Defeating Anti Ad Blockers

By

Waheed ur Rehman

170431-MS(IS)-9-2016

Supervisor

Dr. Syed Taha Ali

Department of Electrical Engineering

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Information Security (MS IS)

In
School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(December 2019)

Approval

It is certified that the contents and form of the thesis entitled "Defeating Anti Ad Blockers" submitted by Waheed ur Rehman has been found satisfactory for the requirement of the degree.

.

Advisor: Dr. Syed Taha Ali

Signature: _____

Date: _____

Committee Member 1:

Dr. Wajahat Hussain

Signature: _____

Date: _____

Committee Member 2:

Dr. Arsalan Ahmad

Signature: _____

Date: _____

Committee Member 3:

Muhammad Imran Abeel

Signature: _____

Date: _____

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by Mr. Waheed ur Rehman, (Registration No 170431), of School of Electrical Engineering and Computer Science (SEECS) has been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: _____

Name of Supervisor: Dr. Syed Taha Ali

Date: _____

Signature (HOD): _____

Date: _____

Signature (Dean/Principal): _____

Date: _____

Dedication

To my family, my supervisor and my friends and who has always been supportive.

Certificate of Originality

I hereby declare that this work is my own and to be the best of my knowledge. It is no published and written by any other person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgment, is made in the thesis. Any research contributions made by other researchers whom I have worked at NUST or anywhere else, is explicitly acknowledged in the thesis.

I also declared that this thesis is the product of my work. Except for assistance from others in the development and layout of the plan or in style, presentation and linguistic is acknowledged.

Author Name: Waheed ur Rehman

Signature: _____

Acknowledgment

Thanks to the Almighty ALLAH for letting me pursue and fulfill this research work. I could not have achieved anything without HIS utmost support and countless blessings.

I am thankful to my parents who supported me throughout my educational career in all circumstances. I dedicate this work to my lovely parents, honorable teacher. They have always supported and encouraged me to do the best in all matters of life.

I am truly grateful to my supervisor Dr. Syed Taha Ali for his guidance, supervision, and direction to complete this task. He inspired me throughout my research by his skills and knowledge about the context; it inspired and motivated me, thank you so very much. I am obliged to all my respectable teachers for providing me their valuable time and considerations. I believe that this work would not have been possible without their guidance and expert suggestions.

I am also grateful to my committee members Dr. Wajahat Hussain for sharing with me his valuable experiences and knowledge of computer vision models, Mr Imran Abel for guiding me through his experience in, and Dr Arsalan Ahmad for their contribution and timely suggestions toward the successful completion of this thesis.

In spite of all the assistance provided by the supervisor, committee members, and others, I take the responsibility for any errors and omissions which may that may exist unconsciously.

Abstract

Perceptual ad blockers came with the claim that they finished the arm race between ad blockers and anti-ad blockers. They used the technique to visually identify advertisement instead of blocking domains. But adversarial attacks successfully defeated all the classifiers that were using to classify ads. But the Classifier that we purposed is not only detecting Ad Choice logo in advertisement images with 99% accuracy but also undefeated by all adversarial attacks that has been performed to bypass perceptual Ad blockers.

In our research we took dataset of 1000 advertisement images with adchoice logos and detect adchoice logo with different object detection models. We found cross correlation classifier with 99 percent detection accuracy. Next task was to evaluate this classifier against different adversarial attacks. We performed all adversarial attacks that "AdVersarial: Perceptual Ad Blocking meets Adversarial Machine Learning" paper performed to defeat perceptual ad blockers. We also used some other scale base noises but our classifier successfully detects adchoice logo in all noisy images.

Machine learning classifiers are easy to defeat using adversarial noise but simple computer vision algorithms are less prone to adversarial attack. So our work proves that it is not easy to defeat perceptual adblockers as our classifier can extract all images from webpage and by detecting adchoice logo it can classify it as an advertisement.

Table of Contents

List of Figures

List of Tables

List of Graphs

Introduction

This chapter will describe the introduction, history, context and working of ad blockers. Key words and terminologies will discuss to build an essential knowledge for all type of audience. It will include few interesting points of arm race between ad blockers and anti-ad blockers. In the context of these points it includes background, motivation and problem statement of this work. Objective and goals, intended audience, and scope of this work are also incorporated. This section is divided into following different sections.

i. Introduction of Ad blockers and Anti Ad blockers
ii. History
iii. Motivation
iv. Problem Statement
v. Goals and Objectives
vi. Intended audience
vii. Scope of the Study
viii. Organization of Dissertation

1.1 Introduction of Ad blockers and Anti Ad blockers

Ad blocker is a program that removes different type of online advertisement from the web page. It blocks banner ads, pop ups and other common form of online advertisement to make web surfing smooth [18]. There are different kinds of Ad blockers. Some are listed in the figure below.

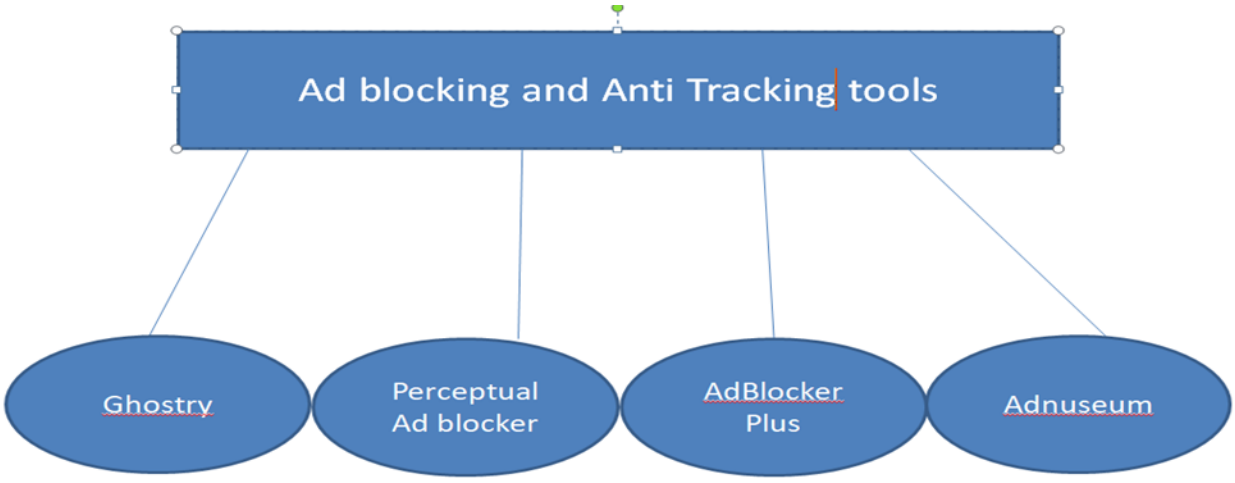


Figure 1: Different Ad Blockers

Recent type of Ad blockers not only block advertisement but also provide protection against trackers and malvertisement. In the figure above all ad blockers belongs to different categories. Ghostry is a tool that saves users from trackers [19]. Perceptual Ad blockers are the most advance type of Ad blocker that detects and block Ads on the base of visual classifiers [8]. AdBlockerPlus is a traditional Ad blocker that detects ads on the base of filter lists [1]. Adnuseum is a type of Ad blocker that not only blocks Ads but simulate clicks on those Ads to spoil the data base of advertisement companies [20].

At one site Ad blockers block Ads on another site Anti Ad blockers block Ad blockers. Anti-Ad blockers detect that Ad blocker is running. If they find active Ad blocker, than they perform action. Some restrict user to view web content unless user do not turn off Ad Blocker.

1.2 History

Ad blockers are browser extension, their main functionality is to block advertisement and trackers. Ad blockers provide users a customize web experience. Ad blockers use filter lists to block web content. Filter lists are set of rules that tell which element should be block [1]. Another type of ad blocker use computer vision algorithms to visually detect ads.

Adblock (0.1) was the original version, written for firefox in 2002. It was hiding ads instead of blocking it to download. Adblock version (0.5) in 2004 was the first version that was preventing ads to download instead of content hiding. Development of the Adblock started after the release of

Adblock 0.5s released. In January 2006, Wladimir Palant released the Adblockplus as a separate extension. It is the most popular extension so far.[2]. At Adblockplus website, the first official version is Adblock Plus 0.7.02,by Wladimir Palant released on 2006-06-08. The newest version of Adblock Plus was released in 2019-03-05 by Mario Konig. Adblock Plus also release Ad blocker called Sentinal. It detects ads using visual classifiers. It is artificial intelligence base Ad blocker. It works specifically for Facebook ads. It collects lot of advertisement from Facebook, train its classifier on the base of that data and detect ads. For better understanding lets categories Ad blokers into two different categories.

1.3 Filter List base Ad blockers

These are basically most famous type of adblockers. They use filter lists to block ads. Filter lists are pre-defined set of rules that tell which element or request to block. Filter lists are two types. Black list filter list block ad base on browser language. Second type is white list filter list. It allows ads that fit in the criteria of adblocker. It is the source of income actually. Companies pay to adblocker and whitelists its ads. [3]

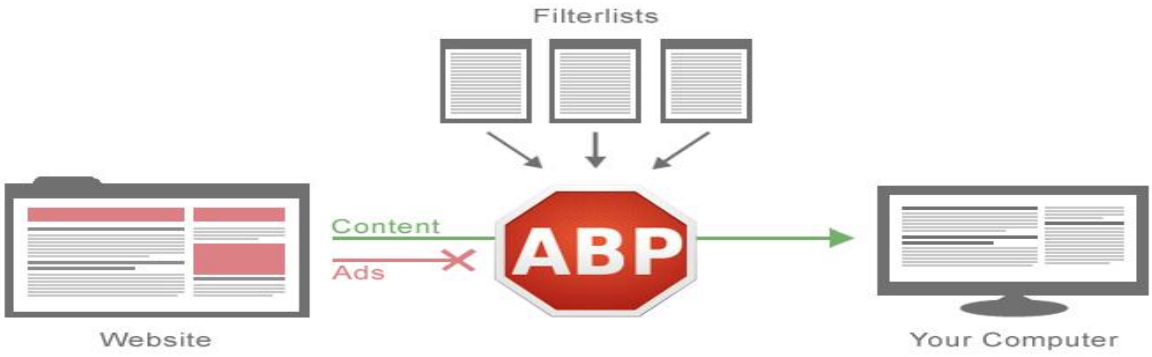


Figure 2: How Conventional Ad Blocker Works

1.4 Perceptual Ad blockers

It is another type of ad blocker. They don't use filter lists. They use some visual properties to block advertisement. For example Adchoice is a company. It uses its logo on each ads they send to user.

And it is standard in many countries. Perceptual ad blocker detects those features using computer vision or other AI base classifiers. On the base of those features it decides whether it is ad or not. In the figure below a perceptual ad blocker is detecting ads and highlighting the ads with "AD CHOICE IDENTIFIED".

1.5 Motivation

Many ads have tracking built in. Some contains malware [4]. Ad blockers play a vital role to block annoying ads and trackers. Many sites using anti ad blockers to forcefully send advertisement to users. Anti Ad-Blockers use different methods and scripts to detect and block Ad blockers. A study found that six point seven percent of top five thousand Alexa websites deploy anti-Ad blockers [5]. So they use different type of checks before and after rendering the page. They use persistent cookies, geographical location third party web request and different other things to track users.

Some researchers found that advertisement and analytical companies used different browser base vulnerabilities to send ads to users. They were using web sockets to bypass Ad blocking [6].

Privacy is the basic right of every one. At one side Ad blockers uses different techniques to secure the privacy of users and on another hand anti ad-blockers uses different techniques to bypass it. It is the arm race between them. If Ad blockers introduce a technique to block advertisement using filter lists than advertisement and analytical companies start detecting this behavior by using honeypots. Means if the property of related div or element in web DOM change it means ad blocker is present. Than perceptual ad blocker let the ads to download but it was highlighting the ads or hiding it by placing another div on it. This behavior was also detected by the anti adblockers. They bypass it by sending ads with different properties to or by adding noise in the images to deceive the adblockers.

1.6 Problem Statement

"Adversarial: Perceptual Ad Blocking meets Adversarial Machine Learning"[7]. In this Paper they perform different adversarial attacks on perceptual Ad blocker and by pass its Ad classification classifiers. So they raise problem for researchers to review their Ad blocking approach.

With the invention of perceptual Ad blocker, they challenged that they finished the arm race between Ad blockers and Anti Ad blockers. They were detecting Ads without blocking web

requests. Their approach was detecting Ads using visual classifiers instead removing those ads from web page. After it they hide it by overlaying it behind another div [8]. So they made it difficult for Anti Ad blockers to detect their approach. Because they let the advertisement URL to hit the advertisement server and Anti Ad blockers previously detecting that if advertisement request is block means Anti Ad blocker is working. In case of Perceptual Ad blockers they don't block web request. So Anti Ad blockers were unable to detect Ad blocker. Second most important part of Perceptual ad blocker was to over lay the Ad div. What Anti Ad blockers were doing that they were placing honey pot divs. As traditional Ad blocker removes those divs Anti Ad blockers know there is Ad blocker. But perceptual Ad blocker successfully bypassed this check. Now Anti Ad blockers have not any server or client side approach to detect Ad blockers. So what researcher did to detect this new approach, they also changed their approach. They used Adversarial attack. Adversarial attack is a type of attack in which attacker creates noisy images to fool the classifiers. So researchers took different classifiers of perceptual Ad blocker and performed attack on each classifier and fool it. So they proved that perceptual Ad blockers can be deceived. So they kept alive this arm race between Ad blockers and Anti Ad blockers.

1.7 Goal and Objectives

Goal of our research is to protect the privacy of web user from Anti ad blockers and trackers. This is only possible if we are one step ahead in this arm race. This is only possible if we are able to purpose a model that efficiently detects Ads features for maximum accuracy. Detection of ads is one aspect of our research. Second aspect is that, that model should not be deceived by the Anti-Ad blockers.

Objective of our research is to detect advertisement with maximum efficiency. Because it is the main objective of perceptual Ad blockers and perceptual Ad blockers are not as much mature to detect ads with 100 percent accuracy. Second and most important objective of our research is to secure Ad detecting classifiers from Anti Ad blockers. Because if Ad blocker is detectable by Anti Ad blockers than they can deceive it. So the classifier we will purpose is tested against all attacks that have been bypassed perceptual ad blocker.

Literature Review

Research community has acknowledged the arm race between ad blockers and Anti Ad blockers. Lot of work is in process from both sides. Researchers purpose a method against Ad detectors than advertisement and analytical companies came up with the hack of that method. Both side investing excessive amount of resources and energy to come up with the best solution. Advertisement is a billion dollar industry so this arm race will continue.

So in this chapter we will discuss the literature related that how advertisement and tracking companies tracking the users. Than we will see different type of solution that researcher's purpose against those trackers. We will also explain in detail that how researchers and advertisement companies bypass methods, used by Ad blockers.

2.1 Online Tracking

Steven Englehardt [9]. He presented his work in ACM CCS IN 2016. This research provided an extensive study of top 1 million web sites. They crawl one million web sites and make 15 types of measurements on each site. They use open source web privacy measurement tool "OpenWPM1". In their research they found different type of trackers and explain briefly how they use different techniques to track users. Their research reveals that news sites have most trackers. They found "Ghostry" as a best tool against these trackers. Tracker companies use different type of techniques to track users. Some techniques are written below.

(1) Persistence Cookies

(2) Local State in browser Plugins.

(3) By Audio Signals

(4) On the Base of geographical location

(5) Third party web requests.

(6) Battery API Fingerprinting.

This paper is important in this way that it grabs they extensive research about tracking, trackers and tracking methods. It provides good information to those who are working on the privacy of internet users. By keeping all those points in mind, better tools and techniques could be developed to protect user's privacy.

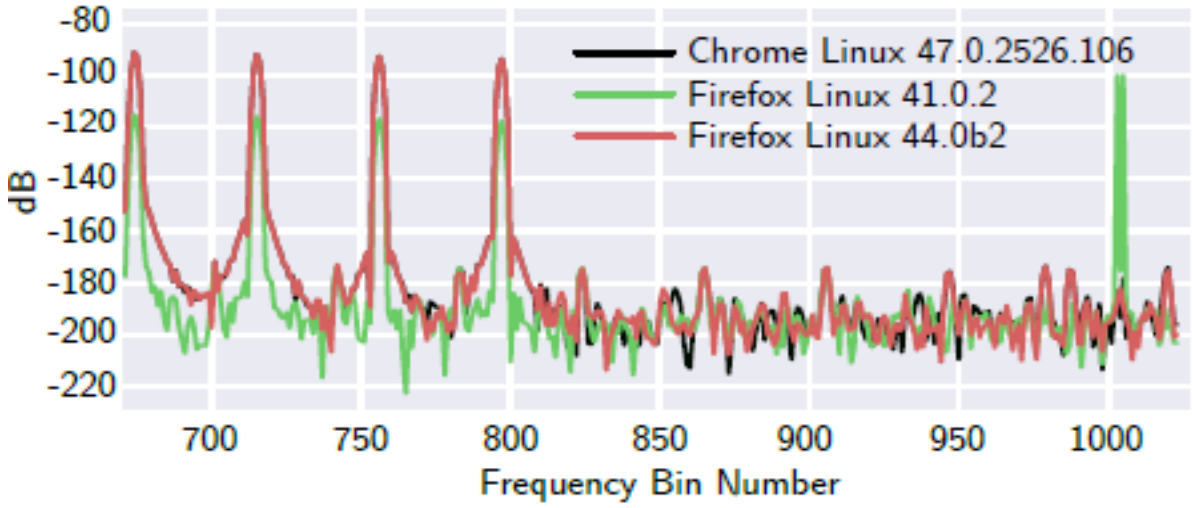


Figure 3: Trackers use audio fingerprinting

The picture above, show the fingerprint of different browsers on the base of audio signals.

2.2 Tracking companies and web socket vulnerability

Muhammad Ahmad Bashir [10]. They crawl 100,000 websites and explained the process of how Advertisement companies used Web Sockets to exploit Ad blockers, intrude user tracking, and send advertisement. A bug called web request bug was first reported in 2012. User reported unblocked ads in 2015 and in 2016. They bug patched in 2017. In the windows of five years analytical and tracking companies used this bug to send advertisement. This research briefly describe that what was the bug and how it is exploited.

Web Request API is used by browser extensions to modify, inspect and outgoing network requests. chrome.webRequest.onBeforeRequest callback is used by Ad blockers to block outgoing network requests to block ads. The bug did not let the websocket connection to trigger the chrome.webRequest.onBeforeRequest. Because this is the main API call used to block

advertisement and if this is not trigger than Ads will not be block. So many Advertisement and Analytical companies used this vulnerability to send ads.

2.3 Web Tracking, Mechanism, implementation and defense

This is extensive survey that deals with different methods used by analytical and advertisement companies. They elaborated the purpose of trackers, implication and possible defense against those trackers. They identified five different mechanism that tracking companies use to track users.

- Session only
- storage base
- cache base
- Finger printing
- Other tracking mechanisms

In session only, there are different technologies used to track session.

I. Session identifiers stored in hidden fields
II. Explicit web-form authentication
III. window.name DOM property

In storage based there are different technologies used to track user.

I. HTTP cookies
II. Flash cookies
III. Flash Local Connection object
IV. Internet Explorer user Data storage

In cache based there are different technologies used to track user.

I. Web cache
II. DNS lookups
III. Operational caches

In finger printing base there are different technologies used to track user.

I. Network and location
II. Device information
III. Operating System instance fingerprinting

IV. Browser version finding
V. Browser instance fingerprinting using web browsing history
VI. Other browser instance finding methods

Other tracking mechanism

I. Headers attached to outgoing HTTP requests
II. Clickjacking
III. Ever cookies (super cookies)

In this paper they also list different tools and techniques to avoid these trackers.

I. Microsoft Tracking Protection List
II. Privacy Badger
III. Adblock Plus
IV. Zend2.com, KPROXY, etc.
V. Tor
VI. Vanish

They also purposed lot of other tool and techniques to evade those trackers. I listed few of them.

2.4 Case Study of AdNauseam

A very nice approach to pollute the data bases of tracking and advertisement companies by simulate clicking on ads and blocking their incoming response.

The main purpose of this software is to infect the data that tracking and analytical companies gathered. Polluted database can lead the companies towards extreme financial loss. AdNauseam also provide protection against malware and malvertising. Technique use to send malware using web advertisement. The main task of advertisement and analytical companies is Aggregation and profiling via clicks on advertisements. The second purpose of those companies is to send the advertisement on the base of analytics that they have gathered in previous step. In the case of AdNauseam, this generates false clicks on the advertisement, separate the Ads in a wallet, stop malvertisement etc.

2.5 Anti Ad-blockers

So far we describe different type of Ad blockers and their working, different ways that advertisement agencies use to track user data. Now this chapter will describe in detail that how advertisement companies detect Ad blockers. Scripts, techniques or other mechanism that is used to detect or block Ad blockers is fall in Anti Ad blockers category. In 2017 a paper published that work explicitly to detect Anti Ad blockers [12]. They used Alexa top 10 million websites. They found six hundred eighty six web sites that making visible changes in their web pages against ad blockers. Most of the web sites using third party scripts to detect ad blockers. They describe in detail that how many websites are using Anti Ad blockers and what techniques they are using. To detect Anti-Ad blockers automatically, these used machine learning approach.

Ad blockers don't let the Ad to download and display. So what Anti Ad blockers do? Its work base on three properties first is time out, condition check and response. When html page start loading, Ad blocker do not let the Ad to display on the page. As Ad blocker start working after some delay to Anti Ad blocker has to wait for some time so Anti Ad blockers set time out. As time out expire the condition check is execute to check the presence and absence of advertisement. It is check by checking the visibility, height and width of the ad frame. After checking the absence or presence of ad, the response step is executed. There are different types of response that can be displayed. Some publisher just prompt that please turn of the Ad blocker while some aggressively said to turn of ad blocker till than they don't let the user to view the content of the page. Many famous websites are using these techniques. The aggressive behavior of Anti ad blockers is shown in the picture below.

Figure 4: Anti-Ad Blockers Behavior

Some researchers found a total of 7264 trackers. In which 524 trackers were unique. On Alexa top 500 websites. They found that few trackers cover a large fraction [13]. Google, Facebook, Twitter, and AdNexus track users across a majority of top 1 million websites. This researcher shows that google alone track 80 percent users of top one million websites [9] . Rafique et al performed manual analysis on top 1000 free live video streaming websites. He found that 163 websites were using Anti-Ad blocker [14].

2.6 Detecting Anti Ad-blockers

As Anti ad blockers prompt a clear visible change in page so researchers developed a system to automatically detect Anti ad blockers [12]. Anti ad blocker detector, first view page with ad blocker plus and without ad blocker. In 2nd step it extracts the features or DOM properties from the page. In 3rd step it compares DOM properties of both the page that it rendered using Ad blocker plus or without Ad blocker. It compare following properties. Website name, nodes added, div, h1,h2,h3, iframe, visibility change, height change and text change. In next step it provides these extracted features to its machine learning base trained model. That model tells that whether there is an Anti ad blocker exists or not.

```
<div class="banner_ads"> </div>
<script>
var ads = document.getElementsByClassName('
    banner_ads'),
ad  = ads[ads.length - 1];
if (!ad || ad.innerHTML.length == 0 || ad.
    clientHeight === 0)
  alert("We've detected an ad blocker running on
      your browser." + ...);
}
</script>
```

Figure 5: Anti-Ad Blocker Script

2.7 The Future of Ad Blocking

Because of arm race between Ad blockers and advertisers, researchers trying continuously to develop different techniques to block advertisement without get caught by Anti Ad blockers. Perceptual Ad blocker [8] is a new technique to block Ads. In this technique developers used computer vision algorithms and some AI base algorithm to visually detect the Ad blockers. In this approach Ad classifier classify Ads on run time without blocking requests. It led the Ad to download and after it, it highlights or hides the Ads under another div. So it is not a traditional method to block Ad. So traditional Anti ad blockers are unable to detect this Ad blocker. On the base of their model researchers of the paper "Future of the Ad blocking: An Analytical Frame Work and New Techniques" [8] claimed that their model is the end game of this arm race. Figure below explain that how perceptual Ad blockers detect and highlight the Ad.

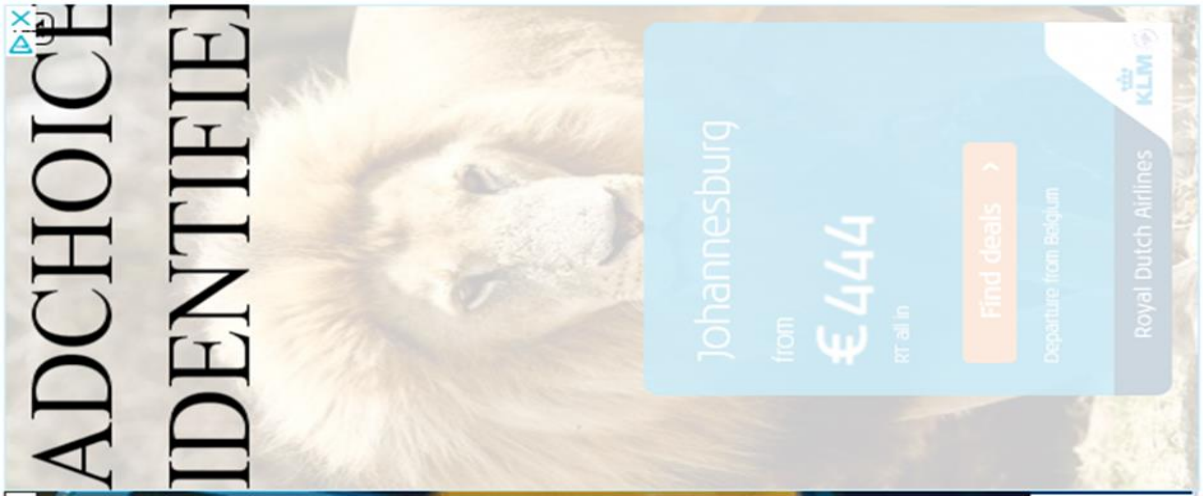


Figure 6: Perceptual Ad Blocker Highlighting Ads

In the image below, Perceptual Ad blocker is highlighting Ads. Facebook Ad detection is based on machine learning base classifiers.

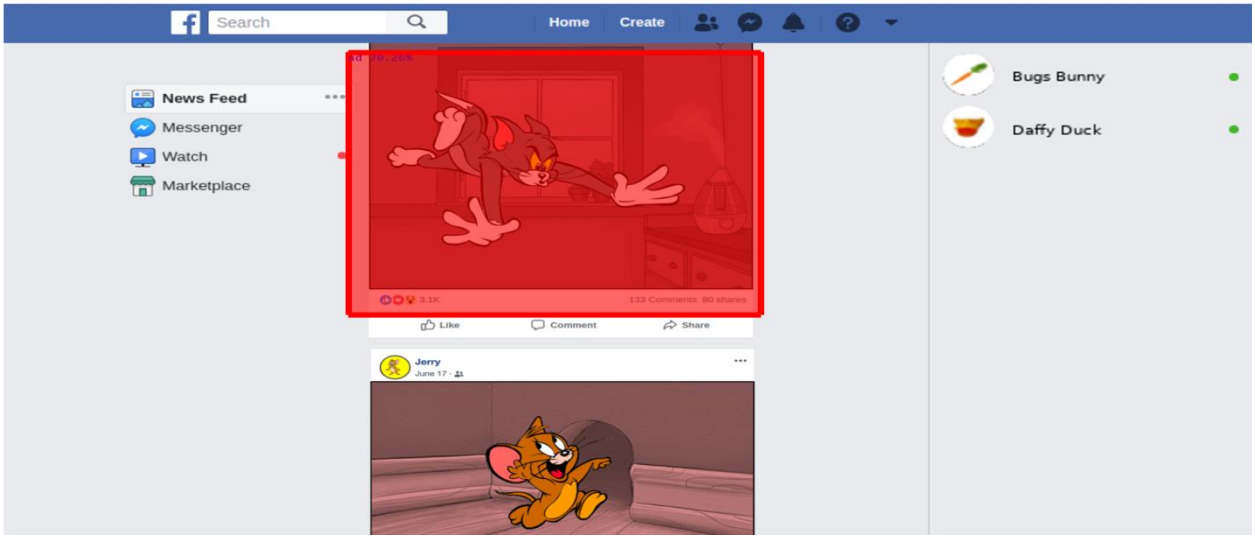


Figure 7: Perceptual Ad Blocker detecting Facebook Ads

In USA Federal Trade Commission in taking care of business practices. It takes action against unfair business practices. Agency has adopted the position in guidelines and enforcement actions that paid advertisements must be clearly recognizable to consumers. European Union has also taken steps to systemize online advertisement. Therefor different type of techniques is used to distinguish Ads from original content. Some advertiser place Advertisement key word on Ad. Some post different type of logos. AdChoices is an industry standard for disclosure of online behavioral advertising. They place their logo on all advertisement.

Figure 8: Ad Choice Logo

So perceptual Ad blocker mostly use those propertise to detect Ads. Second technique that Perceptual Ad blockers use is machine learning. It trained its machine learning model on the base of advertisement images provide by the user. It collect big data set of advertisement images. Process data set and detect Ads. This method is very useful to detect facebook ads.

2.8 Perceptual Ad blocking Evaluation

	Total Number of Ads	Detected by percptual Ad blocker
Facebook	**50**	**50**
Adchoice	**212**	**207**

Table 1: Perceptual Ad Blocker Evaluation

2.9 Adversarial Attacks on Perceptual Ad Blockers

Perceptual Ad blockers is a new method to detect online advertisement base on optical content. Perceptual ad blockers do not use filter lists. They are not using traditional methods to block advertisement so they claimed that they are superior in arm race betweenad blockers and anti ad blockers. But researcher Florian Tramèr [7] performed seven different type of attacks on perceptual Ad blockers by creating perturbed ads, logos and native web content and this researcher successfully mislead perceptual Ad blocker with 100 percent success. They not only bypass Ad blocker detection but also proved that high priviliged level of Adblocker can lead to other attacks i.e DOS attack. They first explained the general architecture of the existing approaches like Perceptual Ad Blocker, Sentinel (perceptual Ad blocker by Ad blocker Plus) and Ad Block Plus. They did it to provide main analysis of design and working of Ad blockers[15].

Ad Highlighter[16] and Percival [17]. They took different classifiers from different Adblockers and performed attack on those classifiers.

AdBlocker	Classifier1	Classifier2	Classifier3
Ad-Highlighter	Perceptual Hahing	OCR	
Percival	Ad-classification Neural Network	SIFT	

Table 2: Perceptual Ad Blockers and Their Classifiers

They also demonstrate that adversarial attacks and on the base of these attacks they perform different exploits, In those attacks a melicious user can utilize the high privilege of Ad blocker to block another user content.

So they proved that perceptual Ad blockers did not finished the arm race but give it an another direction. So now this arm race convert from filters lists towards adversarial attacks.

Design and Methodology

In this chapter, we will present basic design of our solution to detect Adchoice logo with maximum detection rate. The method of our solution is that, we will take images and detect Ad choice logo in that image. The flow of our solution is first extract images from a webpage and detect ad choice logo from those images. Where Ad Choice logo found means there is an Ad. To impliment this scenario we experiment different detection algorithm and choose best one. Below is the list of algorithms that we used for initial testing.

I. Chamfer Matching //detection of bottle
II. Convolution template matching ///detection of box in dinasour image
III. Cross Correlation
IV. Point Feature Matching using speeded up robust features ///

3.1 Chamfer Matching

It is the most reliable, simple and accurate method for segmented images. This classifier is very powerful against missing data, poor segmentation and low resolution [21]. We get its code from git hub repository and run it [22]. It successfully run and show following results.

Figure 9: Chamfer Matching Object Detection

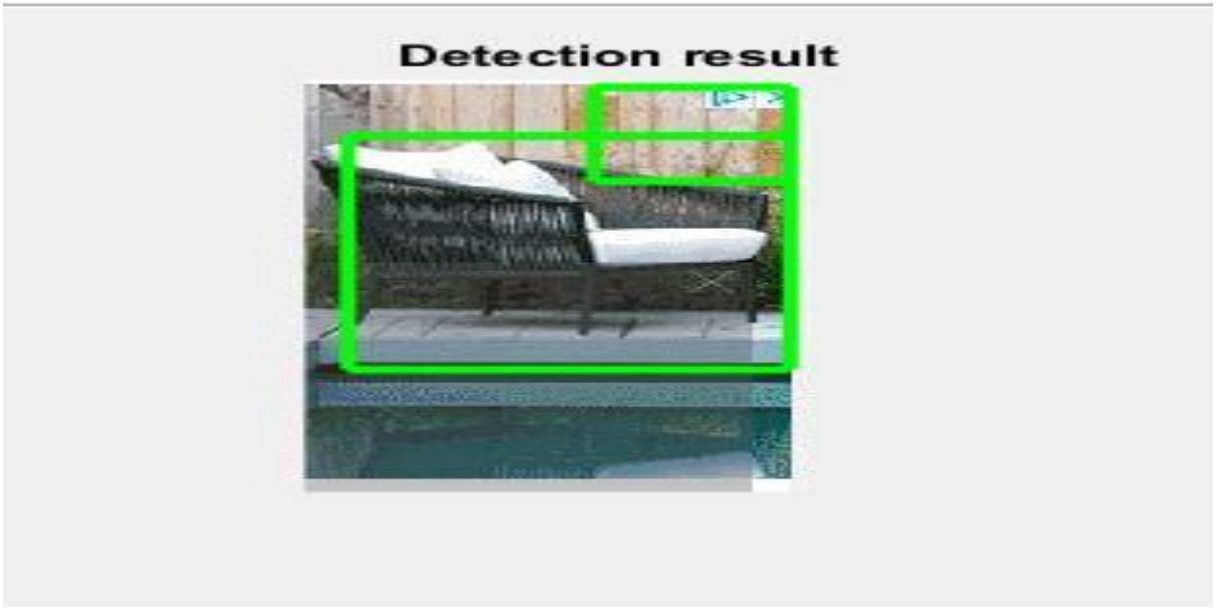


Figure 10: Chamfer Matching Ad Choice Logo Detection

3.2 Convolution template matching

This template matching classifier use a simple but fast correlation based template matching algorithm. Convolution technique is used with the correlation coefficient calculation. Its main focus is on controlling the boundary and selecting region of interest on a given image. The main benefit of using this classifier is that it increases the template matching speed by reducing the computational time [23].

We downloaded the code from github repository [24]. And successfully run it. It shows following results.

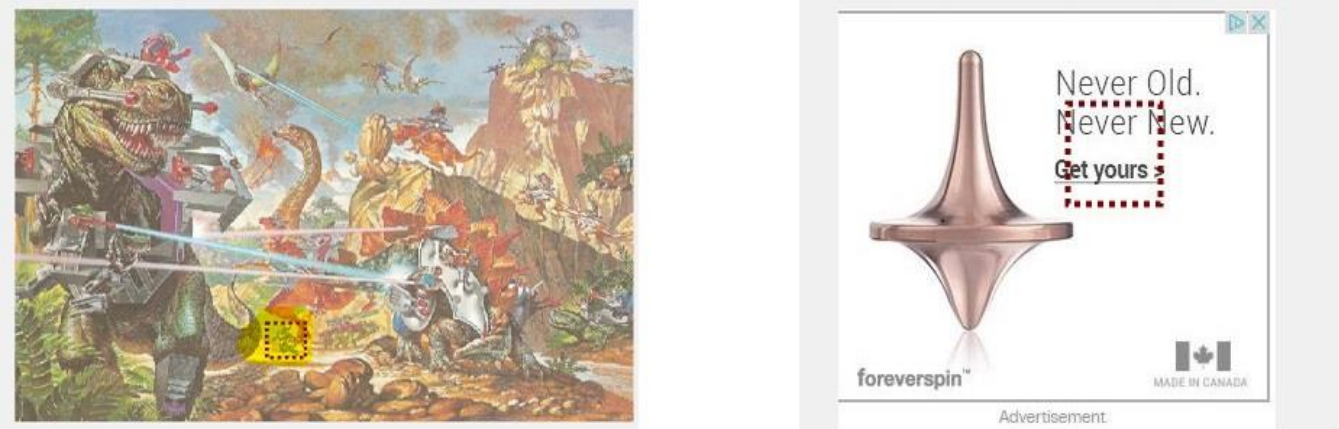


Figure 11: Convolution Template Matching

3.3 Point Feature Matching using speeded up robust features (SURF)

The standard version of SURF is several times faster than SIFT9 (another template matching algorithm) and its authors claimed that it is more robust against different image transformations than SIFT [25]. We got this code from matlab directory [26].

Figure 12: Object Detection Using SURF feature Matching

3.4 Cross Correlation

For experiments I got code from matlab repository [27]. Cross correlation is the basic statistical approach to image registration. It is used for pattern recognition and it is also used for template matching. Template is the part of the image or subimage of the given image. The objective is to find the template image from the given image. Crosscorrelation classifier gives the measure of the degree of similarity between an image and template [28]. Result of the code is in the figure below.

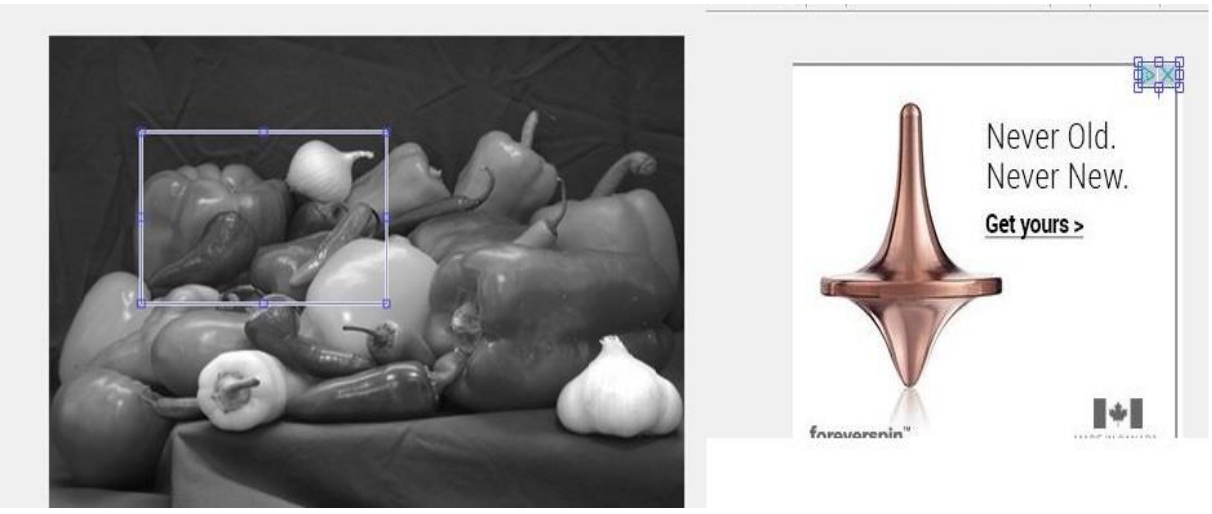
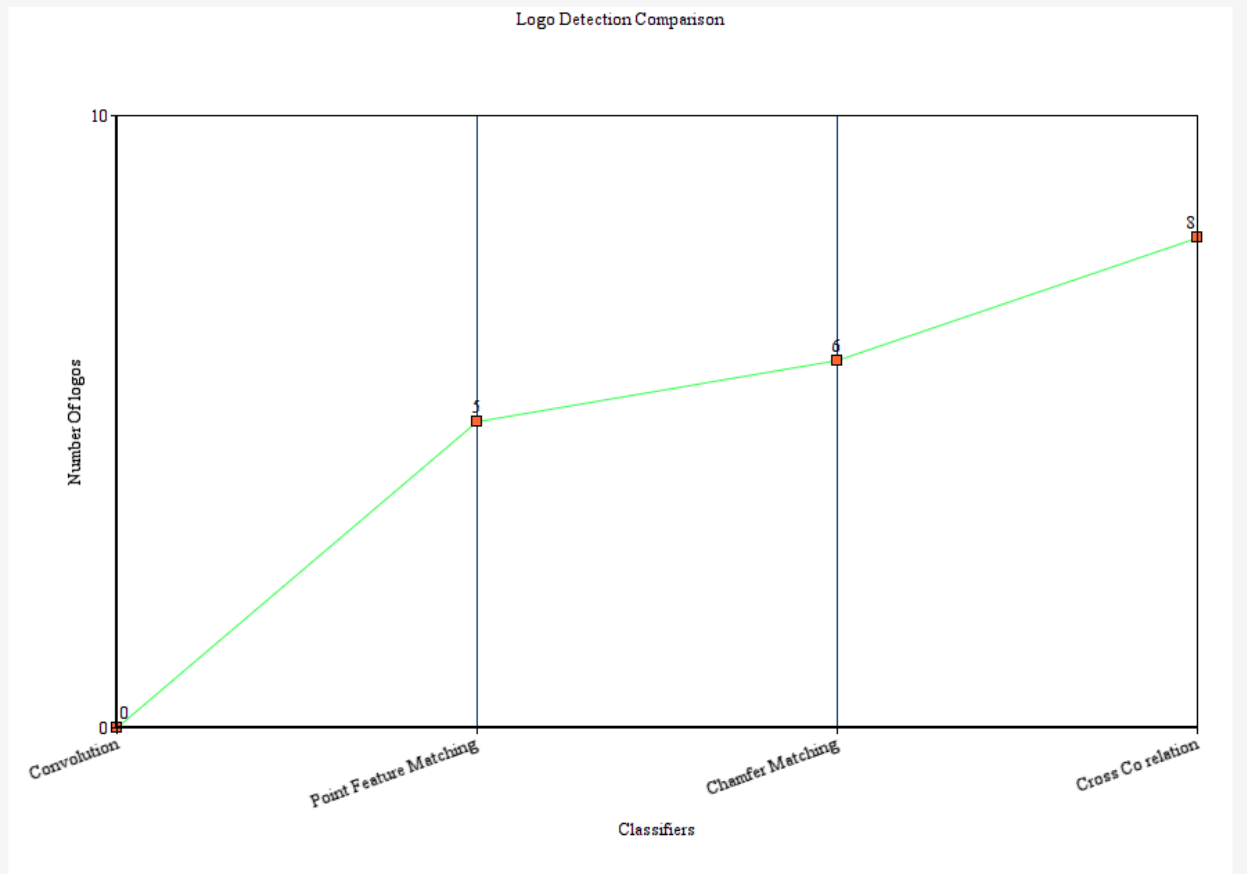


Figure 13: Cross Correlation Template Matching

Classifier	Total Number of Ads	Adchoice logoDetected	comment
Cross Correlation	10	8	
Cross Correlation Modified	1000	993	
Point Feature Matching	10	5	
Convolution template matching	10	0	
Chamfer Matching	10	6	

Table 3 Different Template Matching Classifiers Comparison

Graph to show classifiers detection value:



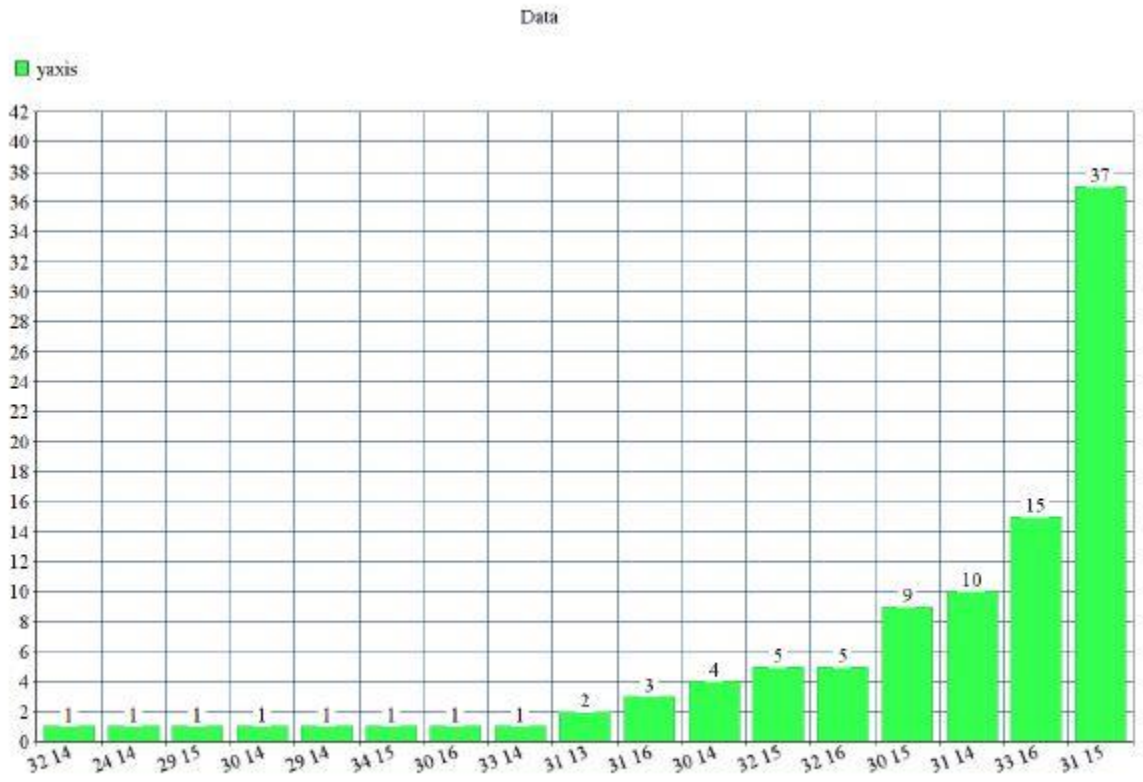
Graph 1 Different Classifier comparison on 10 images

3.5 Ad Choice logo size detection

Data is present into Adversarial → Experimental images → adchoice images → croped logos.

So, for getting maximum detection result we made different experiments. In all advertisement images logo was locating at the upper right corner. In the data set of 1000 images all the logos were found at this position. We took 98 logos and find the size in pixels. Where we found 37 logos of that size of 31,15. We found 15 logos with the size of 31,16. Ten logos of the size of 31,14. Nine logos

with the size of 30,15. 5 logos has size of 32,16, next five has size of 32,15. 4 logos has size of 30,14, three logos 31,16, two logos 31,13. Other 8 logos all have different size. The graph of the size of logos is shown below.



Graph 2 AdChoice Logo Size in Pixels

Implementation of Adversarial Attacks

This chapter will ellaborate different adversarial attacks on different classifiers that has already used to detect ads [7]. There are three classifiers, and we performed six attacks.

1. Average Hashing
2. Sift
3. Yolov3

4.1 Average Hashing

The average image hashing algorithm binaries the pixels based on whether the pixel is brighter or darker than the average gray scale value [37]. Florian Tramèr [7] attack this classifier and bypass it detection mechanism. We performed different experiment by using this classifier. 1st run the classifier to view the template matching results. After it created a noisy image. After it again run average hashing template matching classifier. This time classifier was not able to detect the image.

We also performed false positive attack at that classifier. Created a random image and run classifier again at that image. This time classifier classifying randomly created image as Ad logo.

4.2 Average Hash Matching Result

python –m phash.model ../data/ad_logos/ "../data/web/www/cnn.com/adchoice

Following command is use to run the classifier. In the result, atlas-br.png image showing the maximum similarity with the image aol.png. Similarity rate is 1.00.

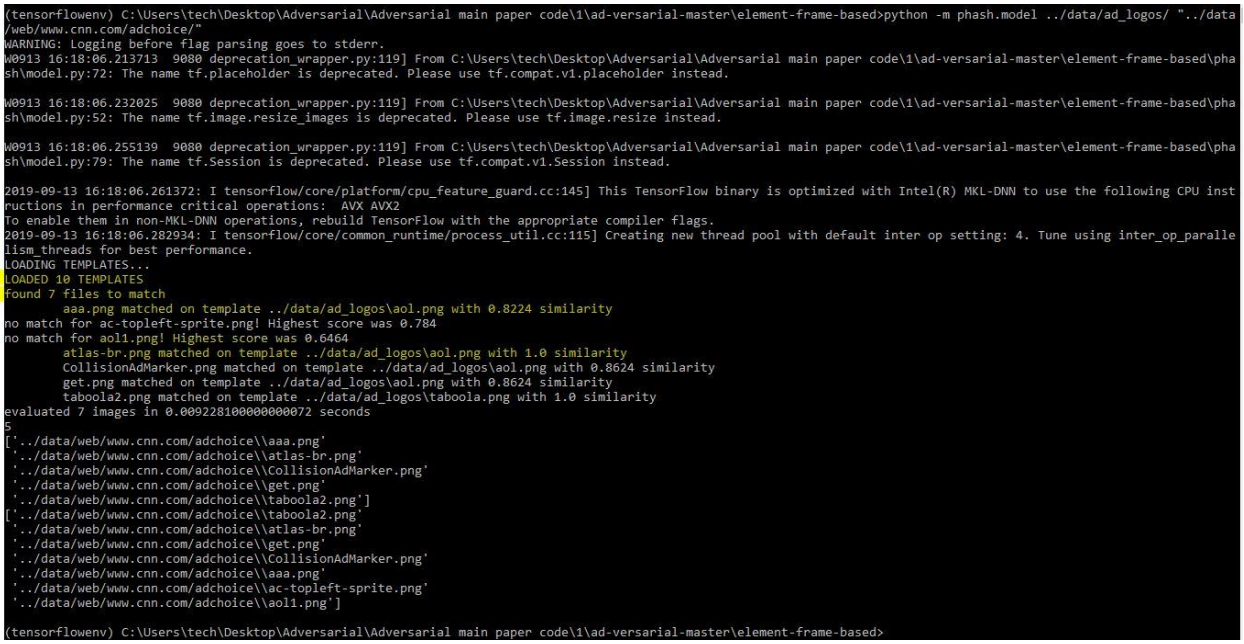


Figure 14: Average hashing logo Matching Result

4.2.1 Creating Noisy Image

After successfully running the classifier next step is to create noisy image to bypass the classifier. To generate the noisy image following command is used.

python –m phash.attack ../data/ad_logos/ aol.png temp ../data/ad_logos/

The image below is the noisy image. It is generated using phash attack. In next step we will evaluate that whether average hashing classifier will be able to detect it or not.

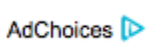


Figure 15: Adversarial Image for Average Hashing Template Matching

In The image below it shows some values after creating noisy image.

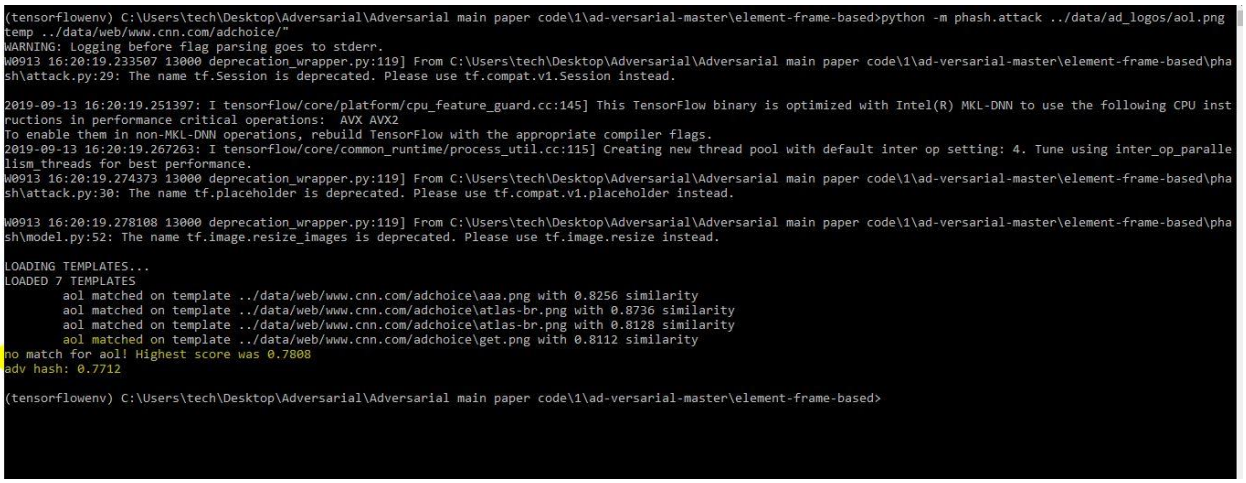


Figure 16: Noisy Image Creation Process

4.2.2 1st Attack

After successfully created the noisy image next step is to evaluate it against average hashing template matching classifier. Than replaced that image with the aol.png. Run the classifier again. In the image below it is clearly written that no match for the aol.png. Attack is successful. And noisy image successfully bypasses the classifier.

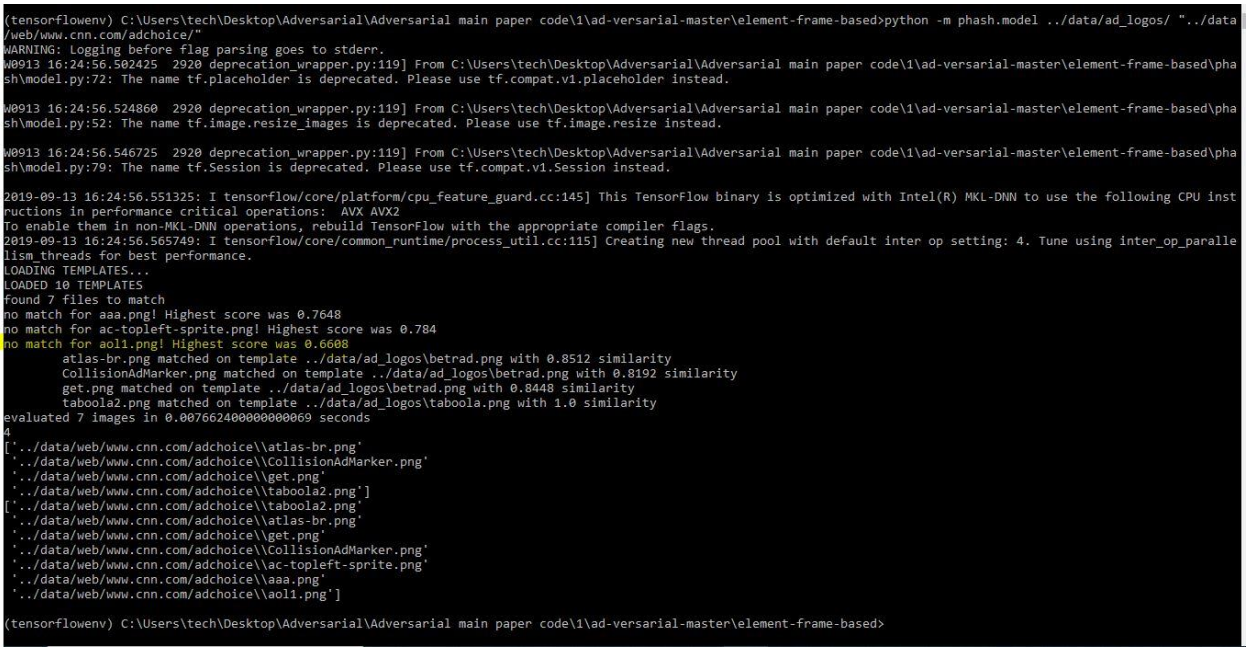


Figure 17: Result of Attack at Average Hashing

4.2.3 Creating False Positive Image

This attack is to create false positive image and compute the classifier result against that image. To generate the false positive image following command is run.

python -m phash.attack_false_positive ../data/ad_logos/aol.png temp

4.2.4 2nd Attack

After successfully created the false positive image next step is to evaluate it against average hashing template matching classifier. Run the classifier again. The result can be seen that classifier detected it with 90 percent accuracy. Hence false positive attack is successful.

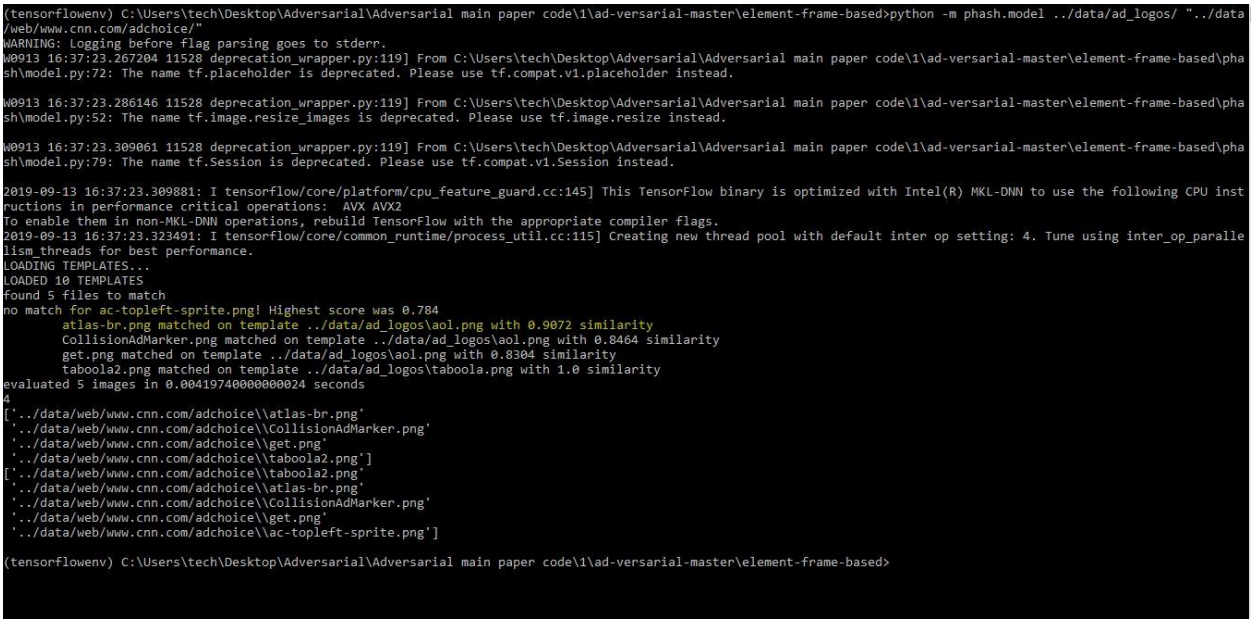


Figure 18: False Positive Attack Result

4.3 Scale Invariant Feature Transform (SIFT)

SIFT make a data base in which it stores key point of objects that it extract from the reference image. Than to find the same image in another image it uses Euclidean distance of their feature vector. It individually compares each feature from the new image to the data base that it already set. From the full set of matches, subsets of key points that agree on the object and its location, scale, and orientation in the new image are identified to filter out good matches [29].

4.3.1 Logo Matching Using Sift

To perform adversarial attack first there is need to run sift classifier to see actual template matching score. Researcher of Perceptual Ad Blocking meets Adversarial Machine Learning [7] put their adversarial attack code at git hub repository [30]. To run that code on new version of windows there is need to install older version of Open CV. After all configuration code successfully run. Code took images from one directory and compares it to images from another directory. By placing the image shown below to both directories. Classifier show the hundred percent match.

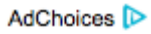


Figure 19: Original Logo Atlas-Br

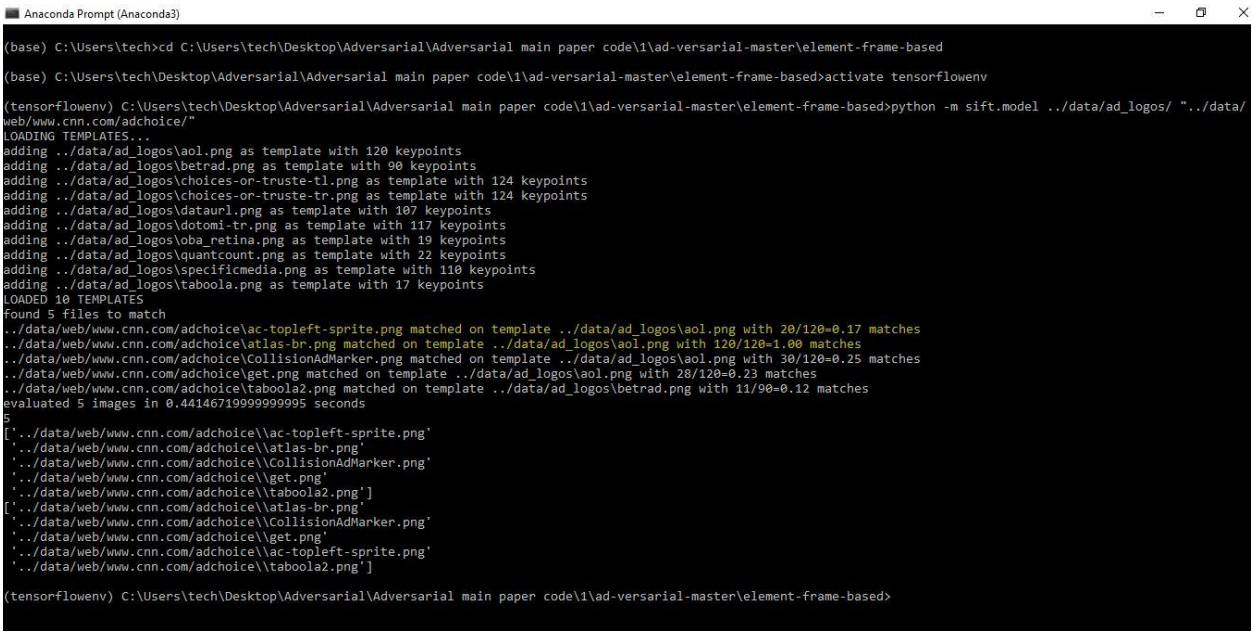


Figure 20: Result of SIFT Matching

4.3.2 Creating Noisy Image

Now we have result of exact matching. Next step is to create adversarial image to fool classifier.

python -m phash.attack ../data/ad_logos/aol.png temp ../data/ad_logos/ [30]. This took more than 6 hours on intel core i7 with 16 GB memory to create a noisy image. After creating noisy image put the image into the directory and run SIFT classifier again to show the result.

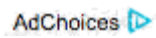


Figure 21: SIFT Noisy Image

4.3.3 Attack`

After creating noisy image, next step is to perform attack and compare the result before and after adding noise. For adversarial attack it took 6 hour to create adversarial image using core i7 7th generation 2.6 ghz processor and 16 GB of RAM. From all the images in the data directory of logos I got succeed to make only one logo fully adversarial. And it made visible changes in adversarial image.

For creating adversarial image put the image into the directory of logos and run the command below.

python -m sift.model ../data/ad_logos/ "../data/web/www.cnn.com/adchoice/" [30].

This command perform SIFT template matching. It will compare the logos place in the folder of ad_logos and compare it to the logos in adchoice

Before adding noise, template matching score was 1.00. After adding noise the score was 0.14. Hence prove that attack totally bypass SIFT classifier.

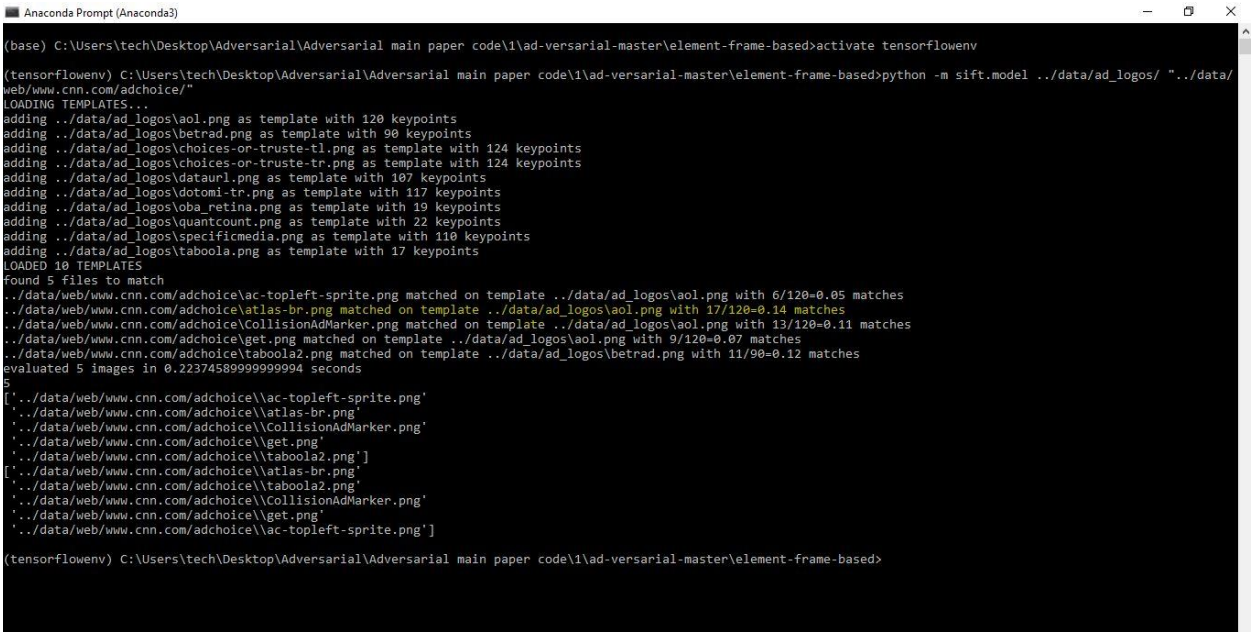


Figure 22: SIFT Attack Result

4.4 Yolo v3

This machine learning model use totally different technique to detect objects. This technique uses single neural network to the full image. First it divides the image into regions, than predict bounding boxes and probabilities for each region. After it those bounding boxes are weighted by the predicted probabilities. At test time it check the whole image and result are informed by global context in the image. This is 1000x time faster than R-CNN [32]. It uses data set of DarkNet to train the model. [33]. DarkNet is an open source framework of neural network. It is easy to use and fast. It is important because it supports both CPU and GPU.

4.4.1 Ad Detection Using YOLOV3

Pre compiled data and model can be found at github repository [34]. Download the model, code and data set from this repository [34]. And integrate it. After it first step was to detect Ad from the web page. So the command below is use to detect Ads.

python classify.py --input_dir=../data/page_based/web/test/ --output_dir=temp

Trained YOLOV3 classifier detects ads and save it to temp folder after highlighting in red. I placed two images below. Img 1 is original image and it has advertisement in it. After running Yolo v3 it

will prove that whether Yolov3 detects Ad from the image or not. If it detects an image than the next step will be add noise in the image and detect it again. We performed different attack at this level and confirm the result by detecting again by Yolov3.

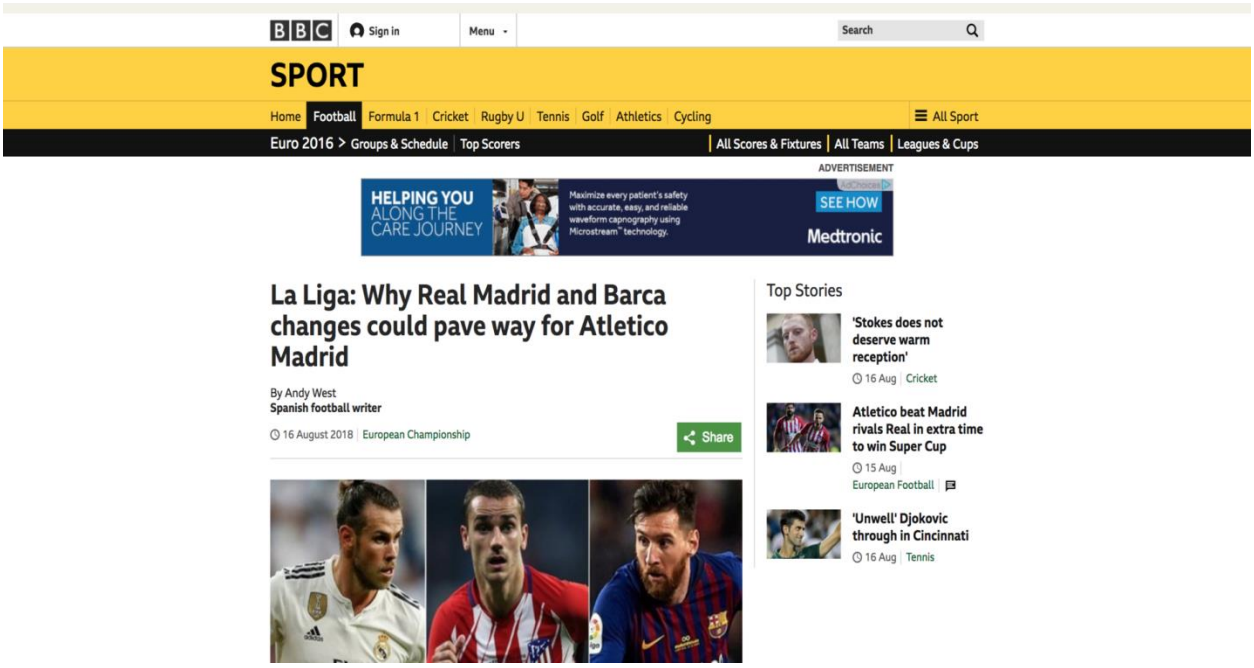


Figure 23: Original Image With Ad

Red highlighter shows that Ad detected successfully. After performing this experiment on many images resides in the data directory we confirmed that Yolov3 v3 result is good to detect Ads. As an example I only put the result of only one image here.

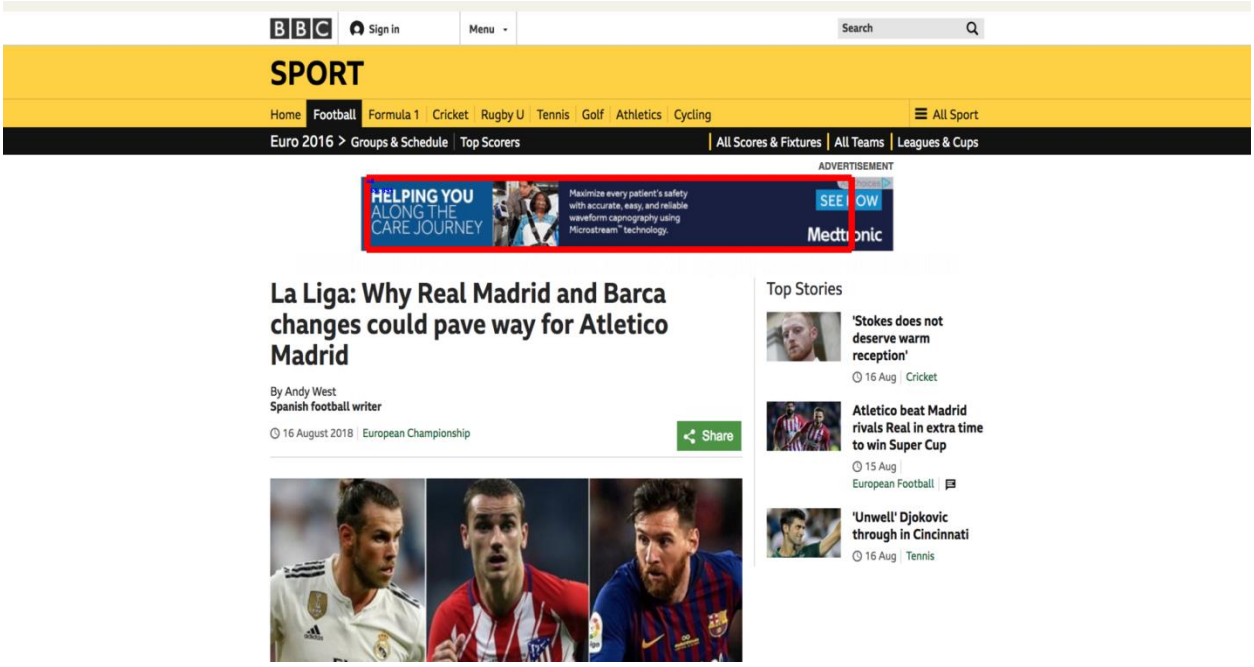


Figure 24: YoloV3 Ad Detection

BBc evade original image from Adversarial > adversarial main paper code > 1 >ad-adversarial-master >data>page_based>bbc>images

4.4.2 Attacks

- Evasion Attack: The publisher perturbs bottom of ad frame to evade ad-blocking.
- Evasion Attack: The ad-network perturbs ads using a universal perturbation to evade ad-blocking
- Detection Attack*:* Publisher perturbs the page header to create a false ad prediction

4.4.3 Attack 1st bbc_evade

This attack creates a folder in output directory with the name of bbc_evade and start iterations 0-19, 10-19, 20-19…..40-19. Than it add noise in those images and start classifying. I.e. in 0th iteration it will manually detect all the ads. Than in 10th iteration it will start evading. In next iteration evasion percentage start increasing.

I took the upper image from the last iteration means from 40th iteration and pass it through the classifier. But it is not detected by classifier.

It is detected by classifier

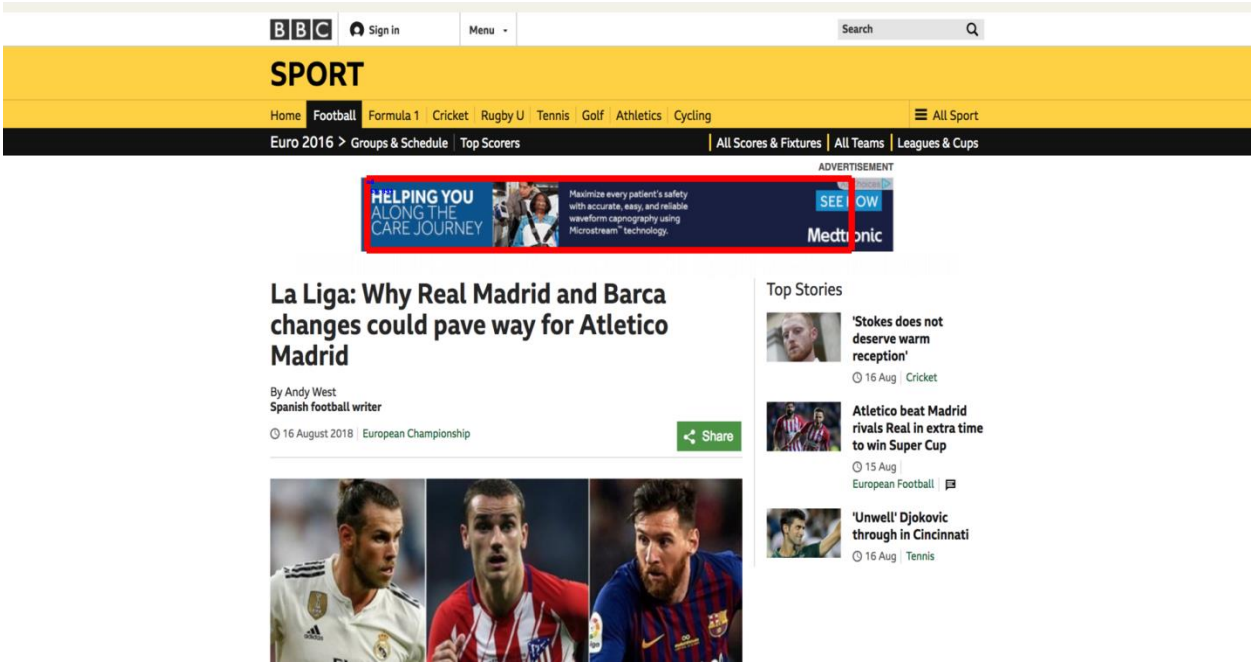


Figure 25: Before Adding Adversarial Noise

The following command use to create noisy images.

python -m attacks.bbc_evade --input_dir=../data/page_based/bbc/

Image below is the noisy image and I collected it from the 40th iteration. After it I run Yolov3 again by putting noisy image and the original image. And 2 images from the iteration 1 and 10. All other images were detected but the image I collected from last iteration Yolov3 was not able to detect Ad in it.

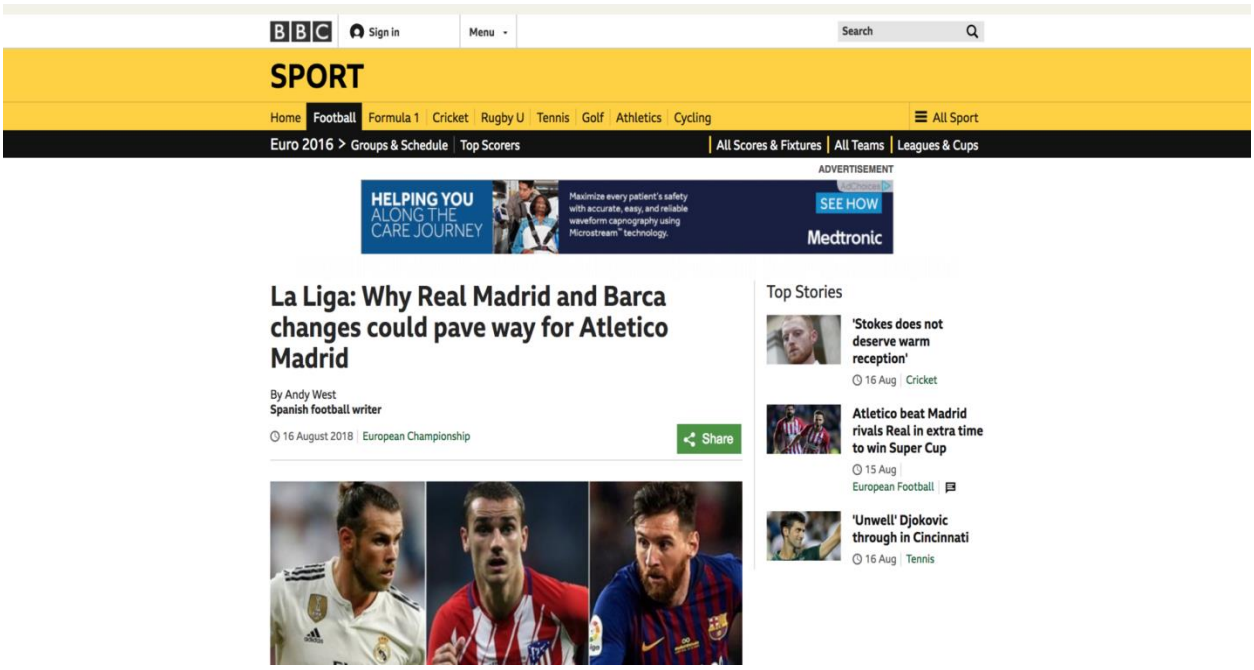


Figure 26: After Adding Adversarial Noise

4.4.4 BBC Evade Ad Network

2nd attack at Yolov3 is Evade network attack.

python -m attacks.bbc_evade_ad_network --input_dir=../data/page_based/bbc/

Output of this attack will be same into the out folder with same pattern as describe before. Iteration limit of this attack is 0-19, 10-19, 20-19……130-19.

4.4.4.1 Original image

The image below is the original image. Got this image from the directory where images are saved for further processing.

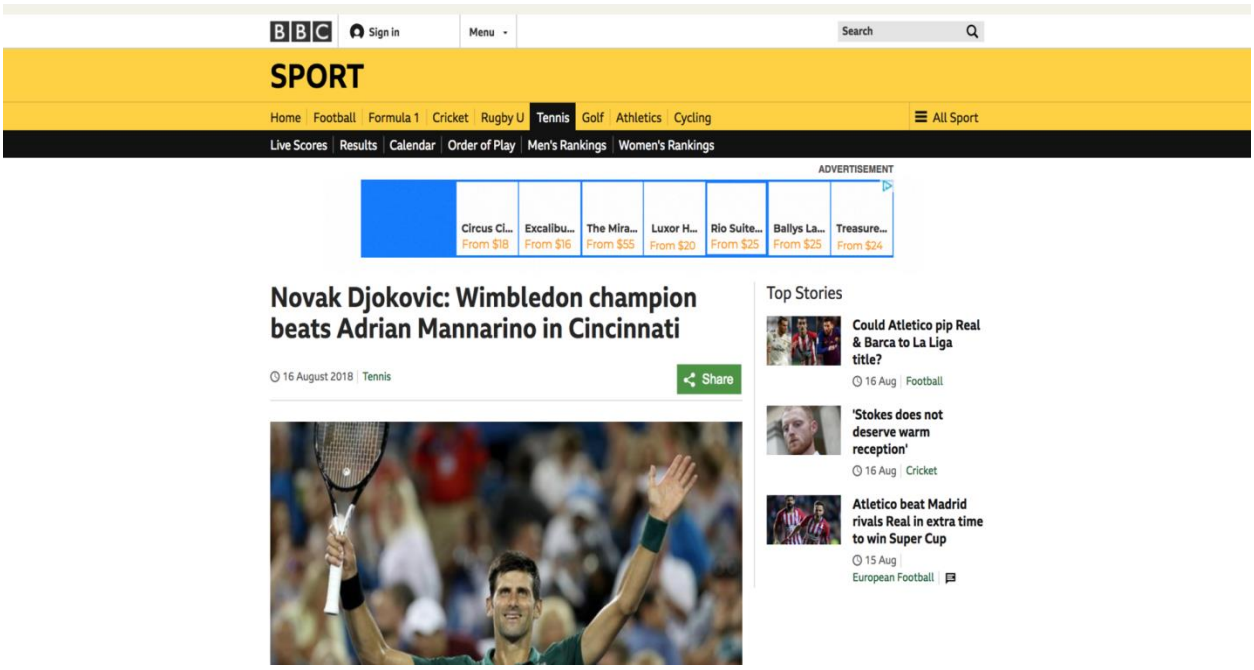


Figure 27: Original Image

4.4.4.2 Detected by yolov3

Detect the image by yolov3 classifier. It is shown that Ad is detected successfully. So the next step is to Ad noise in image and detect again to verify that whether it is passed by the detector or not.

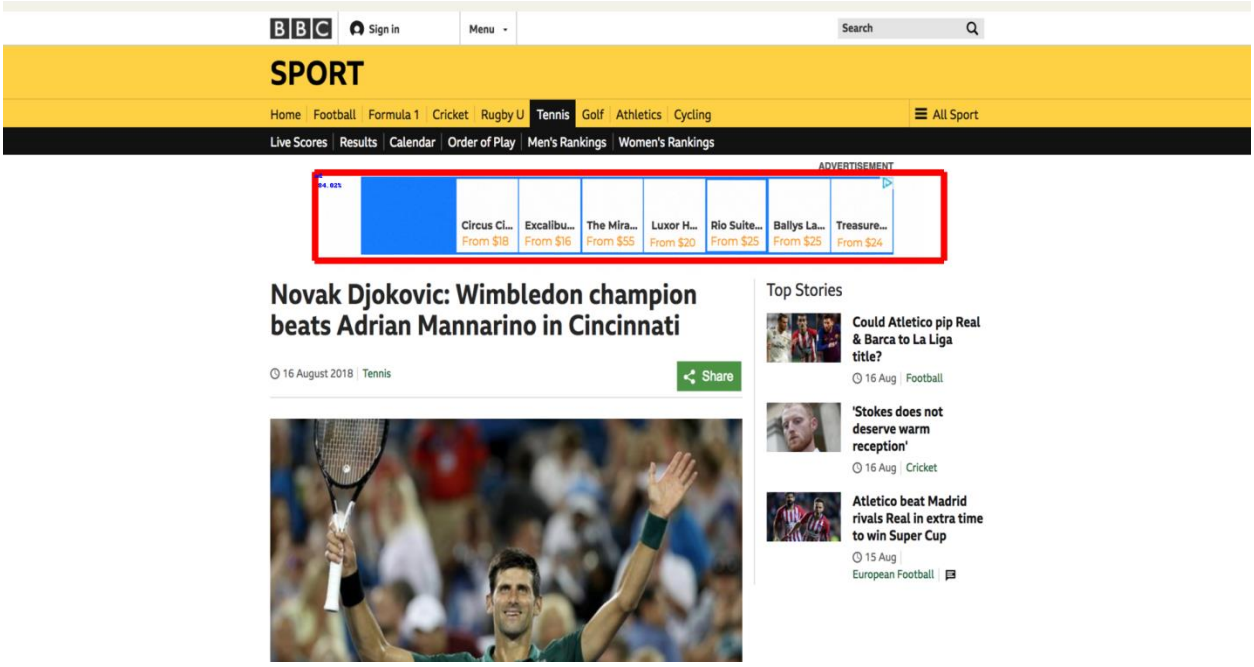


Figure 28: Yolo v3 Ad Detection Result

4.4.4.3 Noisy image

To defeat Yolov3 it runs extensive iterations. 0-19, 10-19, 130-19. Following command is use to run the attack.

python –m attacks.bbc_evade_ad_network-input_dir=../data/page_based/bbc/

Image below is from 130th iteration. Copied this image into the yolov3 directory and run Yolov3 to detect Ad from the image. But this image evade the yolov3 network.

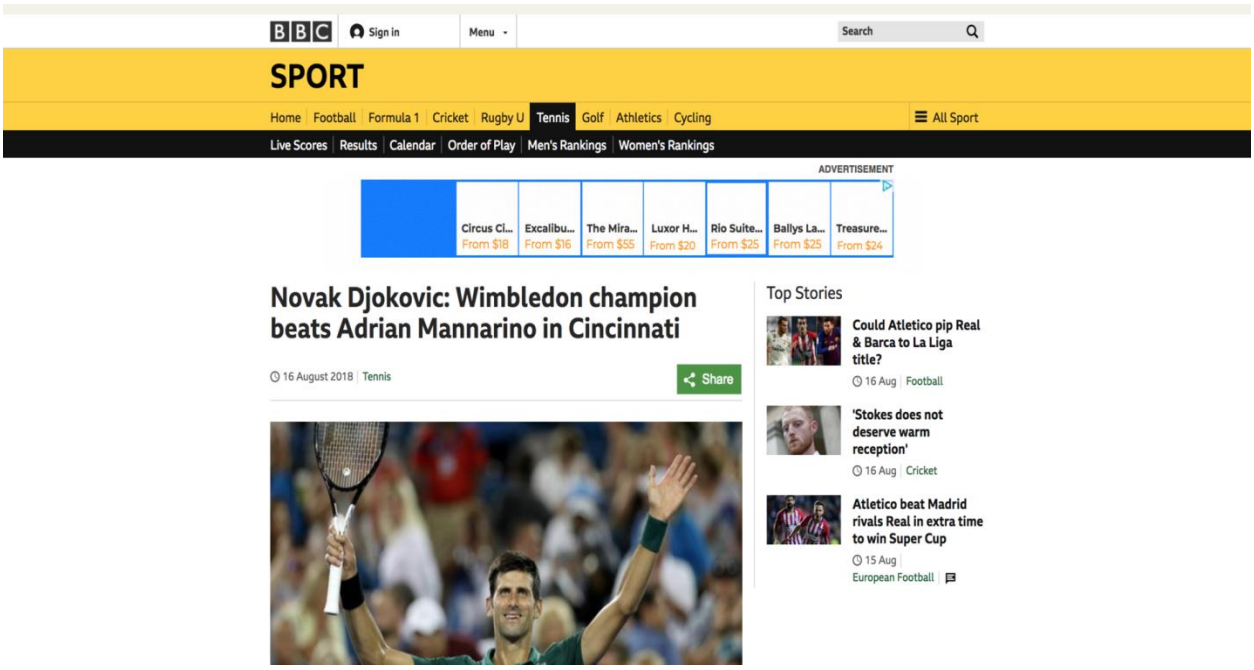


Figure 29: Adversarial Image Result

4.4.4.4 Yolov3 false positive attack

After analyzing two successful attacks the next attack is false positive attack. In these attack attackers change some values at different location of the image and detector detects it as an attack. To run that attack the command below is used.

python -m attacks.bbc_false_positive --input_dir=../data/page_based/bbc/

Image below is the result of the attack. It can see that at the upper right side of the image detector highlighting the yellow strip as an Ad. In the previous attacks we have seen that after performing attack it save the result into the output folder with different iterations. In this case as iteration increase the detection rate increase. Some Ads are detected at 99 percent accuracy.

Figure 30: Yolov3 Result Against False Positive

4.4.5 Crafted Attack

Attacker creates image using some specific values and put it at different location of the page. These small images can fool the detector and detector detects them as an Ad.

Figure 31: Self Crafted Adversarial image

4.4.6 Classifier Result

Image above is the special crafted image. At the footer of the image below it is shown that Yolov3 classifier detected it as an Ad.

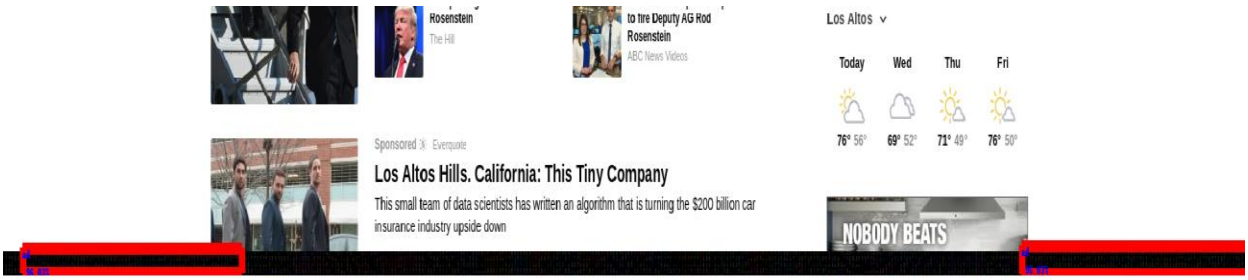


Figure 32: YoloV3 Result against Self Crafted Adversarial Image

4.5 Scale Specific Attacks

We have successfully performed different attacks so far. Now we will perform two scale specific attacks. These attacks are purposed by Zohaib Ali [35]. In his work he uses a classifier that uses SURF (speeded up robust features) to match two images. First it extracts those features from both images. After it, it matches those features. On the base of those features it provides result. SURF uses an integer approximation of the determinant of Hessian blob detector, which can be computed with 3 integer operations using a recomputed integral image. Its feature descriptor is based on the sum of the Haar wavelet response around the point of interest. These can also be computed with the aid of the integral image [36].

Flow of these experiments is that, first we will run SURF classifier and examine it result. In next step perform two scale specific attacks from the seven different attacks from [35]. After it, result will be compare. In last step our main classifier will be run on those noisy images. To check what our purpose classifier perform against these noise.

4.5.1 Surf Feature Matching

The image below showed the result of classifier. Took and image it has Adchoice logo in it. Crop that part of the image as a 2nd image and perform image matching. Result shows many matching strings.

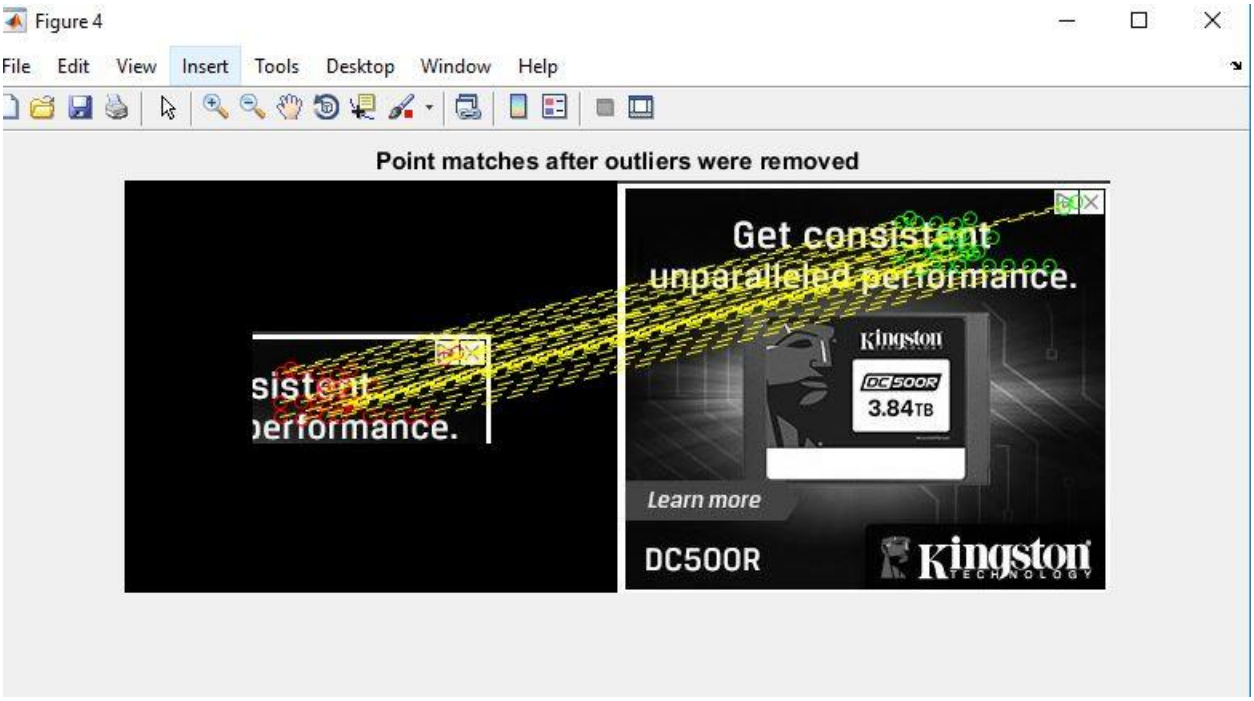


Figure 33: SURF Feature Template Matching

4.5.2 Creating Noisy Image

Took both images that used in previous experiment and added noise in both the image. This attack is targeted attack. It first extract surf features and perform function to make it noisy.

We will use two noises.

- ASMSS (Average Squared Mask Scale Specific Perturbation)
- PPSSS (Pixel-2-Pixel-Scattered Scale Specific Perturbation)

4.5.3 Average Squared Mask

In the image below it can clearly observe pixels perturbation. It is because of adding ASMSS noise. In this case it first extracts SURF features and after it added ASMSS noise into those features.

Figure 34: Average Squad Mask Noise

The image below is the result on SURF matching after adding ASMSS noise. First added noise in both the images, than run SURF feature matching classifier. It is clearly visible that matching points decrease.

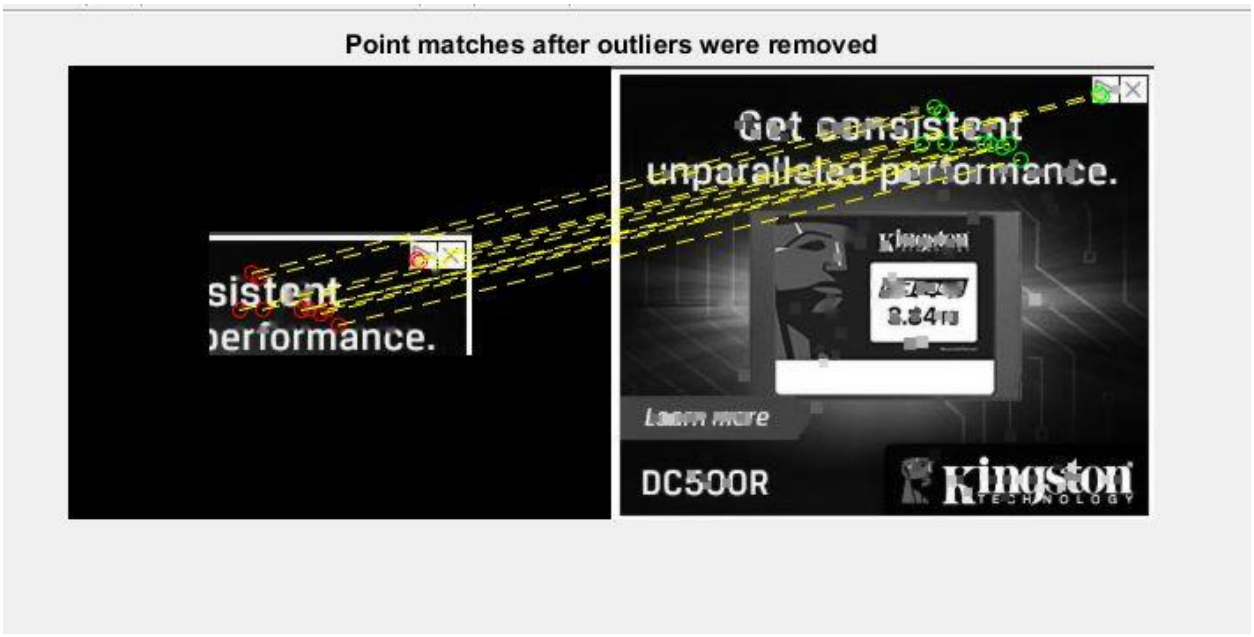


Figure 35: Logo Detection using SURF feature Matching

4.5.4 Pixel-2-Pixel-Scattered

In the image below it can clearly observe pixels perturbation. It is because of adding PPSSS noise. In this case it first extracts SURF features and after it added PPSSS noise into those features.

Figure 36: Noisy Image by P2P Scattered

The image below is the result on SURF matching after adding PPSSS noise. First added noise in both images, than run SURF features matching classifier. It is clearly visible that matching points decrease. Matching points decrease more than the ASMSS noise. So this noise is more effective than ASMSS noise.

Note:

In next chapter we will add both noises into many images and detect it with Cross Correlation classifier. And will check that how our purposed classifier works against those noises.

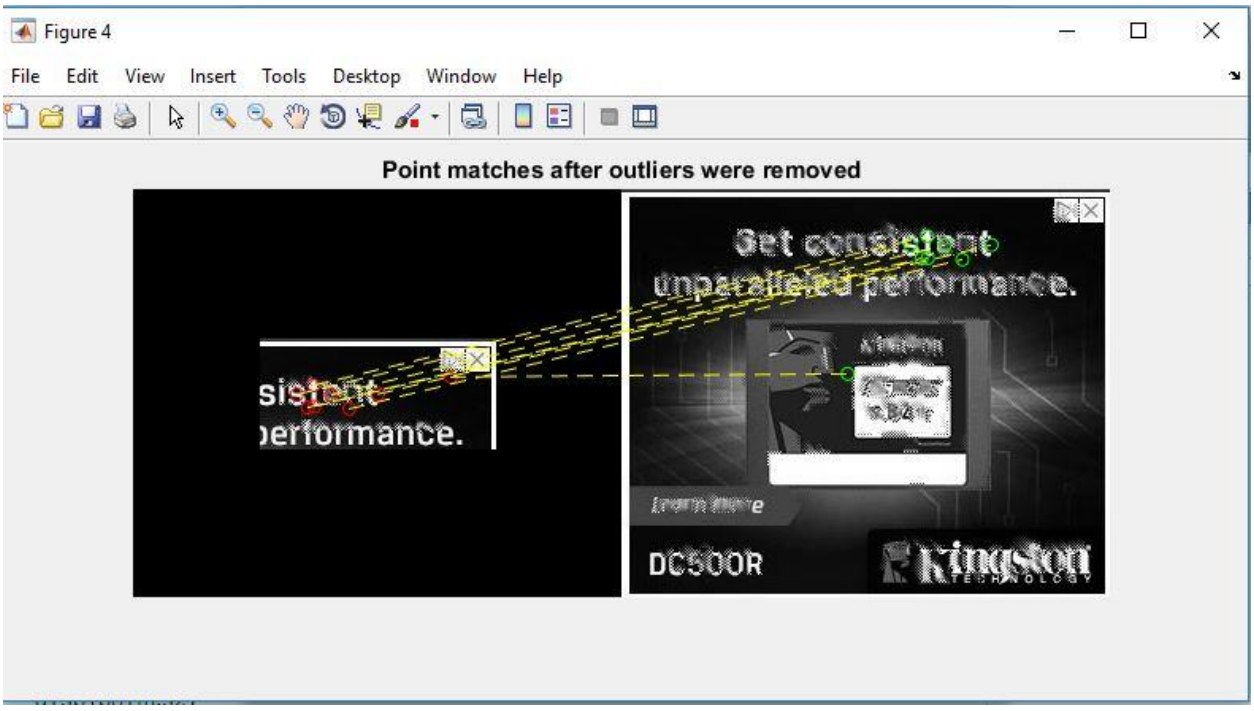


Figure 37: Result of SURF Classifier against P2P Scattered noise

Performance Analysis

In this chapter we will evaluate all the results that we got in previous chapters against our Cross Correlation classifier. Will take the result of each classifier one by one and compare it with the result of our purposed classifier. This chapter will also elaborate the deficiencies in previous solutions and discuss how our classifier is better.

Following were the some objectives regarding our solution against Anti Ad-Blockers

- Our solution will detect Ad choice Ads with maximum detection rate.
- It will have protection against all adversarial noises used in previous work.
- It will also perform well against some scale specific noises.

As per our first objective our solution should detect Adchoice logo in all the images. For this purpose we used Cross Correlation template matching code and change it according to our requirement to get 99 percent result.

Next objective was to test it against adversarial noises that have been used in previous work. We collected different images and add different type of noises in those images. After it we detect Ads in those images using the classifiers that has already used in previous work. In the result noisy images bypassed the classifiers.

In next objective we used some scale specific noises. One was Average Squared Mask and second was pixel to pixel scattered noise. These noises purposed by another researcher [35]. They use another classifier that uses SURF features. We also evaluated those noises against our classifiers.

5.1 Attack against Average Hashing Classifier, evaluation against Cross Correlation Classifier

We have completely observed the classification, obfuscation and false positive behavior against Average Hashing mechanism. Now it's time to evaluate those attacks in Cross Correlation perspective. For this purpose we took an AdChoice noisy logo that we already created.

In next step we took ten random images that have Adchoice logo in it. Run Cross Correlation classifier for 10 images. Detection rate was 90 percent in both the cases. For noisy logo and for false positive logo.

Classifier	Attack Type	Number of Adversarial logos	Total number of Ads	Detected Ads
Cross Correlation	Noisy Logo	1	10	9
	False positive logo	1	10	9

Table 4: Evaluation of Adversarial Attack that defeated Average Hashing Against CrossCorrelation Classifier

5.2 Attack against SIFT classifier, Evaluation against Cross Correlation Classifier

It's a long process to create a noisy image to defeat SIFT classifier. At my machine Intel core i7

2.50 GHZ, 16 GB of RAM at 64 bit operating system it took 6 hours. To generate a noisy logo that was undetectable by SIFT classifier. To evaluate this noisy logo against Cross Correlation classifier we took ten different images that have AdChoice logo in it and evaluate against Cross Correlational

classifier. In this scenario Cross Correlational classifier detects all the logos with 100 percent accuracy.

Classifier	Attack Type	Number of Adversarial Logo	Total Number of Ads	Detected Ads
Cross Correlation	Noisy Logo	1	10	10

Table 5: Evaluation of Adversarial Attack of SIFT against CrossCorrelation Classifier

5.3 Attack against YOLOV3, Evaluation against Cross Correlation Classifier

Yolov3 is a machine learning classifier. It is different from the previous two classifiers. In perceptual AdBlocker it is use to detect Ad in Facebook or to detect Ads that has not any specific properties in it like Adchoice logo. So the images provided to it, it trains itself on the base of those images and detect Ads. To evade this machine learning classifier it is need to create full noisy images instead of creating a malicious logo. So the attack at this classifier adds noise to the whole image. We performed three attacks in this scenario.

- **Evasion Attack 1 (**The publisher perturbs bottom of ad frame to evade ad-blocking**)**
- **Evasion Attack 2 (**The ad-network perturbs ads using a universal perturbation to evade ad-blocking**)**
- **Detection Attack (**Publisher perturbs the page header to create a false ad prediction**)**

5.3.1 Evasion Attack 1

This is very extensive attack. To create fully noisy images that are totally undetectable by Yolov3 classifier is not an easy task. There were lot of loop repetition involve in it. Attack creates a loop of 20 images. After it, it performs lot of iterations at those twenty images. And perform detection as well until that point where images are undetectable by classifier.

In comparison with our classifier we took ten images from different iterations and evaluate against our classifier. Our classifier will successfully able to detect Ads in all images. The difference is there that our classifier work on some specific properties in the images so in this scenario the special property was **Advertisement** key word. So we get this key word from an original image and use it as a template image. Cross Correlational classifier was successfully able to detect that key word from all the noisy images.

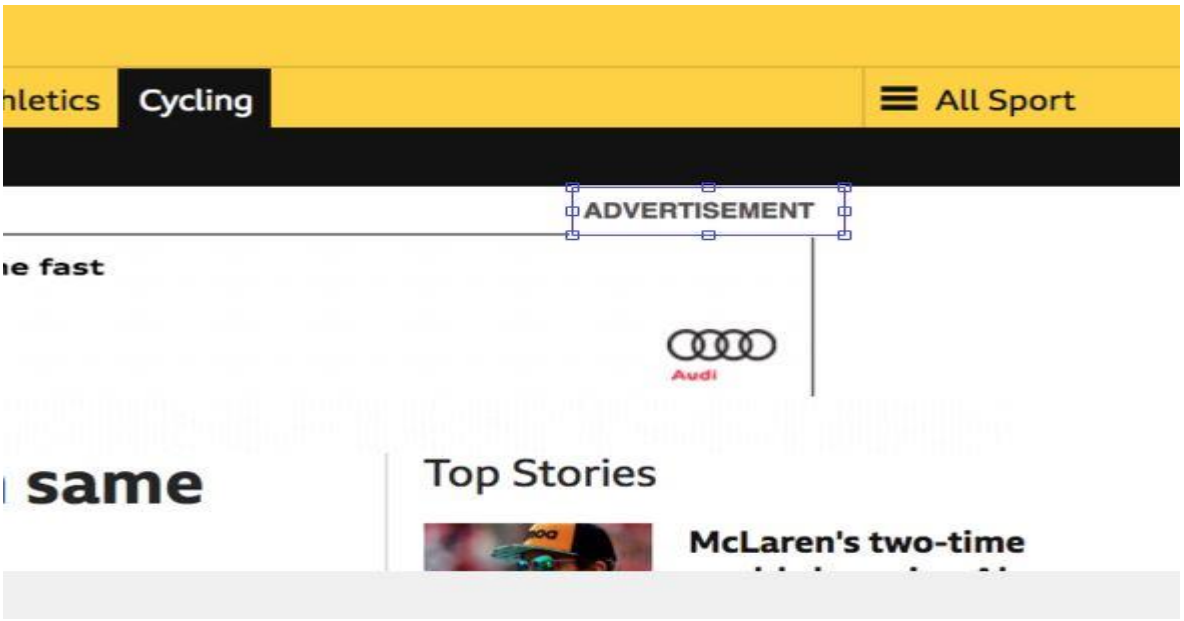


Figure 38: Cross Correlation Classifier result against Evasion Attack 1

Classifier	Type of Attack	Adversarial Logos	Number of Ads	Detected Ads
Cross Correlation	Evasion 1	1	10	10

Table 6: Evaluation of Evasion Attack Against CrossCorrelation Classifier

5.3.2 Evasion Attack 2

In the previous attack attacker was perturbing bottom of Ad frame to avoid detection.

But in this case ad-network perturbs ads using a universal perturbation to evade ad-blocking. In this case it was maximum iterations to create noisy images. Just like the previous case took 10 undetectable images by Yolov3 classifier and examine those images against Cross Correlation classifier.

Classifier	Type of Attack	Adversarial Logos	Number of Ads	Detected Ads
Cross Correlation	Evasion 2	1	10	10

Table 7: Evaluation of Evasion Attack 2 against CrossCorrelation Classifier

5.3.3 Detection Attack

This is third type of attack against Yolov3 classifier. It is false positive attack. In this type of attack, attacker creates different pattern and put it at the header or at the footer of the image. Yolo v3 classifier detects that part of the image as an Advertisement. We have demonstrated active detection attacks in last chapter. There were random images. Separated three images. These three images have maximum false positive result. One image has 81 percent false positive result other has 99 percent false positive result. Detected those images with cross correlational classifier. This classifier detected Ads, not the false positive part of the images.

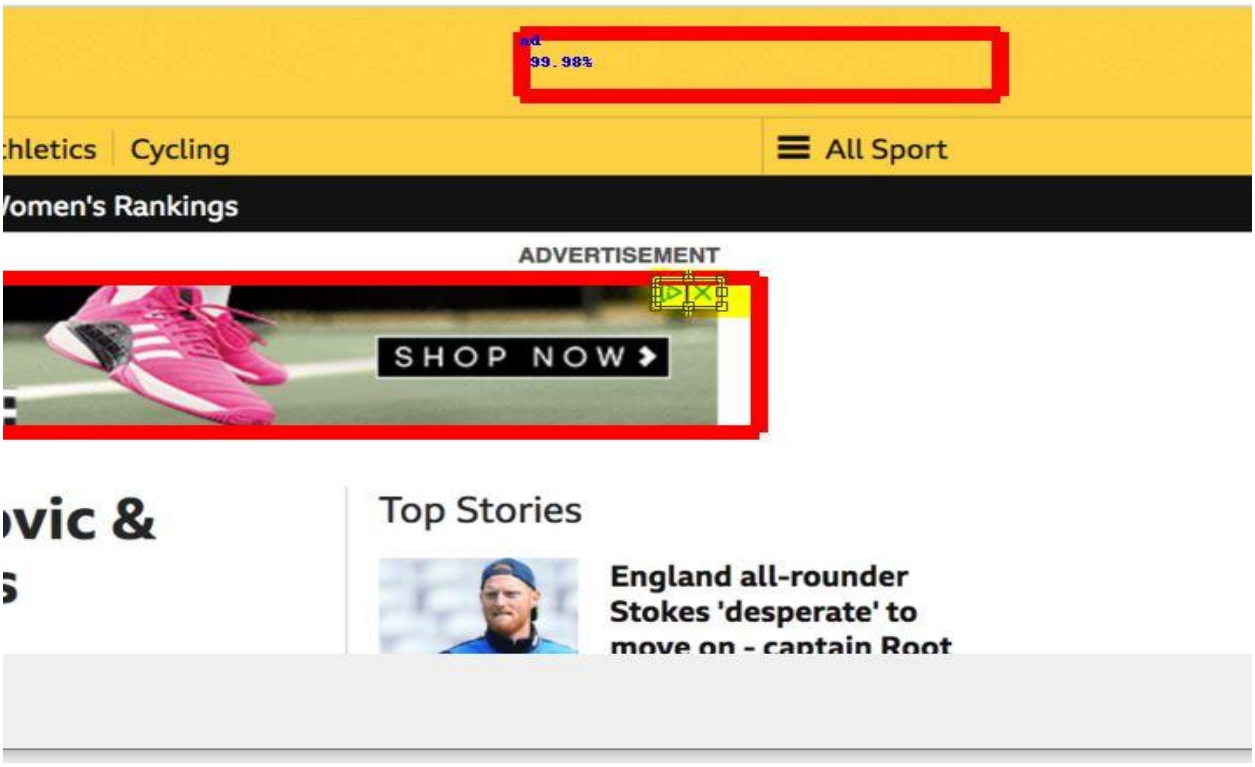


Figure 39: Cross Correlation Classifier Result Against Detection Attack

In the image above there is 99.98 percent false Ad detection. But Cross Correlation classifier is detecting accurate Adchoice logo with 100 percent accuracy.

Classifier	Type of Attack	Adversarial Ads	Number of logos	Detected Ads
Cross Correlation	Detection Attack	3	1	3

Table 8: Evaluation of Detection Attack Against CrossCorrelation Classifier

5.4 Attack against SURF Feature Matching, Evaluation against Cross Correlation Classifier

Classifier	Attack Type	Number of Adversarial Ads	Total number of logo	Detected Ads
Cross Correlation	ASMSS	8	1	8
	PPSSS	8	1	8

Table 9: Evaluation of ASMSS and PPSSS Attack Against CrossCorrelation Classifer

Conclusion and Future Work

In solution we proved that our Cross Correlation classifier is less prone to adversarial attacks. We detected different properties from Ads and evaluate our classifier against one thousand advertisement images. Our classifier detected those advertisement properties with 99 percent accuracy. We extensively studied and performed different attacks that have been done in previous work. After it we evaluate our classifier against those attacks. In this scenario our classifier provides ninety nine percent results against different adversarial attacks.

Future Work

To block user tracking, Malvertisement, fake news, Anti Ad blockers and many other things that advertisement companies do [38], there is need to build a system that is less prone to attack from those companies. All the Ad blockers that we have studied so far are vulnerable to attack by those advertisement and analytical companies. Some attacks have seen who are exploiting browser vulnerabilities to send Ads [39][40]. This is the continuous war between Ad blockers and Anti Ad blockers. To win this war there is need to build an Ad blocker that will not be detectable by advertisers. Perceptual Ad blocker is the first step in this direction but it is defeated by adversarial attacks. Our solution is one step ahead that our classifier is not effected by those attacks. So in future we will collectively use different classifier that will be less prone to adversarial attacks. On the base of those classifiers we will built an Ad blocker. And release it for commercial use.

References

[1] https://adblockplus.org/en/about

[2] https://en.wikipedia.org/wiki/Adblock_Plus

[3] https://appuals.com/whitelist-website-adblock-adblock-plus/

[4] https://www.imperva.com/learn/application-security/malvertising/

[5] M. H. Mughees, Z. Qian, and Z. Shafiq, "Detecting Anti Ad-blockers in the Wild," Proc. Priv. Enhancing Technol., vol. 2017, no. 3, pp. 130–146, 2017.

[6] M. A. Bashir, S. Arshad, E. Kirda, W. Robertson, and C. Wilson, "How tracking companies circumvented ad blockers using websockets," *Proc. ACM SIGCOMM Internet Meas. Conf. IMC*, pp. 471–477, 2018.

[7] Florian Tramèr, Pascal Dupré, Dan Boneh Adversarial: Perceptual Ad Blocking meets Adversarial Machine Learning. In 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19)

[8] G. Storey, D. Reisman, J. Mayer, and A. Narayanan, "The Future of Ad Blocking: An Analytical Framework and New Techniques," 2017.

[9]Steven Englehardt and Arvind Narayanan. Online tracking :A 1-million-sitemeasurement and analysis. In ACM Conferenceon Computer and Communications Security (CCS).ACM, 2016.

[10] Muhammad Ahmad Bashir, Sajjad Arshad, Engin Kirda, William Robertson, Christo Wilson "How Tracking Companies Circumvent Ad Blockers Using Web Sockets" ACM Internet Measurement Conference 2018

[11] Tomasz Bujlow, Member, IEEE, Valentín Carela-Español, Josep Solé-Pareta, and Pere Barlet-Ros , Web Tracking: Mechanisms, Implications, and Defenses

[12] Muhammad Haris Mughees, Zhiyun Qian, and Zubair Shafiq, "Detecting Anti Ad-blockers in the Wild" Proceedings on Privacy Enhancing Technologies 2017

[13] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. Detecting and Defending Against Third-Party Tracking onthe Web. In *USENIX Symposium on Networked Systems*

Design and Implementation (NSDI), 2012.

[14]M Zubair Rafique, Tom Van Goethem, Wouter Joosen, Christophe Huygens, and Nick Nikiforakis. It's free for areason: Exploring the ecosystem of free live streaming services.In *Proceedings of the 23rd Network and Distributed System Security Symposium (NDSS)*, 2016.

[15] Adblock Plus. 2018. Sentinel. https://adblock.ai/.

[16]Grant Storey, Dillon Reisman, Jonathan Mayer, and Arvind Narayanan. 2017. Perceptual Ad Highlighter. Chrome Extension: https://chrome.google.com/webstore/ detail/perceptual-ad-highlighter/mahgiflleahghaapkboihnbhdplhnchp; Source code: https://github.com/citp/ad-blocking.

[17]Panagiotis Tigas, Samuel T King, Benjamin Livshits, et al. 2019. Percival: Making In-Browser Perceptual Ad Blocking PracticalWith Deep Learning. arXiv preprint arXiv:1905.07444 (2019).

[18] https://www.techopedia.com/definition/23090/ad-blocker

[19] https://www.ghostery.com/faqs/how-does-ghostery-work/

[20] Daniel C. Howe, Helen Nissenbaum, "Engineering Privacy and Protest: a Case Study of AdNauseam" Proceedings of the 3rd International Workshop on Privacy Engineering USA 2017

[21] Marcel Van Herk, in Handbook of Medical Image Processing and Analysis (Second Edition), 2009

[22] https://github.com/mingyuliutw/FastDirectionalChamferMatching

[23] Jaydeo K. Dharpure, M. B. Potdar, Ph.D, Manoj Pandya Counting Objects using Convolution based PatternMatching Technique, International Journal of Applied Information Systems (IJAIS), Volume 5 – No. 8, June 2013

[24] https://github.com/sunsided/convolution-template-matching

[25] https://en.wikipedia.org/wiki/Speeded_up_robust_features

[26https://www.mathworks.com/help/vision/examples/object-detection-in-a-cluttered-scene-using-point-feature-matching.html

[27] https://www.mathworks.com/help/images/ref/normxcorr2.html //crossco relation

[28] J.N. Sarvaiya ; Suprava Patnaik ; Salman Bombaywala Image Registration by Template Matching Using Normalized Cross-Correlation, 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies

[29] https://en.wikipedia.org/wiki/Scale-invariant_feature_transform

[30]https://github.com/ftramer/ad-versarial/tree/master/element-frame-based

[31] https://github.com/ftramer/ad-versarial/blob/master/page-based

[32] https://pjreddie.com/darknet/yolo/

[33] https://github.com/pjreddie/darknet

[34] https://github.com/ftramer/ad-versarial/releases

[35] https://github.com/zohaibali-pk/Adversarial-Noises-for-Handcrafted_Features

[36] https://en.wikipedia.org/wiki/Speeded_up_robust_features

[37] https://mfix.netl.doe.gov/doc/tracker/19.1.0/methods/template_matching.html

[38] https://adguard.com/en/welcome.html

[39] https://thehackernews.com/2019/10/malvertising-webkit-hacking.html

[40] https://www.zdnet.com/article/malvertiser-exploited-two-browser-bugs-to-show-over-one-billion-malicious-ads/