An Alternative To Captcha - Video Captcha

B.Tech Seminar report

by

Nawas Abdulla
ETAHECS028

Department of Computer Science And Engineering
Government Engineering College, Thrissur
December 2010

Acknowledgment

I would like to express my sincere gratitude to Prof. Manoj Kumar, the Head of the Department of Computer Science and Engineering, for giving me an opportunity to present this seminar.

Also, I take this opportunity to show my greatest appreciation to Prof.K S Valsaraj, Mr.Ajay James lecturer and Mrs. Baby Syla lecturer, Department of Computer Science and Engineering, for their able guidance and useful suggestions in the various stages of this seminar.

Last but not the least I would like to thank my classmates for being a constant source of encouragement and patient listeners throughout the seminar.

Above all, I thank almighty for all the blessings showered upon me for the successful presentation of the topic.

Nawas Abdulla
December 2010
Govt. Engineering College, Thrissur

Abstract

Video Captcha is a new technique for using content based video as a CAPTCHA task. These CAPTCHAs are presently generated from YouTube videos, which contains tags supplied by the person while uploading that video. These videos are graded using the given videos tags, and also tags from related videos. The human success rate of video CAPTCHA comes upto roughly 70 % to 90 %, while the attack rate at around 13 %. The usability and security of video CAPTCHA is comparable to existing CAPTCHAs, and finds more enjoyable than traditional CAPTCHAs.

Contents

1 Introduction **1**

2 COLLECTING VIDEO SAMPLES **3**

3 CHALLENGE GENERATION **5**

3.1 Related Videos 6

3.2 Cosine Similarity of Tag Sets 6

3.3 Adding Related Tags 7

3.4 Rejecting Frequent Tags 7

4 GRADING FUNCTION **9**

4.1 Preprocessing 9

4.2 Expanding Tags through Word Stemming 9

4.3 Allowing Inexact Matching 10

5 ATTACK SIMULATION **11**

6 USER STUDIES **12**

7 Conclusion and Future Works **15**

7.1 Conclusion 15

7.2 Future Works 16

References **17**

Chapter 1

Introduction

A Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) is a different type of the Turing test in which an online challenge is used to distinguish whether the user is human or a computer program. The purpose of captcha is for preventing the abuse of online services, that is making programs that creates thousands of free email accounts and then using them to send SPAM. Different types of CAPTCHAs are Character recognition, Speech recognition and Image understanding.

Four desirable properties for CAPTCHAs are :

1. Automated: Challenges should be automatically gen- erated and graded by a computer.
2. Open: The database(s) and algorithm(s) used for generating and grading the challenges should be made public. By Kerckhoffs Principle, A system should remain secure even if everything about the system is public knowledge.
3. Usable: Challenges must be easily solved in a reasonable amount of time by humans.
4. Secure: Challenges should be difficult for machines to solve algorithmically.

The widely used type of CAPTCHA requires a user to tran- scribe distorted characters displayed within a noisy image. The algorithms that is used to automatically generate the challenges are publicly available, but many users find them frustrating. Researchers have developed many automated programs that have been successful at defeating them. For example, researchers have developed a program that eventually yielded an attack success rate of 60% against Microsofts Hotmail CAPTCHA. Therefore a need for a new CAPTCHA which is automated, open, usable, and secure arises.

In this method a user has to provide three words (tags) descrsibing a video taken from a public database.We can chose videos from youtube which is presently the largest video database. Words are submitted when the video plays, i.e. the user need not wait until the video finishes. The challenge in its simplest form, is passed if one of the three submitted labels or tags match an author-provided tag related with the video. Exact matching is difficult due to misspelling and inconsistency in tagging. So the algorithm is modified to overcome this problems.

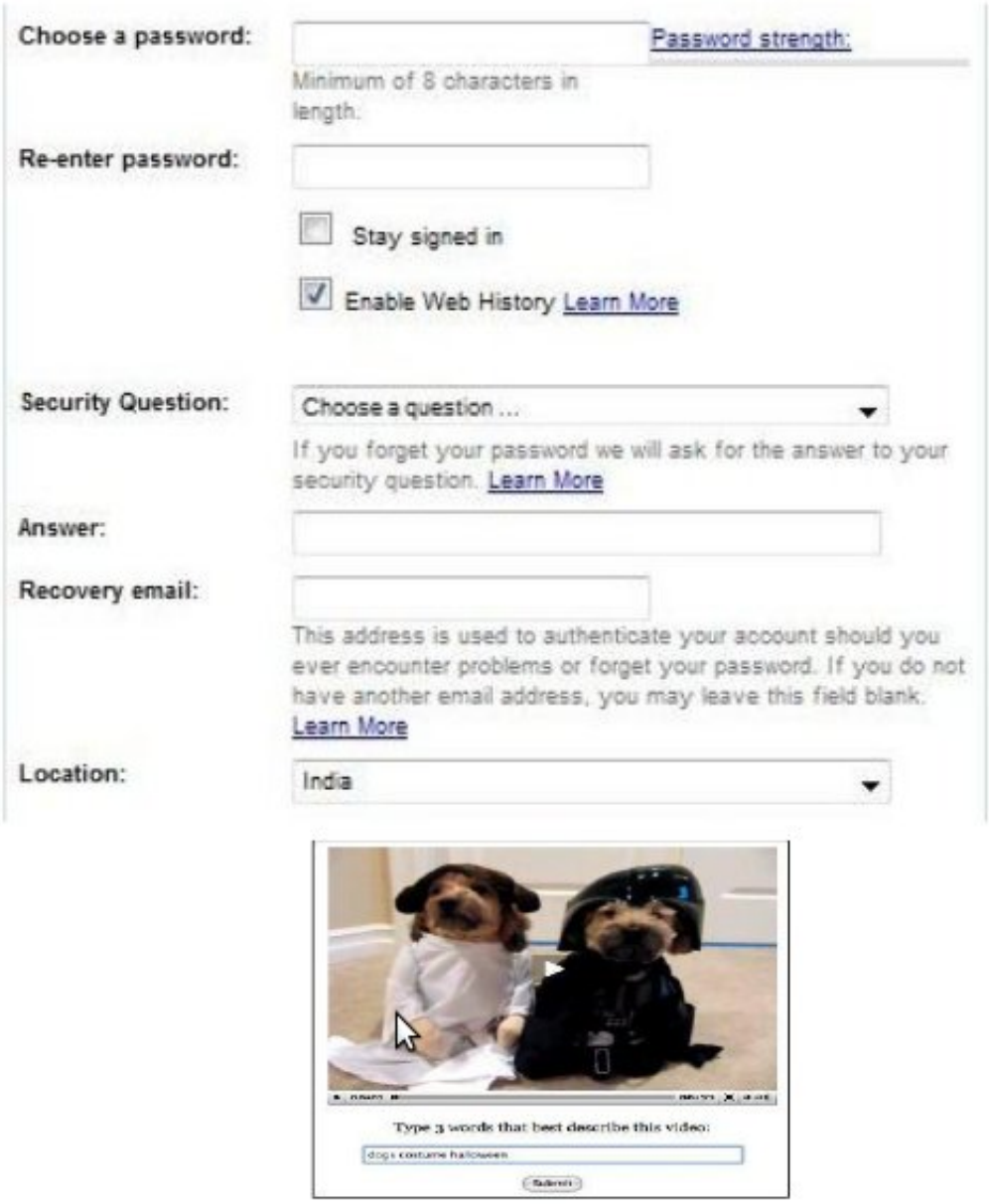

Figure 1.1: Demonstration

Chapter 2

COLLECTING VIDEO SAMPLES

YouTube.com is utilized as the dataset for challenge generation, which is at present the largest user-generated content video system avail- able. It currently stores and indexes about 150 million videos.

Generating YouTube video identifiers (IDs) randomly would yield a true sample, but it is not practicable for collecting a large sample in this fashion. YouTube video IDs are of 11 characters length with a character set comprising of numbers (0-9), lower case letters (a-z), uppercase let- ters (A-Z), underscores (_) and dashes (-) with a total of 64 different characters. On calculation there will be $64^{11} \approx 7.4 \times 10^{19}$ possible IDs. Currently there are approximately 1.5 $\times 10^{8}$ videos on YouTube.com, So the probability of generating a valid video ID randomly is approximately 2 $\times 10^{-12}$.This is not a suitable method for collecting large samples.

The common method used for sampling large social networks is known as snowball sampling. An s stage k name snowball sample is equal to a breadth-first search in which a fixed number of children are selected randomly in the search tree at each node .

A criticism of snowball sampling is that it biases results towards individuals who are connected to the entry points. As a result, random walks are done, which is a kind of randomized local search. This was a technique previously used for sampling video data. Here YouTube is modelled as an undirected, bipartite graph G. The vertices in the graph consists of two disjoint sets: tags U and videos V . The edges in the graph are of the form (u, v) and (v, u) such that u $\in$ U and v $\in$ V ; edges represent associations be- tween videos and tags. Given the YouTube video-tag graph G, a maximum walk depth m, and a dictionary D, the algorithm below returns a random walk of the graph in the form of an ordered list P of video-tags pairs (v, A).

RANDOM WALK(G, m, D)

1. Create an empty list, P $\leftarrow \emptyset$, and counter i $\leftarrow$ 0.
2. Randomly select a walk depth d, where $1 \leq$ d m.
3. Randomly select a starting tag t from dictionary D.
4. Located the tag vertex u corresponding to t in G.
5. While i d:
   (a) Select a random edge (u, v) in G, where v is a video vertex.
   (b) Given the tags A on the video v, append (v, A) to P .
   (c) Select a random edge (v, w) in G where w is a tag as- sociated with video v.
   (d) Assign u $\leftarrow$ w and increment i.
6. Return the list of video-tag pairs P .

The dictionary D, makes use of the English word list available on most of the Unix-based systems.

Chapter 3

CHALLENGE GENERATION

The challenge generation function constitutes of a YouTube graph G, dic- tionary D, maximum walk depth m, tag frequency distribution estimate F , maximum number of related tags to add n, and also a rejection threshold t. The CAPTCHA generation algorithm will return a pair (v, GT) which contains a video and also a set of acceptable ground truth tags.

VIDEO CAPTCHA(G, m, D, F, n, t)

1. A random walk of the video-tag graph G is performed to maximum depth m using dictionary D to choose a video. The last video v and its tags A are stored:
   (v, A) = RANDOM WALK(G, m, D)

2. A list of videos R that are related to video v is obtained from G. The YouTube API, will return at most 100 video-tags pairs for a video: (vi , Ai)

3. Generate up to n additional tags from related videos:
   E = RELATED TAGS(A, R, n)

4. Using the tag frequency distribution estimate F , tags with a frequency greater than or equal to t are removed:
   GT = REJECT FREQUENT TAGS(A ∪ E, F, t)

5. Return the selected video and a preprocessed version of the ground truth tag set:
   (v, PRE PROCESS(GT))

The usability of video CAPTCHA can be improved by adding tags from related videos to the ground truth tag set.

3.1 Related Videos

Currently YouTube provides a list of up to 100 related videos for each video. These Relatedness seems to involve some combina- tion of the similarity of tags, video co-views,the number of viewings a video has received and possibly other factors. Accepting tags from related videos will be help- ful for users and difficult for attackers for constructing or learning models for these social tagging patterns. For example, consider a video tagged with {zidane, football} that has a related video that is tagged with {zinadine,zidane,football}. In this approach we assume that zinadine is likely a valid tag for the given video, even if the person that uploaded the video did not tag it.

Tags from related videos are also helpfull in providing a form of social spell checking. For example, A video of the magician Criss Angel will have many related videos which can be either tagged as Chris Angel or Kris Angel. By including related tags, we both the tags will be valid and allows for common misspellings.

Generally, security impact of adding title words to its security makes no much change in the usabilty. So the tags from title words are not allowed.

3.2 Cosine Similarity of Tag Sets

The tags fom the related videos are sorted in decreasing cosine similarity functon. Selection of tags from those videos which has the most similar tag set to the challenge video are done using this method. Here a sort using the cosine sim- ilarity of the tags on related videos and the tags on the challenge video are performed. The cosine similarity between two vectors A and B can simply be expressed as follows:
$\mathbf{SIM(A,\ B) = \cos\ \theta = A.B\ /}\ (\|A\|.\|B\|)$

In our case, A and B are binary tag occurrences vectors (i.e., they only contain 1s and 0s) over the union of the tags in both videos. Therefore, the dot product simply reduces to the intersec- tion size of the two tag sets and the product of the magnitudes reduces to the square root of the number of tags in the first tag set times the square root of the number of tags in the second tag set.

Consider an example where At = {dog, puppy,funny} and Rt = {dog, puppy, cat}. The cosine similarity of two tag set is 0.6 which contains two tags in common.

3.3 Adding Related Tags

When the related videos are sorted in decreasing cosine similar- ity order, the tags from the related videos are introduced into the ground truth. The maximum number of characters allowed in a YouTube tag set is 120. Therefore, the tag set could theoretically contain up to 60 unique words (each word would have to be a single charac- ter). The maximum number of related videos which YouTube pro- vides is 100. Therefore, adding all of the related tags could poten- tially add up to 6000 new tags. We chose to limit the upper bound by adding up to n additional unique tags from the related videos (sorted in decreasing cosine similarity order). The following func- tion produces up to n related tags, given a challenge videos tags A, and a set of related videos R.

RELATED TAGS(A, R, n)

1. Create an empty set, Z $\leftarrow \emptyset$.
2. Sort related videos R in decreasing cosine similarity order of their tag sets relative to the tag set A.
3. For each related video r $\in$ R:

Add the new tags to Z

This technique will introduce up to n additional tags to the ground truth set. In the case where we have already generated n b related tags and the next related video contains more than b new, unique tags, we cannot add all of them without exceeding our upper bound of n tags. For example, consider the case in which we wish to generate 100 additional tags (n = 100) and we have already generated 99 tags. If the next related video has 4 new tags, we cannot include all of these in the new tag set, and so we randomly pick one to avoid bias.

3.4 Rejecting Frequent Tags

Security against frequency-based attacks (an attack where the three most frequent tags are always submitted) is maintained through the parameters F and t in the challenge generation function. F is a tag frequency distribution and t is a frequency rejection threshold. During challenge generation, after author-supplied tags and tags from related videos have been added to the ground-truth set, tags with a frequency greater than or equal to t in F are removed from the ground-truth tag set.

REJECT FREQUENT TAGS(S, F, t)

1. Initially, GT $\leftarrow$ S.
2. For each tag g $\in$ GT :

(a) If F (g) $\geq$ t, remove g from GT .

3. Return GT .

Chapter 4

GRADING FUNCTION

The generation of a video CAPTCHA returns a challenge video v and a set of ground truth tags GT. Given the challenge video v, the set of ground truth tags GT , the set of user response tags U , and binary variables s and l determine whether to perform stemming and to use inexact matching (l), we grade responses as follows:
GRADE(v, GT, U, s, l)

4.1 Preprocessing

A stop word list is a list of common words which are filtered prior to processing because they are unlikely to add additional information or context. For instance, it has been shown that over 50% of all words in a typical English passage can be constructed using a list of only 135 words . These words are classified as stop words. Prior to grading, all tags are preprocessed using the function PRE PROCESS, described here. The tags are converted to lower case and punctuation is stripped to remove the effects of inconsistent capitalization or punctuation. Additionally, only the first three tags are used in grading.

4.2 Expanding Tags through Word Stemming

To increase the likelihood of passing challenges, the user-supplied tags U may be expanded through word stemming using the STEM function. A stemmer is an algorithm for reducing inflected or derived words to their root. The root of a word is the word minus any inflectional endings, such as s, ing, etc. The Porter Stemmer2 is frequently used in information retrieval systems; it uses a deterministic set of rules to recover word roots. For example both the words dog or dogs can be provided as tags.

4.3 Allowing Inexact Matching

Many users may make spelling or typing mistakes when com- pleting a challenge. Therefore, we can also boost usability by performing inexact matching between user tags and ground truth. We utilized the well known string edit distance, or Levenshtein distance.

Chapter 5

ATTACK SIMULATION

The best way to attack a video CAPTCHA using tag frequency data alone is to submit the three tags which label the largest set of videos (i.e. where the union of the video sets is the largest). Increasing usability by extending the ground truth tag set will typically result in decreasing security because it allows an attacker a larger set of tags to match against.

The attack success rate may be reduced by pruning frequently occurring tags from the ground truth tag set, so that tags with an estimated frequency $\geq$ t are not accepted. However, an intelli- gent attacker would then select the three most frequent tags such that their estimated probabilities are slightly less than the pruning threshold.

Chapter 6

USER STUDIES

From the user study to test the usability and security of video CAPTCHA, to simulate a frequency-based attack against a sample of challenges, The first user study was used to explore possible grading functions and to determine the appropriate parameter values for the second user study.

In the first user study, participants were only instructed to tag the videos and their responses were not graded. However, participants in the second user study were told whether they had passed or failed the video CAPTCHAs. For both user studies and the frequency-based attack, success rates were observed over the space of usability and security parameters.

In the second user study, it was possible to increase human success rates from 70% (exact matching author tags) to 90% while maintaining an attack success rate of approximately 13%. These success rates are comparable to existing CAPTCHAs. In addition, we observed that different balances between security and usability could be achieved by modifying the generation and grading function parameters.

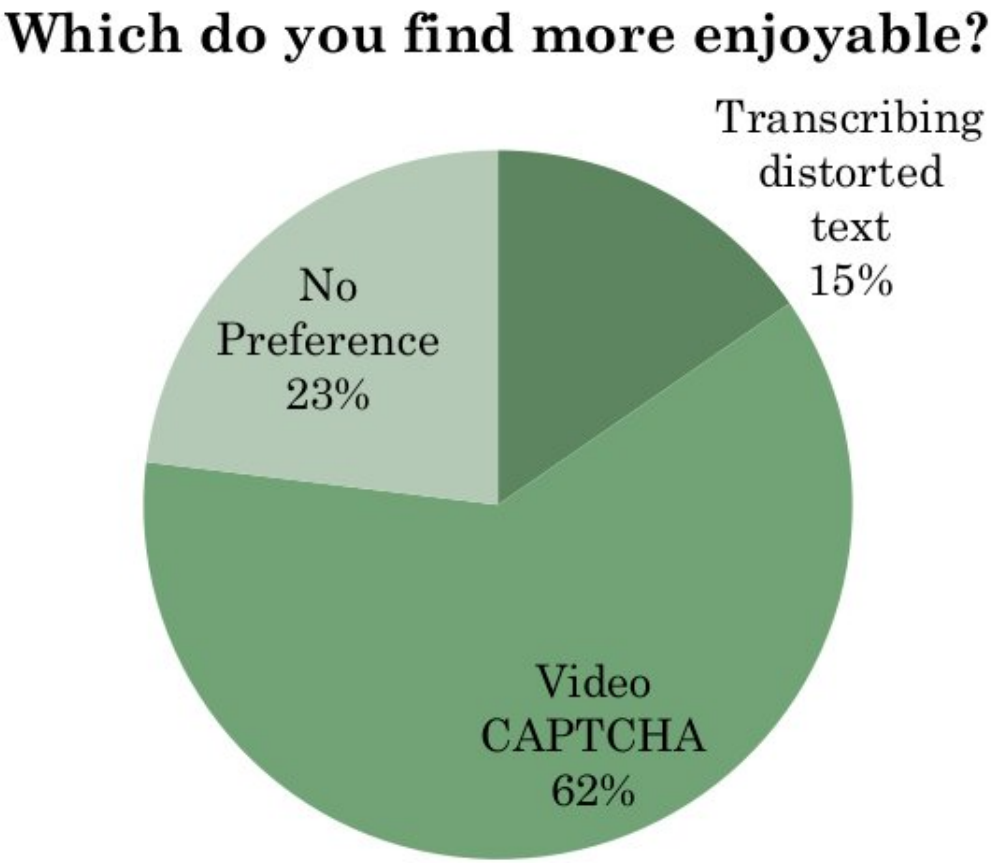

Figure 6.1: Study 1

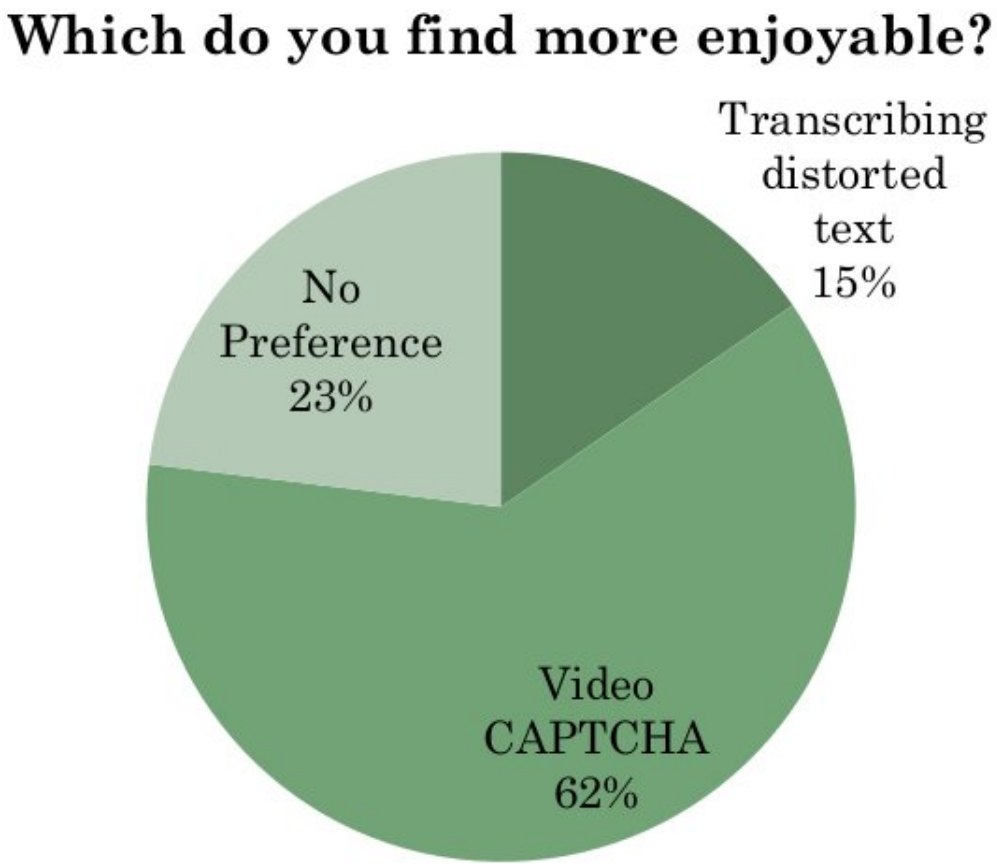

Figure 6.2: Study 2

Chapter 7

Conclusion and Future Works

7.1 Conclusion

Video Captcha is the first CAPTCHA that uses video understanding to distinguish between humans and machines. It has nearly all of the desirable properties outlined in the introduction: chal- lenges can be semi-automatically generated, graded automatically, the challenge design and data are publicly available, and challenge generation and grading may be parameterized in order to achieve a desired balance between usability and security. These results from user studies are en- couraging and suggest that video CAPTCHAs may provide a viable alternative to text-based CAPTCHAs.

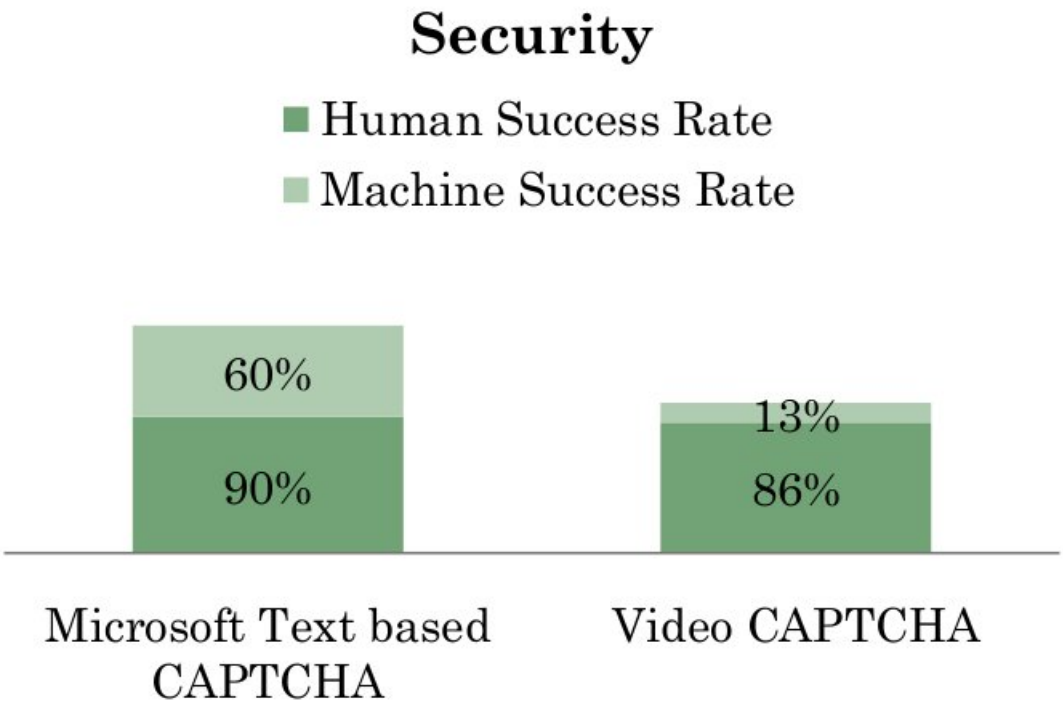

Figure 7.1: Result at a glance

7.2 Future Works

-Remove text segments appearing in the video from GT.

-Attack against audio analysis which will give an indication to content of video.

-In future the algorithm can try to match tags for all languages currently only English is accepted for matching.

References

[1] Kurt Alfred Kluever and Richard Zanibbi. Balancing Usability and Security in a Video CAPTCHA. In Proc. SOUPS July 2009.

[2] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the Worlds Largest User Generated Content Video System. In Proc. IMC 2007, ACM Press (2007), 114.

[3] K. A. Kluever. Evaluating the Usability and Security of a Video CAPTCHA. Masters thesis, Rochester Institute of Technology, 2008.