A Question and Answering system for sports domain using Image Classification.

Ganesh Taduri, Gulnoza Khakimova, Raghava Kundavajjala
University of Missouri - Kansas City

Abstract

Artificial Neural Networks are a recent development tool that are modeled from biological neural networks. The powerful side of this new tool is its ability to solve problems that are very hard to be solved by traditional computing methods (e.g. by algorithms).Artificial Neural Networks (ANNs) are a new approach that follow a different way from traditional computing methods to solve problems. Since conventional computers use algorithmic approach, if the specific steps that the computer needs to follow are not known, the computer cannot solve the problem. That means, traditional computing methods can only solve the problems that we have already understood and knew how to solve. However, ANNs are, in some way, much more powerful because they can solve problems that we do not exactly know how to solve. That's why, of late, their usage is spreading over a wide range of area including, virus detection, robot control, intrusion detection systems, pattern (image, fingerprint, noise..) recognition and so on.

1. INTRODUCTION

The human visual system is one of the wonders of the world. People can recognize images/videos without any effort. That ease is misleading. In each fraction of our brain, we have a primary visual cortex, also known as V1, which contains 140 million neurons, with tens of billions of connections between them. Our vision not only involves V1 but entire series of visual cortices - V2, V3, V4 and V5 - which do more complex image processing. People heads are supercomputer, turned by evolution over hundreds of millions of years, and superbly adapted to understand the visual world. That is why we can easily recognize handwritten digits which is not an easy processes. Humans are colossally good at making sense of what our eyes show us. But all that work is done unintentionally. And we not usually appreciate how difficult a problem our visual system solve.

The difficulty of visual pattern recognition becomes apparent when we try to write a computer program which will recognize images, videos or handwritten digits. What seems easy when we do it ourselves suddenly becomes extremely tough. Simple intuitions about how we recognize shapes - "a 9 has a loop at the top, and a vertical stroke in the bottom right" - turn out to be not so simple to express algorithmically. When you try to make such rules precise, you quickly get lost in a morass of exceptions and caveats and special cases. It seems hopeless.

Neural networks approach the problem in a different way. The idea is to take a large number images, known as training examples, and then implement the system which can learn from those training examples. Neural network uses examples to automatically learn how to recognize images. Furthermore, if we increase the training data, the network can learn more about images and will increase the accuracy.

There are already several existing models which will automatically recognize images or give an annotation to videos. However we can built our own model which will recognize images/videos related to particular field, for example we can train

our model to recognize images related to sport games.

In this paper we will explain our approach in creating a question answering system for sports domain using image classification and recognition.

2. RELATED WORK

Currently, there are several tech startups which partner with broadcasting companies and major league sports. A few of the projects that are related to our project are:

- *NextVR*[11]

Seven years in the making, NextVR has now developed and patented the technology and built the only platform that can deliver live events in virtual reality with the energy and the passion of a truly immersive experience. The response has been so extraordinary that FOX Sports, Live Nation, NBC Sports, HBO/Golden Boy, Turner Sports, and CNN have all partnered with NextVR to create a wide range of scheduled programs and highlights to deliver a truly immersive experience to their dedicated fans.

One of the promises that virtual reality offered was that we would all be able to watch "courtside" sporting events without having to leave our couches. Organizers including those from the Rio Olympics and soccer's English Premier League have touted the new technology as a way to be there without having to deal with the expense, travel and crowds of an in-person game. There is a lot to like about the experience. You do feel very close to the action, and the cameras replicate the vantage point you would get from a close seat. Plus, you get a sense of scale when you're watching -- the height of the players, the baskets, the video board -- which adds a sense of presence you do not get on television. You are not so close to the action that you cannot see everything at once, as that would somewhat defeat the purpose of watching a game. And because you are sitting at the fixed camera's vantage point, you don't necessarily see the game as you would see it. You do not get close-ups of players' faces -- meaning personalities will not always come through as clearly as they might on a typical broadcast.

- *EON Sports*[12]

EON Sports, an athletic Virtual Reality training software company, was founded as a subdivision of EON Reality in 2013. EON Sports launched a virtual reality training program for football players in February 2014. By the end of 2013, EON Reality opened an education and entertainment research and development and production center in Laval, France.

EON Reality launched the EON Innovation Program, a start-up incubator program, in March 2014.The following month, the company established an Interactive Digital Center and EON Entrepreneur School in Moscow, Russia and Muscat, Oman. In October, EON Reality signed an agreement with the Mauritius State Investment Corporation and State Informatics Ltd. to invest $13.1 million US dollars in an interactive digital center in Mauritius.

3. PROPOSED WORK:

We propose a solution where we develop a model for image classification. Using this model we developed an application using Google Conversation for a question and answering system. we developed two models using Decision Tree, Random Forest and also we developed a model using Convolutional Neural Nets using Tensor Flow. A sample question s that can be asked to our system are:

What is the game?

Who is playing?

This application will be able to give a summary to the provided sport image.We also implemented this application using Clarifai API. Before getting to implementation part we would like to give a brief idea on technologies used for this application in the next section.

.3.1 Technologies Used

We tried to implement a variety of technologies for this project. The most most important ones are Spark MLib, Tensor Flow, Clarifai API, API.AI, and Heroku

3.1.1 Spark MLib

The Spark MLib is the spark library which contains many machine learning algorithms and utilities such as logistic regression, naive bayes for classification, Linear regression algorithms, Decision tree, random forest algorithms, algorithms for clustering such as K-means, GMMs etc.

3.1.2 Tensor Flow

Tensor Flow is an open source software library for numerical computation using data flow graphs. It was developed by Google. The tensor flow uses nodes in the graph to represent mathematical operations, the graph edges represent tensors(multidimensional data arrays) communicated between them.

3.1.3 Clarifai API

Clarifai API provides image and video recognition service. Clarifai API is used to generate image tags/captions. Using the frames from the video, generating the key frames, captions for each image are generated.

3.1.4 API.AI

API.AI is a computer-human interaction api used to develop conversation interface to perform tasks and answer questions in natural language. This api.ai is used as a platform for building conversational interfaces for devices. It uses information in the intents, takes inputs as queries which are either in natural language or an event name. The output is in the form of a task or some text in natural language. It uses the information in the intents and machine learning model to perform the output. The output as in JSON response object.

3.1.5 Heroku

Heroku is a cloud service used for deploying program/data. Heroku supports several programming languages that is used as a web application deployment model.

3.1.6 Inception V4

In order to optimize the training speed, Inception V4 uses to tune the layer sizes carefully in order to balance the computation between the various model sub-networks. In contrast, with the introduction of Tensor Flow most recent models can be trained without partitioning the replicas. This is enabled in part by recent optimizations of memory used by backpropagation, achieved by carefully considering what tensors are needed for gradient computation and structuring the computation to reduce the number of such tensors

3.1.7 Show and Tell

The Show and Tell model is used to generate image captions as text in the form of sentences. Show and tell model uses an encoder-decoder type neural network. It works by encoding an image into fixed length vector representation and then decoding the same into natural language representation text. The encoder is inception v3 network for image recognition and LSTM for decoding the images

3.2 Algorithms Implemented

3.2.1 Decision Tree

Decision tree learning used a decision tree as a predictive model which maps observations of an item to conclusions. They can be used both as both classification and regression trees.

3.2.2 Random Forest

Random forests are special types of decisions trees which are used for ensemble learning such as classification, regression etc. It uses Bootstrap aggregation unlike decision trees.

3.2.3 K- Means Clustering

K-means clustering is a clustering technique which uses n number of observations into K number of clusters in which observations belong to cluster with nearest mean.

3.3 System Design

3.3.1 Architecture:

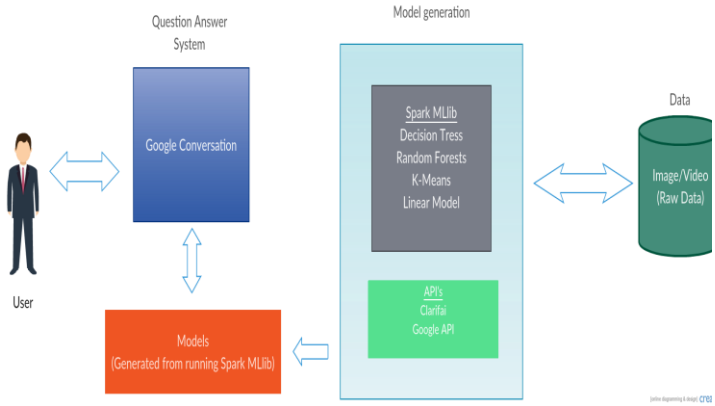


Fig 1. Architecture Diagram

3.3.2 WorkFlow

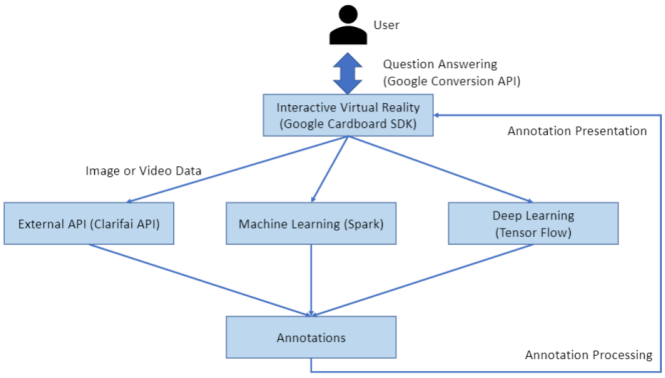


Fig2. WorkFlow Diagram

3.3.3 Sequence Diagrams

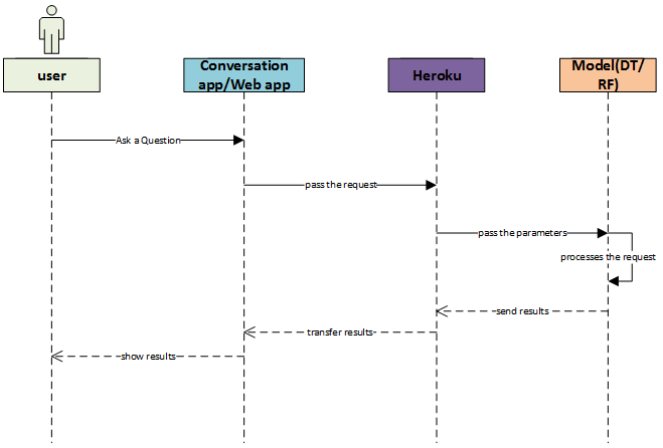


Fig3. Sequence diagram for Conversation app

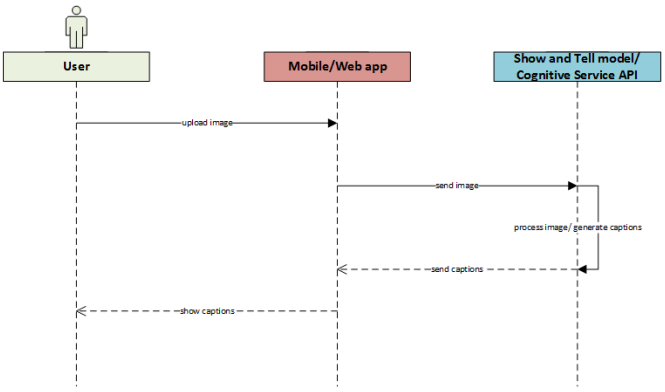


Fig4. Sequence diagram for Sumary App

4. IMPLEMENTATION

In this section, we will discuss the implementation of our application. This section includes overall workflow, model generation, training and testing of the different types of models, First we implemented our application using Clarifai API. Then, generated models using SparkMlib implementing Decision trees and Random Forests. Finally we developed a convolutional Neural Nets for Image Classification using Google Inception V4 and tensor flow. We will discuss the whole process starting from Data collection to model generation in this section.

4.1 Data Collection:

We collected large amount of dataset basically sports images. We classified these images into 6 classes each class specifying a specific sport. The different classes are:

1. Basketball
2. Boxing
3. Cricket
4. Swimming
5. Tennis
6. Skating

We collected around 250 images in each category and used this data to train and test the models.

4.2 Model Building Using Decision Tree and Random Forest (shallow Learning):

Image Classification is one of the task of the project. We implemented decision tree and random forest algorithms on the data collected and generated the models. We used Spark Milb to implement these algorithms.

First, we collected key descriptors in the image data set then used these key descriptors to generate clusters using k- means clustering algorithms. Then, created histograms for the images and finally generated models using Decision Trees by using the clusters and histograms generated from above steps. We repeated the same process for generating a model using Random forests too. We used cross validation while training and testing at 80:20 ratio while building the model. By comparing both the models we found that for image classification Decision Tree provides better results than Random Forests. We observed that Decision Trees has an accuracy of 86% while random forests gave only 77% accuracy. To obtain better results we build a convolutional neural net using Tensor Flow. The implementation is discussed in next sections.

4.3 Convolutional Neural Net Using Tensor Flow(Deep Learning)

In this section we implemented application for video annotation using Tensor Flow. First step was to divide dataset which contains images related to sport into batches which will help us to prevent our machine running out of memory and each batch contains labels and images, for example 10 batches for 10 different types of sport games. After processing images CNN were used on those images, which will train our model to recognize sport images. Our data passed through several neural network layers:

- Convolutional and max pulling layers
- Flatten layer
- Fully connected layer
- Output layer

After training our model in order to know performance/accuracy of our model we tested it against several datasets.

4.4 Implementation of Google Inception V4

For this phase we used transfer learning i.e., implementing the model that has already been generated for another model. Google has created a classification model called Inception V4 which is trained on millions of images for about 2 weeks. This model also uses Tensor Flow library. This

model can classify the images into about 1000 classes. We used the same network by limiting the number classes to 5 based on our own data set. Our model can classify an image into one of the 5 sports categories.

The network used used by Google for building Inception V4 firstly, involves extracting the features from the images in the training data and second part classifies the image based on those features. In this transfer learning we built a new model to classify our own data set by using the feature extraction part and training part from inception V4 own our own data. We trained the last layer in the Network such that it classifies our own image data set using the model generated by Inception V4.Once the model is generated after the training part we tested the model on a new sports image and the results were very effective. We observed that this model has about 94% accuracy.

4.5 Implementation of the application using Spark Client API

In the phase 1 of the project, we implemented a simple web application using a spark Client API. This application , given an image will identify what kind of sport it is. We considered five different categories of sports as five different classes while training the model. Cricket, BasketBall, Swimming, Boxing and Tennis are the five sport categories used for this application. We collected large amount of data set(Sports images) and categorized them into five different classes with respect to the sport the belong and used decision tree model to generate the model for image classification. SIFT features of the images are used for implementing the classification algorithm. Then, we used this model from server end to test an image from client side.

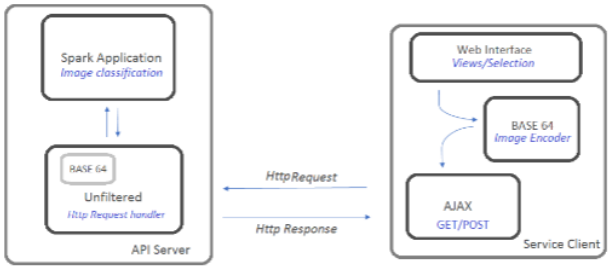


Fig. Architecture diagram for Spark Client API

A client can request a service from the server using GET and POST calls for sending an image to the server and to get a response of the prediction of the Image. Once the model gets an image it predicts the image sports category using SIFT features and displays the result at the client end.

We, also considered Random Forest algorithm for generating the model for image classification expecting it would fetch more accuracy than Decision tree. On Contrary Decision tree model has more Accuracy of 84% than Random forest which has around 72%. Screenshots of the results are included in the documentation.

4.6 Implementation of Google Conversation API

We developed a simple conversation app using Google's API.ai. we generated this conversation app by integrating the Heroku services. First we deployed TensorFlow on Heroku. Then created a google conversation app by importing the zip file which contained intents and actions in JSON format. Then Used heroku Restful Services to connect to Conversation app. by using this app a user can ask a question about a sports image and can get a answer form the model.

4.7 Implementation of Show and Tell model for image captioning.

We implemented Show and Tell model to generate captions for an image. A show and tell model is a deep neural network that learns how to describe

the contents of an image. We used google inception V3 model for encoding images and LSTM model for decoding. We did two phases of training for the first phase parameters of the inception V3 are kept fixed and a single trainable layer is added on top of inception V3 model to transform the image embedding into the word embedding vector. The model is trained with respect to the parameters of the word embeddings, the parameters of the layer on top of Inception v3and the parameters of the LSTM. And in the second phase all parameters including the parameters of Inception v3-are trained to jointly fine-tune the image encoder and the LSTM. It took around 16 hours of training for each phase of the training. this model generates 4 captions for an image. The results obtained were better in second phase of the training.

4.8 Implementation of Clarifai API

Clarifai APi offers video and image recognition. Clarifai API is built around a simple idea – you feed input file and Clarifai API service returns prediction. There are several models available in Clarifai API - for example food, travel, wedding, etc. In our project we are using the General Model.

This part of the project predicts what type of sport game is played on the provided video file. Below are steps used to predict video game using external API:

1. Key Frames detection- take video file as input and iterate over video frames to detect key frames:
2. Detect main frames – after detecting key frames, main frames need to be pointed out so they could be uploaded to Clarifai API server to get predictions.
3. Annotation – after detection main frames and uploading them to Clarifai server for processing we get back predictions on what sport game is shown on the main frame. At this level we need to filter some of the predicted values since we are interested only in what sport is shown on the video. We created the program which can predict the following sport games: soccer/football, tennis, basketball, volleyball, swimming, boxing, badminton and cricket. Other games can be added as needed.

4.9 Implementation of Cognitive Service API

In this phase we implemented Android application which will give us sport image description as a text and as a speech. In order to implement the application Thunkable was used to create the design of application which is connected to Cognitive Service API and Text-to-Speech tool. Our application either can upload picture from gallery or take a picture by phone camera and will give a description of the image in the text below the selected image and also will describe the image by speech.

GitHub link:

https://github.com/gt784/BDA_Project

5. RESULTS AND EVALUATION.

We trained different Image classification models using the data collected and tested them on different data set of sports images. We observed that that among all these Convolutional Neural Nets performs better as it is a deep learning approach. But it requires a high configured hardware and software and requires a lot of time. Another approach for Convolutional Neural Net which is Google inception model works best for image classification. It yields up to 96% of accuracy and takes less time compared to building a Neural Net from scratch using TensorFlow.

In case of shallow learning Decision trees performs better that random forests for Image Classification. From our experiments we observed that model

generated using decision trees has around 84% accuracy while model using random forests has about 72% accuracy. The confusion matrix and other metrics for these models are stated in next sections.

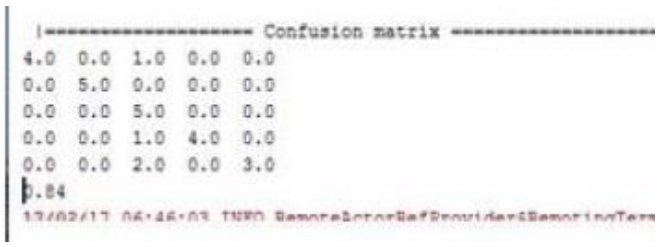
We used two API services one Clarifai API and other Microsoft Cognitive services API to implement our application. The results obtained were better but the services were restricted to only a few operations.

5.1 Observed Metrics:

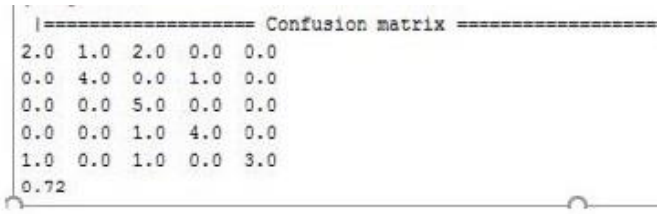
Below are the model evaluation results for the decision tree classification and random forest model generated. We use the conventional evaluation methods for a model such as representing a confusion matrix which shows the actual and predicted results from the model. Other metrics like accuracy of the model, precision, recall and F-measure for each classification class and overall error rate of the model is also calculated based on the conventional formulae involving the True Positive (tp) , True Negative (tn) , False Positive (fp) and False Negative (fn) measures.

Confusion Matrix:

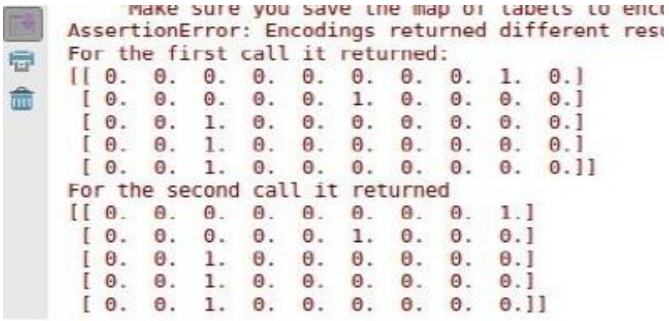
Using Decision Trees:



Using Random Forests:



Using Tensorflow:



Comparison of Accuracy of Various models:

The comparison of accuracy of models is as below.

- Clarifai API: 87%
- Decision tree: 84%
- Random forest: 72%
- Deep Learning : 94%

6. CHALLENGES

During implementation of our sport image/video recognition application we faced several challenges. Firstly it was not easy to find dataset of sport images which would help us to train our models, for some of the models we ended up using video frames as a training dataset. The well known problem with deep learning networks is that their training is difficult, can fall into local extrema, takes long time and powerful computational resources, e.g. GPU, CPU etc. and ones trained the network is not flexible/ adaptive to different new data.

7. CONCLUSION

A set of techniques has been developed that enable learning in deep neural nets. These deep learning techniques are based on stochastic gradient descent and backpropagation, but also introduce new ideas. These techniques have enabled much deeper (and larger) networks to be trained - people now routinely train networks with 5 to 10 hidden layers. And, it turns out that these perform far better on many problems than shallow neural networks, i.e., networks with just a single hidden layer. The reason, of project, is the ability of deep nets to build up a complex hierarchy of concepts. It's a bit like the way conventional programming languages use modular design and ideas about abstraction to enable the creation of complex computer programs. Comparing a deep network to a shallow network is a bit like comparing a programming language with the ability to make function calls to a stripped down language with no ability to make such calls. Abstraction takes a different form in neural networks than it does in conventional programming, but it's just as important.

In our project we tried to implement our application using different models and datasets, each one of them had their own strengths,weaknesses and accuracies. However there is still a room for improvement, we will try to increase accuracy and train bigger datasets which will help us the obtain better results.

8. FUTURE WORK

In the future phases of the project we would like to integrate face recognition techniques to identify a specific player in an image. Also, would like to implement object detection techniques to track the score board so that we can do some analytics. Also we would like to extend this project to video classification so that we can perform some live analysis and prediction about a live sports.

We also want to implement this application in Virtual Reality to have good 360 degree video experience for user.

ACKNOWLEDGEMENT:

We'd like to thank the ideas, guidelines and suggestions given by Dr. Yugyung Lee, Mayanka Chandrashekar, Naga Krishna Vadlamudi, Sudhakar Peddinti, and Manikanta Maddula.

REFERENCES

[1]https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2016-23.pdf

[2]https://ip.cadence.com/uploads/901/cnn_wp-pdf

[3]http://neuralnetworksanddeeplearning.com/chap1.html

[4]https://github.com/googlevr/gvr-android-sdk/tree/master/samples/sdk-videoplayer/src/main/java/com/google/vr/sdk/samples/videoplayer

[5]http://thunkable.com/explore/

[6]https://developer.clarifai.com/

[7]http://neuroph.sourceforge.net/image_recognition.html

[8]https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721

[9]https://www.tensorflow.org/tutorials/image_recognition

[10]https://github.com/tensorflow/models/tree/master/im2txt

[11]https://www.researchgate.net/publication/265917403_VRIO_A_Speech_Processing_Unit_for_Vi

rtual_Reality_and_Real-World_Scenarios_-_An_xperience_Report

[12] http://www.nextvr.com/news

[13]http://fortune.com/2016/04/29/mlb-eon-sports/

[14]http://www.livelikevr.com/#sectionHome

[15]http://virtuallylive.com/

[16] https://docs.api.ai/docs/welcome

[17]https://codelabs.developers.google.com/codelabs/tensorflow-for-poets/?utm_campaign=chrome_series_machinelearning_063016&utm_source=gdev&utm_medium=yt-desc#1

[18]https://github.com/googlevr/gvr-android-sdk/tree/master/samples/sdk-videoplayer/src/main/java/com/google/vr/sdk/samples/videoplayer

[19]https://signup.heroku.com/?c=70130000001xDpdAAE&gclid=CjwKCAjwxPbHBRAdEiwAMRyxS5oDRs6AlwJvU1V2_xs3VPcBh1Op4fAGTjsHz5XPxwW0MHOsEXY3cBoCC1MQAvD_BwE

[20]https://www.cs.swarthmore.edu/~meeden/cs63/f05/id3.html