Activity Recognition Using Video Captioning and Summarization

Deepanshi Bansal, Kshitij Gupta , Aayush Gupta, Pooja Gupta

Abstract: As the technology is improving every day, people are getting more inclined towards it.The busy schedule of people could not let them spare enough time to watch long videos whether it is related to some cricket match or some teaching or some glimpse of some video. Hence, there is a need to build a system to generate a summary of any video for having approximate glimpse without seeing the video. Also, everyday thousands of videos are uploaded on youtube and thus filtering is really necessary when we are searching for a particular video. There also, Video summarisation comes into action where one could skip multiple videos which may not be what the user want by just looking at the summary without watching it and investing time on it. The report discusses about how captioning and summarisation could be done using convolutional and recurrent neural networks and natural language processing.

Index Terms: captioning,neural networks, summary,video summarization,

I. INTRODUCTION

With the advent of the information age, there has been an exponential growth in demand for digital content, particularly in multimedia formats such as text, audio, image and video. Video, in particular, has quickly become the go-to format of choice with the increase in network bandwidth and storage space in mobile devices. Today, anyone and everyone with a smartphone can quickly shoot a small video and upload it to social media websites within seconds, making it accessible across the globe. However, this also poses new challenges. Unlike text and images, which can be analysed comparatively easily to obtain more information and perform complex analysis, videos offer a much more detailed and coherent view of data, with both spatial and temporal information playing a key part. Furthermore, analysing a video often requires a lot of manual effort and is thus a time-consuming task. A fundamental issue is the understanding of video contents without actually going through the video manually. This is where the concept of video captioning and summarization comes in. It enables one to automatically create captions by recognizing the activities, objects, and locations used in the video. What we get is a condensed description of a video that may then be used for further analysis.

Revised Manuscript Received on June 9, 2019

Deepanshi Bansal, CSE Dept., Maharaja Agrasen Institute of Technology(MAIT), GGSIPU, New Delhi, India.
Kshitij Gupta, CSE Dept., Maharaja Agrasen Institute of Technology(MAIT), GGSIPU, New Delhi, India.
Aayush Gupta, CSE Dept., Maharaja Agrasen Institute of Technology(MAIT), GGSIPU, New Delhi, India.
Pooja Gupta, Assistant Prof. CSE Dept., Maharaja Agrasen Institute of Technology(MAIT), GGSIPU, New Delhi, India.

The applications of such an approach are virtually unlimited. Not only will the effort allow for a more rigorous library of videos, it will also simplify the process of searching videos. As of now, a video is searched using the tags and description given to it manually. However, once a video has been summarized and adequately captioned, the newly gained information can be used to search and look through important data in any video, irrespective of whether it was assigned the correct and adequate tags in the beginning or not. Moreover, using this mechanism, viewers can also be allowed to navigate to a particular section of the content they might be looking for. Another application includes automatic activity recognition in videos. Rather than sitting in front of the CCTV footage for hours and manually looking through the same, video captioning and summarization can be used to automatically detect the presence of a fraudulent activity and report it to the concerned authorities. Video captioning allows one to measure how content is consumed by the users. Video managers can analyze which terms were frequently clicked on and which section was re-visited the most. This data can help benefit future content choices and SEO strategies.

II. RELATED RESEARCH

Zuxuan Wu and Ting Yao[2]discusses in their research on the topic Deep Learning for Video Classification and Captioning about the need for video analysis which could be done through classification or through captioning. Different models and strategies used by different authors till date are discussed and the dataset that produces best results with different models are discussed.It discusses about the attention mechanism with LSTM layer to focus on relevant frames to optimize time.

Cheng-BinJin, Shengzhe Li, and Hakil Kim [3] in his paper discusses about knowing the activity being done in the video with the help of three levels that is posture, locomotion and gesture level with multi CNN.The author uses Kalman tracking algorithm for detecting human action regions.It uses 3 CNN.The first network, BDI-CNN, operates on a static cue and captures the shape of the actor. The second network, MHI-CNN, operates on a motion cue and captures the history of the motion of the actor. The third network, WAI-CNN, operates on a combination of static and motion cues and captures the patterns of a subtle action by the subject.Joo-Hwee Lim and Ah-Hwee Tan[4] in thier paper tries to produce good results on customized summarization of video that is different summarization based on different people according to what they want to focus

on , in the video.They demonstrated that AVS strikes a balance between usability and quality of the summary. In the future, they used the user's previously generated summaries will be used to learn his or her preferences.Saarbr¨ucken[5] in this paper a novel dataset of movies with aligned descriptions sourced from movie scripts and DVS was analyzed. First experiments on this dataset were performed using state-of-the art visual features, combined with a recent movie description approach. The approach was adapted and compared against movie scripts. It was found that video captioning and summarization using DVS was a more successful approach than manually made scripts and video descriptions.
BehroozMahasseni, Michael Lam and Sinisa Todorovic Oregon[6] discussed about the problem of unsupervised video summarization, formulated as selecting a sparse subset of video frames that optimally represent the input video. The key idea was to learn a deep summarizer network to minimize distance between training videos and a distribution of their summarizations, in an unsupervised way. The summarizer is the long short-term memory network (LSTM) aimed at, first, selecting video frames, and then decoding the obtained summarization for reconstructing the input video.

III. KNOWING ABOUT MODELS USED

1) *Convolutional Neural Network(CNN)*

CNN is a feed forward neural network which is widely used for image and video recognition, pattern recognition and text processing. It makes use of different layers which include convolutional(Conv3d) layer, relu layer, pooling layer, flattening and then applying hidden layers. The colored image is converted into 3D array denoting red, green and blue channels. In case of videos, 4D array is generated where first dimension denotes the frame and the later dimensions are RGB channel for that particular frame which actually is an image. Now assuming that the input is image, the input is feeded to convolutional layer which gives the collection of feature map. A feature map is combination of different pixels of an image. Rectifier function is applied to the feature map which increases the non- linearity thus removing the negative values. The output of to relu layer is feeded to the pooling layer where the parameters are reduced to prevent overfitting and unnecessary data is removed. Pooling could be maxpooling which takes maximum value, sumpooling which takes average of values and mean pooling which takes mean of all values. This process is done multiple number of times .Input to the neural network is the flattened matrix formed from feature detector after last pooling.Hidden layers are applied and at the output layer, Cross Entropy and softmax is calculated.Back Propagation is done during which both weight and feature detector is trained again and again to reduce the loss and get closer to the best possible accuracy.

2) *Recurrent Neural Network (RNN)*

RNN takes a step forward from a simple neural network. It stores immediate previous output in its memory to carry forward with the next perceptron. This makes it different from other NNs. Hence the input to RNN are two ie. the present input and the immediate output of the previous perceptron.The parameters remain the same as past comes along with present. This model is a way too useful and best suitable when something is related to the previous decisions made. For example, during sentence formation if the sentence started with talking about a boy then the model has to make sure that further more in the sentence it uses the male tone to point that person. Back propagation plays an important role here. It tries to get to the closest possible accurate answer. At each step gradient is calculated based on weights of different parameters. But with recurrent neural networks , if any of the gradient approaches 0 , then next gradients will exponentially approach to 0 and this will eventually not help in improving the model thus will not learn anything.To deal with this, LSTM is there. It is extended version of RNN or it could be said that it is a subtype of RNN.
Long Short Term Memory(LSTM) has a memory way more than that of RNN. It stores a longer poster of past and thus better. It memory is in the form of gates which are of three types .They are – forget gate, input gate and output gate.All the three gates together comprises of the memory of LSTM which makes it possible to remember past for a longer duration.

IV. DATASET

1) *MS- Coco Dataset*

The MS-Coco Dataset is a dataset created by Microsoft which consists of 82,783 images for training purpose and 40,504 images for the testing purpose. There are annotations for all the images and used to find all the possible words in the dataset for the image and build a vocabulary for the model.

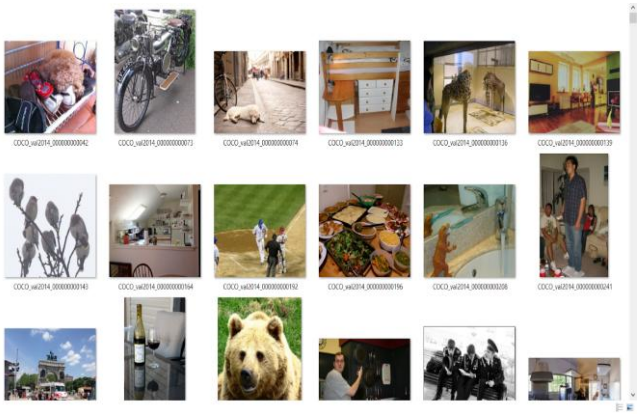

Fig 1. MS-COCO Dataset

2) *SumMe Dataset*

The SumMe dataset consists of 25 videos covering holidays, events and sports. They are raw or minimally edited user videos, i.e. they have a high compressibility compared to already edited videos. The length of the videos range from about 1 to 6 minutes.

Fig 2(a)

Fig 2(b)

Fig 2 (c)

Fig 2(d)

Fig 2. Screenshots for different videos from Summe

V. PROPOSED METHODOLOGY

In this work video summarization methodology has been discussed which enables to extract and isolate the different features from any video. The process starts by doing data preprocessing and doing frame captioning. Then the captioning of all frames are combined and meaningful sentences are kept and rest are eliminated.

Details of each process is discussed as below-

A. *Data Preprocessing*

1) Resizing

Before the model can be trained and applied to new data, the training data needs to be pre-processed for faster and more efficient training. By following a few key pre-processing steps, the overall process becomes much more streamlined without having a major effect on the overall accuracy of the final results. The MS-COCO training dataset consists of 82,783 images from various walks of life, with no specific dimensions for the images. For the purpose of better training at a faster rate, each image from the training data was resized to a fixed size of 256 x 256 pixels. The resultant dataset was made homogenous and stored in a separate directory from where it would be used for the purpose of training the parameters of the model.

2) Building the Vocabulary

When training a LSTM, it is standard practice to tokenize the input. For a sentence model like ours, this means mapping each unique word to a numeric id. The MS-COCO dataset comes with a set of annotations and captions for each of the images in the training data. A basic requirement for the use of this information is to tokenize the vocabulary. These annotations and captions were used to build a simple vocabulary wrapper such that it can be used to train the data. The process involves traversing through the entire captioning data and searching for all possible words in the dataset, and thereafter building a vocabulary list for the model by enumerating all possible captions from the vocabulary. Some special tokens were also added to the list, which included <start> and <end>. These tokens help in the overall functioning of the system by clearly delimiting the start and end of the predicted caption.
Using the entire vocabulary of a dictionary is also not feasible, given the huge number of unique words, with a lot of synonyms. As a result a certain threshold was set to ensure only frequently used terms are added to the dictionary. The captions of the image were initially available in the json file along with the dataset. Some images had more than one annotation given with a rating of how good the annotation is as compared to the actual image. The resultant data was stored inside a separate .pkl file so that it could be used when required. Our model is expected to caption a frame based on the information provided from its snapshot as well as the vocabulary of unique words in the training set.

B. *Frame Captioning*

1) Training

Generally, a CNN extracts the features from the input image and transforms the feature vector to have the same dimension as the input dimension of the RNN/LSTM network. This network was trained as a language model on our feature vector. In images, neighbouring pixels generally represent the same kind of information, except on the edges. This knowledge is used in the CNN to recognize regions representing similar and dissimilar objects, thereby resulting in the power to recognize objects in the image. For example, if the model detects various skin tones and edges that look like an oval, it will most likely be a face. For training our LSTM model, we predefine our label and target text.

The initial task in training is to set up the environment and model, and load the required data wherever required. The image preprocessing check is performed to ensure that the frame has been preprocessed and resized to the required dimensions. Thereafter, the vocabulary wrapper is loaded, along with the information regarding all the directories where necessary files have been stored. The encoder and decoder are initialized for the ResNet 152 model. The initial weights are assigned randomly.
Due to the large size of the image dataset and captioning data, the optimization is done in batches. The training requires a lot of space and time if this process is not followed. This also leads to the requirement of the ability to store and resume the training as and when required.

2) Creating Checkpoints

The time requirements for the training process leads to the need to save the progress and parameters and resume the training from the saved checkpoint. In order to save the model checkpoints, the encoder state parameters are stored in a state dictionary after a fixed number of steps in each epoch. The same is done for the decoder state parameters. For this purpose, the torch.save() function of the PyTorch library is utilized. Moreover, to avoid confusion and to clearly mark each checkpoint, the saved files are clearly demarcated with the epoch and step number which was in progress at the time they were saved.

3) Resuming Training from Checkpoints

There was a requirement to continue the training process from the point it was stopped at. For this purpose, it was necessary to load all the state parameters back to the encoder and decoder objects. In order to achieve this, the encoder.load_state_dict()and decoder.load_state_dict() functions were utilized right after creating the respective encoder and decoder objects. The parameter values were populated from the checkpoint data saved earlier. This allowed for a resumption in the optimizing of the parameters so that better accuracy could be achieved. The code for resuming the training was run only after the training had been started at least once before, and up to a point where a specific checkpoint could be saved.

4) Training on Google Colab

The hardware requirements for the training process involved a very powerful multi-core processor with large amounts of RAM and a powerful dedicated GPU. The lack of any of these requirements rendered the training process very slow to be practically feasible. To overcome such an effect, the services of Google Colab were used to train the model and optimize the parameters. The Google Colab service is a free cloud service based on Jupyter notebooks that supports free GPU usage on a linux system for up to twelve hours on a stretch. It provides its users with a powerful Intel Xeon processor clocked at 2.3GHz and coupled with a dedicated GPU and 13GB of RAM. Shifting the training process to Google Colab resulted in a speed increase of about twelve times from the previous training speed. This brought the total training time down from over forty nine days to just over four days.
The training code was uploaded onto Google Colab and the training dataset was imported into the online service through Google Drive. Thereafter, the optimization process was carried out, with a couple of checkpoints being created during each epoch. This checkpoints were used to resume the training later.

5) Applying model to SumMe dataset

SumMe dataset had 25 videos of different activities and situations. For every 5 sec, a frame is taken and for all videos , 100 frames is taken. The captioning of each frame is done and stored.The encoder used is encoder-44-3236.ckpt and decoder used is decoder-44-3236.ckpt.
This produces a long paragraph having an idea of overall video which may or may not contain repetitive sentences.
For different videos, different sentences are stored in array along with tokens <start> and <end> to differentiate between different sentences.
The output obtained is then applied to remove sentences with less importance or repetitive sentences.

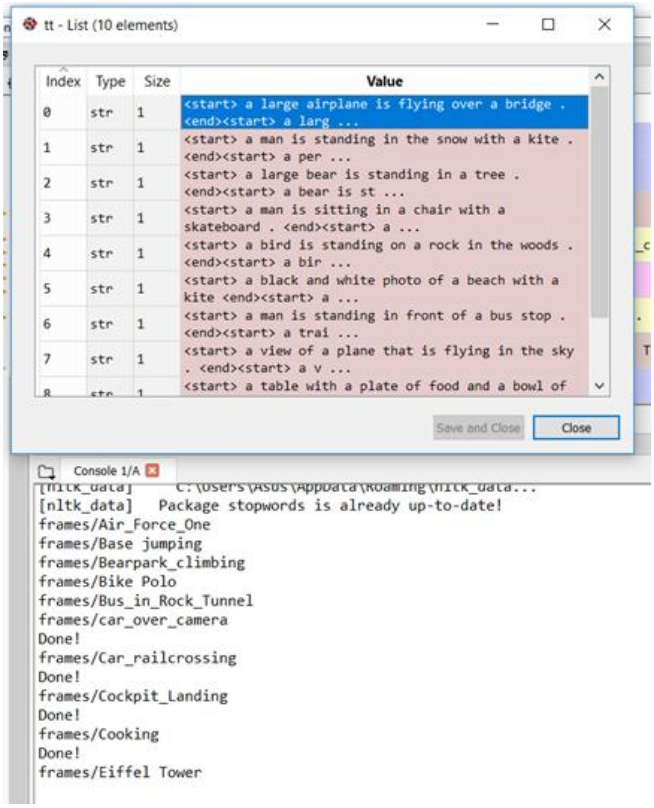

Fig 3. Result for applying frame captioning summe dataset

6) Text Summarization

Text summarization is done by removing repetitive sentences from the different frames of same video and thus making a useful paragraph with the sentences that values while understanding or while getting the video summary. Summarization was done using nltk(natural language toolkit) as well as using general code to remove repetitive sentences if any. It was a necessary step as the result of frame captioning was a way not straight forward. It produced the lines as the number of frames which was difficult to understand and hence this process was done.

Frames)			
Summe Dataset(Videos)	25	7	68%

Table 1 . Result Analysis

As the dataset for videos is not available as whole and preprocessing on a video dataset could be time consuming . Hence our work is able to use image captioning data to train videos and produce good results. The model was trained up to 44 epochs for approx. 82,000 images . 25 videos were available in summe dataset which were used to train to summarize the captions with text summarization. 7 videos created by the authors were taken to check the model applied on summe dataset. The accuracy was checked and came out to be 68%.

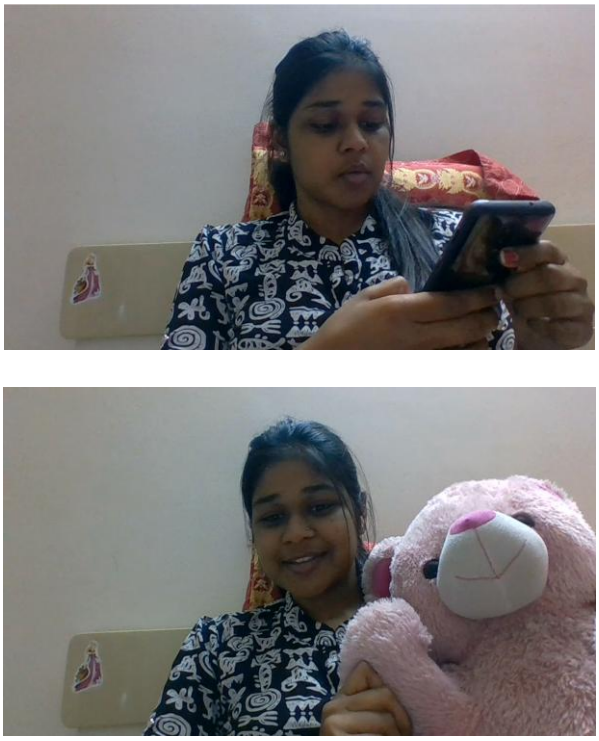

Fig 4. Real Time Video Summarization

VI. ANALYSIS AND RESULT

	Training	Testing	Accuracy
COCO Dataset(for	82,783	40,504	86%

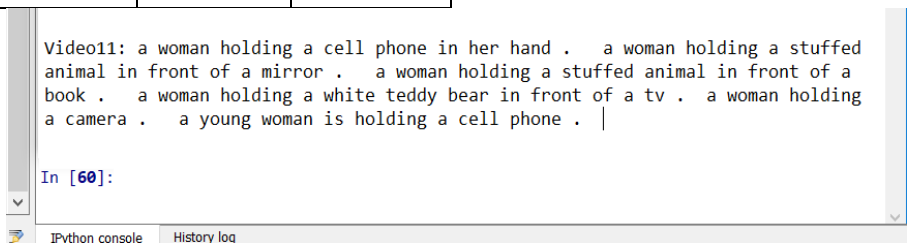

Fig.5 Summarized activity recognition result for fig 4

VII. CONCLUSION

The algorithm used for frame captioning for different images worked quite well on testing dataset and on video dataset as well. Captioning of videos become successful with continuous frame captioning. Summarization of all the frames from each video was done for better and meaningful results.
The preprocessing done before implementation of algorithms was necessary because we needed same embedding matrix size for different frame in videos. Also, building of vocabulary was necessary to produce words for captioning model build with LSTM.
We have planned to now go a few steps further to extract pitch and duration information from the audio signal.
The system for video summarization is built with 70% accuracy. It could be integrated in YouTube to know about the video without investing the complete time watching the video. Summarized text could be useful in recognizing different activities of user in CCTV camera and thus can be used for surveillance for every hour. Not only this, video summarization could be used in sports to get a summarizes report of different shots played, speed and the other components of ball etc.
Further, in future we are planning to add audio feature extraction which could also get the summarized text which could be combined with frame summarized text to produce a better optimized video summarization for practical implementations.

REFERENCES

1. Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., &Baskurt, A. (2011, November). *Sequential Deep Learning for Human Action Recognition*. In International Workshop on Human Behavior Understanding (pp. 29-39). Springer Berlin Heidelberg.
2. Zuxuan Wu (University of Maryland, College Park),Ting Yao (Microsoft Research Asia), Yanwei Fu (Fudan University), Yu-Gang Jiang (Fudan University).*Deep Learning for Video Classification and Captioning* , Feb 2018
3. *Real-Time Action Detection in Video Surveillance using Sub-Action Descriptor with Multi-CNN* by Cheng-Bin Jin*, Shengzhe Li†, and Hakil Kim* *Inha University, Incheon, Korea †Visionin Inc., Incheon, Korea 2017
4. *Unsupervised Video Summarization with Adversarial LSTM Networks* BehroozMahasseni, Michael Lam and Sinisa Todorovic Oregon State University Corvallis, OR,2017
5. *Large-scaleVideo Classification with Convolutional Neural Networks* Andrej Karpathy1,2 George Toderici1 Sanketh Shetty1,2Computer Science Department, Stanford University,2015
6. *Deep Learning for Videos: A 2018 Guide to Action Recognition* Rohit Ghosh , Qure.ai Blog, Revolutionizing healthcare with deep learning June 11, 2018
7. *Large-Scale Video Summarization Using Web-Image Priors* Aditya Khosla Raffay Hamid Chih-Jen Lin Neel SundaresanMasschusetts Institute of Technology, Cambridge MA 02139 eBay Research Labs, San Jose CA 95032 ∗National Taiwan University, Taipei 10617, Taiwan
8. *Gaze-enabled Egocentric Video Summarization via Constrained Submodular Maximization* Jia Xu , Lopamudra Mukherjee§ , Yin Li , Jamieson Warner , James M. Rehg , Vikas Singh† †University of Wisconsin-Madison, University of Wisconsin-Whitewater Georgia Institute of Technology
9.

AUTHORS PROFILE

Deepanshi Bansalhas completed BTech CSE from Maharaja Agrasen Institute of Technology, GGSIPU, New Delhi, India in 2019.
Deepanshi Bansal has been Microsoft Student Partner for 2017-18 and 2018-19. Also she has been treasurer of society International Organization of Software Developers in 2017-18. Now Deepanshi is Vice President of the same society for 2018-19. She has won Deloitte Innovation Award In Smart India Hackathon 2017.She has published a research paper on multiple speaker recognition in scopus indexed journal IJEAT .
Her email ID is deepanshi116@gmail.com

Kshitij Guptais a student of B.Tech in Computer Science and Engineering at Maharaja Agrasen Institute of Technology, GGSIPU, New Delhi, India. Being a rank holder at his college, he also has certifications in Java and Python. He has taken part in various national level competitions including the e-Yantra Robotics Competition in association with IIT Bombay and Ministry of Human Resource Development, Government of India. After completing his graduation in 2019, he plans to pursue the Business Management Program of XLRI Jamshedpur, to improve his knowledge about the functioning of big corporations and conglomerates and put his technical know-how to good use. His prior work includes building a facially recognised recommender system.Email ID: kgupta4497@gmail.com

Aayush Gupta is a student of B.Tech in Computer Science and Engineering at Maharaja Agrasen Institute of Technology, GGSIPU, New Delhi, India. Graduating in 2019, he has a keen interest in Machine Learning and Web Development, and has already worked on projects such as YouTube video analysis and prediction. He is a certified Java and Python developer and was one of the finalists in Compudon 2015. Currently, he is looking forward to working with Tata Consultancy Services as a Software Developer. Email ID: aayushgupta390@gmail.com

Pooja Guptais Assistant Professor at Maharaja Agrasen Institute of Technology(MAIT) , GGSIPU, New Delhi India. She has completed Ph.D. Computer Science & Engineering from JIIT (April-2014) and M.Tech (CE) from YMCA, Faridabad (MDU, Rohtak) (2006).Her area of interest are Information Retrieval and Data Mining, Machine Learning, Computer Networks. She has written various research papers , few of them are “An Improved approach to rank web documents”, Journal of Information Processing Systems (JIPS), Korea, Vol. 9, No.2, pp-217-236 (indexed at DBLP and Scopus.) , “A Novel Technique for Back-Link Extraction and Relevance Evaluation”, International Journal of Computer Science & Information Technologies, Vol 3, No. 3, June 2011 pp-227-238. [indexed in docstoc, pubzone, DOAJ, Inspec, EBSCOS etc.] , “A Novel Framework for Context Based Distributed Focused Crawler (CBDFC)”, Int. J. Computer and Communication Technology, Vol. 1, No. 1, 2009 pp-14-26 (Copyright Inderscience Enterprise Ltd.) etc. Email ID - poojaguptamait@gmail.com