non-spatial and range queries.
2. Study the indexing methods like R-Tree, B-Tree, Bitmap for different queries.
3. Scalability analysis for large datasets.
4. Extensive evaluation of the performance of Sharded MongoDB cluster on range, spatial and search queries.
5. Record the results and infer the limitations of the systems based on the query latency.



Results







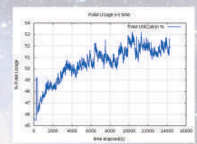


Table 3.2: Performance for Search Queries on different databases

Search Query Performance	Mean (s)	Min (s)	Max (s)
SQL	0.0341	0.0306	0.0407
Druid	0.0333	1.4430	0.0040
MongoDB	1.8712	2.1201	1.8003
MongoDB (Sharded)	0.0011	0.0023	0.0009

Recommendations

1. Since data loading is periodical hence partial indexing must be supported in this framework to quickly add the new data to search index.
2. The framework must support multiple indexing techniques at the same time, this is to make different types of queries work efficiently.
3. Indexing takes a lot of time hence a parallel hadoop job like implementation of building indexes on the distributed cluster is important and necessary.
4. The schema must be flexible to change the data model for different observations records, be it catalog, images, or normal spectral data.

Poster Credits: Samriddhi Jain

Classification of Musical Instruments from Audio

SCHOOL OF COMPUTING AND ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MANDI
MANDI - 175005, INDIA

Sparsh Saurabh [B13334], Ritesh Kumar [B13133]; **Mentors**: Padmanabhan Rajan, Arnav Bhavsar

INTRODUCTION

Around the world we have lot of musical instruments but they can be broadly classified into four major groups, viz., Percussion, Plucking, String and Wind. The aim of the project is to classify the instrument not just according to the group but also, according to the instrument itself. The project comprises of mainly three steps:

- Preprocessing of data.
- Feature extraction.
- Classification using classifiers.

Monophonic sound of every instrument is being used. The aim of the project is also to analyze the effect that recording session variability plays in classification process.

DATASET

- There are total 5 group of instruments (Flutes can be considered within wind instruments), with total of 16 instrument for our project as listed in Table 1.

Group	Instruments
Percussion	Drums, Mridangam, Tabla
Plucking	Guitar, Mandolin, Sitar, Veena, Banjo
String	Cello, Sarangi, Violin
Wind	Saxophone, Shehnai, Trumpet
Flutes	Bansuri, Western Flutes

Table 1: Groups and Instruments used in the Experiments

- Audio Files were collected from different sources on internet, a significant part of which came from YouTube. The audio files had sampling rate of 44.1 kHz.
- The audio files were segmented into 5s segments and cleaned manually using the steps shown in Diagram 1.

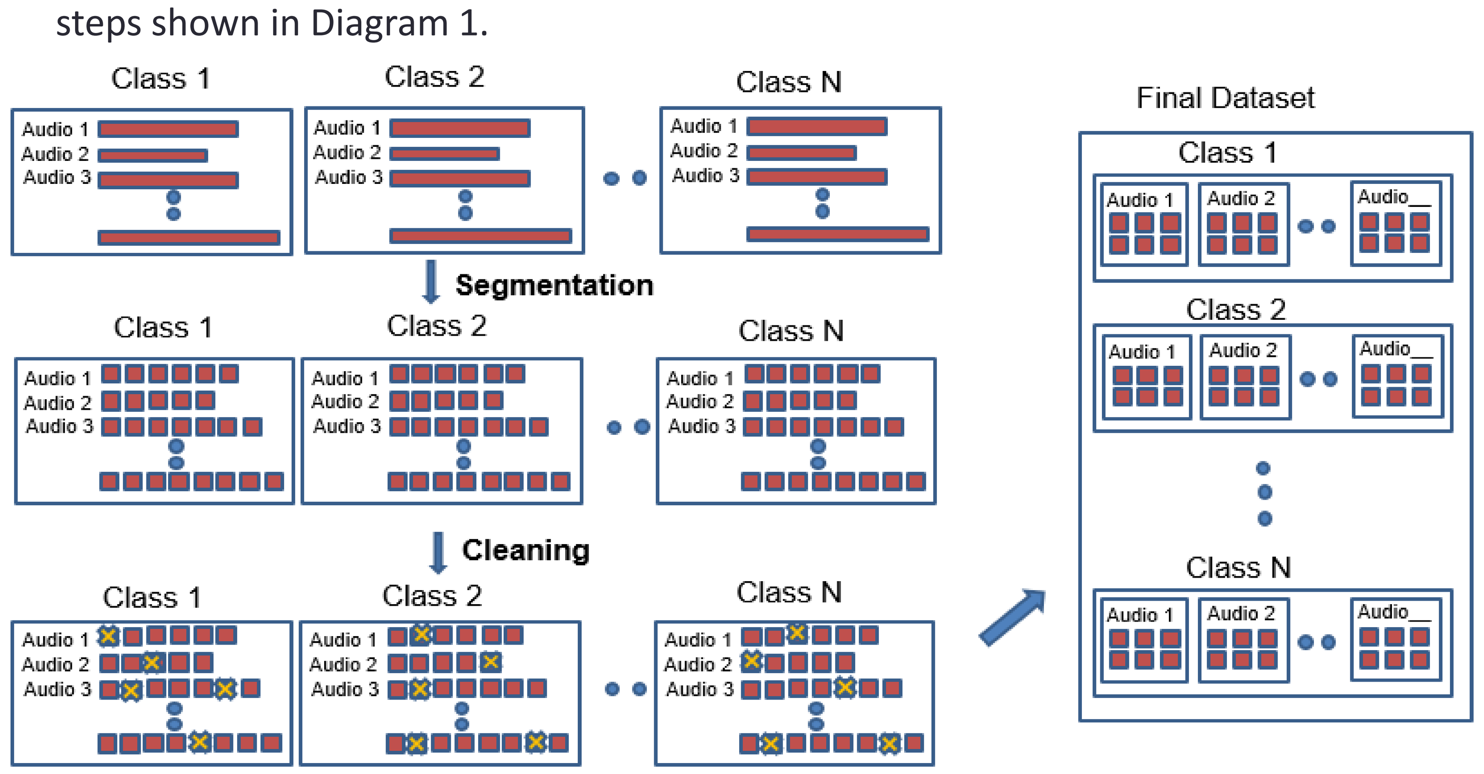


Diagram 1: Segmenting and Cleaning of Data

- To analyze the effects of session variability across different audio sources, four experiments were conducted:

1. Training has seen Testing audio source
   a. Training and Testing come from same audio source (Diagram 2).
   b. Training using all sources and Testing from any audio source (Diagram 3).
2. Training and Testing from different sources
   a. Training from a single source (Diagram 4).
   b. Training from multiple sources (70% for Training + 30% for Testing) (Diagram 5).

Diagram 2: Same Source for Train and Test
Diagram 3: Train using all Source
Diagram 4: Training using Single Source
Diagram 5: Training using Multiple Sources

For these experiments, dataset was divided once and this common dataset was used for conducting these experiments across different feature types and classification methods.

Apart from these four experiments, a dataset was also created for GMM-UBM, where 25% sources were used for UBM[1], 25% sources for training and rest 50% sources were used for testing.

FEATURE EXTRACTION AND CLASSIFIERS

For feature extraction, four different methods were used:

1. Mel-frequency cepstral coefficients (MFCC)[2]
2. MFCC with cepstral mean and variance normalization (CMV)[3]
3. MFEC
4. Chroma[4]

For classification, three methods were used:

1. Gaussian Mixture Model
2. Deep Neural Network
3. GMM-UBM (Universal Background Model)

RESULTS

The result of experiments done for checking effects of session variability are shown below in Table 2.

Expt. No.	Experiments	MFCC		CMV		MFEC	CHROMA
		GMM	DNN	GMM	DNN	DNN	GMM
1.	Training and Testing from same source	100.00%	99.07%	97.32%	94.67%	99.53%	93.13%
2.	Training using all sources	79.38%	95.59%	71.25%	87.538%	98.56%	85.63%
3.	Training from single source and Testing from other source	38.13%	37.73%	40.63%	46.90%	44.78	31.25%
4.	Training using 70% of the sources and Testing using rest 30%	56.30 %	62.44%	61.96%	67.846%	74.07%	42.96%

Table 2: Results for experiment to check effects of session variability

Results are shown for GMM with 10 mixtures in Table 2. Also, it must be noted that MFCC used in GMM was 39 dimensional while, in DNN, it was only 13 dimensional.

Following can be observed from Table 2:

- In Experiment 1 and Experiment 2, the highest accuracies are 100% and 98.56% respectively, while for Experiment 3 and Experiment 4, the highest accuracies were 46.9% and 74.07%.
- The better accuracies of Experiment 1 and 2 is because, in training step the recording environment of the testing files are already seen , which is not the case in Experiment 3 and 4.
- **Thus, session variability is playing an important role in determining the accuracies.**

Few more experiments were performed with GMM-UBM (with mean-adaptation) and results for the same are shown in Table 3.

Expt. no.	Experiment	Source Distribution	MFCC	CMV	Chroma
1	Single UBM for all classes	25% U + 25% Tr + 50% Te	53.96%	44.53%	31.46%
2	Group Wise UBM *[see Table 1]*		50.36%	47.66%	31.51%

Table 3: Results for GMM-UBM ***[U: UBM, Tr: Training, Te: Testing]***

Following can be observed from Table 3:

- Experiments for GMM-UBM have performed poorly as compared to experiments with GMM (with 70% sources for training and 30% for testing).
- This is mainly because, despite 25% sources being used for training UBM and then 25% being used for adapting UBM for each class, the process is not as effective as training a simple GMM using 50% sources.

CONCLUSIONS

- **The analysis of the effect of session variability on classification process is one of the important aspect of the project.**
- Both DNN and GMM have shown similar kind of behavior in all the experiments. In Experiment 1 and Experiment 2 both of them performed well in comparison to Experiment 3 and Experiment 4.
- No classification method or feature extraction method can give good performance until it is prone to session variability. Also, such a method can't be practically useful for classification of sounds.
- In the experiments conducted, DNN with MFEC features were the best performer, while GMM was next, and GMM-UBM was last. But, the results obtained in all these experiments were of the same order.

REFERENCES

[1] Douglas A. Reynolds, Thomas F. Quatieri et al., *Speaker Verification Using Adapted Gaussian Mixture Models*, 2000

[2] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, ser. Signal Processing Series. Englewood Cliffs: Prentice-Hall, 1993 .

[3] Olli Viikki and Kari Laurila, Cepstral domain segmental feature vector normalization for noise robust speech recognition, Speech Communication, vol. 25, no. 1, pp. 133–147, 1998.

[4] M. A. Bartsch and G. H. Wakefield, To catch a chorus: Using chromabased representations for audio thumbnailing, in Proc. IEEE Workshop Applicat. Signal Process. Audio Acoust. (WASPAA), New Paltz, NY, 2001, pp. 15–18

Drone Video Analysis and Summarization

An exploration of Bird's-eye view

Ashish Kumar Bedi & Samya Ranjan Patro

Dr. Aditya Nigam & Dr. Arnav Bhavsar

IIT Mandi | School of Computing and Electrical Engineering

Contact Information:

Phone: 9816923885 | 9816923604

Email: akbedi2000@gmail.com | samya.ranjan.patro@gmail.com

Objectives and Challenges

In this project, we achieved the following two objectives, in drone video data:

- Multiple Object Annotation and Identification.
- Object Tracking.

We faced the following challenges while achieving the objectives:

- Object Re-identification.
- Zooming-in and out of the bounding box with moving object.

Data Collection

The data has been collected using DJI Phantom 3 Advanced Drone. The high resolution videos are captured at 40 fps. We collected this data by flying the drone inside and in surroundings of our Kamand campus. Figure 1 shows the type of data collected.

Data collection has been done such that the collected data included a variety of different objects, such as: buildings, pedestrians, playing ground, basketball and football.

Figure 1: Images of IIT Mandi, collected using DJI Phantom 3 Drone.

Object Annotation and Identification

State-of-the-art techniques have been implemented in YOLO v2 and SSD. Both of these are single shot detection techniques. We combined the output layers of the CNNs in YOLO v2 and SSD, and the corresponding results are better than those obtained from each of them individually. The area corresponding to the Jaccard Index of the outputs obtained from YOLO and SSD is being considered as the final output bounding box. Figure 2 shows the comparison between the architectures of the neural networks implemented in YOLO v2 and SSD.

Calculating accuracy: Let Jaccard Index of the output bounding box with the bounding box containing the actual object be J.
$J > 0.85$ implies *Correct Annotation* and $J < 0.85$ implies *Incorrect Annotation*.

Description	Number of Images
Images Tested	2000
Correctly Annotated as B2	852
Incorrectly Annotated as B2	40
Correctly Annotated as D2	912
Incorrectly Annotated as D2	24
Total Correctly Labelled	**1692**
Total Incorrectly Labelled	**136**
No Label	**172**
Accuracy	**88.2%**

Table 1: Results of the combination of YOLO v2 and SSD on our data-set.

SSD model adds several feature layers to the end of a base network, which predicts the offsets to default boxes of different scales and aspect ratios and their associated confidences. Below, we can see a comparison of the architectures of YOLO v2 and SSD:

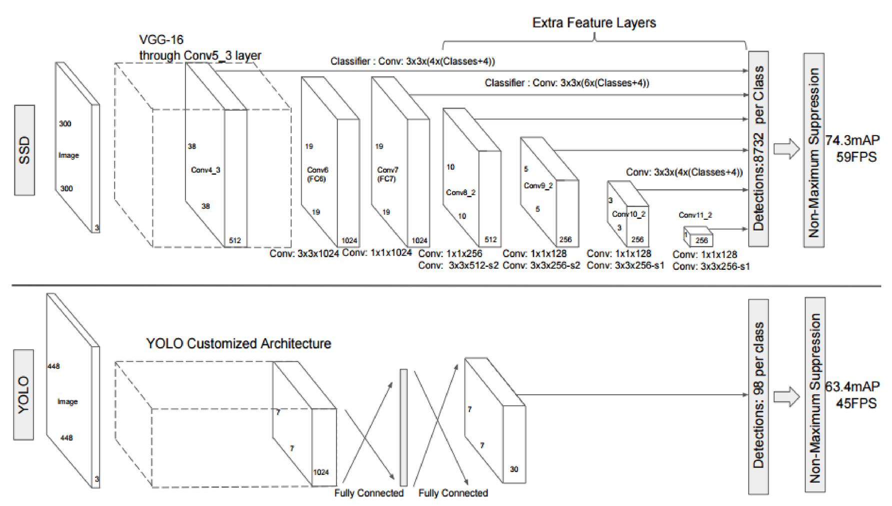


Figure 2: Image showing the comparison between architectures of YOLO v2 and ssd.

Figure 3: Image showing results of YOLO v2, SSD and combination of YOLO v2 and SSD on B2 Prashar Hostel, IIT Mandi. The red box corresponds to YOLO v2, blue to SSD and yellow to their combination.

Object Tracking

For object tracking in drone video, we have used state-of-the-art techniques implemented in STRUCK and MEEM

Video No.	MEEM	Modified MEEM
1	0.65	0.68
2	0.71	0.75
3	0.53	0.58
4	0.55	0.63

Table 2: Results of MEEM and Modified MEEM tested on our Drone Video data.

After modifying MEEM and STRUCK, we are now able to track objects in Bird's-eye view data. MEEM was performing slightly better, but it fails in certain scenarios, such as, in a video where background and foreground look similar in color. So, we incorporated the HOG, MBH and HAAR features of STRUCK into MEEM, to handle such cases. Modified MEEM gives better results as compared to MEEM, as can be seen in Table 2. These results show the average IoU of the output with the ground truth.

Figure 4: Results of Meem and Modified Meem tested on Drone Video data.

Both STRUCK and MEEM have a serious limitation,i.e., size of the bounding box is fixed. As the object moves towards the camera, size of the object increases as compared to the previous frames. Both the trackers fail to track the object in this case. We tackled this problem by computing the optical flow values, and training 3D Convolutional Neural Network on these values.

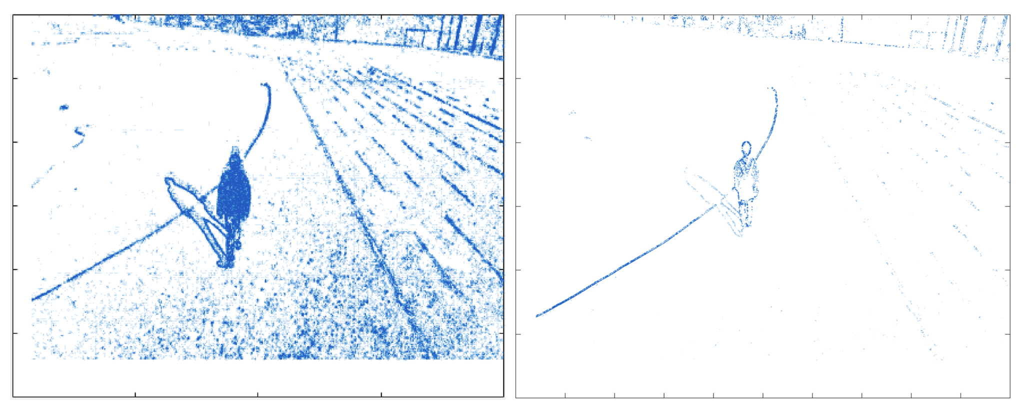


Figure 5: Left image depicts the optical flow between consecutive frame. Right image depicts the refined optical flow using first order derivatives.

Figure 6: MEEM results after training on our 3D-CNN model.

Our model of 3D-CNN consists of two convolutional layers with 32 and 64 filters, one fully connected layer with 1024 neurons and one output layer predicting a single float value.

Conclusions

- We were able to improve the accuracy of the state-of-the-art neural networks, on the bird-view data. In order to achieve greater Accuracy for Object Annotation, we combined YOLO v2 and SSD and this worked with an accuracy of 88.2% on our data-set.
- We were able to solve the problem of Object Tracking, by studying and modifying the state-of-the-art techniques implemented in STRUCK and MEEM.
- We were able to solve the task of dynamic bounding box sizes with change in Object's size in a video, by training 3D-CNN on optical flow values, computed between two consecutive frames.

Ongoing Research

- We tackled the problem of Object Identification and Annotation using YOLO. But, YOLO provides only spatial understanding of the object. In order to get both spatial and temporal understanding, we are modifying and training ROLO on our drone video data-set. ROLO is an LSTM layer applied over YOLO. We are modifying ROLO to also tackle the problem of Object Re-identification in a drone video.
- Improving accuracy of our 3D-CNN model by evaluating the values of hyper-parameters which work best on the 3D data.

References

[1] Joseph Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013–2016.

[2] Sentdex. 3d cnn in tensor flow. Explanation available at https://www.kaggle.com/sentdex/first-pass-through-data-w-3d-convnet/.

[3] Jianming Zhang, Shugao Ma, and Stan Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2014.

Effective Query and Visualization of Big Data IN PHARMA

Kulendra Kumar Kaushal, Guide: Dr. Varun Dutt

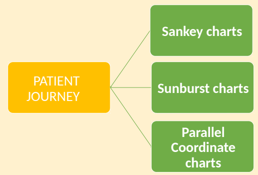
AIM AND OBJECTIVES

- We aim to understand patient journeys via certain powerful visualization charts, which help mine patterns in Big-Data concerning patients at the individual and aggregate levels.
- Patient journeys indicate the journey of patients from sickness to recovery.

BACKGROUND

- Process-mapping technique maps the entire patient's problem with different sequential steps involved [1].
- Due to the continuous real-time and multivariate nature of patient journey data, one-time 2-D X-Y visualizations are not sufficient. Rather, we need to have an automated software solution which can help us in visualizing the multivariate patient journey data in real time.
- Conventionally, researchers have used visualization methods like 2-D X-Y plots, scatter-grams, histograms, which are limited in the sense that they can be used to visualize only two-dimensions at a time [2].

VISUALIZATION CHARTS



Competitor Analysis - N by N Correlation Matrix

Sales Rep Performance - Whisker Box Plot

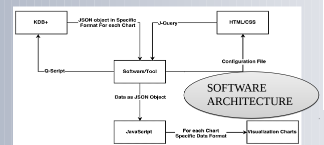


FIGURE: 1. ARCHITECTURE DIAGRAM OF THE SOFTWARE TOOL

```
{
    "GlaxoSmithKline":[13.98],
    "Pfizer_Inc":[16.39],
    "Purdue_Pharma.":[5],
    "SANOFI_AVENTIS":[0]
}
```

Figure: 2. Data Format for N by N Correlation Schematic Scatterplot Charts

```
{
    "Expenditure":[
        {"Vijay":[980]}
    ],
    "NumberOfVistis":[
        {"Vijay":[5]}
    ]
}
```

Figure: 3. Data Format for Whisker-Boxplot

```
{
   "Patient_ID":"1",
   "Diagnosis_Code":"V5812",
   "Diagnosis_Date":"2014.08.01",
   "Procedure_Code":"J4568",
   "Procedure_Date":"2014.08.22",
   "Drug_NDC":"00054455025",
}
```

Figure: 4 Data format for parallel coordinate axes

```
{
   "nodes":[
        {"name":"node0"},
        {"name":"node1"}
   ],
   "links":[
        {"source":"node0","target":"node1","value":"2"}
   ]
}
```

FIGURE: 5. DATA FORMAT FOR SANKEY DIAGRAM

```
l1: string(select distinct name:ndc_code from x);
l2: string(select distinct name:diagnosis_code from x);
nodes1: l1,l2;
```

FIGURE: 6(A). Q SNIPPET USED TO DERIVE DRUGS AND DIAGNOSIS AS NODES OBJECT

```
links1: select source: ndc_code, target: diagnosis_code,hvid from x;
links2: 0!select lvalue:count hvid by source,target from links1;
```

Figure: 6(B). Q Snippet used to derive links object consisting of drug as source node, diagnosis as target node and count of patients between them as weight.

```
graph:{[x; y]
`nodes`links!(x ; y)
}[nodes1;links1];
```

FIGURE: 6(C). Q SNIPPET USED TO DERIVE GRAPH OBJECT BY COMBINING NODES AND LINKS OBJECT OF FIGURE 5(A) AND FIGURE 5(B)

```
{
  "name": "Patient Journey",
  "children": [
    {
      "name": "diagonosis1",
      "children":[
        {
          "name": "Medication1",
          "children":[
            {"name": "Procedure1", "size": 3938}
          ]
        }
      ]
    }
  ]
}
```

Figure: 7. Data Format for Sunburst Visualization

DATA FORMAT AND Q QUERY

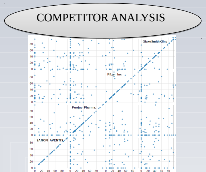


Figure: 8. Y axis of M[0][i] is expenditure by GlaxoSmithKline and , Y axis of M[1][i] is expenditure by Pfizer Inc., Y axis of M[2][i] is expenditure by Purdue Pharma and Y axis of M[3][i] is expenditure by Sanofi S.A. X axis of M[i][0] is expenditure by Sanofi S.A., X axis of M[i][1] is expenditure by Purdue Pharma, X axis of M[i][2] is expenditure by Pfizer Inc. and X axis of M[i][3] is expenditure by GlaxoSmithKline. I varies from 0 to 3.

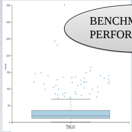


Figure: 9 (A). Whisker-Boxplot showing the performance of sales representatives

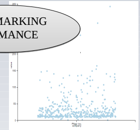


Figure: 9 (B). When double click on boxplot of Figure 9 (A) it changes in to a scatter plot as shown

Injection infliximab, 10 mg : 222 (25.46%)
Injection, abatacept, 10 mg : 156 (17.89%)
Injection, tocilizumab, 1 mg : 129 (14.79%)
Injection, rituximab, 100 mg : 125 (14.33%)
Injection, golimumab, 1 mg : 102 (11.70%)
Injection, certolizumab pegol, 1 mg : 80 (9.17%)
Injection, adalimumab, 20 mg : 29 (3.33%)

Figure: 6. Finding most popular procedure Used for Rheumatoid Arthritis

CORRELATION MATRIX

Correlation Matrix	days_supply	diagnosis_code	hvid	proprietary_name	short_description
days_supply	1			0.110 (0.00)	-0.066 (0.00)
diagnosis_code		1			
hvid	-0.042 (0.00)		1	-0.058 (0.00)	0.016 (0.12)
proprietary_name				1	-0.076 (0.00)

Figure: 5. Correlation matrix show correlation among dimensions

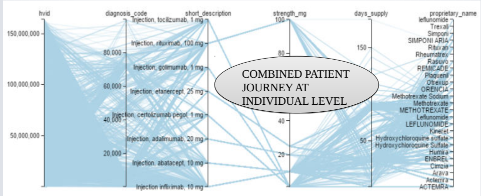


Figure: 2. An example of Patient Journey of individual patients with parallel coordinates chart.

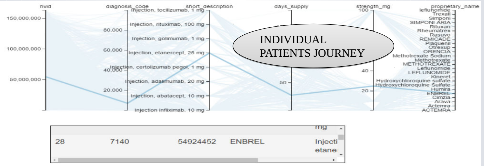


Figure: 3. Individual Journey of a patient with id 54924452

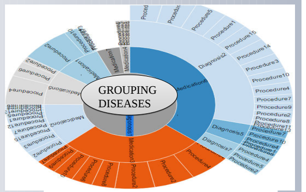


FIGURE: 9. PATIENT JOURNEY VISUALIZED WITH SUNBURST CHART. DIAGNOSIS I REPRESENTS THE I[TH] DIAGNOSIS CODE. SIMILARLY PROCEDURE I REFER TO I[TH] PROCEDURE CODE. WHILE MEDICATION I REFERS TO THE I[TH] MEDICATION CORRESPONDING TO CERTAIN DIAGNOSES AND PROCEDURES.

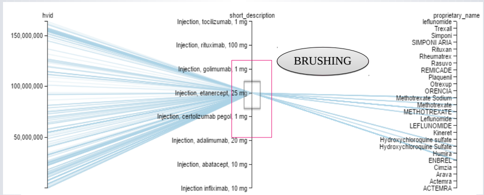


FIGURE: 4. BRUSHING AND VISUALIZATION OF SELECTIVE DIMENSION IN PARALLEL COORDINATE AXES

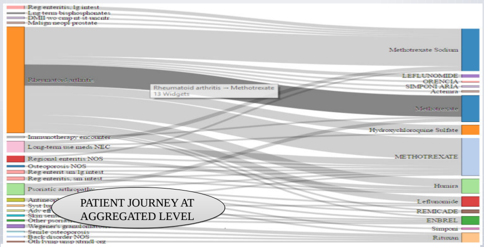


FIGURE: 7. SHOWING THE JOURNEY AND FLOW OF PATIENTS FROM RHEUMATOID ARTHRITIS TO DRUG METHOTREXATE.

CONCLUSIONS

- Some of these visualizations allow us to visualize multivariate patient journey data both at the individual patient level as well as at the aggregate level.
- To satisfy different stakeholder interests, we highlighted the potential of using three different visualizations.
- The Parallel Coordinate charts show patient journeys on individual patient level and thus could help physicians in understanding journeys of patients across several demographic variables
- Sankey charts allow pharmaceutical companies to visualize mapping of medications across a group of patients undergone specific diagnoses and specific procedures

ACKNOWLEDGEMENT

The project was supported from a grant (awards: #IITM/CONS/RxDSI/VD/07) from Rx Data Science Inc, USA, to Varun Dutt

REFERENCES

[1] Trebble, T. M., Hansi, N., Hydes, T., Smith, M. A., & Baker, M. (2010). Process mapping the patient journey through health care: an introduction. BMJ, 341(7769), 394-397

[2] Yuan, X., Guo, P., Xiao, H., Zhou, H., & Qu, H. (2009). Scattering points in parallel coordinates. IEEE Transactions on Visualization and Computer Graphics, 15(6), 1001-1008.

Submitted for Review in Visual Analytics and Health Care Conference which is held in association with IEEE
Kulendra Kumar Kaushal, Krish Vishwanathan, Balajee Chellapa, Sayee Natarajan, Varun Dutt

Opening Lead Recommendation in Contract Bridge

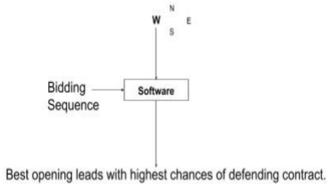
Gopal Krishan Aggarwal & Naman Gupta

Prof. Deepak Khemani

School of Computing and Electrical Engineering (SCEE), IIT Mandi, Himachal Pradesh, India

Goal

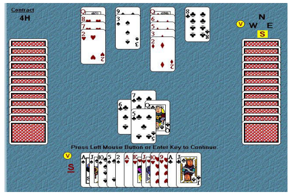
Develop a program to determine the opening lead of the popular cards game of 'Contract Bridge' to maximise the chances of defenders winning the game.



Motivation

- No world class computer bridge unlike for Chess, Go.
- Games are perfect platform for developing and testing AI algos.
- Bridge is much more complex game than chess.
- To enhance opening lead of existing bridge games.
- Can be also used as a learning tool for bridge players.

The Game



- Discrete
- Partially observable
- Stochastic
- Adversarial game

- Each player has 13 cards
- Ordered set Ace ranked 1, followed by K, Q, J till 2
- Bidding - Contract
- 13 rounds - trick
- Dummy
- Win maximum number of rounds

Objectives

- Generating hands according to given bidding sequence and West's hand.
- Determining if EW pair is able to defend the contract given a deal and contract.
- Replicating results given in Bird and Anthias book.
- Making a program to predict the opening lead for a random new hand and using the above the methods to verify the results.

Generating Data For verification

- Used Deal
- Input TCL rules for various bids
- Generate 20,000 West hands
- For each fixed West hand:
  - Generate 5000 hands
  - Total hands generated: 100,000,000
  - Monte Carlo Simulation
  - Percentage time each of the 13 cards is able to defend contract.
  - Compare the data with book.

ML Approach

- Other approach, rule based.
- Input: ordered set of 52 binary numbers.
- e.g. 0,0,0,1,0,1,0,0,1,1,0,0,0,0,0,0,0,0, 1,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,1,1, 0,0,0,0,0,0,1,0,1,0,0

Regression using Neural Network

- Training data
- e.g. 0,0,0,0.0496,0,0.0476,0,0,0.0498, 0.0498,0,0,0,0,0,0,0,0,0.058,0.058,0,0,0.05 42,0,0,0,0,0,0,0,0,0.035,0,0,0,0,0,0.1124, 0.064,0.064,0,0,0,0,0,0,0.0568,0,0.0506,0,0
- 20,000 training samples
- Neural networks ranging from 4 to to 10 hidden layers.
- Very low accuracy

Naive Bayes Clustering

- Training data
- e.g. for above input '39' i.e. A of Hearts
- 20,000 training samples
- Better accuracy

Classification using Neural Network

- Same training data as for Naive Bayes.

Simple Logistic

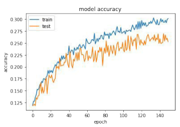
- Other classification algorithms such as Random Forest were also tried.

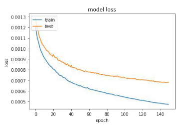
Results

- Regression using neural network gave accuracy of just 37% but with sample output as follows:

[0.00669532 0.00805629 0.00790471 **0.04841855** 0.00656826 **0.04627116**
0.01230518 0.01621353 **0.04922389** **0.08186292** 0.01272484 0.00381293
0.00306514 0.00420591 0.00823677 0.00268892 0.00450318 0.00389304
0.06063421 **0.05804513** 0.00774613 0.00816881 **0.05995162** 0.00687261
0.00597707 0.00030584 0.00078916 0.00098978 0.00386256 0.00598644
0.00157384 0.00274211 **0.04550658** 0.00752675 0.00439383 0.0062809
0.00139312 0.00373178 **0.11187352** **0.06205911** 0.07075668 0.00510138
0.00345985 0.0042186 0.00249029 0.0007015 0.00576708 **0.05446858**
0.00567879 **0.04878691** 0.00030414 0.00520478]

- Naive Bayes Clustering produced accuracy of 48.9%
- Classification through neural networks produces accuracy of 30.13%





References

- [1] Smith, S. J. J., Nau, D., & Throop, T. (1998). Computer bridge: A big win for AI planning. AI magazine.
- [2] "How Long Before Computers Beat The Best At Bridge?"(2016). Bridge Winners. N.p., Web.
- [3] BridgePro - The Ultimate Contract Rubber Bridge Game. Retrieved from www.bridgepro.com.
- [4] GNUBridge - Available on https://github.com/pslusarz/gnubridge
- [5] Deal program, documented at http://bridge.thomasoandrews.com/deal/
- [6] Bo Haguland's Double Dummy Solver: http://privat.bahnhof.se/wb758135/bridge/index.html

Acknowledgements

- Guidance and mentorship by Prof. Deepak Khemani and his colleagues Chief Scientist Mr. Arun Bahulkar at TCS Innovations Lab TRDDC, TCS Pune and Mr. Rajendra Gokhale is deeply acknowledged.
- Tushar Gupta and Swapnil Sharma were extremely helpful in applying various neural network algorithms.
- Dr Sriram Kailsam, Dr Dhiraj Patil, Dr Prasanth Jose for providing servers for computations.

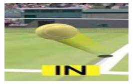
Line Fault Detection in Tennis and Video Summarization

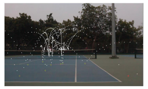
Rohit Bishnoi(B13134)

Mentor: Dr. Renu M. Rameshan and Dr. Aditya Nigam

Problem Statement

Detecting line fault and video summarization in tennis without human input. This project uses 3D tracking to solve this problem. This technique will only work on data recorded with two calibrated cameras. Speed of ball, exact spot of the ball on ground are some challenges faced during project.





Data Capturing

A setup of two camera is used in this project which are placed at certain distance and then calibrated. Each pair of camera covers both lines in corner. So for whole ground four such pair will be needed.

Rotation vector : R =

0.9998	-0.0011	0.0174
0.0037	0.9886	-0.1507
-0.0170	0.1508	0.9884

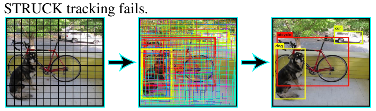
Translation vector: T = [148.39276 -1.55872 -6.53204]mm

Technique used in this project requires images from both the cameras to be in same plane which is done using rectification

2D Tracking and Detection

The ball in image(2D) is tracked using STRUCK. STRUCK is a tracking-by-detection technique. STRUCK is applied on frames by left camera. Then using result by STRUCK, NCC is used to calculated position of ball(2D) in right image and depth of the ball.

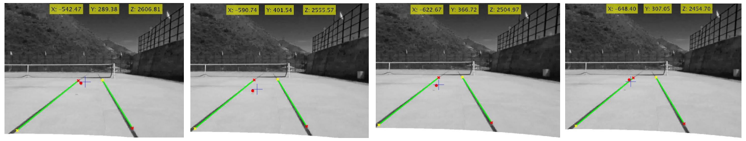
When the ball goes out of frame then YOLO is used to detect the ball when it comes back. YOLO is also used when STRUCK tracking fails.



Results

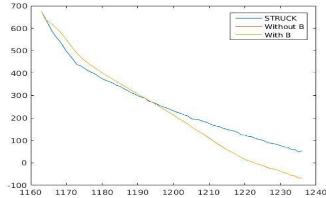
Results without using Kalman Filter are not acceptable because the amount of error in the measured value leads to a large amount of error projected 3D location of the ball.

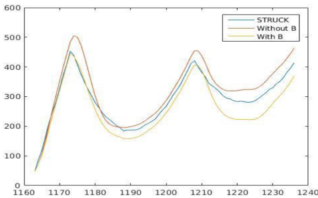
Kalman Filter results are not that good. One reason for this is modeling. Kalman Filter model require some more changes for specific cases. Other reason is the unsynchronized data. Data captured using two cameras was not properly synchronized which leads to error in results.



The point where ball touches the ground, the velocity of ball changes due to non-elastic collision. This special case is added in Kalman Filter using control input model. This control input model only effects the Kalman Filter for the point where ball touches the ground. Controlled input consider the collision to be inelastic and uses coefficient of restitution for calculating velocity after the impact.

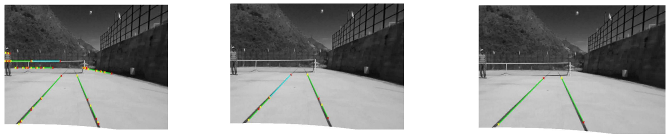
The improvement in result after using controlled input model is significant. But still Kalman Filter requires more effective model to track with precision.





STRUCK and Kalman Filter performance was better when they were used parallel rather then using sequentially. Reason for this STRUCK can use the predicted values of Kalman Filter and use it to track ball in next frame.

After tracking the ball lines in tennis ground is detected using Hough and Canny. Initial result given by Hough and Canny included all lines in ground. Then horizontal lines and very small vertical lines are ignored. After this overlapping lines are merged into line to detect the required line tennis field.



3D Tracking

Then using all theses results are provided to Kalman Filter to track the ball in real world co-ordinate(3D). Kalman Filter is a algorithm which based on series of measurements, estimates unknown variable. Due to non-linear transformation matrix in Kalman Filter Extended Kalman Filter is used.

$U = [X \quad Y \quad Z \quad \dot{X} \; \dot{Y} \; \dot{Z} \; \ddot{X} \; \ddot{Y} \; \ddot{Z}]^T$ *State Variable*

$z = [x1 \; y1 \; x2 \; y2 \; Z]^T$ *Measurments*

$P_{t|t-1} = E[(U_t - U_{t|t-1}) \cdot (U_t - U_{t|t-1})^T]$ *Convariance Matrix*

H is transformation matrix

F is the model

$Q = E[w_t \, w_t^T]$

$R = E[v_t \, v_t^T]$

Prediction Step

$$U_{t|t-1} = F \cdot U_{t-1|t-1} + w_t$$

$$P_{t|t-1} = F \cdot P_{t-1|t-1} \cdot F^T + Q$$

Update Step

$$U_{t|t} = U_{t|t-1} + K_t(z - z_e)$$

$$z_e = H \cdot U_{t|t-1} + v_t$$

$$K_t = P_{t|t-1} \cdot H^T \cdot (H \cdot P_{t|t-1} \cdot H^T + R)^{-1}$$

$$P_{t|t} = (I - K_t \cdot H) \cdot P_{t|t-1}$$

Conclusion

Modeling of Kalman Filter is a step by step process.Once all these steps are covered then it is expected that even if STRUCK fails at some point Kalman Filter will not so tracker will eventually work good. We tried to cover most of the cases and correct most of the reasons for error but still the Kalman Filter is not working as expected. With Kalman Filter not working properly it is hard to predict the exact spot on which the ball touches the ground and without that the B(control-input model) would not work. So these points are provided manually.

Modeling Human Decisions with Deep and Conventional Reinforcement Learning Algorithms in Games

Himanshu Singal (Btech, CSE, Final Year)
Advisor: Dr. Varun Dutt

Objective

We aim to try different deep and conventional reinforcement learning algorithms in static and dynamic games and analyze their performance, learning speed and deviation from humans.

Introduction

- People learn by experience and imitation [1].
- There are a class of reinforcement learning algorithms like Q learning and Instance Based Learning or imitation learning algorithms which allows us to model human decisions. [2][3][4].
- They are known for their human like approaches [3].
- Several Deep Q approaches also possible [5].
- It is not known if they perform in a human-like fashion.
- We aim to compare and analyze different AI algorithms on a given test environment.
- Analyze their learning speed, flexibility and deviation from humans.

Static Platform Jumper Game

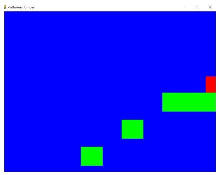
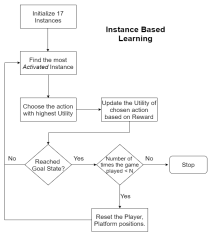



Figure 1.1a(left): Shows the initial position of the player
Figure 1.1b(right): Shows the goal or the final state which the player need to achieve.

- Figure 1 shows the game on which initial testing was done.
- Red block is the player, green is the platform.
- Four actions possible: **Left, Right, Jump Left, Jump Right**
- We applied *Instance Based Learning (IBL)* on this game [3].
- Different models made to play the game 10 times and the results averaged over 10 trials.



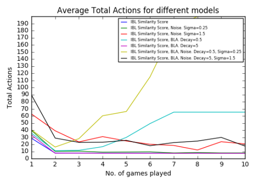


Figure 2a(left): Shows the flow chart of the algorithm
Figure 2b(right): Shows the results with different parameter settings

Moving Platform Jumper Game

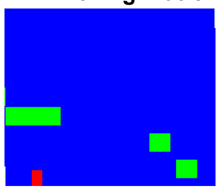
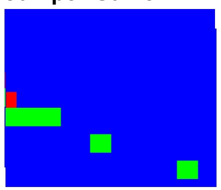



Figure 3a(left): Shows the initial position of the player
Figure 3b(right): Shows the goal or the final state which the player need to achieve

- Figure 3 shows a more challenging game.
- Five actions possible: **Left, Right, Jump Left, Jump Right, Stay**
- The platform in the middle is a moving platform
- Two versions of game**: Slow moving platform, Fast moving platform**
- Collected human data, applied different algorithms on both versions.
- Action Vector of model M in ith game, $AV_{M,I}$ = [N(left), N(right), N(jumpl), N(jumpr), N(stay)]
  - N(A) means count of action A from start to goal state.
- $Score = N(jumpl) + N(jumpr) + N(left) + N(right) + 0.3 * N(stay)$

Methods

Nearest Neighbor: It is like IBL used previously but considering only similarity score and initializing 33 states.
Complex DeepQ: Tried four architeling DeepQ (fig 5b) along with experience replay, separate target network.

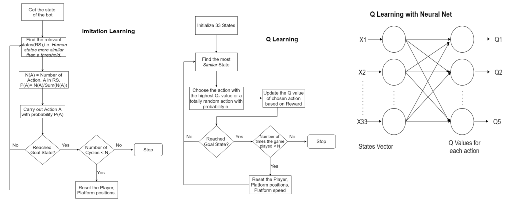


*Figure 4a(left): Flow chart of the **imitation learning,** Figure 4b(middle): Shows the flow chart of the **Q learning,** Figure 4c(right): Shows the network architecture for **Q learning with neural network***

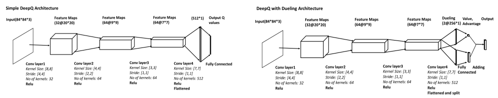
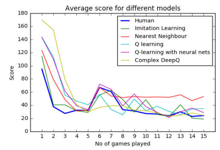


*Figure 5a(left): Architecture for **DeepQ,** Figure 5b(right): Architecture for **DeepQ with Dueling[5] [6]***

Results



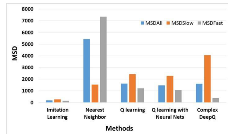


*Figure 6a(left): Shows the average score for different models, Figure 6b(right): Shows the Mean Squared Deviation (MSD), * Complex DeepQ has equivalent score.*

- The first five games in figure 6a are slow version, last 10 are fast version.
- Complex DeepQ required 45 slow and 55 fast games to reach human like scores.
  - For comparison, scores over 9 slow games and on average 5.5 fast games is clubbed together to get equivalent score over 5 slow and 10 fast games.
  - For a model M, MSDAll $= \sum_{i=1}^{15} \frac{(AV_{M,i}-AV_{H,i})^2}{15}$, MSDSlow $= \sum_{i=1}^{5} \frac{(AV_{M,i}-AV_{H,i})^2}{5}$

    MSDFast $= \sum_{i=6}^{15} \frac{(AV_{M,i}-AV_{H,i})^2}{10}$, *where $AV_{M,I}$ is the action vector of model M, at ith game*

Conclusion and Discussion

- On Static Platform Jumper Game, *Best Results:* IBL with similarity score
  - Remembering every instance led to bad results
  - Noise leads to random instances being chosen
- For Moving Platform Jumper Game,
  - Local maxima at 6th game as the speed of platform increased.
  - The best performance and lowest MSD: Imitation Learning
    - It simulates what humans did. So low MSD expected.
  - If we do not consider imitation learning, then MSD calculations shows
    - Overall MSD, Quite similar for all Q learning approaches, Worst for *Nearest Neighbor* as it is unable to learn the fast version of the game.
    - Slow cycles, Best: *Nearest neighbor,* Worst: *Complex DeepQ.*
      - Nearest Neighbor chooses no random actions, reward directly added to utility.
      - Complex DeepQ have to learn about the image, spatial relations between platform and players
    - Fast cycles, Best: *Complex DeepQ*, Worst: *Nearest Neighbor.*
      - *Nearest Neighbor* does not carry out any random actions
      - Complex DeepQ has learnt about the image. Better features are extracted. Trained over more games.
  - Conventional algorithms learn at a pace similar to humans, DeepQ algorithms require more training and computation.
  - In conventional algos, the states pre-initialized by the user, DeepQ automatically extracts the features.

References

[1] Aosaki, T., Tsubokawa, H., Ishida, A., Watanabe, K., Graybiel, A. M., and Kimura, M. (1994). Responses of tonically active neurons in the primate's striatum undergo systematic changes during behavioral sensorimotor conditioning. *J. Neurosci.* 14, 3969–3984.

[2] Richard Sutton and Andrew Barto. Reinforcement Learning: An Introduction. MIT Press, 1998.

[3] Dutt V. and Gonzalez C., The role of inertia in modeling decisions from experience with instance-based learning: Front. Psychol., 06 June 2012

[4] Schaal S, Is imitation learning the route to humanoid robots? Trends in Cognitive Sciences 3(6): 233–242, 1999.

[5] Mnih V. et al., Playing Atari With Deep Reinforcement Learning, NIPS' 13 Workshop on Deep Learning, 2013.

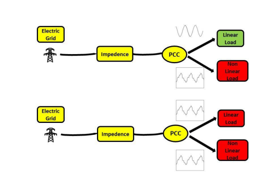
[6] A. Juliani, DeepQ Networks and Beyond, Medium, 2016.

Power Quality Analysis in Power Distribution System

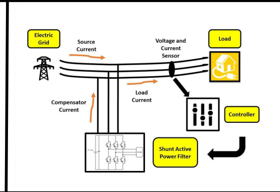
Avnish Kumar, Nikhil Kaushik, Rajat Valecha
iitmavnish@gmail.com,nikhilkshk1@gmail.com,valecharajat@gmail.com
School of Computing and Electrical Engineering
Indian Institute of Technology Mandi, India

Project Background

Problem Defined

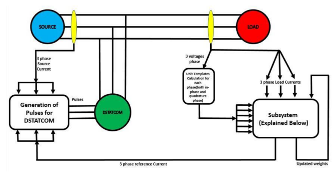


Solution to the problem



Our Novel Contribution

Design of solution



Modified Weight Update Method

Momentum update: Taking use of gradient of the gradient of the change to be done in weights

WLpa(k) = WLpa(k − 1)

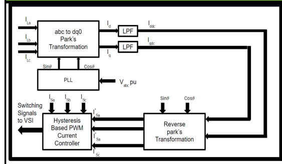
+

[μ(γ {iLa(k-1) − WLpa(k − 2) xspa(k-1)} xspa(k-1))]

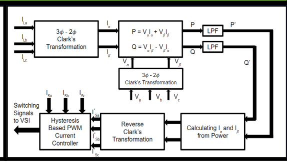
-

[(γ {iLa(k) − WLpa(k −1)xspa(k)} xspa(k)]

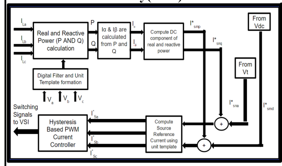
Discussion on different algorithms



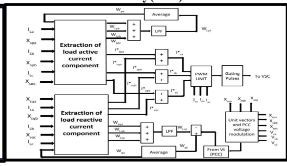
Synchronous Reference Frame Theory(SRF)



Instantaneous Reactive Power Theory(IRP)

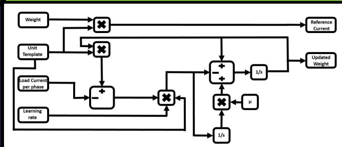


Power Balance Theory(PBT)

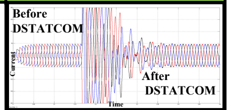


Neural Network Based Theory

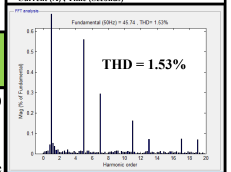
Design of subsystem



Results



Current (A) , Time (Seconds)



Real Time Digital Simulator Implementation

Real Time Digital Simulator (RTDS)

- validates and test operations for various power and control models.
- Integrated with hardware equipments such as relays, amplifiers, converters and electric drives.

Technical Specifications

- 20 Triple processor card (ADSP 21062)
- 1 RISC Processor card (PC70CXe)
- Work in Time Step of 50 to 60 us.

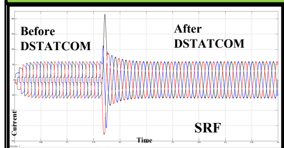
(*All Specifications for 1 Rack*)

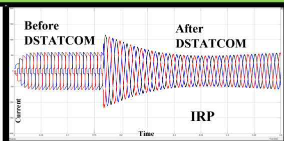
Specification of equipment

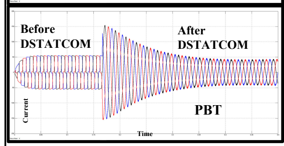
- Source : 415 p to p rms,5 Hz
- Load: Uncontrolled Rectifier with R-L load with R=12 ohm and L=200mH
- DCLink Capacitance: 7mF
- RC Filter:R=5ohm, C=80uF
- Interfacing Inductor: 2mH
- Sampling time : 0.000001s

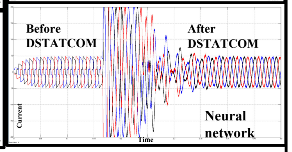
****Note: All Simulations done with above specifications**

Simulation Results



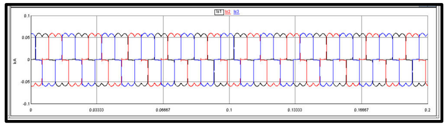




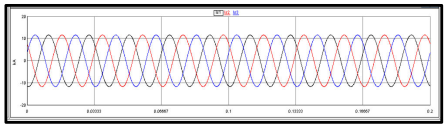


Current (A) , Time(Seconds)

RTDS Results

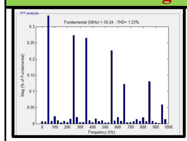


Uncompensated Source Current Waveform

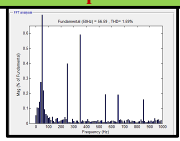


Compensated Source Current Waveform

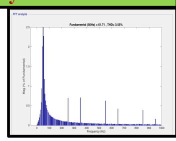
Algorithm Comparison Analysis



SRF

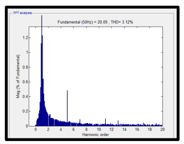


PBT



IRP

	SRF	IRP	PBT	Neural Network
Total Harmonic Distortion(THD)	1.23%	3.55%	1.59%	3.12%
Settling Time (seconds)	0.12	0.47	1.07	0.34
No. of Sensors Required	10	10	10	10
Computation Level (Time taken for 1 sec simulation)	105s	120s	53s	116s



Neural Network

Conclusions

- Control Algorithms based on SRFT,IRPT,PBT and Neural Network are compared.
- Neural Network based method modified by modifying the weight update arithmetic.
- Results with the modification proven to be better.
- Results also validated using RTDS.

Selected References:

[1] A Comparison of Control Algorithms for DSTATCOM by *Bhim Singh*, and *Jitendra Solanki*.

[2] Power Quality Improvement in Isolated Distributed Power Generating System Using DSTATCOM by *Sabha Raj Arya,Ram Niwas, Krishan Kant Bhalla, Bhim Singh,Ambrish Chandra* and *Kamal Al-Haddad*,

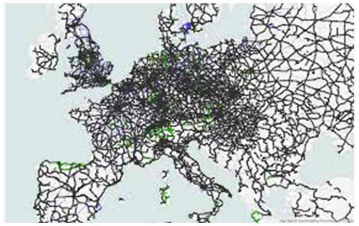
[3] DSP based implementation of a three-phase, four-wire DSTATCOM for voltage regulation and power quality improvement by *P. Jayaprakash, Bhim Singh, DP Kothari*.

Quantum Maxflow Mincut Problem

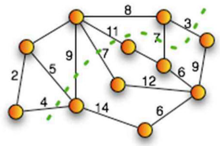
Under the guidance of Dr. Samar Agnihotri

About the Project:

- In a graph, the maximum flow is equal to the bottleneck capacity of the network.
- It takes quadratic runtime to calculate minimum capacity by the best known algorithm.
- Our aim is to improve this runtime using Quantum Computation.



a typical rail network



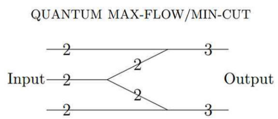
A classical network

Existing Algorithm:

Best known algorithm given by Stoer and Wagner gives a time complexity of:

$$O(|V||E| + V^2 \log |V|)$$

In case of large networks, where V and E are huge, time complexity almost becomes cubic.



A tensor network

The Proposed Algorithm:

To calculate mincut in a graph, the problem is divided into 2 parts:
Classical part: it enumerates all the cutsets in a graph using quadratic time .
Quantum part: It calculates the minimum among them by implementing a quantum algorithm built upon Grover's search algorithm in squared root run time.

Classical part

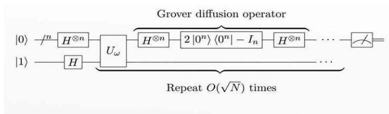
We first enumerate all cutsents using the classical algorithm given Tsukiyama and Shirakawa.
Its time complexity is:

$$O((|V|+|E|)(|V|-\log r)r)$$

Now apply quantum part to evaluate min of these using sqaured root complexity.

Quantum part

Built upon Grover's algorithm:
1. Choose a value p from the set of cuts enumerated by the classical part.
2. Repeat:
i. Apply Grover's algorithm and find q such thatq<p.
ii. If we find such a q, reset p as q. Else, output p as result.



Result

Speedup is achieved in the calcualtion of mincut from almost cubic time to quadratic (in the classical part) and squared root in the quantum part.

Farah Anjum

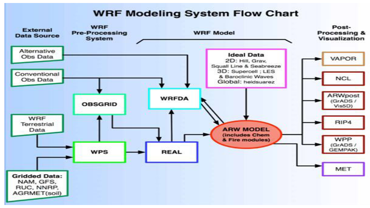
Setting up of WRF-GFS Model for Regional Climate Prediction

Abhimanyu Mittal

Under the guidance of Dr. Sarita Azad

OBJECTIVE

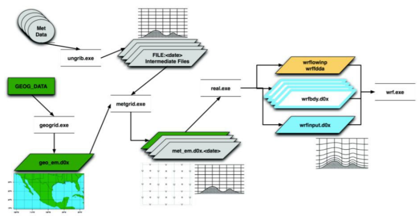
- To obtain the high resolution numerical prediction of precipitation over the Indian region using Weather Research and Forecasting (WRF) System
- Examining the 6-hour global forecast output may not give a satisfactory picture of the likely forecast uncertainty on the local scale. This poster describes a system that uses the (WRF) model to effectively downscale the GFS members over a small region.



WRF Modelling System Flow Chart

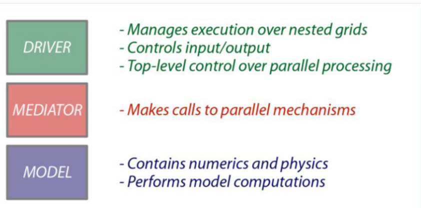
METHODOLOGY AND DATA

- Use of Advanced Research WRF (ARW) core of the Weather Research and Forecasting System (WRF)
- 6h Global Forecasting System (GFS) at 1.0 X 1.0º grids generated by NCEP's global forecast system.
- May 15, 2017 0000 hrs to May 23, 2017 0600 hrs



WRF Workflow

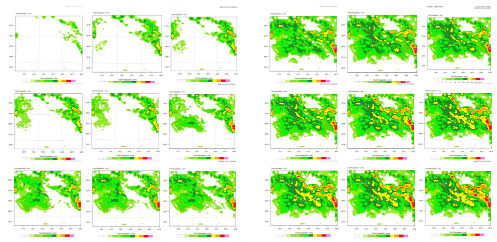
WRF Domain and Configuration



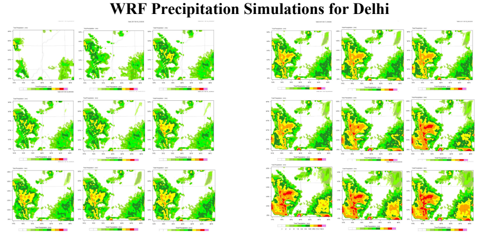
WRF Model Architecture

- **ARW Version:** 3.4.1
- **Long-Wave Radiation Scheme:** RRTM
- **Short-Wave Radiation Scheme**: Dudhia
- **Surface Clay Physics**: Monin-Obukhov similarity theory
- **Surface Physics**: 5 layer thermal diffusion model from MM5
- **PBL Scheme**: YSU
- **MP Physics**: WSM3

Forecast Product



WRF Precipitation Simulations for Delhi



WRF Precipitation Simulations for Bangalore

RESULTS

- The likely default configurations using the GFS domain gives more or less the same result for all the planar areas
- The ~10 km spatial resolution simulation could be made in approx 2-3 mins while the ~3 km spatial resolution took about 24-25 mins per location.
- Forecasting of precipitation and surface temperature for the local Himachal Pradesh region was tried but the main challenge attached with it is the drastically changing terrain height. To counter this problem, the WRF-IBM module is needed with the alternative parameterization methods to obtain real forecasting results.

FUTURE WORK

- Assimilation of satellite observations of the atmosphere to improve skill in precipitation forecasts
- Evaluation and Verification of model performance using forecasted boundary data from NWP models other than NOAA models
- Exploring further refinements to the proposed default set of parameterization configuration

REFERENCES

- Sikder, S., and F. Hossain (2016), Assessment of the weather research and forecasting model generalized parameterization schemes for advancement of precipitation forecasting in monsoon-driven river basins, J. Adv. Model. Earth Syst., 8, 1210–1228, doi:10.1002/2016MS000678.
- Rajeevan M, Bhate J, Kale J D and Lal B 2005 Development of a high resolution daily gridded rainfall data for the Indian region (Met Monograph Climatology, 22/2005, 26 pp, available from India Meteorological Department, Pune, India).

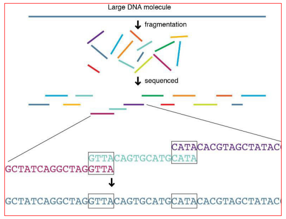
Streaming Based Genome Mapping on Apache Flink

Ankush Jindal, Sagar Ghai

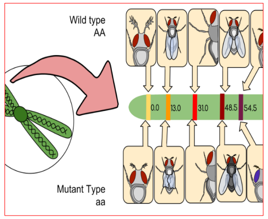
Under the guidance of Dr. Arti Kashyap

INTRODUCTION

- Genome Sequencing is process of determining the complete DNA sequence of an organism's genome.
- Genome Mapping is aligning the reads (small sequence of base pairs) onto a reference genome.



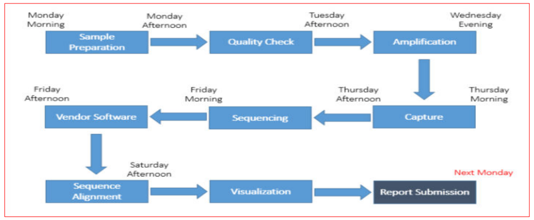
Whole Genome Sequencing



Gene Mapping in Drosophilla

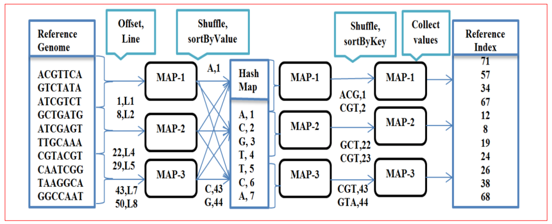
OBJECTIVE

- **Challenge**: Reduce the read-mapping time
- **Objective**: Develop the flow for streaming manner read mapping, while implementing AVLR-Mapper (sequencing and mapping) in Apache Flink (both for streaming data and batch processing) and compare results with current state of the art.

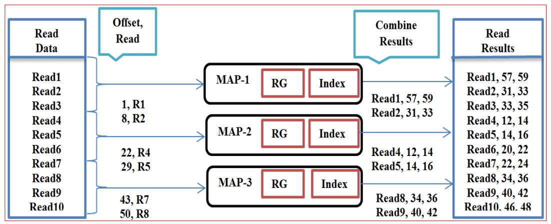


Agilent's Sequencing and Mapping Workflow

AVLR-Mapper Algorithm

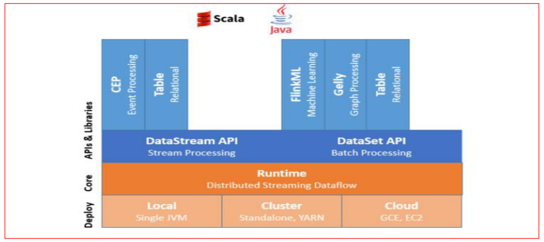


Iteration 1: Suffix Generation



Iteration 2: Mapping

Apache Flink

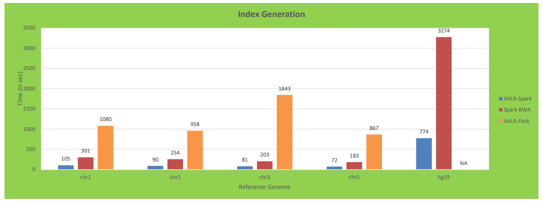


Apache Flink's Core Programming Model

DATASETS

Dataset	Size (in MB)	AVLR-Spark	Spark BWA	AVLR-Flink
chr1	254	105	301	1080
chr2	248	90	254	958
chr3	201	81	203	1843
chr5	184	72	183	867
hg19	3157	774	3274	-

RESULTS : AVLR-Flink vs Others



Index Generation time for AVLR-Spark, Spark BWA, AVLR-Flink

CONCLUSION

- Spark clearly outperforms Flink in doing the sequence alignment by AVLR-Mapper
- AVLR-Flink's results were accurate both for batch processing and streaming data
- The streaming based sequence alignment tool will take almost 'zero' additional time for mapping when the mapper is run in parallel to the sequencer using streaming capabilities. Even though Spark outperforms Flink's implementation of AVLR-Mapper in index generation, it will take smaller net time in the mapping of the reads.

REFERENCES

- Rathee, Sanjay, AVLR-Mapper: An accurate and variable length read-mapping tool on Apache Spark Platform, IIT Mandi, (submitted).
- Abuín, José M., et al. "SparkBWA: speeding up the alignment of high-throughput DNA sequencing data." PloS one 11.5 (2016): e0155461.

WIRELESS POWER TRANSFER 2.0

Shri Kisna Mahajan B13230
Munindar Kumar B13216
Avanish Yadav B13205

Faculty Mentors
Dr. Bhakti Joshi
Dr. Ramesh Oruganti

School of Computing and Electrical Engineering , Indian Institute of Technology Mandi, H.P., India.

OBJECTIVES

- To develop a design for the secondary coil to decrease sensitivity of the setup because of misalignment of primary and secondary coils.
- Develop a theoretical model of the multidimensional secondary coil setup.
- Experimentally validate the theoretical model.
- Design a rectifier circuit to provide a DC voltage as output from the set of secondary coils.

Theoretical modelling and Inferences for single secondary coil setup

$$\overrightarrow{dB} = \frac{\mu_o * I * \overrightarrow{dl} X \overrightarrow{d}}{4\pi * |\overrightarrow{d}|^3}$$

Fig. Biot-Savart's Law used to map the magnetic flux density at any point because of a current carrying element.

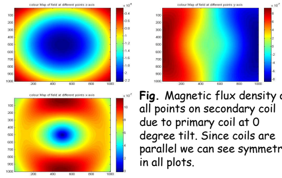


Fig. Magnetic flux density at all points on secondary coil due to primary coil at 0 degree tilt. Since coils are parallel we can see symmetry in all plots.

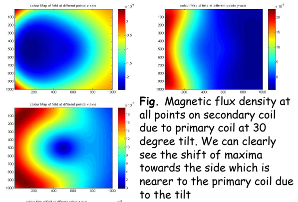


Fig. Magnetic flux density at all points on secondary coil due to primary coil at 30 degree tilt. We can clearly see the shift of maxima towards the side which is nearer to the primary coil due to the tilt

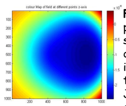


Fig. Z-axis magnetic flux density plot at 210 degree tilt. We can see that it is similar to the 30 degree tilt plot as the difference in tilt is 180 degrees which means the coil just did half a rotation. Just the direction of the maxima changed.

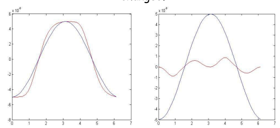


Fig.(left) Total flux passing through coil at different angles (red) compared to a cosine wave (blue) considering only z-axis component of flux density. **(right)** FLux varying with tilt compared to a cosine considering only y-axis component.

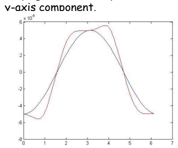
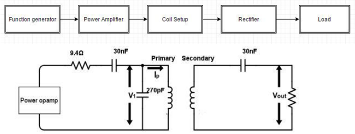


Fig. Flux varying with tilt when both z and y-axis components considered.

The flux doesn't follow the cosine function we thought it would considering uniform flux density. The flux has a peak when both coils aren't parallel which is an astonishing result. This happens because of the factoring in of the y-axis components as well.

Experimental Results

The **figure** below shows the flow diagram of the experimental setup. The function generator acts as a high frequency voltage source. Since it is not able to provide high current , we then used LT1210 current feedback amplifier because of its high output drive current which makes it perfect for our use. The coil setup is as shown in the image. The induced sinusoid in the secondary is then sent to a rectifier individually for both the secondary coils and we get as their sum a DC voltage.



The load resistance taken here is 1.2kohms. After using 30nF compensating capacitors and by hit and trial getting to the maximum output frequency i.e. the resonant frequency, we can calculate the mutual inductance between the coils which comes out to be around 35uH.
We did experiments a distance 10 cm and took observations for various tilt angles at the same input power.

V(primary) rms (Volts)	V(secondary) rms (Volt)	Tilt angle (degrees)	I (secondary) rms mA
5.11 V	2.44	0	2.033333333
5.12V	2.51	22.5	2.091666667
5.1V	2.72	45	2.266666667
5.12V	2.1	67.5	1.75
5.11V	0.2	90	0.166666667
5.12V	-1.86	112	-1.55
5.11V	-2.68	135	-2.233333333
5.11V	-2.43	158	-2.025
5.11 V	-2.3	180	-1.916666667

Observations made for the secondary side voltage without using rectifier at primary side voltage of 5.11 V; operating frequency Of 150kHz and distance between coils of 10 cm.

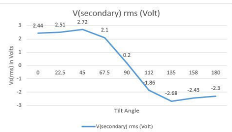


Fig. Plot between emf and tilt angle of the secondary coil w.r.t. The vertical axis

The plot of emf obtained is clearly following the same trend as obtained in the theoretical model and therefore the results are validates

Pair of secondary coils setup simulation results and Rectifier circuit

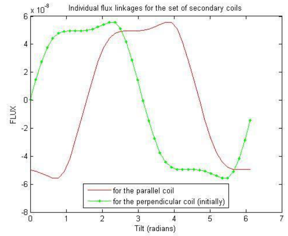


Fig. This shows the variation of flux for both of the secondary coils simultaneously as we vary the tilt angle. Since the coils are perpendicular to each other we can see a phase difference of 90 degrees in the plots.

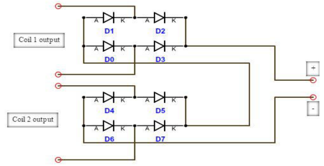


Fig. Dual rectifier circuit which individually rectifies outputs from both the secondary coils and then adds them up to give us a higher Dc voltage.

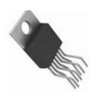


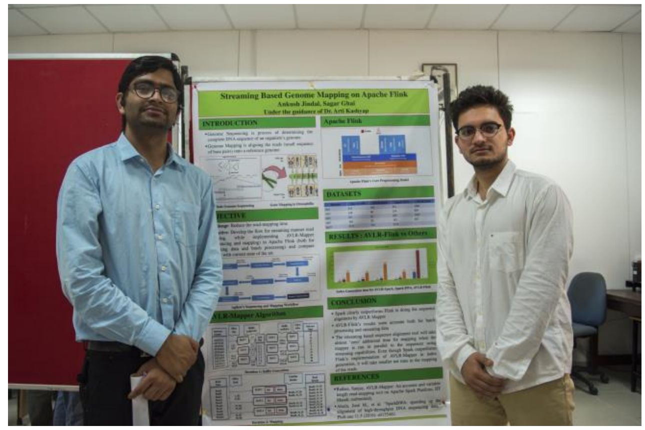
Fig.
LT1210 Current Feedback Amplifier
- 1.1 A Output Drive Current
- 35 MHz Bandwidth
- 900V/us Slew Rate

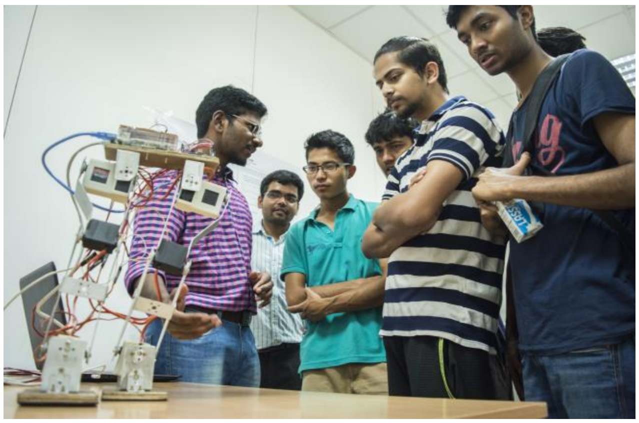
CONCLUSION

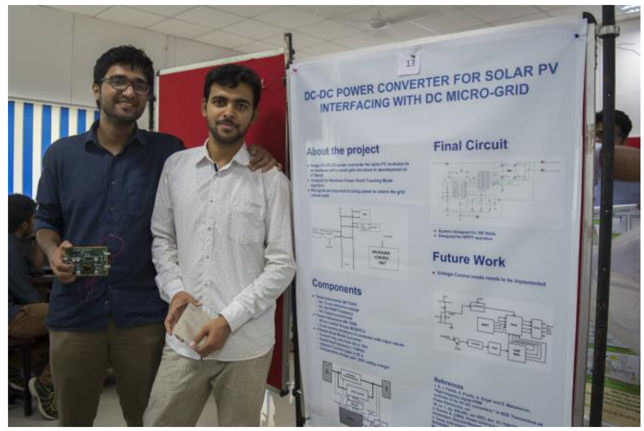
From the above results we can easily see the similarity in the behaviour of the secondary side voltage with the variation in the tilt of the single coiled setup. The results are almost same when compared to the simulation results in the previous subsection.
Hence the simulation results which stated that the generated emf in the secondary would not follow a cosine function when plotted against the tilt of the setup are successfully validated.

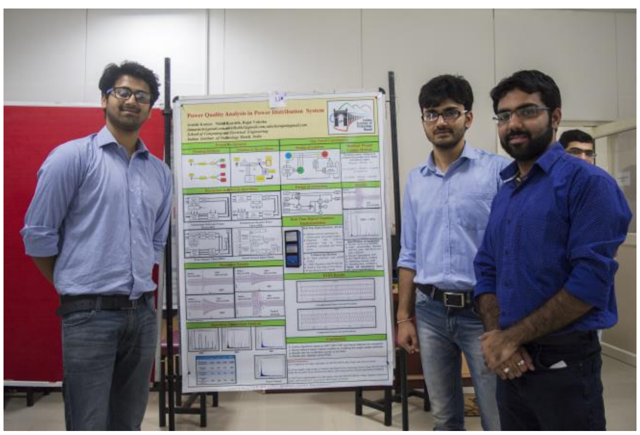
REFERENCES

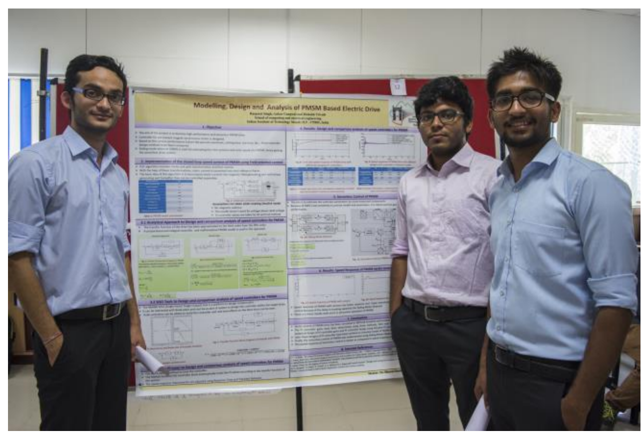
1. **André Kurs, Aristeidis Karalis, Robert Moffatt, J. D. Joannopoulos, Peter Fisher, Marin Soljac˘ic´ .** Wireless Power Transfer via Strongly Coupled Magnetic Resonances
2. **http://pdf1.alldatasheet.com/datasheet-pdf/view/70358/LINER/LT1210.html**
3. **Siqi Li.** Wireless Power Transfer for Electric Vehicle Applications

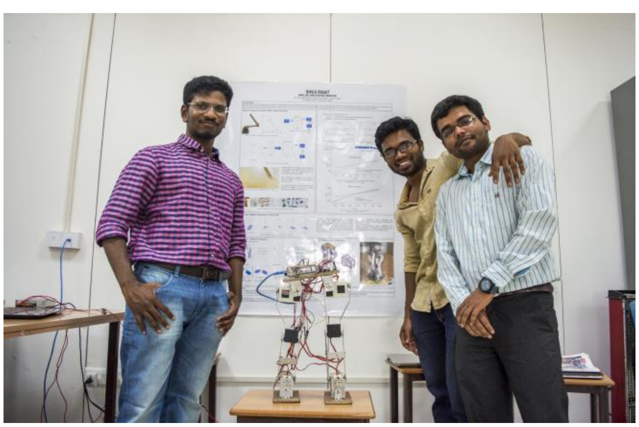

Streaming Based Genome Mapping on Apache Flink
INTRODUCTION
Apache Flink
DATASETS
CONCLUSION
REFERENCES

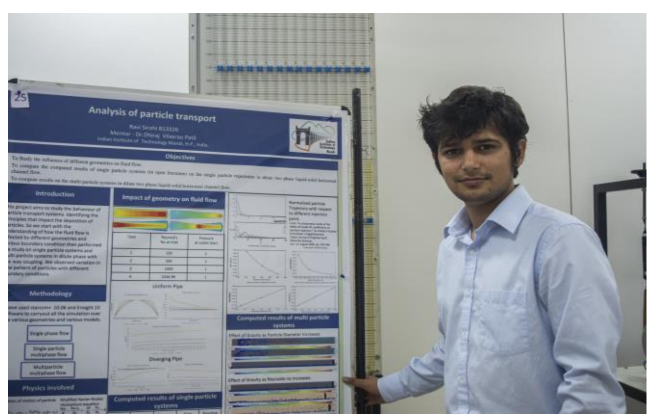


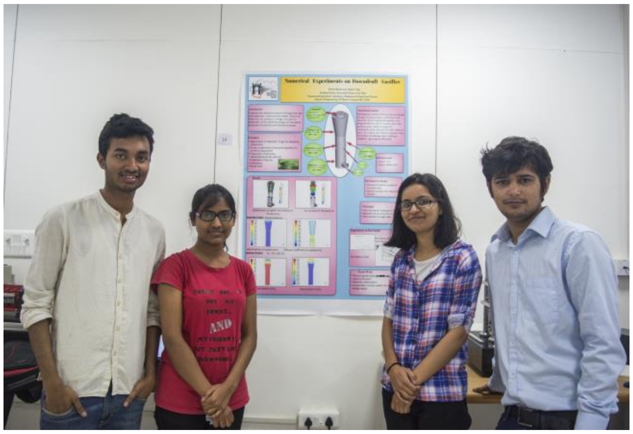

DC-DC POWER CONVERTER FOR SOLAR PV INTERFACING WITH DC MICRO-GRID
About the project
Final Circuit
Future Work
Components

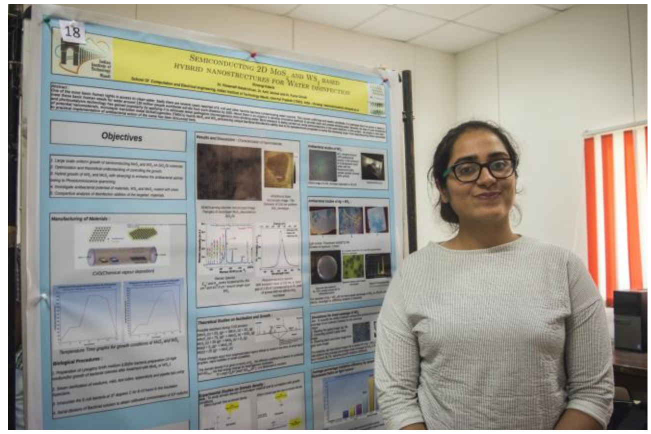

Power Quality Analysis in Power Distribution System

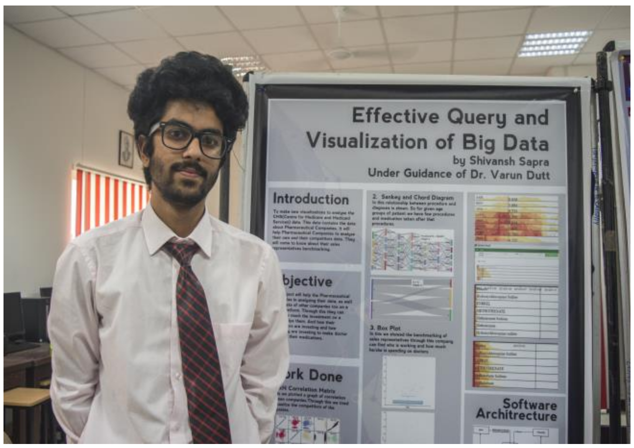

Modelling, Design and Analysis of PMSM Based Electric Drive

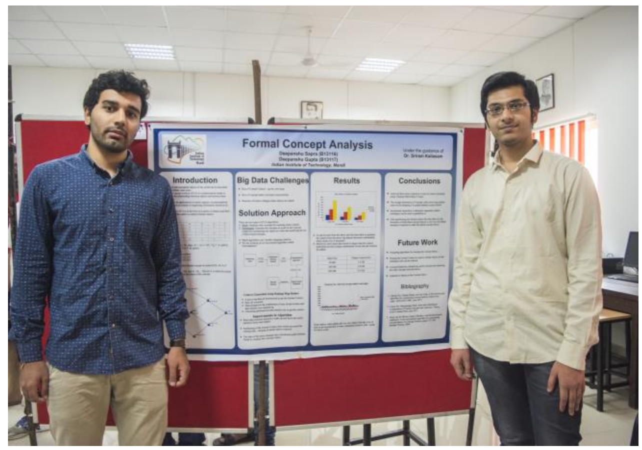


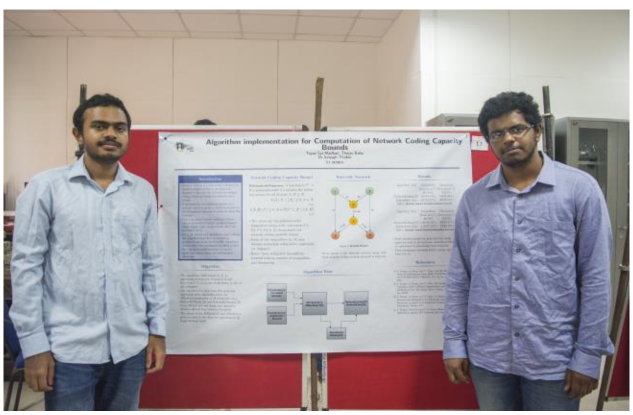

Analysis of particle transport
Objectives
Introduction
Impact of geometry on fluid flow
Methodology

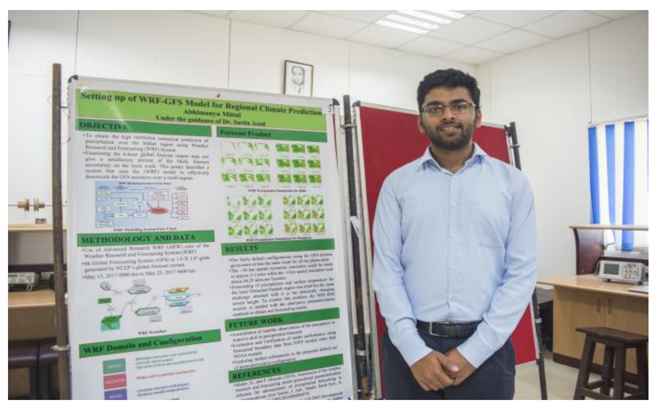


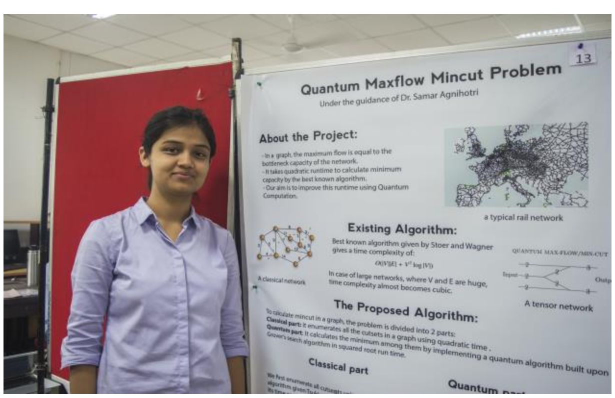

18
Objectives

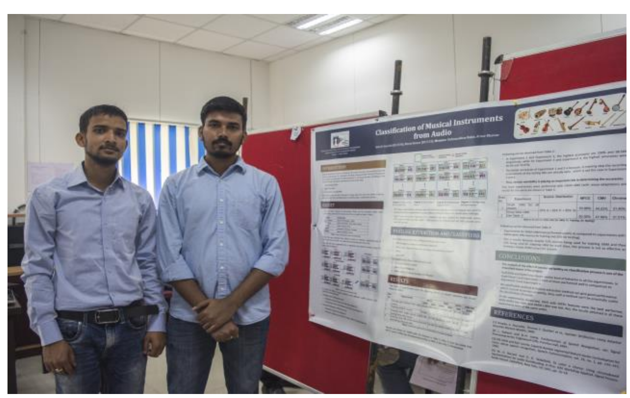

Effective Query and Visualization of Big Data
by Shivansh Sapra
Under Guidance of Dr. Varun Dutt
Introduction
2. Sankey and Chord Diagram
3. Box Plot
Software Architecture

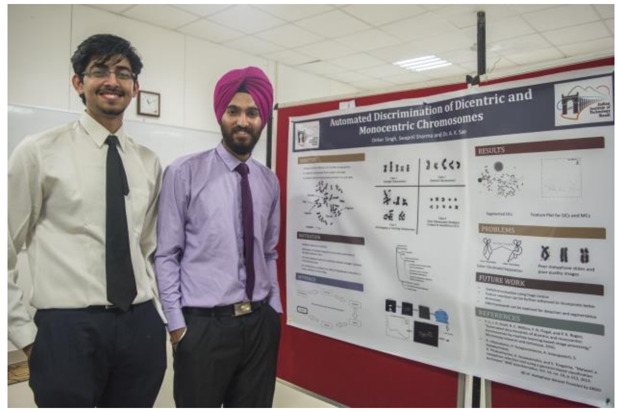

Formal Concept Analysis
Introduction
Big Data Challenges
Solution Approach
Results
Conclusions
Future Work
Bibliography


Algorithm implementation for Computation of Network Coding Capacity Bounds

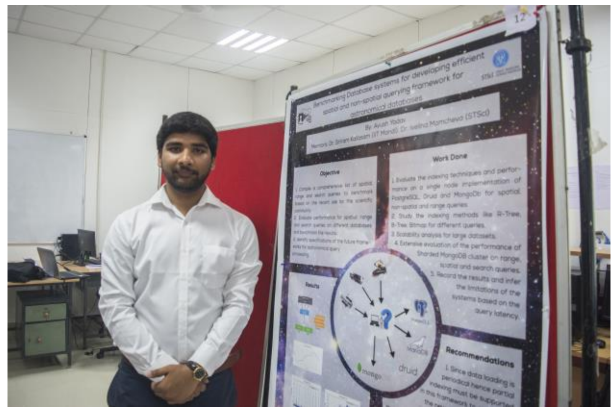

Setting up of WRF-GFS Model for Regional Climate Prediction
Abhimanyu Mittal
Under the guidance of Dr. Sarita Azad
OBJECTIVE
Forecast Product
METHODOLOGY AND DATA
RESULTS
FUTURE WORK
WRF Domain and Configuration
REFERENCES

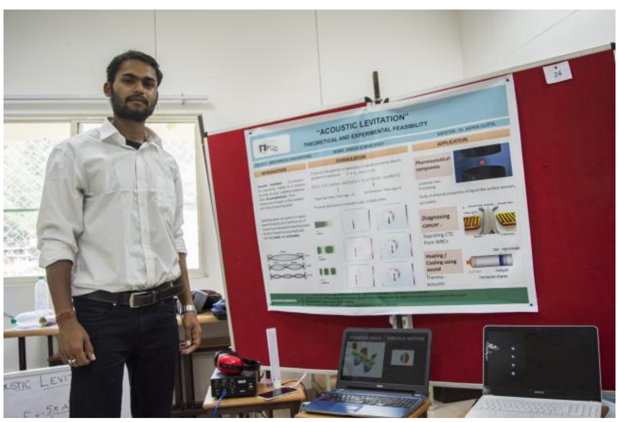

13
Quantum Maxflow Mincut Problem
Under the guidance of Dr. Samar Agnihotri
About the Project:
- In a graph, the maximum flow is equal to the bottleneck capacity of the network.
- It takes quadratic runtime to calculate minimum capacity by the best known algorithm.
- Our aim is to improve this runtime using Quantum Computation.
a typical rail network
Existing Algorithm:
Best known algorithm given by Stoer and Wagner gives a time complexity of:
A classical network
A tensor network
The Proposed Algorithm:
Classical part

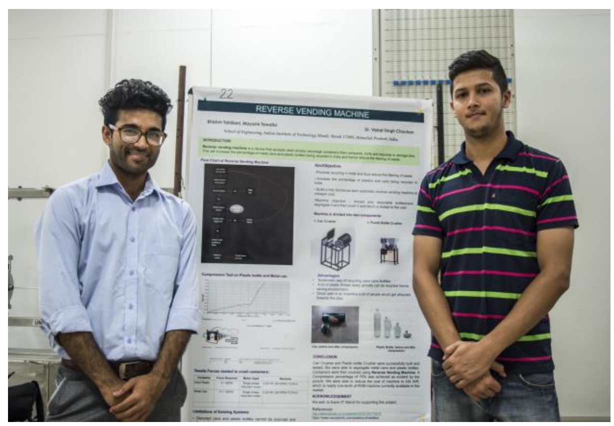

Classification of Musical Instruments from Audio

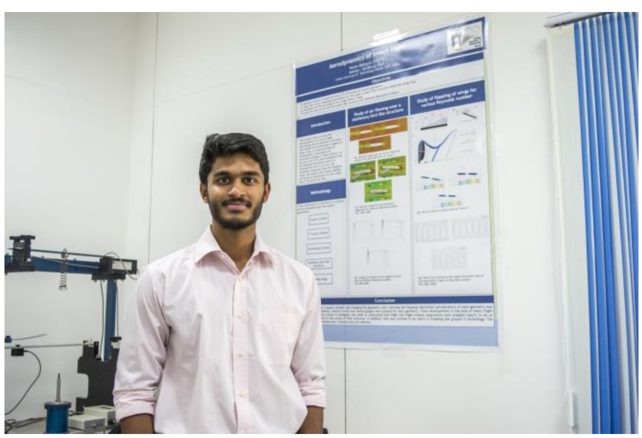

Automated Discrimination of Dicentric and Monocentric Chromosomes


Opening Lead Recommendation in Contract Bridge




"ACOUSTIC LEVITATION"
THEORETICAL AND EXPERIMENTAL FEASIBILITY


REVERSE VENDING MACHINE



Contact Us:

If you have any queries, then mail us at

chairscee@iitmandi.ac.in

sceeoffice@iitmandi.ac.in

Contact Number: +91-1905-267133

+91-1905-267046