Image-Based News Recommendation

An Online and Offline Approach

F. Corsini

Technische Universiteit Delft

TU Delft Delft University of Technology

Challenge the future

Image-Based News Recommendation

An Online and Offline Approach

by

F. Corsini

in partial fulfillment of the requirements for the degree of

Master of Science
in Computer Science

at the Delft University of Technology,
to be defended publicly on Monday August 29, 2016 at 10:30 AM.

Supervisor:	M Larson	
Thesis committee:	E. Hendriks,	TU Delft
	A. Hanjalic,	TU Delft

TU Delft Delft University of Technology

Contents

1

Introduction

1.1. Problem Statement

Typical online news content providers publish images along with their news items. Our work is motivated by the conjecture that these images play a role in the effect of the recommendation, especially whether a user will click on the item. Content providers are well aware of the importance of images and are already taking advantage of them (e.g., both their informative potential, and their potential to act as clickbait). However, the effect of images for automatic recommendations is currently understudied and not well understood. Our research looks for the effect of such images, in order to determine if they can play a crucial role in the definition of a more refined recommendation. Our hypothesis is that people tend to click on news articles because they are curious about the image, as the image catches their eye, and some images depict things clearly making it very easy to see what the article is actually about. An screenshot from the New York Times [1] can be seen in Fig.1.1 where an example of images used in online recommendations have been circled. As can be seen, almost all of the articles suggested have an image accompanying the title.

No prior research in this field has been done at this time. To our knowledge, this is mainly due to the fact that images have never been seen as special features, but rather as standard attributes of the item (article) as categories or the authors. Advertising images have indeed been studied to some extent, and some work have been used in this thesis, esp. [2]. By taking a look at any news website, we can easily spot the importance of images in the showing of the articles: each article is shown to the user only as a title plus an image (thumbnail), therefore we felt that these two attributes should be regarded as ”special” and investigated further. In this work we try to exploit only the information extracted from images. Another reason image based-recommendation has not been extensively investigated is the computing resources required. As the totality of news recommender service runs in an online environment, speed is usually critical to the correct functioning of the system: image analysis tend to not perfectly fit into the ”easily computable” requirement.

Important tools in this work have been gained by joining the CLEF-NewsREEL (see 1.4) competition held during the period of this thesis work. The competition has given us the access to an online platform which allowed us to test and benchmark our finding in a living lab environment. Our approach to the problem has tried to look in both the challenges and advantages of the online and offline environments.

The work explained this this thesis is presented as following: A general overview is present in the current chapter, followed by a more in depth dive into the related work and useful background in chapter 2. The approach in chapter 3 shows how we decided to tackle and solve the problems to support our claim. An exhaustive overview of the strategies and limitations of the implementation is presented in chapter 4, followed by more in detail explanation of the online in chapter 5 and offline in chapter 6 approaches and their respective results. A discussion and conclusion is presented in the last chapter 7. Following are a few appendixes which include: the tables with the raw results of the evaluation 9, example of images from the ORP 8. At the very end, after the bibliography it is attached the paper

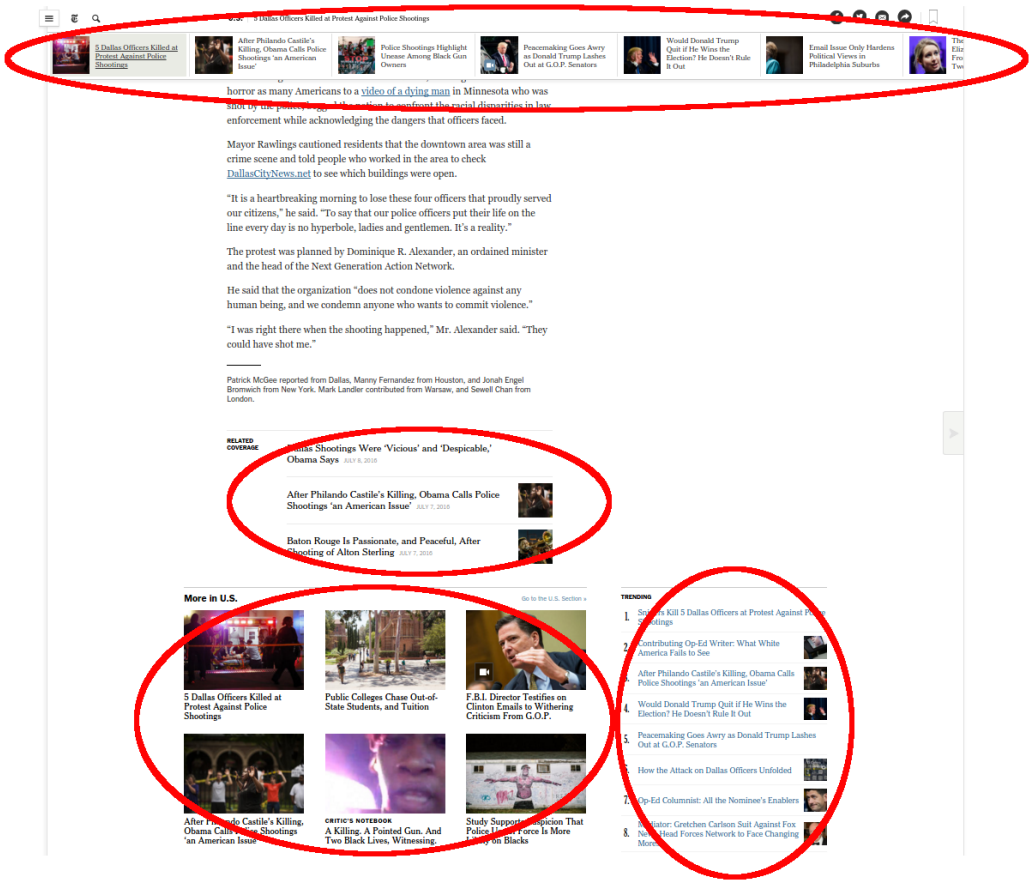


Figure 1.1: News recommendation Example

published in CEUR-WS [3] for the CLEF 2016 - Conference and Labs of the Evaluation forum. The paper itself is a partial explanation of what was done during this thesis and an online version can be found at [4].

1.2. Images in our Research

Figure 1.2: A sport Image

What is the role of Images in this research, and why users are pushed toward showing an interest towards it?

- It catches the eye. In contrast to the rest of the page, the image attracts people's gaze. An image can be in itself attractive. This can be for several reasons: because of the content or it can be because of the way in which the image is taken, i.e., it "glows" or has an interesting composition. An eye catching example is show in Fig. 1.2
- Additional information: it provides information additional to that of the title. This information is only effective if it can be understood. Some images are easier to "parse" than other because they present their content in a clearer way. The ability of a person to parse an image depends on the context. To a certain extent the title might activate the context. However, people's background knowledge and experience will also play a role.

There are at least two reasons whypeople are attracted to content shown in images:

1. The image is interpretable, but only to a certain extent. The user clicks the image because they feel that they should be able to interpret it, but can't.
2. The image is of something familiar, but it presents a puzzle that makes them curious (like an unfinished story)

The user attitude towards image will vary depending on the domain and how images are presented. The domains and the news topics included in this research are limited and therefore our conclusions are limited to the kind of images shown in these domains. From the list of websites and domains used in this research 1.4.1, a few examples have been given below. As can be seen, different domains have different images: the frame, the presentation and the aim of each domain is different. While General news information websites tend to show images that reflects somehow the content of the article (Fig. 1.3), Automotive (Fig. 1.4) and Sport (Fig. 1.5) try to use appealing shots rather than informative. On the other hand IT (Fig. 1.6) try to always shows standard shots of some familiar brand or device.

Attention is required from the user in order to ensure the click. There are a few different levels of attention in regards to a shown image

- Full attention: a user perfectly visualize and parses an image, and consciously decides if it triggers his/her own interest.

Figure 1.3: General news recommendation

Figure 1.4: Automotive news recommendation

- In field but not focused: Most of images that pass along a user gaze but are not processed consciously.

- Out of field: The image cannot be processed due to being covered or not in the user field of vision.

Logic would dictate that a user is more likely to click on an image if its level of attention is higher. It is worth noting that level of attention and reason of interest for an image are correlated. Full attention is needed in order to be able to parse the image for additional information, since an out of field image cannot give meaningful information. Images that ”catch the eye” might more likely shift the user attention from the ”in field” level to full.

As previously mentioned, each different news category and domain use a different kind of images which have a better effect on the respective users. General News, by trying to communicate trough the image, will try to reach the full attention of the user, which than will be able to detect the addition information cantained in the image. Professional photos will usually have no problem in communicating the topic of the additional information contained in it, however the extremely reduced size of the shown thumbnail makes this problem actual. Automotive and Sports domains will try to use images to catch even partial attention and direct the user to the title which usually easily explain the content of the article (the name of the team or the car). This latter domains will not have the problem of interpretability, as the average image will tend to follow always the same pattern.

This introductory analysis of the composition of images leads to the informal questions that drove a more in detail research. Are there some features in the images which make the image more interpretable and help users parse the images for additional information? Are there some features which make easier for the attention of users to shift to full?

This section was included to convey the complexity of the overall problem. Acknowledged the fact that this is something beyond academic computer science research, we make the assumption that users clicks have something to do with the image “catching the eye”, and we will not explicitly address the other considerations.

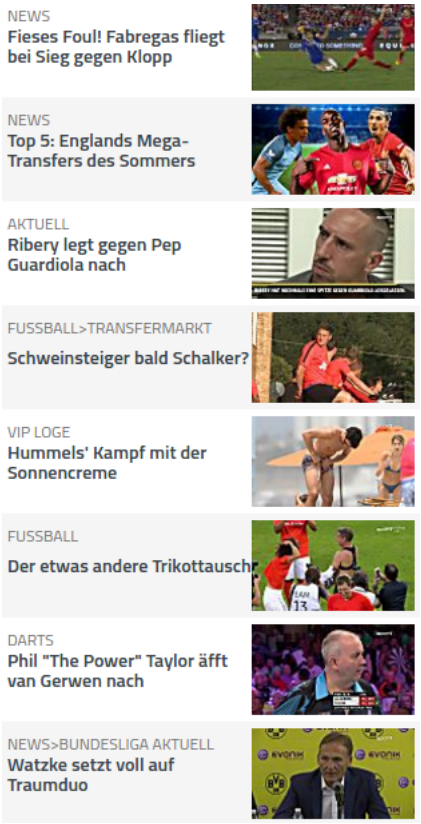


Figure 1.5: Sport news recommendation

Figure 1.6: IT news recommendation

1.3. Research question and Assumptions

Research questions:

- Can a non-trivial news recommender that extracts features from an image feasibly be run online in a real world environment?
- Can information extracted from images accompanying items in a recommendation system help improve the Click Through Rate of such systems, specifically in the news domain?

The reason for these questions can be found in our interest to analyze image based news recommendation. The feasibility of such heavy computational task is not guaranteed in an online environment, as no previous work has been done in this field; therefore an evaluation which takes technical constraints into account is needed, such the 3d recommender evaluation [5]. From here the logical next step is to investigate the effect that these images have on the performance of the recommender, specifically the CTR.

Table 1.1: ORP domains

domain	topic
www.ksta.de	general
www.sport1.de	sports
www.gulli.com	IT
www.tagesspiegel.de	general
www.computerwoche.de	IT
www.cio.de	IT
www.tecchannel.de	IT
cnet.de	IT
www.zdnet.de	IT
www.silicon.de	IT
www.motor-talk.de	automotive

Assumptions:

- Users are more likely to click on news links accompanied by images that they find interesting.

We decided to write down this assumption in order to check our claims. We think that this is a reasonable and logic assumption, linking Click Through Rate (CTR) and user interest in the image.

1.4. The CLEF-NewsREEL Challenge

The CLEF NewsREEL [6] News Recommendation Evaluation Lab challenges participants to come up with an original and effective solution for providing recommendations for users in the news environment. Our participation is both for Task 1 (Living Lab Evaluation) and Task 2 (Evaluation in Simulated Environment). An overview of this year challenge results can be found at [7]. This challenge was intended to be one of the major accomplishments of this thesis: a paper regarding our experience was submitted to the CLEF 2016 Evora conference [8] and accepted. The overview paper of this year results is still to appear.

The participants in this competition are given access to the Online Recommendation Platform (ORP) where they can test and benchmark their algorithm. ORP is a web based platform which redirects part of the traffic from a recommendation company, plista [9], to participants' experimental algorithms. This allows participants to get a share of real user traffic from real websites. The redirected traffic is around less than 5% (according to plista). The typical page layout of the pages of these domain can be seen in Fig.1.7.
This platform was critical tool in this work, both because all data comes from there and all online testing has been done there. A more detailed overview on ORP functions can be found in chapter 5.

This platform allows participants to actually generate recommendations for real world users, by creating list of articles which are displayed in the web page to users as shown in Fig. 1.3 1.4 1.5 1.6, exactly the same way as the plista recommendation would.

1.4.1. Domains

The ORP platform serves around 10 news domains, which cover a broad range of topics from newspapers to tech blogs. Table 1.1 is a list of all the domains which are served by ORP. As can be seen, all domains are in German, however that does not affect our research. More information about the data from these domains can be found at [10]

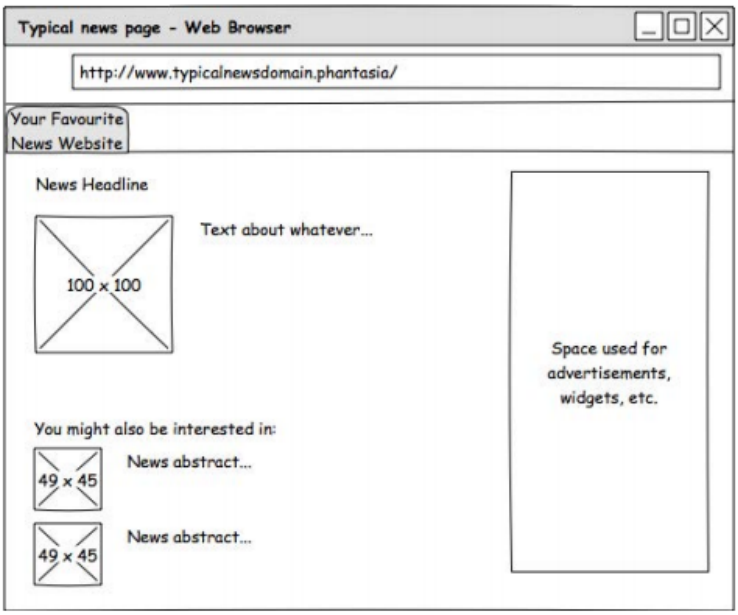


Figure 1.7: Exemplary news article website, source [10]

2

Background and Related Work

2.1. Recommendation Systems

Recommendation Systems (RSs) collect and hold many information about users and items, and try to make the user interested in a few new items which the user didn't look explicitly for. RS are usually implemented in web shops (Ebay, Amazon), search providers (Google, Yahoo) and News ('You may also be interested in...'). The basic paradigm of a RS is filtering objects based on the user in order to present a short list of recommendations to the user.
In order to filter items, many different approaches can be taken. Depending on the task complexity and number of items/users, opposite methods can be useful. Personalization can be introduced only if there is enough information about the user which can help to filter out items.
Commonly used methods to filter are: popularity, recency, content-based, collaborative and hybrid [11].

2.1.1. Content based Filtering

Content based filtering systems are a really popular method of filtering items [12] [13]. This consists of including an item in the recommendation based on one or more fields which have something in common with the one the user is viewing. A perfect example is movie RS: when a user is viewing a thriller movie, the system suggests either other thriller movies or movies from the same author. By selecting an important field (genre and author in this case), a handful of items can be selected as more appropriate.
This method became extremely popular with movies due to the Netflix competition [14], which run in 2009 and awarded 1M dollars in rewards. The winner basically used a tweaked content based filtering algorithm [15].

Advantages:

- User Independence: since it is based on the features of the items rather than the use of the other users, this method can be helpful even if the target group is only one user.
- New Item: newly arrived and fresh items can be recommended even if there has been no interaction with users yet.
- Transparency: A recommended item can be easily traced back to understand how and why the original recommendation was generated. This is possible since the used fields for filtering are usually explicit

Disadvantages:

- Limited content analysis: there are only a finite number of explicit field which can be used to predict. If those are not enough, other features can be generated from attributes of the item. However, domain knowledge is needed and often domain anthologies are needed. If an item has poor description and features, it will be probably filtered out due to not enough features to be analyzed and recommended.

- The bubble problem: basing recommendation on common features means that all the recommendations will be similar to the first item. This leads to an over-specialization problem since the user is going to be recommended items similar to those already visited.

2.1.2. Collaborative Filtering

Collaborating filtering is widely known for its use in all kind of E-commerce systems [16][17][18]. It is one of the most successful approaches to building a recommender system, as it uses the known preferences of a group of users to make recommendations of the unknown preferences for other users. When a new user enters the system, A, is matched against the database to discover neighbors. This neighbors are other users who have historically had similar taste to A in regards to items visited.

Advantages:

- No domain knowledge required: since the recommendations are generated from other user behaviour, there is no need to know anything specific to the item being recommended. No knowledge of the features of the items is needed as well.
- Good results: this is the reason why it is so widely used in all E-commerces online. It is an easy and fast implementation which provides basically the best results overall.

Disadvantages:

- Scalability: depending on the size of the system, it might be really computationally intensive to find the nearest neighbour, especially with the size of the items and the user table growing. Recently new techniques such user factorization have almost resolved this problem.
- Sparsity: especially due to the large number of items, it may be hard to find a close enough neighbour to make a reliable recommendation prediction
- New user: this kind of technique is especially prone to the cold start problem as the user can receive reliable recommendations only after a few interactions with the items

2.1.3. Cold start problem

This problem is one of the major issues with today recommending systems. It occurs whenever there is a new entry in the system (user or item) and there are not yet enough data to properly address the recommendation. Collaborative filtering is especially weak again this problem, since in the beginning, whenever a new user joins, the system has no interactions from which gaining a neighbour point, as the use has not interacted yet with the system. This problem exists even on the item side, even if less catastrophic.

2.1.4. News Environment

News recommendation is a sub domain of classic recommendation, but what is special about it? Here are a few peculiar features.

- Short item life: The usual life spawn of a news article is around 3 days. After that it is either deleted by the publisher or it is not visited anymore (user lost interest in what is not ”news”). This introduce a series of new problems, since it is a continuous cold start from an item point of view.
- ”Weather Problem”: the interest of people shifts rapidly, especially in the news field. Depending on what’s going on around them, users can abruptly change their interests. One example is that people who have never cared about soccer, may suddenly be interest in it only during the world championship. See Section 3.5.4 for more details.
- No registration system. Even if there are a few exceptional cases, user landing on newspaper pages can only be tracked through cookies, which are often either disabled or unreliable. There are a few news portals which require authentications, however these a relativity tiny number. This leads to a user side continuous cold start since no historic data is available.

In general, existing news recommender systems can be categorized into the three groups detailed above [19]: content-based methods [20], collaborative filtering [21] and hybrid version of several techniques [22]. However, many of the standard ways of creating recommendations tends to have problems.
First, user based filtering cannot be carried out properly because of the massive cold start problem, both from a point of items and users. Items, which get updated continuously, cannot be filtered since almost no users have visited and interacted with them. Users which recently joined, having no history, cannot be addressed with proper recommendation since they cannot get matched.
On the other hand, content based could work however the ”weather effect” does take its toll by making the interest of the user shift continuously depending on what is happening and what is hot in the news world. Moreover, content based may recommend old items which have something in common with items the user is currently browsing, however they are not interesting anymore since they might be old or outdated. If a recency time threshold is imposed, it might end up with not having a match for recommendation.
For the above listed problems, the news field is a sub domain of the recommending field which does not gain much improvement from standard personalization techniques, which are actually counterproductive in many cases.

By looking at current and available papers about implementations of news recommendation systems [23][24] and from surveys on the topic [25], it seems that the best performing and the most widely used algorithms are popularity based and recency, or a mix of both with external information about the user. The relevant literature on the topic seems to show that standard personalized models do not fully work in the news environment, making these ”naive” algorithms work better that the classical content-based and collaborative filtering techniques. This is probably due to the underlying problems of the news sub domain. In order to overcome this and introduce personalized models, new approaches have been taken, which are described in the next section

2.2. Related Work

2.2.1. Interest

Many studies over attention on images. What triggers our eye to lay on an image?

With content-based image retrieval on the rise, there is an increase in the study of cues that could help in ranking the retrieved images. A sound measure that would help to automatically rank is how interesting people find an image. Some work has been done about the Internet and especially with Flicker images [26], however this interestingness is different since it implies some sort of community and social behaviour which is totally absent in a news recommendation environment. Flickr’s interestingness is based on social parameters linked to the behavior, i.e. according to the uploader’s score reputation and ratio between views, favorites and comments. As example, images with a positive connotation (smile, bright), tend to always have a higher level of interestingness in a social media.

The literature about attention goes way beyond informatics related journals. Psychology, with its studies on cognition and emotions, is an obvious field of research in which this topic has been much investigated.

Silvia [27] in his paper has studied the mechanism of appraisal in interest. He found novelty-complexity and coping potential to be the most important appraisals factors in the process of feeling interest toward an image. Unsurprisingly, he has also identified that people with a higher level of familiarity with the subject depicted in the image have a higher level of interest in more complex forms of the stimuli. He also tries to classify people into two interest categories: the first group, with higher curiosity and openness traits, is more likely to be interested by novel and more complex stimuli. The second group, however, was mostly triggered by coping potential and comprehensibility.
Gygli [28], following psychological studies, identify various cues for “interestingness”, namely *aesthetics*, *unusualness* and *general preferences*. Soleymani [29] shows an interesting approach on how to create a model that learns from the images what’s a person is triggered about. In his research,

affective content, quality, coping potential and complexity are shown to have a significant effect on visual interest in images.

2.2.2. Attention toward images

Images tends to pick our attention easier, as most publishers will leverage images that attract our interest or curiosity. Most of us are visual creatures at heart; interesting a 65 percent of us are visual learners, according to the Social Science Research Network [30] , therefore are inclined to follow any new visual stimulus.
This is strengthened by the findings over the recent addition of twitter: imaged showed directly on tweets rather than the bare link. A blogger [31] showed promising improvements from her twitter account after the adoption of the images: tweet with an image received 18% more clicks, 89% more favourites and 150% more retweets over those without image.
This is of course reflected by how the market is tackled. Marketers have finely tuned strategies to attract new customers which often relies on heavy visual assets. 46% of marketers say photography is critical to their current marketing and storytelling strategies [32], which often incorporate online clicks.

2.2.3. Advertisement Theory

An accurate prediction of the probability that users click on ads is a crucial task in online advertisement business. Even if with different methods, both our thesis research and ads business share the same goal: predict (and increase) how many clicks an image (or an ad) receives. State-of-the-art click through rate prediction algorithms rely heavily on historical information collected for advertisers, users and publishers. However recent work have seen the integration of multimedia features extracted from display ads into the click prediction models [2] [33].

2.2.4. Clickbait

Facebook considers clickbait as "a headline, especially that of a sensational or provocative nature, which withhold information necessary to understand what the story is about (You'll Never Believe Who Tripped and Fell on the Red Carpet…)" [34]. Wikipedia also adds: "a term describing web content that is aimed at generating online advertising revenue, especially at the expense of quality or accuracy". Content publishers of all kinds discovered clickbait as an effective tool to draw attention to their websites. Clickbait on social media has been spreading quite a lot in recent years, due to its virality and easiness of spread, and even some news publishers have adopted this technique in order to squeeze some more clicks. Clickbait refers mainly to the title of the article promised, however big part of it is the image itself. This works because it includes some kind of intriguing hint which makes us thirsty for more, however leaving the user dissatisfied with the content.

In his paper over clickbait detection, Potthast [35] (it refers only to text however the concept explains images as well) explains how clickbait works: it is widely attributed to teaser messages opening a so-called "curiosity gap," which increase the likelihood of readers to click the target link to satisfy their curiosity. Loewenstein's information-gap theory of curiosity is usually cited as the psychological reason on clickbait success: "the information-gap theory views curiosity as arising when attention becomes focused on a gap in one's knowledge. Such information gaps produce the feeling of deprivation labeled curiosity. The curious individual is motivated to obtain the missing information to reduce or eliminate the feeling of deprivation." [36]

2.2.5. Personalization

With the development of social media and general ability to recover information about users, new models which take in consideration external information have been developed. Many auxiliary data source can come into play when to model a user: social media feedback, click and browsing behaviour, etc. The most recent approaches especially include the topic of context aware recommendation, which can depicted from the available user metadata. The most basic way of adding a model is by adding user profiles: the user is required to create and maintain the details of what interest him the most. This can be a straight forward way which can enchant on already existing recommendation methods. As example, this research [20] has showed that content-based filtering can successfully use this paradigm. However the effort required from the user may lead to the system not being used at all. Another no-

table example which try to keep user awareness out of the equation is Google news search [37] which makes use of user clicks to create a model of interests.

Although promising, personalization was not implemented in this thesis work for several reasons. This is a experimental work with the objective of testing the hypothesis rather than reaching the best possible CTR, and a working personalized system would have added little to no benefit to our purpose. Additionally, personalization has been discarded due to the fact that ORP is reached by only 5% of total plista traffic, therefore making it unlikely that the same user gets constantly redirected to our server. This leads to always have fresh users while hardly retaining old one. The implication is that a personalization system would result in minimal advantages due to the continuous cold start. As a final addition, a recent research [38] has found that personalization in recommendation is not powerful and groundbreaking as previously thought, with contextualization bringing better results.

Table 2.1: Discarded Feature Ranking

Feature Ranking		
Feature	Comments	References
1. Novelty	Hard to detect with the feedback we have	[27], [28]
2. Text	Even if the presence of text does affect the CTR of images, in reality text barely appears in news images.	[2]
3. Global features	Many small local/global features, which are too weak to be of any importance (Contrast, saturation, segments)	

2.3. Image Features

In the tables referenced in this section we present the features that have picked our interest during the literature research. This is intended to be a list to show them, while a deeper analysis can be found in 2.4.

Although the literature found is exhaustive on the effect of such features to estimate the beauty/ interest / appeal of an image, little work was done on exploring their correlation with CTR and the reaction of the user. However, logic would suggest us that there is a direct correlation between beauty / interest / appeal and CTR. This is why one of our assumption (see 1.3) is that such a correlation exists. The following tables are just a brief index for an overview of the considered features. The features shown in Table 2.2 are the one that were considered during this work (although not all picked) for further investigation. A few features were discarded from the begging without much further research, as considered too weak or too hard to be detected, and can be found in Table 2.1

This ranking was done after an intense research in the literature in the related fields. The ranking was created following the number of reference found and the reliability and strength of the claims shown in the papers. Next section dives into the analysis of each feature.

2.4. Analysis of Features

In this section, for each feature, we describe the feature, cover the related work and then provide a technical analysis. The technical analysis is used to inform our choice of features for the approaches developed further along this work.

2.4.1. Face

Interestingly, Cheng [2] brings out many important features that can bring a user to click on the image, thus increasing the CTR. Faces have been found to be an important factor; however surprisingly the presence of one single face has a better CTR than many faces. Users might be interested in faces because we are inherently ”social animals”. Another explanation could be that single faces in news

Table 2.2: Feature Ranking

Feature Ranking	
Feature	References
1. Presence of people / Face	
• General reference	[39], [26], [33]
• Only one face is better than many faces	[2]
• Gaze direction	[40]
2. Saliency Map / Object Detection	
• Less Points of Interest (POIs) easier to be understood	[2]
• POIs in the centre	[33]
• Background to POI ratio	[2]
• Aesthetics as saliency map with low complexity and low compression	[28]
3. Size / Quality of Image	
• General reference	[29], [39]
• Images small and complex are hard to process	[41]
• Occlusion negative correlation	[39]
• Pleasantness	[29]
2. Website Related	
• Position score	
• Navigation models	[42]
• Color of the website environment	[43]
2. Simplicity and Familiarity	
• General reference	[44], [28]
• Simpler color / color harmony	[2]
• Minimum level of complexity, abstract not appealing	[27]
• Familiarity	[45], [29]
• Copying potential	[27], [29]
2. Miscellaneous features	
• Number of connected components	[2], [33]
• Illumination attribute	[26], [2], [33]
3. Composition (not strictly defined)	
• General reference	[39], [44], [28]

images might often be of a famous and renown person, therefore triggering the interest of the user through familiarity. This problem is analyzed in section 3.5.3

Technical Analysis:
Detection of faces with state of the art algorithms have become a common task, for example Viola and Jones [46]. However, gaze direction may be tricky and not so straight forward to gain [40]; therefore it has been discarded for this work.
Challenges include the selection of the right classifier for the task and choosing if side faces needs to be included as well along frontal faces. The biggest challenge for this feature, as for other features, is the computation time required to process the image as a whole. The fact that it has to be done only once per article can make the whole project doable. The idea is to lunch the parallel computational task whenever a new update is received.

2.4.2. Saliency Map

Saliency maps are a mixed blessing, as they could be really helpful for our task, however there is no clear rule or preferred way on how to create them. In the news recommendation environment, the

gaze given at the thumbnail of an image from a user usually lasts for a fraction of second, only gaining relevant information from the salient parts of the image. The website itself could be possibly treated as a saliency map as well, although this is not done in this work.
Good and useful hints for appeal and interest over image can be gathered, especially in the world of advertisement [33] [2]. As example, Cheng [2] shows that the change of point of interest distribution and rate can strongly affect the CTR. Other analysis of saliency maps [44] show that visual attention does change depending on the composition of the points of interests.

Technical Analysis:
This task is divided in two subtasks: 1-how to obtain the saliency map and 2-what features to extract from the generated map.

Saliency Map generation:
The initial approach was to make the saliency map from starting from the brightness, however the results were not good therefore another approach was required. A few other interesting algorithms were investigated, and the final decision was made to use spectral residuals [47].

Feature Extraction:
A few basic computations can be done over the image, namely: number of points of interest, ratio POI(Point of Interest)/background, position of the points, and a few others [33] [2]. According to the cited papers it is directly linked to a good improvement of the CTR. Many other features are presented in the papers, however only the one that showed the most promising results were selected to be investigated further.

2.4.3. Size/Quality of Image

Quality of image is one of the most referenced features for generating interest and attention over an image [39] [29] [41]. Quality is not strictly defined, and it changes depending which was the purpose the displayed image in first place, however common traits can be extracted, as the one which quality is strictly connected to the size and the compression of the image itself.

Technical Analysis:
A few features included in the broad term ”quality of image” are already part of other topics (example: ”occlusion” is part of ”Saliency Maps” with the interest points ratio and position). An interesting thing to look into is the relation between simplicity and the size of the image [41]. Of the many papers found during the literature research which talk more in abstract terms of ”pleasantness” [29] (many which come from the psychology field), not one of them provides a actual way on how to gather it; therefore we have no actual or sound computational way of coming up with the feature starting from an image. This not clear definition has lead us to prioritize other features first.

2.4.4. Website Related

Part of the trick of catching the eye of the user is how the news portal websites presents the related articles (recommendations). The place, size and way of presenting the images are to be considered meta data of the image itself.

Technical Analysis:
The ORP items (articles) have the domain as a field: it could be possible to go and examine all the domains websites, especially on the topic of color [43] and possible navigation model [42]. An even better approach would be to make a ”score” in order to rank how well the providers are showing their recommendations, since a highly visible recommendation is more likely to be clicked. However the reality is that we have no more information other than the domain, therefore we don’t know where the recommendation is shown, or in which widget (there are usually more than one). This, coupled with the fact that the layout changes drastically across all domains, heavily reduces the reliability of this feature.

2.4.5. Simplicity and Familiarity

Simplicity has been addressed as one of the main generating causes of interest [44] [28]. At its basic, simplicity can be seen as an easy use and disposition of colors and points of interests. However too much simplicity can bring the opposite effects [27]. This feature is evenly more stressed in the news environment, as images and pages are browsed really quickly. An average user lay eyes on a single image for a fraction of a second, disregarding it if too complex to catch the eye. Soleymani [29] finds an interesting relationship between familiarity and complexity: if the user is familiar with the image depicted, complexity positively influences interest. Familiarity attracts users, as we are usually attracted to what we already know. This is connected to the inherent interest we have when evaluating image for copying potential [27], [29].

Technical Analysis:
Although not explicitly said in any papers, this feature could be possibly extracted from the generated saliency map 2.4.2.
Gray simplicity maps can be created following Azimi [33], who claims that the proposed gray level features are very effective in predicting the CTR of ads. Similarly can be done with color, as Luo shows [48]. Familiarity cannot be found through multimedia features, therefore it is discarded.

2.4.6. Miscellaneous features

Advertisement empirical studies have showed that both the number and the size of connected components in the image matter for the CTR [2], [33]. The same studies and a few other related to interest in images [26] looks over the illumination and brightness attributes of both parts and the whole image.

Technical Analysis:
Although not really crucial and important, these smaller features can bring same more information and CTR to the project. Techniques that looks over this smaller features are usually trivial and well documented. However the effect of such small features has to be proven

2.5. Selection of Features

Not all the above features were adopted in order to be implemented. The two main limiting factors were the following:

- Time Constrains: Each of the above features required quite a big effort both to set up properly and for implementation and testing. With a limited time frame, not all feature could be selected.
- Online Environment: this is a tradeoff between completeness and speed. The more features are added to be computed, the slower the algorithm gets when processing images. If overloaded too much, online testing would be not feasible anymore, as the errors caused by the algorithm delay would be greater than the CTR increase.

Due to the above, only two features have been selected, namely Face Detection and Saliency Map. These two were chosen above the others as regarded to be most important factors according to the literature research in section 2.4.

2.6. Initial Checks

The following section goes over a few preliminary checks that have been done before the implementation work began, in order to ensure that the hypothesis we were going to test were sound.

2.6.1. User Reaction Check

This little experiment was done in order to do a "sanity check" over the idea that the selected features in section 2.5 were good factors to test. The aim of the check is to show images to users to see if people are sensitive to the images with the specified features. This was intended to be a small check that the research was headed in the right direction, rather than an experiment in itself. Instead of testing each single feature alone, which would have required a high number of participants due to the weakness of the effect of the single feature, we decided to check if user do chose randomly in respect to interested for an image. If users presented with selected images with the specified features

would not chose randomly, it would mean that the image features have an impact on the generation of interest for the image in the person.
A dataset was extracted mainly from the ORP domains (see 1.4.1) and some other widely known news providers: Guardian[49], BBC [50] and The New York Times[1]. This last three domains have been added in order to add variety to the possible news images, as the ORP domains are sometimes too specific (technology and sports mainly). Each image in this dataset was processed and labelled with the features found: face vs not face, salient vs not salient. The dataset was created in order to have an equal class of labels: 20 for each combination of the features.
Five participants were asked to rank the images from the most ”interesting” to the ”least interesting”. It was clearly stated that ”interesting” referred to the fact that they would be interested in know the news story behind it and therefore click on it if presented in a newspaper website. After the data were gathered, a Mann–Whitney U test (as well known as Wilcoxon rank-sum test) test was carried out in order to asses if there was a significant difference in the distribution between the images with the faces/saliency features and those without. The Mann–Whitney U can test if there is a significant difference in the ordinal distribution (ranking) between two given distributions created by the user.

- H_0: the distribution of scores for the two groups are equal
- H_1: the distribution of scores for the two groups are not equal

The two groups tested here are one the images with features (faces and saliency) and the other the images without features.

Table 2.3: User Reaction Check Results

Setting	H_1
Sal+Face	3/5
Face	2/5
Sal	3/5

As can be seen in Table 2.3, in 3 out of 5 people a difference has been found in the distribution, therefore they were more interested in salient and images with faces. This result is relevant: the tested features (faces and saliency) do have a significant impact on the generation of interest toward images.

2.6.2. ORP image availability

An important and required attribute for the research to succeed was to have enough images in the ORP in first place. The ORP is the platform where our algorithms (and therefore our hypothesis) are tested: a substantial number of images with the selected features (faces and saliency) are indeed required. A small test to check how many images and which features were in day of traffic on the ORP was set up. The results can be seen below in Tables 2.4:

Table 2.4: Updates

Total Updates	100%
Updates without images	10.8%
Updates with images	89.2%

Total Image Updates	100%
Updates only with Faces	22.1%
Updates only with Salience	6.6%
Updates with Faces and Salience	1.6%
Updates with Faces or Salience	30.2%

This is enough images, images with faces, images with saliency, images with faces and saliency in order for them to generate an effect because of these features.

3

Approach

3.1. Image Classification

Our approach is based on a straightforward binary image classifier, which classifies the image of the target item (thumbnail) as either ”interesting” or ”not interesting”. We felt the need of using this naive approach versus a more complex classification due to the easiness of verifying results.

The approach can be summarized simply as follows: According to our research an image is interesting if it either have:

- The presence of a person: A single central person (portrait) is preferred over multiple people all over the image
- Saliency map: A single cluster in the middle of the image with a flat background. A single object is preferred over multiple objects

3.2. Algorithms

Our approach was designed to validate our hypothesis that images impact user clicks on recommendations rather than reaching the best possible performance or CTR. This was done by comparing baseline algorithms with their own ”image enhanced” version. The difference in performance between the two can potentially show us the effect of the images on recommendations. The algorithms were developed and tested in the two different environments: Online (ORP) and Offline (data downloaded from ORP). The algorithms developed and tested are the following:

- Online: Baseline1
- Online: Baseline1 + Faces
- Offline: Baseline1
- Offline: Baseline1 + Faces
- Offline: Baseline1 + Salience
- Offline: Baseline1 + Faces + Salience
- Offline: Baseline2
- Offline: Baseline2 + Faces + Salience

Baseline1 is a Popularity with a freshness windows of 100 items, while Baseline2 is Random based with the same freshness windows. For the remaining part of the paper these two algorithms will be called Pop100 and Rand100. These algorithms will always prioritize the images with the specific features for recommendation on top of their baseline. As example Pop100+Faces will always recommend by picking top popularity, and prioritizing the ones that features faces. Faces+Salience will prioritize images with both features over the one with only one. By looking at the difference between the image enhanced algorithm and the relative baseline we can understand the effectiveness of image-based recommendation in the news environment.

3.2.1. Baselines

Two baselines were chosen for this work: Popularity-based recommendations and Random recommendation.

-Popularity was chosen as baseline as it is considered to be a ”strong baseline” in the news environment, as it has the best performing CTR. Thanks to an in depth discussion with the NewsREEL organizers, we found out that the best currently performing algorithms in the state of the art are same sort of popularity based recommendation. This is a really challenging baseline, as its CTR performance is already quite high. In this work only the basic version of popularity has been tested, however it should provide quite a good insight even for ”refined” versions. The danger of this baseline is that since it is already quite strong on its own, it might hide the effect which images bring to the table. Randomness, as an second alternative baseline, was chosen to reduce the strength of the baseline to the minimum (it is a ”weak baseline”) and therefore be able to test only the effect of images without a large impact from the baseline.

3.2.2. Online vs Offline Testing

The reason behind the adoption of both Online testing, through ORP platform serving recommendations to real news providers, and Offline testing, through recorded offline data, is because they complete each other. While online testing provides a real world opportunity to really benchmark our finding, it has numerous drawbacks which require much time and effort to overcome (harder implementation, much bigger technical challenges, domain specific requirements). On the other hand, Offline testing allow us to test our algorithms on a large amount of data without the time constrain, however the absence of a real user reduce the effectiveness of the feedback. More in depth analysis of these two settings and their respective implementation can be found in the online 5 and offline 6 sections.

3.3. Algorithm Logic

Even if the algorithms deployed in the online task differed from the one in the online, the logic behind them is quite similar and can be summarized in the two specific functions: Update and Recommend.

Update:
A freshness window for each combination of category/domain is created, each window encompassing 100 items. Every time a new update comes in, it is processed by taking the url_img field and scraping the corresponding image from the website. Features for the image are computed with our image processing algorithms, namely Viola-Jones [46] for face detection (see implementation 4.2) and spectral residuals [47] for the saliency map (see implementation 4.3). The saliency map involves the extraction of several sub-features (e.g., number of objects and their positions, background to foreground ratio) which are then used to detect if the image satisfies the requirement of being a single cluster in the middle of the image. This newly processed item is then added to the possible recommendations list, while the oldest item in the list is discarded (if full). A pseudocode of the Update process, which is called anytime a domain sends an update to an article, can be seen below

Process `Update(`*id,domain,category,url_img,...*`)`
  $fresh\_wind \leftarrow$ getFeashnessWindow($domain, category$);
  `/* scrape and process the image */`
  $im_properties \leftarrow$process_image(url_img);
  `/* create article Object */`
  $new_article \leftarrow$process_update($id, im\_properties, \ldots$);
  `/* update freshness window */`
  if *size*($fresh\_wind$) == *100* **then**
    deleteOldestItem($fresh\_wind$);
  end
  $fresh\_wind$.add($new\_article$);

Algorithm 1: Update Process

Recommend:
The recommendation is triggered by a request communication from ORP for N items. The recommendation is expected by ORP as an answer shortly after sending the request, along with the N items. The specific recommendation implementation differs for each algorithm listed above in section 3.2.
For the Pop100 algorithm: These items are sorted by a popularity score, which is an aggregation of how many impressions the item has received plus how many clicks it received in previous recommendations. Whenever a recommendation request arrives, the top N items are selected and only picked if they individually satisfy the "visual requirements" (see 3.1). If not enough items have been gathered before the top C (see 3.3.1) elements have been considered, then standard popularity is used instead, without taking in consideration the "visual requirements" in order to fill the remaining spots. For the Rand100 algorithm, the logic is the same, however the ranking step is replaced by a random picking of items. The first C random times the item will be picked only if it satisfies the "visual requirements", after C times this restriction decays. A pseudocode for Recommend process, which is called anytime a domain sends a recommendation request, with the Pop100 baseline is shown below:

```
Process Recommend(rec_request_N)
    domain, category, N ←process_request(rec_request_N);
    /* get correct freshness window                                    */
    fresh_wind ←getFeashnessWindow(domain, category);
    /* sort based on popularity                                        */
    sort(fresh_wind);
    /* create list with N recommendations                              */
    for article in fresh_wind do
        /* the first C articles are added only if satisfy the visual
           requirements                                                */
        if article.satisfyVisualRequirements() then
            rec_list.add(article);
        end
        /* exit For loop if you reach C or the maximum number of
           recommendations N                                           */
        if counter >=C or size(rec_list) >= N then
            break;
        end
        counter + +;
    end
    /* if N is not reached complete the list with the top popularity
       articles which has yet not been taken)                          */
    if size(rec_list) < N then
        completeList(rec_list,fresh_wind,N);
    end
    return rec_list
```

Algorithm 2: Recommend Process

3.3.1. The variable C

The constant C can be interpreted as a tradeoff between "being interesting" and "following the baseline". In case of Pop100, the smaller C the most the items will be popular and less "visually interesting". This variable has been determined from empirical testing, by initially looking for the minimum number of recommendations that needed to satisfy the"visually interesting" requirement. However, since our intention here is to test if the visual component has an effect, C has been intentionally exaggerated during experimental testing in order to make the effect more notable.

The size of C was chosen in order to usually have the full number (N) of recommendations composed by faces/saliency. Since the usual required number of recommendations is around 5, C needs to be big enough to have around 5 articles with faces or saliency. In order to calculate C, we looked at the average number of articles with faces and saliency. This was done by simply collecting all the articles in a small dataset (the same as the initial experiments in section 2.6.2) and looking at its results

Experiment has been run on a small dataset of several non consecutive weekdays:
Total: 13413
Saliency: 879
Faces: 2967
Saliency and Faces: 204

Therefore the probability of having a ”interesting image” (either with saliency, faces or both) is 30.2%. In order to get on average at least 5 (the average number of required recommendations) hits in the top C recs, C needs to be of size of 16,6 (which makes it 17).

3.4. Resizing

The field extracted from the item (article) which reference the image shown as the thumbnail in the recommendation is a link which shows the full size picture. The downloaded picture is usually much bigger in resolution the the actual showed size. In order to speed up the computation and recreate the actual scene which is seen by the user we need to reduce the images to an appropriate scale. However, as stressed in the problem section 3.5.2, it is not clear what size the images end up being. Therefore a brief empirical study over the served domains 1.4.1 has been done in order to assess the average size of the thumbnails in the recommendation sections. This has been done over the specific domains of the ORP (see 1.1). Initially a plan to include a few extra domains to increase flexibility was intended, but upon seeing an extreme variation in sized, only the ORP domains were selected.
The following are the sizes of the most common online thumbnails. This are the size of the image shown on the domain, which is the the size shown to the user in a desktop visualization of the website.

- 250 ×250 (common)
- 320 ×100 (less common)
- 119 ×67 (less common)
- 300 ×150 (common)

So the average image is something around 250×150. The resing has been done in a way which prevents the proportion to be broken: the biggest dimension (height or width) is resized to 250, while the other follow accordingly.

3.5. Theoretical Limitations and Problems

3.5.1. Feedback

The biggest limiting factor encountered during this work is the limited availability of feedback, as the only feedback that can be extracted from ORP and the offline data is the user click. The limited access to plista’s stream is very constrained in the types of recommendations (sports and news); plista, being a recommendation and advertisement company, cares only about CTR and not about users.

All conclusions are based on the only truth we can extract: CTR. CTR is a good representative of the interest and the intention of the user, although biased [51]. Only the effect of the click can be detected, not the reason for it. No more in depth analysis of the satisfaction of the user can be made, as no other parameters (time a user stay on the page, satisfaction) can be extracted. This is one of the major issues with the results as all our findings are based on a limited feedback.
Limited feedback can be dangerous for the long term user experience. Even if something produces more clicks in the immediate future, it doesn’t mean it fully satisfies the user. Clickbait articles are the example: they do generate lot of traffic, but the user is always left dissatisfied with the content. This could be dangerous, especially for small content providers which rely on a small number of loyal costumers which are interested into the specific topic, and might get dissatisfied when valuable content is replaced with ”more appealing” but less worthwhile content. This limited feedback issue is a more general extremely complex weakness of current recommender systems evaluation method rather than only in this specific domain.

Figure 3.1: Different widgets for recommendations

3.5.2. Image Size

Problem: we don't know the size the images are shown. Our recommender can recommend something on the ORP platform however it is not known exactly where it ends up: we know the domain and the "widget number", however we have no way of knowing which number is which widget on the page layout (example in Fig. 3.1). When being asked, plista answered that they do not know as well, as they just know the persistence of the widget number (recommendation of same widget means apparition in the same spot). Therefore no information about the size of the image shown can be used; this could have been quite helpful, as according to our literature research (see 2.4.3), the bigger the image the more complex depiction the user can "tolerate". A bigger image meas that is more notable as well, which will probably lead to more clicks.

3.5.3. Familiarity

A theoretical problem encountered is the familiarity problem. We have been looking at the face feature as a standalone when it comes to the effect it brings. The hypothesis we formulated states that the presence of a person in an image makes the image more interesting, therefore users will likely click on it more. But is the added interest brought by the presence of the face or because that face might be someone familiar to the user (a star, a famous person)? In news related websites, an image with a single face facing the camera have a high chance of being of some renown or popular person. Sadly, there is no way to understand the difference between the two from the limited feedback we have: we cannot know why a person clicked on an image. Although to our ends of proving our hypothesis this difference doesn't interfere much (as we only rely on clicks without knowing the "why), it does preclude us to have a deeper understanding of the reasons which move the users to behave the way they do.

3.5.4. The Weather Problem

With the term "weather problem" we refer to the dynamic nature of the user interest. News consumption varies across the world depending on the events and the society interests on a daily basis. This temporal change has been well studied and addressed in the Web queries semantic context [52], as Web search is strongly influenced by time. The queries people issue change over time, with some specific queries occasionally spiking in popularity due to current events. Examples are are a spike in search about earthquakes the day after an actual earthquake shacked the earth, or events following the normal seasonal changes as 'black Friday'. Much study has been done in the web search domain, and methods to avoid and detect such spikes for semantic correction have been developed [53]. In a recommendation system the problem of dynamic change of interest is the same. Although the semantic understanding of the object is not as a pressing problem as in the web search, this abrupt change in user interests might lead to a drastic change in what users are interested to click. If not properly addressed, the resulting CTR might be strongly affected.

4

Implementation

4.1. Initial Setting

4.1.1. Hardware

In order to set up experiments, benchmark findings, test algorithms etc.., more than one running machine was needed. The following were used:

1. Personal Laptop: development of algorithms
2. TUDelft Insight cluster VM: main server, used to deploy algorithms and test them against ORP in an Online environment.
3. Plista VM: backup server, used to evaluate offline runs, mainly for Offline tests. Late addition to the project.

Initially only the personal laptop was used, shortly followed by the TUDelft server. Quite a few problems arose during the used of this server, therefore a third additional server was required. This last server was provided by plista.

4.1.2. Initial Client and Recommenders

An initial working client and attached base recommenders was obtained from recommenders.net [54]. The code can be found at the github repositories for the Client [55] and for the Recommenders [56]. The Client is needed to interface with the ORP platform and to support the deployed recommender algorithm on top of it; only minor fixes have been applied from the initial version throughout all this thesis work, mainly to support image scraping. The Recommenders are a collection of various recommendation algorithms which have been mostly discarded but for the popularity-based algorithm. This has been taken and used as baseline (see section 3.2) with only minor tweaks.
The initial popularity recommender works with a recency windows of 100 items. These items are sorted in popularity score. The score stacks and is based on the number of times users interact with the item: 1 point from every user landing, 5 points for every click to the item recommendation, 3 points for every click on another item which is visualized in the current item recommendation. From this initial algorithm, our version was developed which can be found in section 3.3

The programming language used by the already provided client and recommender was Java, therefore this project has been using Java for all the development which is related to this field: ORP communication and client, Image scraping, Popular and Random Recommenders. The analysis of the images however has been done using python, for its easiness in dealing with multimedia data and its clear and well documented interface with OpenCV.

4.1.3. Obtaining the Images

Images links could be easily obtained by extracting the relative field from each item (article) in the ORP platform. This is done every time a new update or a new item comes in the ORP. Once the link is obtained, the image needs to be downloaded from the domain hosting the file (which usually is the news

provider itself) to our server, as ORP does not keep the files. However scraping the images from the links has revealed itself not trivial, for mainly two reasons: ip black listing and 2-empty or incorrect links.

Black listing:
Many domains use protections against DDOS attacks by looking for recurring IP. If an IP makes many requests in less than the normal "human" time, the IP is denied service for some time (usually 10 minutes). This means that if our server acts too fast for a human, it gets blocked!
Lucky, this problem has not been detected for online testing, as updates and new items are not extremely frequent and are divided between different domains. However, this proved to be a source of missing images when testing offline, as updates and new items were coming in much faster (at the rate of reading a file) and therefore our server IP got blocked. This has been solved by putting pauses in the code once in a while.

Links:
Often image links got out of items are missing, broken or just leading to nothing. This has been solved by created a list of broken and fake links (many of which were recurring) which was used to filter out items which were not meant to have an image in first place. As example, one minor domain in ORP always put a link to their homepage whenever they didn't have an image in the article.

4.2. Face Recognition

For this task we used OpenCV library, which makes already available a Haar features Cascade classifier for faces (following the Viola Jones algorithm [46]). Since the task of detecting faces is not trivial and depends on many factors, many parameters needed to be tweaked in order to get an optimal detection. A dataset composed by one hundred images coming from news websites had been created in order to empirically calibrate the parameters, to choose the face model and to calculate the accuracy. The images in the databased have been manually labeled as face/not face.

As for the choice of the models, OpenCV makes already available a trained classifier for the following models: $haarcascade\_frontalface\_default$, $haarcascade\_frontalface\_alt$ and $haarcascade\_profileface$. The documentation is lacking and briefly describes them as specific for different images typologies, however no model can excels with such a big variety of pictures (of the news field). Therefore the empirical check is necessary to assess which one (or which combination) performs best under the variety of news images.

4.2.1. Results

The results of the testing over the 100 images for the face profiles can be seen as confusion matrices in Table 4.1. The profile adopted for frontal faces is $haarcascade\_frontalface\_alt$, while the profile faces detection was discarded because of the poor performance.

Table 4.1: Confusion Matrices

		Prediction outcome		
		p	n	100
actual value	p′	20	1	21′
	n′	14	65	79′
	100	34	66	

Table 4.2: haarcascade_frontalface_default

		Prediction outcome		
		p	n	100
actual value	p′	11	2	13′
	n′	23	64	87′
	100	34	66	

Table 4.3: haarcascade_profileface

	Prediction outcome		
actual value	p	n	100
p′	9	0	9′
n′	25	66	91′
100	34	66	

Table 4.4: haarcascade_frontalface_alt_tree

	Prediction outcome		
actual value	p	n	100
p′	22	2	24′
n′	12	64	86′
100	34	66	

Table 4.5: haarcascade_frontalface_alt

The same experiment setup was used to calibrate parameters, for example the scale and neighborhood parameter. This is needed since the type of pictures that can appear on an online newspaper is not uniform, and faces can be really up close or small and distant depending on the context. Therefore the best solution has been to empirically optimize the scale of the faces on an average of images. After the experiment the best scale factor resulted to be either 1.3 or 1.1 depending on the classifier

As can be seen from the confusion matrix, generally Type I error is always smaller than Type II error. This means that is mostly common for the classifier to not find a face rather than to find one where there is none. This is usually due to the typology of pictures belonging to the news category: when there are faces it is either a full single face facing the camera (celebrity, politician, interview) which is easily recognizable, or a group picture of faces not facing the camera, as people are not posing for the picture (crowd with action, car crash, riot). The latter is usually harder to be classified as face, therefore generating a rather big Type II error. If we couple the above with the fact that one face is better than many faces (as explained in section 2.4.1) from a CTR point of view, we can see that the Type II error is not such a big problem in comparison with the Type I. The models and the parameters have been chosen accordingly to reflect this imbalance between the errors.

4.2.2. Face Recommender

A first implementation (before the current algorithm logic presented in section 3.3) was done using a ”score amplification” for images on top of the popularity. The initial adopted algorithm had already implemented a score for the popularity baseline(see 4.1.2): whenever the article had a face detected this score got ”amplified” $*2$ if with one face, $*1.5$ if with multiples. The ranking stayed the same, therefore images with faces got boosted in score and therefore more likely to be on top of the ranking (and be picked for recommendation). This first version run only for a few days and it was made to see if things were working properly. The obvious downside of this approach is that it is just a tweaked popularity algorithm: there is no guarantee that there will be faces recommended.
The new version was implementing the C variable as explained in section 3.3.1, in order to ensure the maximum contrast between the baseline and the condition using visual features. While with the previous implementation we could run the risk of never recommending articles with faces if they were barely popular, now in the moment of choosing the top recommendation the algorithm looks into the first C top recs and chose the one with faces. If the limit is not hit, the most popular one are taken.

4.3. Saliency Map

The objective of the saliency map is to detect how interesting the image is. This is done by looking for a few factors 2.4.2: an image with a single object in the centre with monochrome background tends to attract more the eyes than a confused and too complex image, especially in the variety of a news recommendation.

The process over the image is the following:

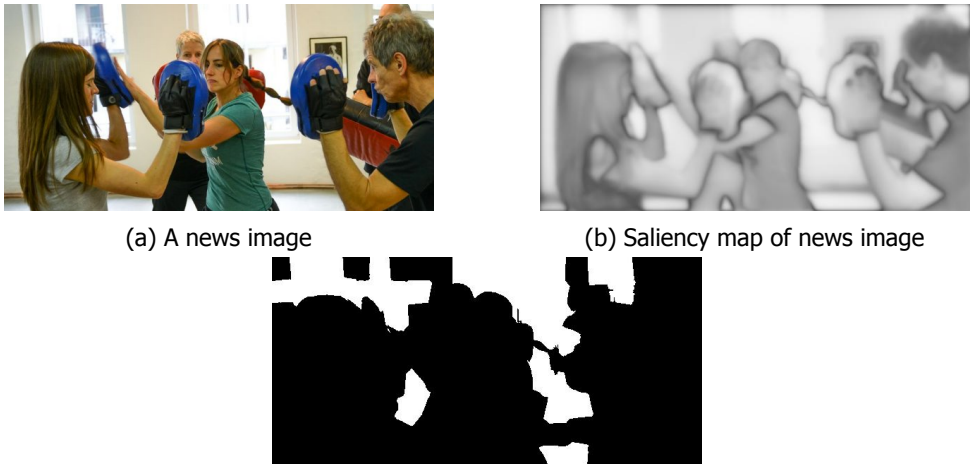
1. create saliency map
2. threshold saliency map to create a binary Object/attention map
3. extract features from the binary map
4. combine the features in order to get to a final classification: ”salient” vs ”not salient”

The following features were chosen over multiple, especially taking in consideration the work of Chen [2], and balancing out CTR and implementation workload as parameters. The features selected and used in the remaining part of this work are the following:

- foreground/background ratio
- number of components (number of salient objects)
- centre of mass of the objects

There are multiples ways on how to get a saliency map. A first approach was done using the brightness of pixels (see 4.3.1), however the limits of this method were quite clear. A better version was implemented soon after, following the spectral residual technique (see 4.3.2)

4.3.1. Brightness method



(a) A news image (b) Saliency map of news image

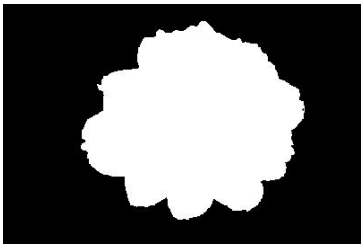
(c) thresholded

Figure 4.1: A figure with two subfigures

This first implemented method was a quick and easy approach, however it revealed itself as a ”mistake” during the thesis process, where we blindly trusted the literature without testing on our specific domain first. The Saliency map was generated from the contrast and the brightness of pixels. This was tested on a standard set for saliency from the literature, usually with objects highlighted in the middle (light pointing at it), and the results were great. However it performed extremely poorly when deployed on news images, which are more diverse and light is not staged. The results were showing as ”interesting parts” even walls and backgrounds.

As can be seen from examples, in Fig.4.2 the binary map seems good as it is part of the ”standard set”, however when deployed on news image, the results were dissatisfying (Fig.4.3). The whole process with the intermediate step of the saliency map can be seen in Fig.4.1. Although many different thresholding methods were tested, none really performed in an acceptable way, therefore it was decided to try a different approach to the creation of the saliency map.

(a) Original Image



(b) Saliency thresholded map

Figure 4.2: Saliency set Image

(a) Original Image

(b) Saliency thresholded map

Figure 4.3: News Image

4.3.2. Spectral Residual method

Our attention shifted to a method which could detect the change of the pixels, therefore highlight the interesting parts. This was found by following another algorithm for saliency detection, namely Hou [47]. This method is based on the spectral residual method. The spectral residual is defined as $\mathcal{R}(f)$

$$\mathcal{R}(f) = \mathcal{L}(f) - \mathcal{A}(f)$$

where $\mathcal{L}(f)$ is the log spectrum of the image and $\mathcal{A}(f)$ is the average spectrum. The average spectrum is calculated by convoluting the image with a local average filter.

With this model, the spectral residual contains the innovation of the image. The Inverse Fourier Transform is used to get back from spectrum to spatial domain, by creating the resulting *saliency map*. This generated map is how much the area is "new" and unusual with respect of the rest of the image. This map is then thresholded to create the *object map*. The paper suggests that the best empirical threshold found from extensive testing was $threshold = E(\mathcal{S}(x)) \times 3$ where $E(\mathcal{S}(x))$ is the average intensity of the saliency map.

After the creation of the binary map, the features listed in section 4.3 are extracted from this map. This method is great to find uncommon things which attract the eye. However, given its functioning, it naturally does not detect whole objects, rather interesting parts of objects. If the object is small, it is detected as a full object as it is a change over the rest of the picture, but if the object is big enough it is not detected as a whole body and instead its most interesting parts are highlighted, making it fragmented in patches. The overall results are quite great, as all the objects are detected, even if the part detected might be smaller than the whole object itself, or multiple parts are detected in one object. This fragmentation is excessively stressed in news images, where there is usually not defined hierarchy, and multiple objects tends to be revealed as interesting.
In order to avoid this problem, a filtering technique has been implemented to clean the image of unwanted and noisy patches and improve the detection of the single objects.

4.3.3. Filtering

Filtering has been necessary to isolate the real interesting part that usually corresponds to the parts of objects which attracts the eye. As news images tends to be really varying in form and shape (see

(a) A news image

(b) Saliency Map

(c) Object map

Figure 4.4: The saliency map process

the variety and movement of a news image like 4.4a vs the stillness of a staged shoot like 4.8), so saliency detection might be too fragmented in small parts, as seen in image 4.4c.

The rationale behind the filtering is a cleaning of small patches and joining patches which belong to the same object. By using standard morphological transformations over a binary map, we can obtain much better results. Specifically, the following transformations are applied:

1. A Closing with a 2x2 kernel is applied in order to glue together really close patches.

2. An Opening is performed in order to delete the remaining small patches. This leads to the patches smaller than the 2% of the image being deleted.

As results, let's compare the previous image 4.4a. As can be seen, the original object map 4.5a created has many small spots, however after the filtering 4.5b, the attention sections are clearer.

(a) Original Object map

(b) Filtered Object map

(a) A news image

(b) Object Map

(c) Filtered Object map

Figure 4.6: The filtering process

4.3.4. Image Classification

As previously mentioned, the goal of this binary classification is to come up either with a ”interesting” or ”not interesting” class. During this saliency phase, we investigate if the image follows the ”A single cluster in the middle of the image with a flat background” rule. In order to do this, we check if a single or multiple patches are present. If multiple central patches close together are detected, a closing is performed to see if they can be joined in a cluster. Then we analyze the features extracted from the saliency map with a cascade of decision stumps classifiers which check if each feature satisfy the requirements. The classification steps are reproduced here as pseudocode:

```
Result: classification variable isSalient
isSalient ← false;
/* not if too many objects                                          */
if object_count() < N then
    /* check if it is a single central cluster                      */
    isCentralCluster ← true;
    for element do
        if !isinCentralBox(element) then
            isCentralCluster ← false;                  /* if central */
        end
    end
    if isCentralCluster then
        doClosing();                          /* closing to unite cluster */
        /* if now it is a single object                             */
        if object_count() == 1 then
            /* if the ration is between the good values             */
            if min_ratio < getRatio() < max_ratio then
                isSalient ← true;
            end
        end
    end
end
```

Algorithm 3: Saliency classification procedure

4.3.5. Training

”Training” refers to the process used in order to come up with good parameters values used in this technique, specifically the parameters introduced by our approach which were not covered by theory already in the Hou [47] work. These are the filtering and classifications variables as they have been introduced only in this thesis work:

- filtering variables (see 4.3.3)
    1. patch size for closing (to connect same object)
    2. patch size for opening (to delete small patches)
- classification variables (see 4.3.4)
    1. patch size for closing (clustering)
    2. N (maximum patches for cluster)
    3. Central Box (outside is not ”central”)
    4. min_ratio and max_ratio (band for the background/foreground ratio)

This values were determined using empirical testing over a dataset created for this purpose. The dataset was created by downloading 100 images from news portals, namely the ones in the ORP platform (see 1.4.1), The Guardian[49], BBC [50] and The New York Times[1]. This last three have been added in order to add variety to the possible news images.

The training was done in a qualitative way by labelling by hand these images, by checking if they were following the ”central single cluster” rule, and see how the classification performed over those images during the tweaking of the parameters. Since these parameters are quite straight forward to understand and it is easy to see the effect, no additional steps were required. Usually the values of the parameters could be easily guessed only by looking at the implementation, and the training proved that only minor changes were needed. The values are the following:

- filtering variables
  1. closing patch: 2x2 for filtering and 5x5 for clustering
  2. number of openings: variable
  3. size of patches: 2x2
- classification variables (see 4.3.4)
  1. closing patch: 5x5
  2. N: 3
  3. Central Box: see image 5.1
  4. min_ratio and max_ratio: between 1% and 40%

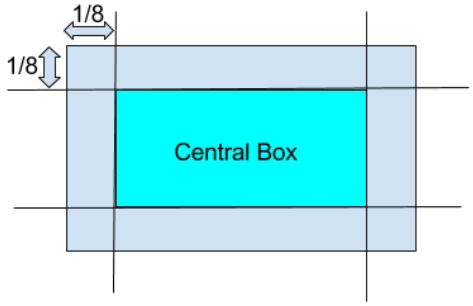


Figure 4.7: Central Box of Image

Although these are probably not the best possible values for the parameters, both the training and the final results showed acceptable results.

4.3.6. Results

The adaptation of this algorithm has proven to be quite successful from a qualitative evaluation. Sadly, there is no real ground truth dataset on which we can test our results, as the saliency / object / interest map varies heavily depending on the purpose of the application. As this is not a plain object detection task, but rather an attention grabbing task, no real dataset has been found. However news images are already quite diverse and different in their domain, therefore a qualitative testing over around 50 images from the news domain (from different categories) have shown more than decent results. This was used to ”test” the final results after training.

As expected, the algorithm performs good when there is a single staged object in the center as in Fig.4.8 . Surprisingly, even when there is chaos and not a uniform background, the algorithm still manages to find the most salient object (player, kick, ball) in Fig.4.9, although the classification fails here.

4.4. Technical Challenges

In this section we describe the technical challenges and annoyances which we were faced with. Many minor however quite time consuming problems were generated by the only initially available server from insightlab TUDelft (referred only as server from now on). The second server (see 4.1.1) was added later in order to get relief from the constant technical problems faced.

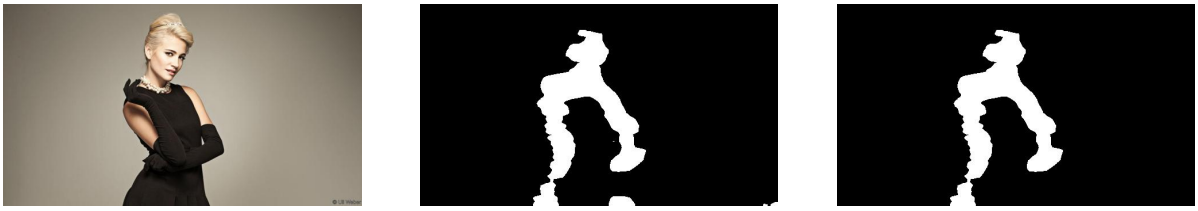


Figure 4.8: Salient: Central clear object

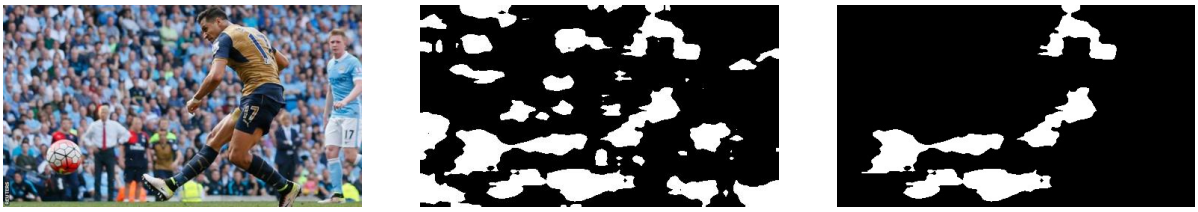


Figure 4.9: Not Salient: Central not clear object

Figure 4.10: Not Salient

Figure 4.11: Salient: Central single object

Figure 4.12: Salient: Central single cluster

4.4.1. Server Set Up

The initial setting up of the server to make it correctly communicating with the ORP platform and behaving the correct way revealed itself not easy as planned. The initial VM which we obtained from TUDelft had only 10GB of storage size, therefore making it quite a challenge as the data outputted daily by ORP are quite bigger. Initial tests revealed that if we intended to save all the data about items, impressions and information sent by ORP (for a more detailed list see 5.1.1) as a plain log file, the storage required was around 2.5GB per hour. The problem was resolved by setting up a database which would filter out most of the unimportant and redundant data which were forwarded to our server. This database however required constant maintenance as it needed to be manually copied to another location (our personal computer hard drive) around once a week to stay below the 10GB limit. The function of the database is explained in section 4.5.1.

Another small problem was the unavailability of the administrator access to the server, which delayed all the deployment, as new and experimental software needed to be approved and installed by the technician of the lab, not always available. As example the database took more than 3 weeks to be set up, as the communication pipeline (request installation, wait installation, detect problems, communicate problems, wait for fixing) was extremely slow due to this restrain. Additionally the server would be extremely unstable due to the small storage (10GB) used as buffer, which would constantly fill up if left unattended for too long and crash the client service. This on top of general occasional slow down caused by other used using the cluster (it is a VM in the TUDelft insight cluster).

4.4.2. Server Maintenance

Since the server used to run the recommender is a TUDelft Virtual Machine, during the weekend of 30/01 and 31/01 the server was taken down for maintenance.
Due to the architecture of the recommendation algorithm, which holds everything in memory, the shutting down could have had serious consequences. Losing the recommendation memory (which was not written or saved anywhere) means losing all the information about all the articles, impressions, clicks, therefore not being able to make an accurate prediction anymore.
In order to prevent this a backup plan was introduced:

- A temporary weaker recommender was deployed in our personal computer, which started with the data retrieved from the saved data from the database. This data were partial, however it was a better start than empty memory.
- When the TUDelft server was reactivated, the memory was filled with the previous database and all the interactions the temporary recommender had gathered

This approached allowed to not totally lose everything that was in memory. Even if the replacement server(our PC) didn't perform good as the real server (500 recs a day vs the 4500 of the TUDelft server), it allowed the recommender algorithm to not fully lose all the data and therefore saving it from a new cold start.

4.4.3. Concurrency

A big challenge was to modify the initial client and recommendation algorithms to be supporting multiple threads meant to solve the problem of having many possible ORP updates at the same time, which imply many parallel computational tasks to evaluate images. This was not included in the initial client and recommenders as they were not dealing with computationally heavy task such image computation. This was done using standard Java libraries, however it wasn't easy to guarantee the avoidance of synchronization problems like the need of having the image computation before the possible recommendation. Part of this problems have been solved by assuming that images are "not interesting" until the thread has finished the computation and outputs the label for the image.

4.5. Server Architecture

Following the initial client and recommender implementations, the architecture developed for our server can be seen in Fig. 4.13. The components are the Client, which handles the communication with the outside world (usually ORP), the recooomender layer which can be switched depending on which algorithm needs testing, the log cleaner script which frees space deleting the logs and transferring the important data into the database.

4.5.1. Database

Our research required the use of offline data in order to test our hypothesis in the Offline environment. These data were obtained by recording all data which were sent to the recommender from the ORP. In order to do this, a database was implemented.
Since the server we had at our disposal was only 10GB of storage, which were filling quite fast due to the extensive data logs produced by the recommender, we decided to move everything to a database (MongoDB) by using a script in Python which parses the logs once every hour, update accordingly the

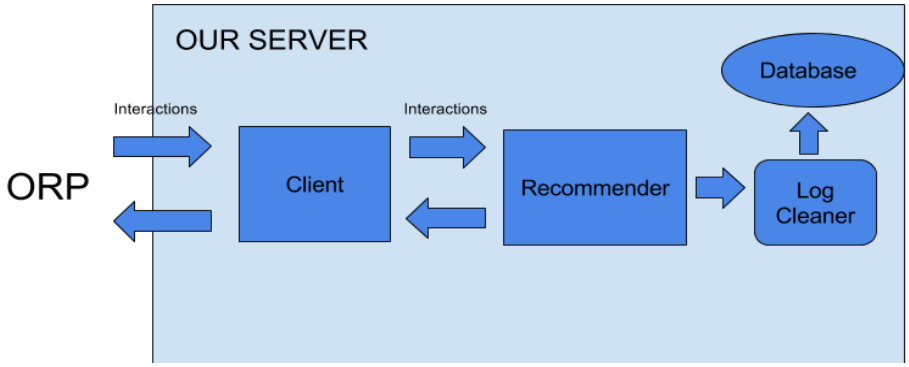


Figure 4.13: Server Architecture

db, and then deletes the logs. This script filters out redundant data and selects only the appropriate. Example is all the overhead info coming from impressions and updates, which details all the information about the sender, however quite useful for our aim.
Although the space required by the database was way less than the plain logs, manual maintenance was required around once a week as the small storage space in the server was filling quite quickly: all data were moved once a week to an external hard drive therefore freeing up space in the server. This solution was adopted over the direct implementation of a database in the recommender since the use of a db could greatly slow down the computation speed of the algorithm, therefore increasing the likelihood of failures due to the 100ms response timeout.

5 Online

The Online Scenario was executed on the Open Recommendation Platform (ORP) by plista. Part of plista's traffic is redirected the ORP. The ORP makes it possible to deploy and test algorithms in a real environment. This platform has enabled us to test and benchmark our algorithms, however technical difficulties typical of the online environment have limited the effect of our results.

5.1. Online Architecture

The news Domain (Newspapers, news channels, etc..) talk directly to the ORP platform. Whenever a domain needs to generate a page when a user requests it, it makes a call to the ORP platform for which recommendation to show. ORP forward this request to our server which generates the recommendation and answer. Only when the answer is received the domain generates the page for the user. This makes timing and fast response a critical part of this system.

Figure 5.1: Online Architecture

5.1.1. How it works

The platform uses HTTP protocol supporting JSON format for data. Communication is handled from ORP by four types of messages:

- Recommendation requests: sent by ORP whenever a user lands on a domain where a page with recommendations is generated. The expected answer is a list of items, which is the recommendations which are going to be shown.
- Impressions: sent by ORP about every navigation action a user is taking. Every page visited from a user is fired as an impression. This can be seen as the user log, as all the actions are recorded.
- Item Updates: sent by ORP whenever a new article is written and uploaded or an old one is modified. This message is the one which carries the image link information.
- Error Messages: sent by ORP whenever something goes wrong with the domains or with our server. An example is the "timeout" error event which gets fired whenever our server is not able to answer in time.

The timeout from ORP on waiting for the response from our server is 100ms: if our system does not answer within this timeframe, the request is counted of having caused an "error"

5.2. Evaluation

The advantage of testing the recommendation algorithm online in a real world environment is that it is possible to have real feedback from the user. The feedback is given in the form of click through, therefore hypothesis can be directly tested and proved/disproved here. Since the traffic redirected from plista to ORP is less than 5%, it is guaranteed that the results are not biased as the users coming are always different and new to the recommendation our algorithm is making.

Testing was done following the A vs B algorithm paradigm, where a baseline algorithm competed against baseline+imageFeatures algorithm. The baseline is a popularity based recommender with only the 100 most recent items counted as valid for recommendations (Pop100). At the time of the evaluation we had at our disposal only one server, therefore testing of the two algorithm in parallel at the same time (and therefore on the same dataset) was not possible. The recommender were deployed one after the other on the same server for the same amount of time. The testing on the baseline Pop100 algorithm was carried out during the month of March 2016. The testing on the experimental algorithm was carried out during the evaluation window for the CLEF-NewsREEL contest, from the 28-04-2016 to 28-05-2016. A visual overview of the results of the contest and our performance can be seen at Fig. 5.2. Sadly, due to unexpected technical difficulties, the saliency image feature was not ready to be implemented into the final recommender at the beginning of the evaluation period, therefore it was not adopted throughout all the evaluation window for consistency. The tested algorithm were

- Online: Pop100 baseline
- Online: Pop100 baseline + Faces

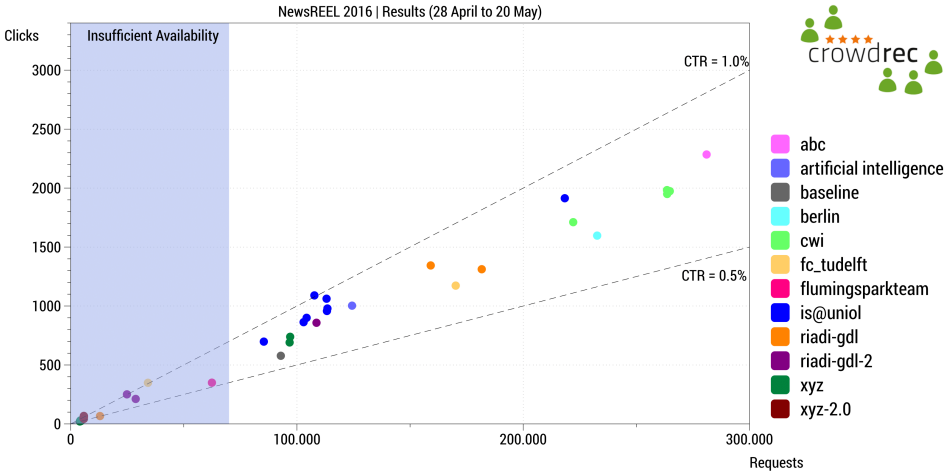


Figure 5.2: NewsREEL Evaluation window results

5.3. Results

The Online results show the data obtained from the scoreboard in the ORP during the evaluation window. Although the evaluation itself has run for around 40 days, not all the days have been taken in consideration due to issues which resulted in the recommender receiving a low volume of requests. As a result, only 24 days have been considered for the results. In order to answer our research question we decided to benchmark our image enhanced algorithm against its own baseline without image information. As for the Online, Pop100 is the baseline.

As can be seen from the image 5.4: the baseline performed better in CTR value over long period of time. The Pop100+Face sees a 28% decrease in CTR over the baseline Pop100. The two recommenders have received around the same number of cumulative clicks 5.3, however the CTR is different: this means that the Pop+Face has received overall more requests. This underline the difference between the different testing timeframes which might have caused the weather problem (see 3.5.4).

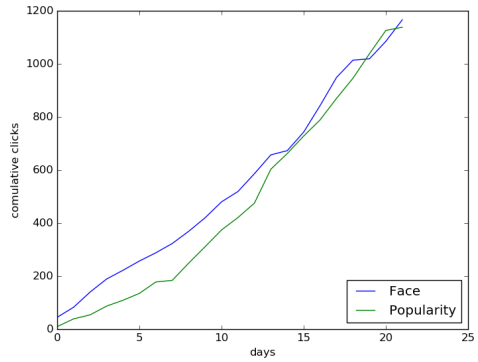


Figure 5.3: Cumulative clicks

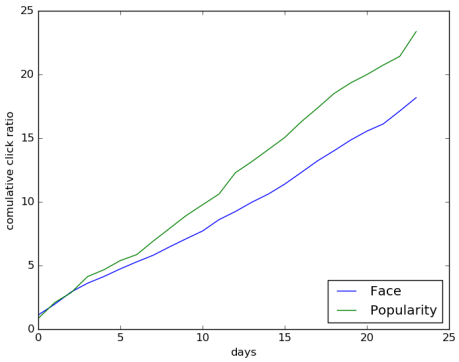


Figure 5.4: Cumulative CTR

Figure 5.5: Online Pop100 vs Pop100+FacesDetection

Our conclusion is that the lower result is actually due to a mixture of technical problems and the weather problem which most likely undermined the performance of the algorithm. A rundown of the problems can be found in the next section 5.4

5.4. Problems

The first challenge encountered was how to deal with images. We decided to intervene on links only provided by update messages, rather than the one provided by the impressions (which are more but redundant).

5.4.1. Timeout

One of the problem encountered was to make the algorithm fast enough to keep up with the ORP rate of updates. The adoption of the update message as trigger for the image computation solved the "too many messages" problem however created another one. While the requests sent by the platform do follow the performance of the algorithm (if the algorithm is struggling less requests are sent), this does not apply to the updates; therefore all the updates are sent anytime. Updates are the "computationally intensive" part in our algorithm, as each update usually comes with an image that needs to be downloaded and analyzed. Updates tend to come in groups of 10 or more, making it necessary to queue them. Therefore it wasn't uncommon that the next batch of updates came before the queue was all processed, making the queue longer and the processing time even longer, thus making the problem worse: if repeated enough times the server would crash and in need of manual rebooting, therefore losing time and going through a new cold start period. Longer queue and longer processing time meant longer delay to answer recommendation requests as well, thus failing due to the timeout time. The strategy adopted to solve this problem was initially to not process all images, by putting a maximum limit on the number of concurrent processes. This lead to a quite big percentage of images laid out (around 20%/30%). This initial solution was implemented only in the early days of the algorithm as a dirty and quick fix, but it was soon dropped. The time resources available for this research were necessary limited and not all solutions to this problem have been explored.

5.4.2. Miscellaneous

Since the point of having an Online evaluation was to have real feedback, this brought along small problems when to deal with real domains. In the specific many small problems typical of the real life environment were faced. Example of notable troubles were:

- A single domain was directing all the requests to a single category. This lead to the algorithm not knowing what to recommend since the category was empty due to the fact that had never received updates, just requests. This was resolved by addressing the requests from this specific domain as general requests rather than category specific. This problem was found only after deep debugging of the logs file as all the recommendation to one domain were failing.

- A few domains didn't host the images themselves, but rather used a third party host for their data. This initially lead to problems as our algorithm was faced with multiple links for images with only one working. This was solved by filtering out the recurring links which were not actually hosting the images.

5.5. Conclusion

From the results presented above, two important facts can be seen:

- Feasibility: The algorithm has successfully run online for more than one month.
- CTR: The performance of the experimental algorithm is lower than the baseline

Although the results show a decrease in performance, we consider this experiment to be successful to our end. The algorithm has been on and running for around one month, so we can successfully say that it is feasible to run this sort of computationally intense recommendation algorithm online. The evaluation itself has run for only 23 days, however the algorithm has proved to be reliable over many more days.

Many factors can be explaining the decrease in CTR performance. We think that the bigger factors negatively effecting the outcome have been two: poor technical implementation and the weather problem (see section 3.5.4) caused by testing of the baseline and the baseline+imageFeatures during two different time slots. A more in depth discussion is described in the discussion chapter 7.1

6 Offline

The Online Scenario was executed by testing and evaluating our algorithm on an offline dataset, by looking at recorded user interactions. Although the testing environment is easily controllable to be shaped in order to test our hypothesis, the feedback obtained from the data cannot be interpreted as the same: we cannot test anymore over direct CTR by offering recommendations as there is no actual user clicking on the article after we recommend. However useful information can be extracted from the behaviours of such users.

6.1. Dataset

The NewsREEL organizers provided a dataset in order to test in the Offline Scenario [57]. However, this official dataset did not have a crucial field which was required by our image-based algorithm: the img_url. Although the field itself is present, the official data set was collected in June 2013 and the most of the links have disappeared since the images are hosted by the domain themselves. Domains tend to remove the items (especially images) after some time of inactivity, by cleaning their databases of old dated articles, as they take much space and do not generate any kind of traffic.

The dataset was generated from two different sources: ORP daily dumps from TUBerlin and daily logged activity from ORP (to our server).

1. ORP dumps from TUBerlin: This is the same structure and data as the ”official dataset” provided during the NewsREEL contest, however with a much recent date. This makes all the links which points to images reliable. This is a high quality dataset as all the interactions and all the details of the ORP can be found here.

2. ORP logged activity: this was logged from interactions from ORP to our server with the database structure (see 4.5.1). The quality is worse that the ORP dumps, both because as not all the data were logged (the server was not constantly up) and not all the fields were saved, as many were filtered out.

The dumps cover around 1 months of activity: May 2016. The logged activity dates back to the first implementation of the database, which is around March 2016. This is around 5 months of data.

In order to translate one dataset structure into the other a translator script in Python was set up; this translation was necessary in order to standardize the input for the evaluator framework to work.

6.2. Evaluation

In order to evaluate the success of our algorithm a sound evaluation metric needs to be implemented. Feedback from the NewsREEL organizers and TuBerlin revealed that a commonly used metric is to see the logged activity of the users for possible hits. Therefore the evaluation metric adopted was the following: a recommendation is a successful hit if, by looking at the user behaviour who the recommendation was sent to, the user lands on the recommended page within 10 minutes of navigating the

website. This is done by looking at the navigation history of the specific user following a recommendation. This kind of evaluation has been adopted in order to stay in line to the evaluation done by NewsREEL

The offline evaluation process had initially started using Idomar [58], but later changed to a proprietary evaluator provided by TUBerlin used during the NewsREEL competition. The components used in this process are three: Sender, Evaluator and the Recommender.

- **Sender**: This component replays a set of data through HTTP to the specified address. This simulates the behaviour of ORP. The answer from the Recommender are logged. The interactions (see section 5.1.1) are retrieved directly from the dataset and replayed.
- **Recommender**: it is the Server with our recommendation algorithm loaded and ready to answer with recommendations. The architecture in this server is explained in section 4.5.
- **Evaluator**: it takes the answers logged by the Sender and evaluates them against the original interactions from the dataset by looking at the user behaviour.

This allowed us to test the exactly same implementation of the algorithm both online and offline using two separate servers. A visual representation of the Architecture can be seen in Fig. 6.1.

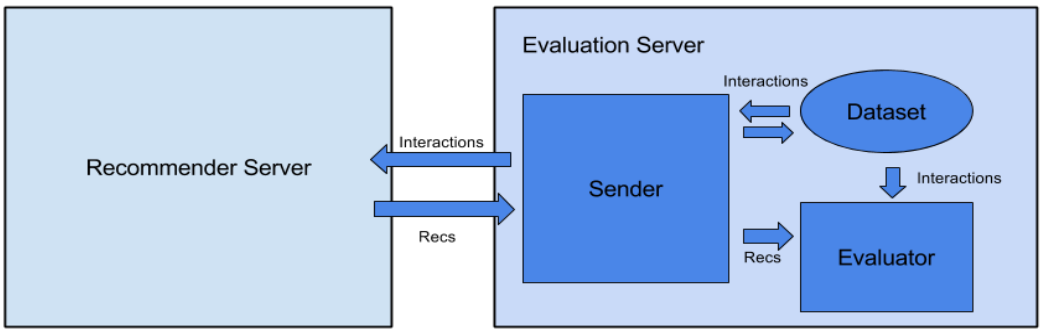


Figure 6.1: Offline Architecture

The approach to testing is to have a training period before the evaluation window in which the points to be tested (recommendation requests) are picked. As can be seen in Fig. 6.2, starting from the original single dataset (a day dump from ORP), the evaluation window (green) is only picked after a initial training period (orange). This means that at anytime a testing point has a training of all the initial interactions from the training period plus all the interactions up until that moment in the evaluation window. On this approach we have conducted our main experiment using extended window evaluation which uses the daily dumps from ORP as single data point 6.2.1. Additionally, we have conducted a smaller check using and a distributed sampling approach 6.2.3, using separate tiny samples from the whole dataset. We decided to introduce this second check to see if the initial finding applied to the whole dataset. More details can be found in the respective sections.

6.2.1. Extended Window Evaluation

Extended window evaluation (or daily evaluation) was our first and most straightforward approach of testing: using the ORP daily dumps and logged data sample and evaluate directly on them. This means that the size of the evaluation window the same as the whole data (a day), therefore there is no space for initial training period where no recommendations requests are taken. A few non-consecutive days have been used as a test set. We consider the used days to be the minimum-sized data set large enough to provide a reliable comparison as the number of requests is around the same obtained during the Online Evaluation (175.000).

Since this daily testing was started just after the first negative looking online results were obtained,

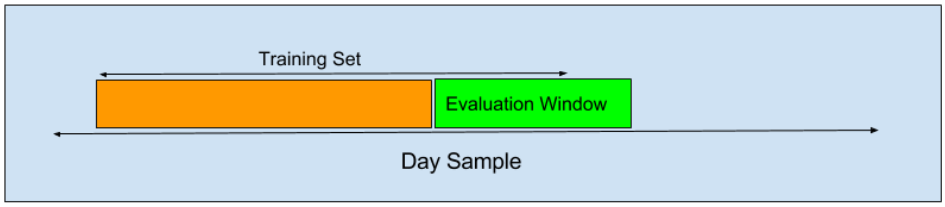


Figure 6.2: Testing Window

it was decided to introduce a new baseline along the popularity. Rand100 was introduced in order to ”weaken” the strength of the baseline algorithm in order to better show the effect of the Image features.

6.2.2. Results

In this testing we conducted tests over two different baselines: Rand100 and Pop100. The raw results can be seen in the Table 6.1, while a visual clue can be seen in Fig. 6.3

Table 6.1: Task 2 Results

Algorithm	Clicks	Requests	CTR
Rand100	258	204456	0.13%
Rand100+Face+Salience	390	202254	0.19%
Pop100	630	204120	0.31%
Pop100+Face	857	203893	0.42%
Pop100+Salience	816	203935	0.40%
Pop100+Face+Salience	771	203979	0.38%

Introducing Image-based recommendation leads to a clicks increase of 51% with respect to the baseline Rand100, while the increase is 36% with respect to the Pop100 when considering only faces, 22% with both features.

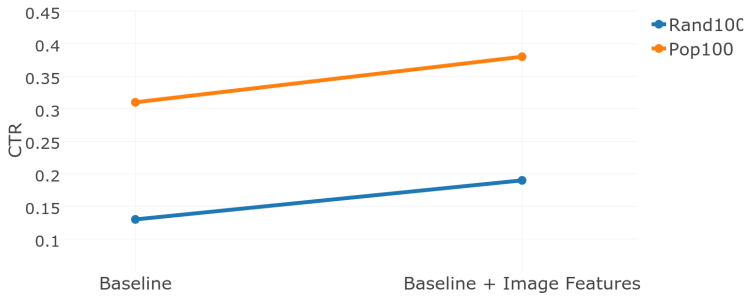


Figure 6.3: Offline Results

6.2.3. Distributed Sampling Check

This check was aimed to see if the results found in the previous experiment can be found throughout the whole dataset.
The need of a more granular testing arose from our dissatisfaction with the forced choice (from NewsREEL) of having one full day as data point to be tested. A full day per se is not a problem, however

intensive testing over the dataset revealed its flaws: offline testing a full day took approximately between 6 to 8 hours of work on our server. Due to the limited amount of time resources, validation over of the whole dataset was not possible. We looked at a few solutions in order to extend our results to the whole dataset rather than to the only days tested. The obvious solution is to select only a sub sample where to test, however this lead to several problems:

- Selection: how to perform it? Random selection would have spread out the days, however to have a reliable sized sample from different timeframes too many samples were needed.
- The Weather Effect (see 2.1.4): The results of the testing change based on many outside factors specific for that day, example if something relevant happens in the world than the news will spike and recommendation will have a different effect. Days close together might have this problem hidden in them. A basic "weather problem" can be seen in the difference between a weekday and a day from the weekend, as seen in Fig. 6.4 6.5

Due to the above listed problems, we decided to abandon the day as standard sample in favour of a smaller and granular data point. This allowed us to gain a few benefits:

- Avoid the weather problem: distribute many more separate data points though out the dataset, in order to even out the weather offect.
- Validate on a bigger scale: more testing points spread out can used to validate the results of the extended window into a larger dataset
- Smaller computational task: having a smaller window means less computational task, even if with the added training period.

The number of slots tested in the end are not enough to make this a proper experiment, therefore it was intended to be more of a "sanity check" to see if the results apply all around. The number of slots were based on our time resources to conduct experiments. We decided to test over more slots rather than over more combinations of different algorithms (different baselines and different features): only over Pop100+Face+Salience was evaluated as it is the most representative algorithm (as it has both the image features).

Due to the fact that samples are much smaller, the initial training period as in Fig. 6.2 was required. This is because it might happen that the cold start period was longer than the evaluation window itself, therefore compromising evaluation. Decision was taken to start the evaluation window only after the minimum training needed to overcome the cold start was finished.
The following numbers were found over an average of 5 runs, with a marked difference between weekday and weekend.
Cold start was measured as the number of updates needed before at least one category of one domain had the freshness window filled. It was found to be 3282 updates. On the other hand the time needed to fill all of the freshness windows is more than one day. The data which need to be sent to the recommender for the initial training are calculated on the average to obtain the 3282 updates. This has revealed to be around 2.672.855 communications from ORP (impressions, requests, updates, etc.) on weekdays, and 5.408.109 during the weekend. This is a big number, and the average time needed to finish the cold start period is from 00:00 to 13:10. That's 13 data hours of training, however they mostly are during the night, which usually means not many updates are incoming. If the training is done starting the during the day, it usually takes around 5 data hours.
The evaluating windows must be a tradeoff between being small enough to be accurate and being big enough to not require a big computation. The size has been decided to be 1000 requests, which usually translate in 280.238 interactions. This is roughly 90 minutes of interactions during an average busy time of the weekend, and more 150 minutes during the week. This size has been chosen to be small enough to be granular and big enough to have a meaningful timeframe

6.2.4. Results

The only difference in the testing set with the previous one is that the Rand100 was not tested, as we felt no need to further prove the effect on a weak baseline. Therefore directly Pop100 was used.

Table 6.2: Task 2 Results

Algorithm	Clicks	Requests	CTR
Pop100	21	8000	0.26%
Pop100+Face+Salience	26	8000	0.32%

Introducing Image-based recommendation leads to a clicks increase of 24% with respect to the baseline Pop100,

6.3. Problems

No major problems were encountered during this stage of the implementation, however a few details need to be mentioned.

Space Constrain: Daily dumps and logged data would occupy around 7GB to 11GB in log text file once downloaded from ORP (in case of daily dumps) or generated from the database (in case of logged data) for each day to be evaluated. Due to our limited space in our server, the whole evaluating task became tricky and hardly automatised (due to limited or non-existent space buffer) through scripts. This lead to an extensive use of manual labour in loading the right testing data and following the server during the computation.

Time Constrain: Computation and evaluation of the offline dataset requires a lot of computing time as the data used grows. The daily dumps testing required quite a lot of computation timewise, as a single dump took from 6 to 8 hours to be replayed and around 30 minutes to be evaluated. Since there are 5 different conditions, the time limiting factor became apparent on the number of days we were able to test. Although the data used are enough to test our hypothesis, we would have liked to use more.

Servers: The minimum required servers to run this offline testing is two, preferably three. The Sender and Recommender needed a server each to communicate to each other and perform the evaluation. A third server was occasionally set act as the Evaluator on the run (a VM was usually used for this task). In case that was not possible, the evaluation could be carried out later through the log files rather than live. Although fine on paper, the problem was that if one of these components had a trouble (usually related to running out of space) than all the process would fail.

Weekend vs Weekday: The clear difference between the typologies of the days can be seen in Fig. 6.4 and Fig. 6.5. Although the interactions (impressions, updates, requests) overall tend to be more, the updates are less. This is more likely because there is less staff working for the news domains during the weekend, however there are much more users. The overall CTR performance is decreased slightly, due to the big increase of interactions and only a partial increase in clicks through. Therefore there is a significant difference between weekday and weekends in terms of CTR.

Small Sample: The approach conducted during the distributed sampling check brought us in front of a little problem which was discovered only after the evaluation was done. Although the updates received during the training phase were enough to fill the freshness window of at least one category, many other windows were lacking in content and items to be recommended. This obviously led to a decrease in material available with a likely repercussion on quality of recommendations therefore a decrease in CTR. Increasing the training period to a longer period would have possibly solved the problem, however it would have made the overhead computation needed for each sample much bigger, bringing it (almost) in line with a day dump. Since the appealing part of the distributed sampling check is the less resource required, it was decided not to increase the training further and accept this small loss in CTR.

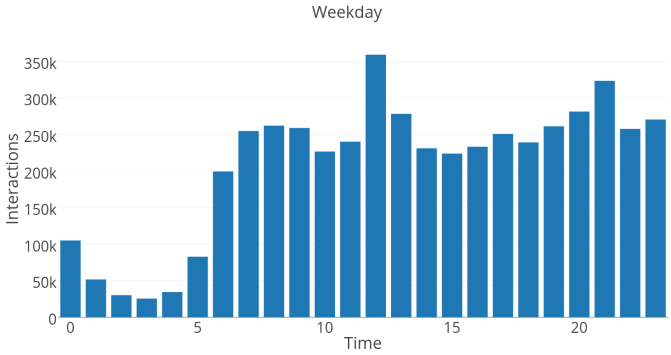


Figure 6.4: Interaction Time Distribution: Week Day

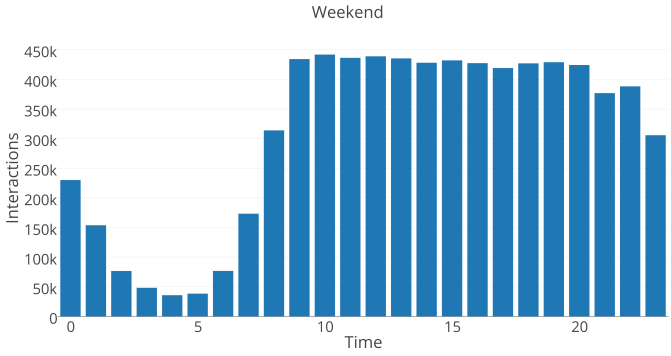


Figure 6.5: Interaction Time Distribution: Weekend

6.4. Conclusion

One fact from the results can be deducted:

- Image-based recommendations do increase the CTR of news recommendation systems

The results shown in section 6.2.2 show a good improvement over CTR when adopting image based recommendation. The extended window experiment shows this improvements, and the distributed sampling try to generalize them: although with a few differences, they both agree on the outcome. This result is important especially when compared with the results of the similar experiment conducted Online. This strengthen our idea that the Online evaluation was cursed by bad technical implementation and weather problem rather than negative image feature effect.

7

Discussion

In this section we analyze the findings, while lying out our conclusion and the aspects that we suggest for future work.

The testing done was organized in order to be presented as contrastive conditions: each recommender has been tested against it is own baseline in a A vs B testing. Because of this, in this chapter we are able to reliably detect and discuss the difference between the recommenders as the effect of the added image features.

7.1. Online

Two important facts can be seen from the results:

- Feasibility: The algorithm has successfully run online for more than one month.
- CTR: The performance of the experimental algorithm is lower than the baseline

Feasibility

The successfully online run has clearly showed us that using computationally intensive recommender which makes use of images information is feasible. Following the 3D evaluation framework [5], we stress the importance of analyze our algorithm from a technical prospective for Online feasibility. This result is interesting mainly for future work, as next generations algorithms will have to include time complexity and feasibility in their evaluation, especially if needed to be run online and in a real word environment rather than just in offline lab experiments.

CTR performance

The results gathered during the evaluation windows of a month suggest that the baseline (Pop100) performs better than the image enhanced algorithm. This can be inferred from the contrastive condition between the popularity baseline and the baseline + Image features. This decrease in performance can have three possible explanations:

1. Technical problems which the algorithm faced when running online which jeopardized the final result.
2. The weather problem caused by having two different periods for testing (baseline vs baseline+imageFeatures).
3. Negative effect of the images features (Face or Saliency).

In order to investigate to what extent technical problems were affecting the performance of the algorithm, extensive logging was enabled for the online algorithm. Sadly, the dynamic and not constant nature of the technical problems encountered could not allow us to fully grasp the detail impact of all the errors and problems on the CTR performance. The detected problems clearly showed us that the

timeout response was a critical part, as explained in section 5.4.1. A new and improved version of the algorithm was developed to try to solve the technical problems, however testing of this new improved version was not possible due to plista closing the ORP platform.

The weather problem, with the random chance of having a different CTR for each period, might have affected the outcome. The two different periods of 23 days are close but the hidden effect might have changed the result. There is no way for us to detect this shift.

We are keen to believe that the two above problem are the cause of the decrease in CTR, rather than a negative effect of the image features. This has lead us to exclude this specific result for the CTR performance conclusion.

7.2. Offline

One important fact can be seen from the results:

- Image-based recommendations do increase the CTR of news recommendation systems

The main experiment done using the extended window approach shows a big improvement when adding image features into the recommendation. The distributed sampling check confirms this finding. Their results, although slightly different, do agree on the outcome: Image-based recommendation increase the CTR of the system.

Offline evaluation is a limited approach to the problem. No direct CTR performance can be taken as an absolute result, but rather only the change in performance. The Offline approach, by making use only of the recorded log of the user, cannot detect the effect of the influence of the use, as items recommended might not have appeared online and consequentially not appear in the log in the first place. However the offline trending can be a good indication of the general state of affairs. The evaluation method used in this task does make the CTR quite lower than the Online one; this is probably due to the evaluation method on the logs and the absence of a real user to answer directly to the recommendation shown. However the difference between the baseline and the baseline+visual information can be used to infer the effect of such features.

7.2.1. Extended Window Evaluation

The contrastive conditions setup allowed us to clearly see the effect of the image features on the CTR. For both baselines Rand100 and Pop100 we can see a significant improvement (see 6.1) of the CTR when we make use of the Image information. As expected the increase is bigger with the ”weaker” baseline, Rand100, since the effect of the image feature is not hided by an already good performing baseline.

Another interesting result is the recommendation featuring only faces or only salience performing slightly better than the combined faces+saliency. This unexpected result might be explained by the specific data and the settings of our domains. The logic behind Saliency+Faces algorithm dictates that the recommender is most likely to show images with both features rather than only one. This is useful for general purpose news recommender, however it does not give benefits for a few specific domains in this specific case: sport and automotive. As can be seen from the examples in Fig 7.1 7.2, many times the coupling of these two features results into a less attractive picture as result. This is for the specific domain:

- Sport: The coupling of salience and face usually result in a picture which clearly not show the body. Our guess is that much of the attractiveness of the image is the team colors and brand which clearly gives the additional information which can appeal the user.

- Automotive: The combination of this two features leads to the exclusion of the car or motorcycle which usually (we conjecture) is the base of the appeal in this kind of images.

(a) Saliency

(b) Face

(c) Saliency and Face

Figure 7.1: Sport Images

(a) Saliency

(b) Face

(c) Saliency and Face

Figure 7.2: Automotive Images

7.2.2. Distributed Sampling Check

Only Pop100+Face+Saliency has been used in this check, as it is both considered the most representative algorithm (it has both features) and has the lowest performance from the previous evaluation, therefore generalizing the results of the other two (they will supposedly perform equal or better). This check was done to see if the results found in the extended window evaluation could be representative of a larger scale dataset. The results extracted from this more granular testing confirmed the same results could be found across different days: introducing image features increase the CTR.

Digging deeper in the actual numbers (see 6.2) we see a 24% increase, which is less than the daily evaluation increase. The CTR results of the extended window evaluation and the distributed check

differs significantly. The cause for this needs to be searched for the difference between the datasets used for testing. Although the Extended evaluation was carried on a bigger dataset on the number of requests, only a few separate days were included, with none of them being a weekend day (a discussion over weekday vs weekend differences can be found at 6.3 "weekend vs weekday"). On the other hand, the distributed sampling check tends to cover every kind of day, weekend included. Having dataset made harder by the inclusion of the weekend, coupled with the fact that the small sample might have reduced the pool of possible recommendations (see 6.3 "small sample") might explain the small but significant drop in CTR performance. By taking this into consideration, we believe that this results can confirm the one found during the extended window evaluation. This implies that the results initially found can be applied to a larger dataset of a few months rather than only a few days.

7.2.3. Offline Discussion

Although with a small difference in the performance of the CTR, the distributed sampling check confirms and generalizes the results found in the extended window experiment. Especially when taking into account the two problems (weekend and small sample) which might have reduced the effect of the distributed check, and by looking at the fact that they show the same trend in image feature improving the CTR, we can say that the results confirm our hypothesis: Image-based recommendation increase the CTR of new recommendation systems.

This strengthens our idea, when comparing this results to the Online, that the Online implementations results were jeopardized by the poor technical performance and other problems rather than a negative image effect.

7.3. Examples Analysis

This small section wants to dig into the a few examples in the images obtained from ORP to see in which cases we were successful in in which there were some troubles. We do this to confirm that there are no obvious confounding factors that we overlooked, and also to allow to formulate suggestions for future work A few thousands images were scraped and analyzed to see what were the unexpected effects of the recommended images. This analysis has been made only at the end after we have obtained the results, therefore there is no danger that by looking at the data beforehand we could have somehow bias our result for this particular data (thus limiting its generalizability). Two investigations were made: the images which caused the most clicks and a random sampling to see if the feature classification was correct.

7.3.1. The Most Clicked

The absolute mostly clicked was strongly correlated with the number of user each domain had. The most used domain was a sport domain, therefore all the most clicked news were from the sport field, and all had images attached. Here we present the top three images, a bigger dataset can be seen in appendix 8. Fig. 7.3 shows the absolute top three.

Figure 7.3: Top Sport Images

Interestingly the first three all have the image features analyzed in this work: two have face and one has salience.
By excluding the domain which refers to sport, we found that the second most clicked are the one about general news. Out of the top 100 not sport items, 9 did not present an image to the reader. The top three can be seen in Fig. 7.4: one salience and face, one with no features and one with salience.

Figure 7.4: Top General Images

7.3.2. Feature Classification

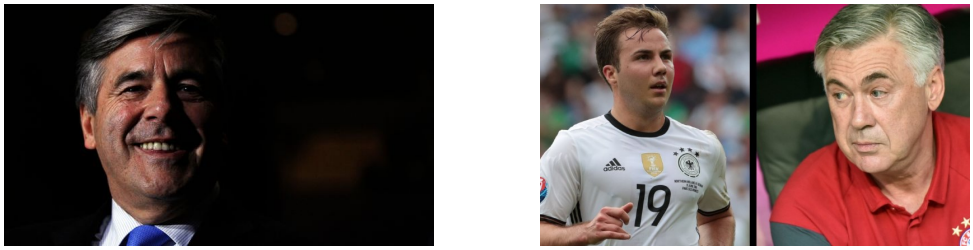
Following are a few important features classification failure and success examples, taken from a random sampling and comparing the algorithm label with a human one. To have a more exhaustive list of the whole random sample of images see Appendix 8. Fig. 7.5c shows images which have successfully adopted the saliency feature and obtained a decent number of clicks. Fig 7.5d is an example where the salience has been detected, but the face not. The most clear example of saliency not properly working is in Fig. 7.5e, where for some strange reason an image containing text has been considered salient. For faces the story is different, as the Viola-Jones classifier works quite well and there are no miss detections or big failures 7.6; the only few failures can be detected as a few images having the saliency feature from the human label while the algorithm does not detect it. The combination of the two features presents a few failure cases such as Fig. 7.7b where although the face detection is correct, the saliency is erroneously detected.

(a) (b) (d) (e)

(c) Successful Salient Images (f) Failure Salient Images

Figure 7.5: Salient Images

Figure 7.6: Face Images



(a) Successful Sal+Face Image (b) Failure Sal+Face Image

7.4. Future Work and Known Issues

In this section we go over the known issues which we feel might have been approached differently or revised. This naturally leads to the future work and the next development.

7.4.1. Known Issues

Although satisfied with the work done during this thesis work, a few points have been left unclear both for wrong technical decisions and temporal constrains. In this section we are going to analyze the problems which have been encountered during our work.

Offline Testing Set: Although the testing done is enough to partially support our hypothesis, a much bigger dataset would have been preferable, especially during the generalization through the distributed sampling evaluation approach. The reason for such a reduced dataset is mainly due to the time resources available which were consumed as manual labour to test each section. An automated testing implementation was planned, however this implementation was not possible due to hardware problems such the extreme and not existent memory space in the servers (see section 4.4) which caused the scripts to always run out of space; therefore continuous manual intervention was required.

Second Online Evaluation: A new and improved algorithm was developed in order to deal with the technical difficulties detected during the first evaluation, however a new round of testing was not possible due to plista taking down the ORP platform.

Confounding Features: CTR is the industry evaluation standard and it also is how most companies that serve ads charge their clients. This is a limitation to our understanding of the reasons which bring a user to click on an image. As previously stated (in section 3.5.3), face and familiarity can be confounding features which we might not be able to detect. Other examples could be the saliency in the IT domain which reflects the presence of single new devices ex. Fig. 7.8, which causes IT readers to be interested in.

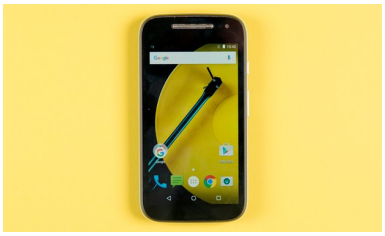


Figure 7.8: IT image with saliency feature

These two examples are the specific representation of a larger problem: we could not guarantee that the detected features were what actually caused the increase in performance in the recommender. If the real effective features are different from the one tested, but correlated with the face and saliency, we would still have positive results although wrong. This problem cannot be solved through the limited feedback we receive from ORP, but only through a more in depth analysis, which encompass different users tests with expanded feedback (e.g. questionnaires, controlled user tests).

7.4.2. Future Work

The algorithm and the approach developed during this challenge was intended to be an exploratory task. More work is still needed to investigate the real effect of images on the recommendation, and all its possible application and uses.

The unsolved problems mentioned in the previous section 7.4.1 needs to be analyzed and solved in any future work. Both Online and Offline testing needs to be continued and deepened on all the possible combinations of baselines and features used in this thesis work, in order to test both the single effect of the features independently and their strength against different baselines. This is especially needed in order to investigate further the difference between Online and Offline, especially in light of the results obtained in this work. Due to time constrains, the dataset used for testing and validation (especially in

the Offline part) were not the size we were initially hoping for. A more deep testing over larger dataset needs to be performed.

Improvement in efficiency and running times are needed in order to allow the algorithm to properly work in an online environment, especially when adding even more image features than the ones tested in this work. The current implementation has many flaws that likely resulted in many delays and worse CTR.

7.4.3. Classification

A basic improvement needed in the approach presented is the introduction of a more refined classification procedure. In this thesis work, naive binary classification is used when dealing with images: images are classified either ”interesting” or ”not interesting”. First thoughts suggest the use of a better classification approach that includes scores or different grades for the ”interestigness” could bring big improvement of CTR and user experience. This would resemble human mind, as we tend to be more interested by certain pictures however with different degrees. This could be coupled with an initial personalization attempt, in order to accommodate, for each user, a different levels of attention for different topics and different types of visual feature.

7.4.4. Features

Although this thesis work has focused its attention on the exploitation of high level visual clues (people, saliency map), a more in dept analysis of other features classes may reveal useful insights. Many other high level features are worth testing, e.g. presence of animals, aesthetic of the image. Additionally, lower level features might be reveal its usefulness: notable global features include colorfulness, brightness and saturation. Another interesting approach could be the inclusion of visual information of how and where the recommendation is displayed (website related features).

7.5. Conclusion

In this work two environments have been used to test our hypothesis: Online and Offline. This has been done in order to analyze our algorithm both from a user prospective (CTR - Offline testing) and from a technical prospective (Online feasibility). Online testing has proved that the time complexity can be handled by our current implementation; Offline testing has proved us that the performance of the recommendation can be enhanced by the use of images.

Let us take a look again at our research questions:

- Can a non-trivial news recommender that extracts features from an image feasibly be run online in a real world environment?
- Can information extracted from images accompanying items in a recommendation system help improve the Click Through Rate of such systems, specifically in the news domain?

7.5.1. First Question

- Can a non-trivial news recommender that extracts features from an image feasibly be run online in a real world environment?

Our running algorithm has provided an affirmative answer: it is feasible, even with an computationally intense load. Time complexity is not a blocker, although it makes the required implementation harder. The current implementation done in this work has many flaws which jeopardize the final CTR performance, but it was sufficient to provide the proof of concept necessary to answer the research question.

7.5.2. Second Question

- Can information extracted from images accompanying items in a recommendation system help improve the Click Through Rate of such systems, especially in the news domain?

Offline results would provide an indication that there is a notable improvement of performance when adding image based features. Information extracted from images, in this case the presence of a person

and the presence of a single object in the centre, can be used to predict and refine the recommendation in a news recommendation environment, specifically in the general, sports, IT and automotive categories.

7.6. Final Remarks

This thesis work has attempted to shed some light in an not-yet well understood area of the field of recommender system. We hope that this work will be continued in the future and more in depth studies will verify and integrate our findings.

7.6.1. Code Release

In hope of lying down the road for future research, all the important code is released as open source, in order to facilitate academic research in the near future. This includes the recommender, the client, the communication system and the feature detectors. The code can be found at https://github.com/FranCorsini

Bibliography

[1] *The New York Times,* http://www.nytimes.com/, accessed: 2016-06-5.

[2] H. Cheng, R. V. van Zwol, J. Azimi, E. Manavoglu, R. Zhang, Y. Zhou, and V. Navalpakkam, *Multimedia Features for Click Prediction of New Ads in Display Advertising,* in *18th ACM SIGKDD international conference on Knowledge discovery and data mining* (2012) pp. 777–785.

[3] *CEUR-WS,* http://ceur-ws.org/, accessed: 2016-08-10.

[4] *CLEF NewsREEL 2016: Image-based Recommendation,* http://ceur-ws.org/Vol-1609/16090618.pdf, accessed: 2016-08-10.

[5] A. Said, D. Tikk, Y. Shi, M. Larson, K. Stumpf, and P. Cremonesi, *Recommender systems evaluation: A 3d benchmark,* in *ACM RecSys 2012 workshop on Recommendation utility evaluation: beyond RMSE, Dublin, Ireland* (2012) pp. 21–23.

[6] F. Hopfgartner, T. Brodt, J. Seiler, B. Kille, A. Lommatzsch, M. Larson, R. Turrin, and A. Serény, *Benchmarking news recommendations: The CLEF NewsREEL use case,* SIGIR Forum **49**, 129 (2016).

[7] B. Kille, A. Lommatzsch, G. Gebremeskel, F. Hopfgartner, M. Larson, J. Seiler, D. Malagoli, A. Sereny, T. Brodt, and A. de Vries, *Overview of NewsREEL'16: Multi-dimensional Evaluation of Real-Time Stream-Recommendation Algorithms,* in *Experimental IR Meets Multilinguality, Multimodality, and Interaction 7th International Conference of the CLEF Association, CLEF 2016, Evora, Portugal, September 5-8, 2016.*, edited by N. Fuhr, P. Quaresma, B. Larsen, T. Goncalves, K. Balog, C. Macdonald, L. Cappellato, and N. Ferro (Springer, 2016).

[8] *The CLEF Initiative,* http://www.clef-initiative.eu//, accessed: 2016-07-10.

[9] *plista,* https://www.plista.com/, accessed: 2016-07-10.

[10] B. Kille, F. Hopfgartner, T. Brodt, and T. Heintz, *The plista dataset,* in *NRS'13: Proceedings of the International Workshop and Challenge on News Recommender Systems*, ICPS (ACM, 2013) p. 14–22.

[11] J. Bobadilla, F. Ortega, a. Hernando, and a. Gutiérrez, *Recommender systems survey,* Knowledge-Based Systems **46**, 109 (2013).

[12] R. V. Meteren and M. V. Someren, *Using Content-Based Filtering for Recommendation,* ECML/MLNET Workshop on Machine Learning and the New Information Age , 47 (2000).

[13] P. Lops, M. de Gemmis, and G. Semeraro, *Recommender Systems Handbook*, Vol. 54 (Springer, 2011) pp. 479–510, arXiv:arXiv:1011.1669v3 .

[14] N. Company, *Netflix Price,* "http://www.netflixprize.com/.

[15] Y. Koren, *The Bellkor solution to the Netflix grand prize,* Netflix prize documentation , 1 (2009).

[16] X. Su and T. M. Khoshgoftaar, *A Survey of Collaborative Filtering Techniques,* Advances in Artificial Intelligence **2009**, 1 (2009).

[17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, *Item-based collaborative filtering recommendation algorithms,* in *Proceedings of the 10th International Conference on World Wide Web*, WWW '01 (ACM, New York, NY, USA, 2001) pp. 285–295.

[18] M. Hristakeva and K. Jack, *A Pratical Guide to building Recommender Systems,* .

[19] L. Li, D. D. Wang, S. Z. Zhu, and T. Li, *Personalized news recommendation: A review and an experimental investigation,* Journal of Computer Science and Technology **26**, 754 (2011).

[20] R. Carreira, J. M. Crato, D. Gonçalves, and J. a. Jorge, *Evaluating Adaptive User Profiles for News Classification,* the Proceedings of the Ninth International Conference on Intelligent User Interfaces , 206 (2004).

[21] A. Das, M. Datar, A. Garg, and S. Rajaram, *Google news personalization: scalable online collaborative filtering,* Proceedings of the 16th international conference on , 271 (2007).

[22] L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan, *SCENE: a scalable two-stage personalized news recommendation system.* Sigir , 125 (2011).

[23] I. Verbitskiy, P. Probst, and A. Lommatzsch, *Development and Evaluation of a Highly Scalable News Recommender System,* (2015).

[24] G. G. Gebremeskel and A. P. D. Vries, *The degree of randomness in a live recommender systems evaluation,* in *Working Notes of the 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France, September 8-11, 2015, Proceedings.* (2015).

[25] Kille, Benjamin and Brodt, Torben and Heintz, Tobias and Hopfgartner, Frank and Seiler, Jonas, *Overview of CLEF NEWSREEL 2014: News recommendation evaluation labs,* in *Experimental IR Meets Multilinguality, Multimodality, and Interaction 5th International Conference of the CLEF Association, CLEF 2014* (2014).

[26] S. Dhar, T. L. Berg, S. Brook, V. Ordonez, and T. L. Berg, *High level describable attributes for predicting aesthetics and interestingness,* Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition , 1657 (2011).

[27] P. J. Silvia, R. a. Henson, and J. L. Templin, *Are the sources of interest the same for everyone? Using multilevel mixture models to explore individual differences in appraisal structures,* Cognition & Emotion **23**, 1389 (2009).

[28] M. Gygli, H. Grabner, H. Riemenschneider, F. Nater, and L. Van Gool, *The Interestingness of Images,* Computer Vision (ICCV), 2013 IEEE International Conference on , 1633 (2013).

[29] M. Soleymani, *The quest for visual interest,* in *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15 (ACM, New York, NY, USA, 2015) pp. 919–922.

[30] W. C. Bradford, *Reaching the visual learner: teaching property through art,* Indiana University School **11** (2004).

[31] B. B. Cooper, *How Twitter's Expanded Images Increase Clicks, Retweets and Favorites,* .

[32] CMO Council, *The Role of Visual Media in Impactful Brand Storytelling,* (2015).

[33] X. Fern, *The Impact of Visual Appearance on User Response in Online Display Advertising,* Proceedings of the 21st international conference companion on World Wide Web , 457 (2012), arXiv:arXiv:1202.2158v2 .

[34] L. Bershidsky, *Facebook, I Want My Clickbait Back,* http://www.bloomberg.com/view/articles/2016-08-05/facebook-i-want-my-clickbait-back.

[35] M. Potthast, S. Köpsel, B. Stein, and M. Hagen, *Clickbait Detection,* in *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*, edited by N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, M. G. Di Nunzio, C. Hauff, and G. Silvello (Springer International Publishing, Cham, 2016) pp. 810–817.

[36] G. Loewenstein, *The Psychology of Curiosity: A Review and Reinterpretation,* (1994).

[37] J. Liu, P. Dolan, and E. Pedersen, *Personalized news recommendation based on click behavior,* Proceedings of the 15th international conference on Intelligent user interfaces , 31 (2010).

[38] R. Pagano, P. Cremonesi, M. Larson, B. Hidasi, D. Tikk, A. Karatzoglou, and M. Quadrana, *The contextual turn: from context-aware to context-driven recommender systems, TO APPEAR,* in *Proceedings of the 15th international conference on Intelligent user interfaces, RecSys2016* (2016).

[39] A. E. Savakis, S. P. Etz, and A. C. P. Loui, *Evaluation of image appeal in consumer photography,* Proc. SPIE 3959 **3959**, 111 (2000).

[40] P. Ricciardelli, C. Iani, L. Lugli, A. Pellicano, and R. Nicoletti, *Gaze direction and facial expressions exert combined but different effects on attentional resources,* Cognition and Emotion **26**, 1134 (2012), pMID: 22900946, http://dx.doi.org/10.1080/02699931.2011.638907 .

[41] M. Farah and S. Kosslyn, *Structure and strategy in image generation,* Cognitive Science **5**, 371 (1981).

[42] H. van Oostendorp, S. Karanam, and B. Indurkhya, *Colides+ pic: a cognitive model of web-navigation based on semantic information from pictures,* Behaviour & Information Technology **31**, 17 (2012), http://dx.doi.org/10.1080/0144929X.2011.603358 .

[43] N. Bonnardel, A. Piolat, and L. Le Bigot, *The impact of colour on Website appeal and users' cognitive processes,* Displays **32**, 69 (2011).

[44] J. A. Redi and I. Povoa, *The Role of Visual Attention in the Aesthetic Appeal of Comsumer Images: a Preliminary Study,* in *Visual Communications and Image Processing (VCIP)* (Intelligent Systems, Delft University of Technology, The Netherlands, 2013).

[45] R. Datta, D. Joshi, J. Li, and J. Z. Wang, *Studying aesthetics in photographic images using a computational approach,* in *Proceedings of the 9th European Conference on Computer Vision - Volume Part III*, ECCV'06 (Springer-Verlag, Berlin, Heidelberg, 2006) pp. 288–301.

[46] P. Viola and M. Jones, *Rapid object detection using a boosted cascade of simple features,* Computer Vision and Pattern Recognition (CVPR) **1**, I (2001).

[47] X. Hou and L. Zhang, *Saliency detection: A spectral residual approach,* Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition , 1 (2007).

[48] Y. Luo and X. Tang, *Photo and Video Quality Evaluation,* Quality **8**, 386 (2008).

[49] *The Guardian,* `http://www.theguardian.com/international`, accessed: 2016-06-5.

[50] *BBC NEWS,* `http://www.bbc.com/`, accessed: 2016-06-5.

[51] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, *Accurately interpreting clickthrough data as implicit feedback,* in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (Acm, 2005) pp. 154–161.

[52] A. Kulkarni, J. Teevan, K. M. Svore, and S. T. Dumais, *Understanding temporal query dynamics,* in *Proceedings of the fourth ACM international conference on Web search and data mining* (ACM, 2011) pp. 167–176.

[53] J. Kiseleva, E. Crestan, R. Brigo, and R. Dittel, *Modelling and detecting changes in user satisfaction,* in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (ACM, 2014) pp. 1449–1458.

[54] *recommenders.net,* `http://recommenders.net/`, accessed: 2016-07-10.

[55] A. S. Alejandro Bellogín, *net.recommenders.plista - client,* `https://github.com/recommenders/plistaclient` (2014).

[56] A. S. Alejandro Bellogín, *plistarecs,* `https://github.com/recommenders/plistarecs` (2014).

[57] B. Kille, A. Lommatzsch, R. Turrin, A. Serény, M. Larson, T. Brodt, J. Seiler, e. J. Hopfgartner, Frank", J. Savoy, J. Kamps, K. Pinel-Sauvagnat, J. G. Jones, E. SanJuan, L. Cappellato, and N. Ferro, *Stream-based recommendations: Online and offline evaluation as a service,* in *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 6th International Conference of the CLEF Association, CLEF'15, Toulouse, France, September 8-11, 2015, Proceedings* (Springer International Publishing, Cham, 2015) pp. 497–517.

[58] *Idomaar crowdrec reference framework,* http://rf.crowdrec.eu/, accessed: 2016-07-10.

8

Appendix A: Images

In this appendix we show random sample of images scraped from ORP, with the features labels specified.

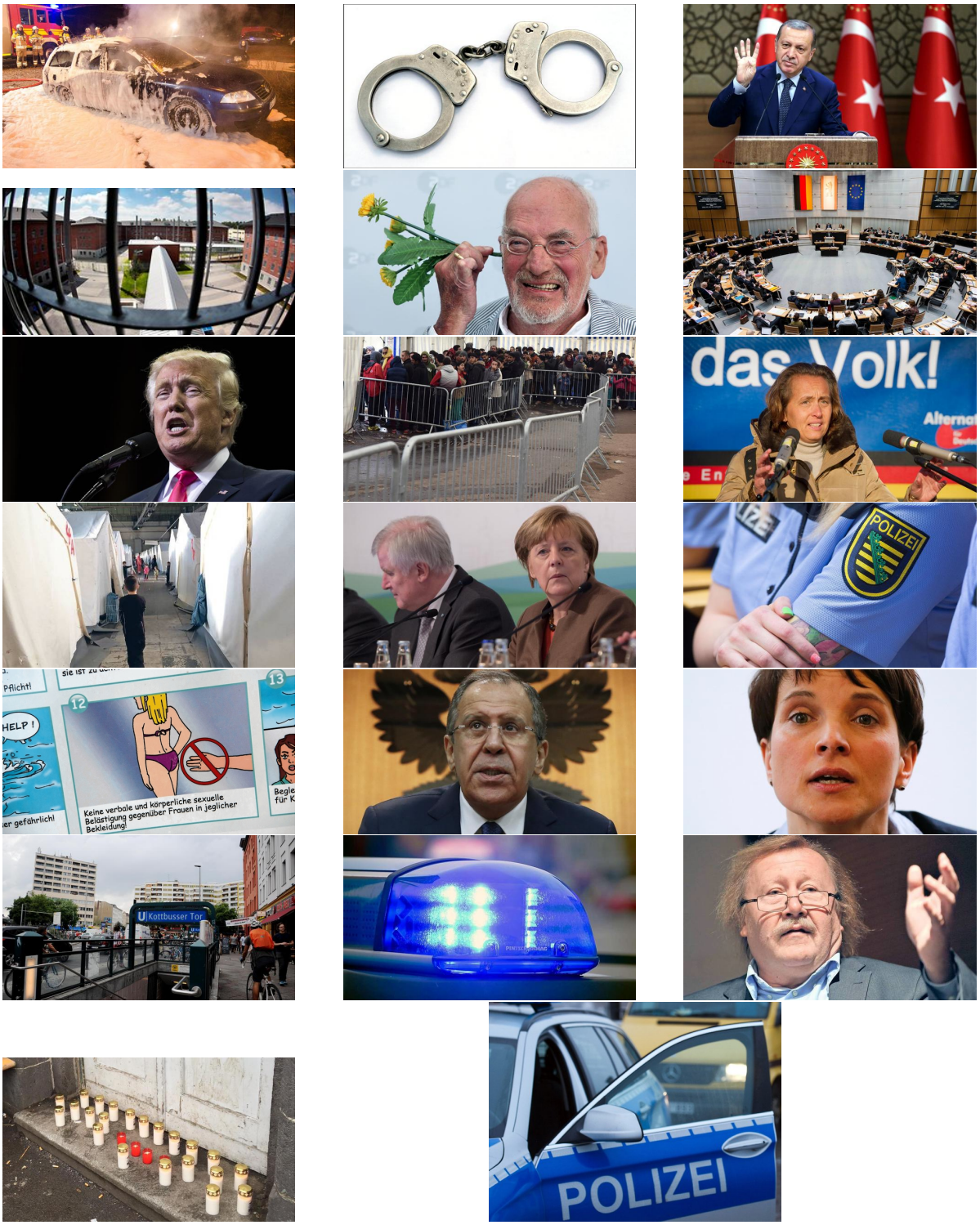
Figure 8.1: Random Sample Salient Images

Figure 8.2: Random Sample Face Images

Figure 8.3: Random Sal+Face Sample

Figure 8.4: Absolute Top Clicks

Figure 8.5: Top Not Sport Domain Clicks



9

Appendix B: Tables

In this appendix we present the Online daily results extracted from ORP

Table 9.1: Daily ORP Results Popularity Baseline

Day	Requests	Clicks	CTR
2016-03-10	1,270	6	0.47%
2016-03-11	1,178	10	0.85%
2016-03-12	2,333	29	1.24%
2016-03-13	1,924	15	0.78%
2016-03-14	1,689	33	1.95%
2016-03-15	1,746	22	1.26%
2016-03-16	4,858	26	0.54%
2016-03-17	5,956	43	0.72%
2016-03-18	1,270	6	0.47%
2016-03-19	6,106	65	1.06%
2016-03-20	6,224	62	1%
2016-03-21	6,298	63	1%
2016-03-22	5,507	47	0.85%
2016-03-23	6,413	54	0.84%
2016-03-24	7,637	128	1.68%
2016-03-25	6,750	59	0.87%
2016-03-26	7,010	66	0.94%
2016-03-27	6,427	61	0.95%
2016-03-28	6,580	81	1.23%
2016-03-29	6,980	76	1.09%
2016-03-30	8,127	92	1.13%
2016-04-01	10,665	88	0.83%
2016-04-02	1,809	12	0.66%

Table 9.2: Daily ORP Results Popularity+Face

Day	Requests	Clicks	CTR
2016-04-21	4,023	45	1.12 %
2016-04-22	4,547	38	0.84 %
2016-04-23	5,916	57	0.96 %
2016-04-24	7,114	49	0.69 %
2016-04-25	6,204	33	0.53 %
2016-04-26	5,863	35	0.6 %
2016-04-27	5,676	31	0.55 %
2016-04-28	6,712	35	0.52 %
2016-04-29	7,022	46	0.66 %
2016-04-30	8,064	51	0.63 %
2016-05-01	9,851	60	0.61 %
2016-05-02	4,379	39	0.89 %
2016-05-03	10,462	67	0.64 %
2016-05-04	9,690	71	0.73 %
2016-05-09	2,486	16	0.64 %
2016-05-11	8,979	70	0.78 %
2016-05-12	11,001	100	0.91 %
2016-05-13	11,494	106	0.92 %
2016-05-14	8,242	65	0.79 %
2016-05-15	604	5	0.83 %
2016-05-16	9,285	66	0.71 %
2016-05-17	14,319	81	0.57 %
2016-05-18	15,664	66	0.42 %

10

Appendix C: Paper

This appendix contains a research paper, written during this thesis project. The paper is published in the CLEF 2016 - Conference and Labs of the Evaluation forum, and can be currently found online at CEUR-WS [3]

CLEF NewsREEL 2016: Image-based Recommendation

Francesco Corsini[1] and Martha Larson[1 2]

[1] Delft University of Technology, Netherlands
[2] Radboud University Nijmegen, Netherlands
corsinifrancesco0@gmail.com, m.a.larson@tudelft.nl

Abstract. Our approach to the CLEF NewsREEL 2016 News Recommendation Evaluation Lab investigates the connection between images and users clicking behavior. Our goal is to gain a better understanding of the contribution of visual representations accompanying images (thumbnails) to the success of news recommendation algorithms as measured by standard metrics. We experiment with visual information, namely Face Detection and Saliency Map, extracted from the images that accompany news items to see if they can be used to chose news items that have a higher chance of being clicked by users. Initial results seems to suggest great CTR improvement in the Simulated Environment task, while some decrease in performance has been found in the Living Lab task. The latter result must be further validated in the future.

Keywords: Recommender System, News, Image Analysis, Face Detection, Saliency Map, Evaluation

1 Introduction

The CLEF NewsREEL [5] News Recommendation Evaluation Lab challenges participants to come up with an original and effective solution for providing recommendations for users in the news environment. Our participation is both for Task 1 (Living Lab Evaluation) and Task 2 (Evaluation in Simulated Environment). An overview of this year challenge results can be found at [9].

Typical online news content providers publish images along with their news items. Our work is motivated by the conjecture that these images play a role in the effect of the recommendation, especially whether a user will click on the item. Content providers are well aware of the importance of images and are already taking advantage of them (e.g., both their informative potential, and their potential to act as clickbait). However, the effect of images for automatic recommendations is currently understudied and not well understood. Our research looks for the effect of such images, in order to determine if they can play a crucial role in the definition of a more refined recommendation. Our hypothesis is that people tend to click on news articles because they are curious about the image, as the image catches their eye, and some images depict things clearly making it very easy to see what the article is actually about. Specifically, in this work, we

will focus on the usefulness of information about faces appearing and saliency in images. The Open Recommendation Platform (ORP) by plista provided a unique framework to test and benchmark our approach. Given the constraints of the online environment (100ms timeout response time, unpredictable load on the server), new and innovative architectures and algorithms were developed in order to deal with the heavy computational load caused by the image analysis. Our research also investigates whether features extracted from images can be used in a real-time recommendation pipeline.

The rest of the paper is organized as following: in section 2 we discuss the related work on how to trigger interest on images presented, plus the background needed to understand our approach to image classification. Section 3 describes our approach to solve the challenges presented in Task 1 and 2 and here our algorithm is presented. The outcome of our experiments and the results of the evaluations is presentend in section 4. The discussion 5 follows presenting future work and a wrap up for the conclusion.

2 Related Work and Background

2.1 Grabbing Attention

In this section, we discuss factors that trigger our eyes to land on an image. With content-based image retrieval on the rise, there is an increase in the study of cues that could help in ranking the retrieved images. A sound measure that would help to automatically rank is how interesting people find an image. Much research has been devoted to the study of interestingness on the Internet, especially with Flicker images, e.g., [2]. However, this sort of interestingness is different from what we investigate here. Specifically, it implies some sort of community and social behavior that goes beyond the effect of images merely catching the eye. The presence of this kind of behavior cannot be assumed to be present in news recommendation environment, where the images come from the news provider, rather than being contributed by community members. Flickr's interestingness is based on social parameters linked to the behavior, i.e., according to the uploader's score reputation and ratio between views, favorites and comments. As example, images with a positive connotation (smile, bright), tend to always have a higher level of interestingness in social media.

Other related research comes from the area of advertising. An accurate prediction of the probability that users click on ads is crucial for the online advertisement business. Even if with different methods, both our work and ads business share the same goal: predict (and increase) how many clicks an image(or an ad) receives. State-of-the-art click through rate prediction algorithms rely heavily on historical information collected for advertisers, users and publishers. However, recent work has seen the integration of multimedia features extracted from display ads into the click prediction models [1] [3]. The features related to an increase in CTR are numerous. In particular, Cheng et al. [1] present an extensive list of image features and their correlation with CTR. In

this study, we focus on key features from [1], chosen because of their promise and their feasibility in being deployed in an online environment. From a study of the literature, we found two of most interesting and investigation-worthy features: the presence of a person [13] [2] [3] [1] [12], especially when having a face clearly visible facing the camera, and the analysis of the saliency map to detect aesthetics and simplicity [1] [3] [4] [11]. However, due to unexpected technical issues during the implementation of these features, only the presence of a person (face detection) was fully developed at the start of the Task 1 challenge. For this reason, it was the only one adopted for consistency throughout all the Task 1 evaluation window. However both features have been tested together in the Task 2 part of the challenge.

2.2 Image Classification

Our approach is based on a straightforward binary image classifier, which classifies the image of the target item (thumbnail) as either "interesting" or "not interesting". The motivation behind this choice of binary classifiers is the lack of time resources and easy management of the results; a better and more refined approach to the classification (e.g. degrees of interestingness) is planned in future work 5.3. The classification process can be summarized simply as follows: According to our research an image is interesting if it either has:

- The presence of a person: A single central person (portrait) is preferred over multiple people all over the image
- A single cluster in the middle of the image with a flat background. A single object is preferred over multiple objects

As for example, the Fig. 1a and 1b are considered "interesting", 1a for the presence of a face and 1b for satisfying the single object in the center. While 1c does not satisfy either of the two requirements.

(a) Face (b) Salient (c) Not interesting

Fig. 1

3 Approach

Our approach was designed to validate our hypothesis that images impact user clicks on recommendations rather than to reach the maximum possible CTR.

The Living Lab Evaluation [6] (Task 1) was executed on the ORP, where part of plista's traffic is redirected. The ORP makes it possible to deploy and test algorithms in a real environment. The platform uses HTTP protocol supporting JSON format for data. Communication is handled by four types of messages: Recommendation requests, Impressions, Item Updates, Error Messages. The timeout for the waiting for the response is 100ms: if the system does not answer within this timeframe, the request is considered as an "error"

The Evaluation in Simulated Environment [10] (Task 2) officially makes use of a set of data provided by the NewsREEL organizers. The set includes item updates and event notification [8]. However, this official dataset did not have a crucial field which was required by our image-based algorithm: the img_url. Although the field itself is present, the official data set was collected in June 2013 and the most part of the links have disappeared since the images are hosted by the publishers themselves. Domains tend to remove the items (especially images) after some time of inactivity, by cleaning their databases of old dated articles, as they take much space and do not generate any kind of traffic. Our participation in CLEFNewsREEL using the "official" dataset is, for this reason, compromised. However, this fact did not prevent us from testing our algorithms on another offline dataset. The data used are daily dumps from the plista ORP platform, just like the original dataset with a much more recent date (May 2016).

The algorithms developed and tested are the following:

- Task 1: Baseline1
- Task 1: Baseline1 + Faces
- Task 2: Baseline1
- Task 2: Baseline1 + Faces
- Task 2: Baseline1 + Faces + Salience
- Task 2: Baseline2
- Task 2: Baseline2 + Faces + Salience

Baseline1 is a Popularity with a freshness windows of 100 items, while Baseline2 is Random with the same freshness windows. For the remaining part of the paper these two algorithms will be called Pop100 and Rand100. By looking at the difference between the image enhanced algorithm and the relative baseline we can understand the effectiveness of image-based recommendation in the news environment.

3.1 Algorithm

Although the algorithms deployed in the Living Lab Evaluation (Task 1) differed from the one deployed in the Evaluation in Simulated Environment (Task 2), the logic behind them is quite similar and can be summarized as follows:

A recency windows for each combination of category/domain is created, each window encompassing 100 items. Every time a new update comes in, it is processed by taking the url_img field and scraping the corresponding image from the website. Features for the image are computed with our image processing algorithms, namely Viola-Jones [14] for face detection and spectral residuals [7] for the saliency map. The saliency map involves the extraction of several subfeatures (e.g., number of objects and their positions, background to foreground ratio) which are then used to detect if the image satiefies the requirement of being a single cluster in the middle of the image. This newly processed item is then added to the possible recommendations list, while the oldest item in the list is discarded (if full).

For the Pop100 algorithm: These items are sorted by a popularity score, which is an aggregation of how many impressions the item has received plus how many clicks it received in previous recommendations. Whenever a recommendation request arrives, the top N items are selected and only picked if they individually satisfy the "visual requirements" (see 2.2). If not enough items have been gathered before the top C elements have been considered, then standard popularity is used instead, without taking into consideration the "visual requirements" in order to fill the remaining spots. For the Rand100 algorithm, the logic is the same, however the ranking step is replaced by a random picking of items. The first C random times the item will be picked only if it satisfies the "visual requirements", after C times this restriction decays.

The constant C has been determined from empirical testing, and it can be interpreted as a tradeoff between "being interesting" and "following the baseline". In case of Pop100, the smaller C the most the items will be popular and less "visually interesting". As our intention here is to test if the visual component has an effect, C has been intentionally exaggerated in order to make the effect more notable.

4 Results

4.1 Living Lab Evaluation

The Online results show the data obtained from the scoreboard in the ORP during the evaluation window. Although the evaluation itself ran for around 40 days, not all the days have been taken in consideration due to issues which resulted in the recommender receiving a low volume of requests. As a result, only 24 days have been considered for the results. In order to answer our research question we decided to benchmark our image enhanced algorithm against its own baseline without image information. As for the Online, Pop100 is the baseline.

As can be seen from the Fig. 2: although the image enhanced recommender had more overall clicks, the baseline performed better in CTR value over long period of time. The Pop100+Faces sees a 28% decrease in CTR over the baseline Pop100. Our conclusion is that the lower result is actually due to a mixture

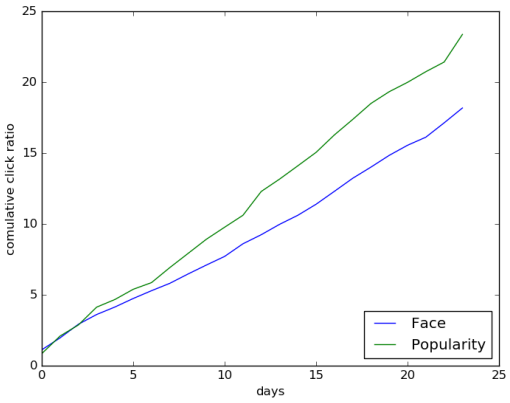


Fig. 2: Online Pop100 vs Pop100+FacesDetection: Cumulative CTR

of technical problems that most likely undermined the performance of the algorithm. A rundown of the problems can be found in the discussion in section 5.1

4.2 Evaluation in Simulated Environment

The Task 2 evaluation was done by using the dataset from ORP daily dumps. Three non-consecutive days have been used as a test set. We consider three days to be the minimum-sized data set large enough to provide a reliable comparison. Each day has an average of 68.000 requests. Since the algorithm running in the Task 1 environment accumulated a total of 175.000 requests over a month, we needed three days to reach approximately the same number of requests to have a comparable size for the dataset. Further testing is planned over a larger dataset in the future. The evaluation metric works as following: A recommendation is a successful hit if the user lands on the recommended page within 10 minutes of navigating the website. In this testing we conducted tests over two different baselines: Rand100 and Pop100. Rand100 was introduced in order to "weaken" the strength of the baseline algorithm in order to better show the effect of the Image features. The results can be seen in the Table 1

Introducing Image-based recommendation leads to a clicks increase of 51% with respect to the baseline Rand100, while the increase is 36% with respect to the Pop100 when considering only faces, 22% with both features.

Table 1: Task 2 Results

Algorithm	Clicks	Requests	CTR
Rand100	258	204456	0.13%
Rand100+Face+Salience	390	202254	0.19%
Pop100	630	204120	0.31%
Pop100+Face	857	203893	0.42%
Pop100+Face+Salience	771	203979	0.38%

5 Discussion

The results from the Task 1 and Task 2 evaluation differ: we think that this may be due to the inherent difference between the testing environments. We discuss this with more details in this section.

5.1 Living Lab Evaluation

The results gathered during the evaluation window of a month suggest that the baseline (Pop100) performs better than the image-based algorithm. This can be partially attributed to the technical problems which the image-based algorithm faced when running online.

One of the problem encountered was to make the algorithm fast enough to keep up with the ORP rate of updates. While the requests sent by the platform do follow the performance of the algorithm (if the algorithm is struggling less requests are sent), this does not apply to the updates; therefore all the updates are sent at anytime. Updates are the "computationally intensive" part in our algorithm, as each update usually comes with an image that needs to be downloaded and analyzed. Updates tend to come in groups of 10 or more, making it necessary to queue them. Even when trying to solve the matter with various strategies, it sometimes happened that the next batch of updates came before the queue was all processed, making the queue longer and the processing time even longer, thus making the problem worse: if repeated enough times the server would crash and get rebooted, therefore going through a new cold start period. Longer queue and longer processing time meant longer delay to answer recommendation requests as well, thus failing due to the timeout time. The time resources available for this research were necessary limited and not all solutions to this problem have been explored.

5.2 Evaluation in Simulated Environment

The Evaluation method used in this task does make the CTR quite worse than the one obtained in Task 1, as there is no actual user answering directly to the recommendation shown. Therefore no direct CTR comparison can be made.

However the difference between the baseline and the baseline+visual information can be used to infer the effect of such features.

For both baselines Rand100 and Pop100 we can see a significant improvement of the CTR when we make use of the Image information. As expected the increase is bigger in the "weaker" baseline, Rand100. However the most striking difference is the improved performance over the Pop100, especially when compared with the results of the similar experiment conducted Task 1. This strengthens our idea that the Task 1 implementations results were jeopardized by the poor technical performance rather than the Image-based recommendation model.

5.3 Future Work

The algorithm and the approach developed during this challenge was intended to be an exploratory task. Much is still needed to indeed prove the real effect of images on the recommendation.

Both Task 1 and Task 2 testing needs to be continued on all the possible combinations of baselines and features used in this paper, in order to test both the single effect of the features independently and their strength against different baselines. This is especially needed in order to investigate further the difference between Task 1 and Task 2, especially in light of the results obtained in this paper. A larger dataset (including images) needs to be used for testing in Task 2. This is our aim in the forthcoming future. Improvement in efficiency and running times are needed in order to allow the algorithm to properly work in an Living Lab environment. The current implementation has many flaws that likely resulted in many delays and worse CTR. A possible approach could be to not compute images until they reach a minimum level of popularity: this would filter out many "socially uninteresting" images.

Although this paper has focused its attention on the exploitation of high level visual clues (people, saliency map), a more in depth analysis of other feature classes may reveal useful insights. Notable global features include colorfulness, brightness and saturation. Another interesting approach could be the inclusion of visual information of how and where the recommendation is displayed (website related features). All of this on top of a more refined approach to the classification, by introducing different degrees of interestingness in the process.

5.4 Conclusion

Task 1 and Task 2 results seems to contradict each other at the first look. Task 2 shows an increase of the recommender performance while Task 1 shows a decrease. We can partially explain the difference by the fact that early Task 1 implementation ran in technical difficulties typical of the online environment, which partially jeopardized the final outcome.

By looking at the Task 2 results we can clearly see an improvement of the CTR when introducing image-based recommendations. This initial result seems to suggest a great improvement even when combined with already strong baselines (Popularity/Recency). More experiments with different baseline combinations and settings are required in the future to definitively prove the effectiveness of image-based recommendation in the news environment. We think that the results shown in this paper provide a good initial confirmation of its potential.

References

1. Haibin Cheng, Roelof Van Zwol, Javad Azimi, Eren Manavoglu, Ruofei Zhang, Yang Zhou, and Vidhya Navalpakkam. Multimedia Features for Click Prediction of New Ads in Display Advertising. In *18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 777–785, 2012.
2. Sagnik Dhar, Tamara L. Berg, Stony Brook, Vicente Ordonez, and Tamara L. Berg. High level describable attributes for predicting aesthetics and interestingness. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1657–1664, 2011.
3. Xiaoli Fern. The Impact of Visual Appearance on User Response in Online Display Advertising. *Proceedings of the 21st international conference companion on World Wide Web*, pages 457–458, 2012.
4. M Gygli, H Grabner, H Riemenschneider, F Nater, and L Van Gool. The Interestingness of Images. *Computer Vision (ICCV), 2013 IEEE International Conference on*, (iii):1633–1640, 2013.
5. Frank Hopfgartner, Torben Brodt, Jonas Seiler, Benjamin Kille, Andreas Lommatzsch, Martha Larson, Roberto Turrin, and András Serény. Benchmarking news recommendations: The CLEF NewsREEL use case. *SIGIR Forum*, 49(2):129–136, January 2016.
6. Frank Hopfgartner, Benjamin Kille, Andreas Lommatzsch, Till Plumbaum, Torben Brodt, and Tobias Heintz. *Benchmarking News Recommendations in a Living Lab*, pages 250–267. Springer International Publishing, 2014.
7. Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (800):1–8, 2007.
8. Benjamin Kille, Frank Hopfgartner, Torben Brodt, and Tobias Heintz. The plista Dataset. In *2013 International News Recommender Systems Workshop and Challenge*, pages 16–23, 2013.
9. Benjamin Kille, Andreas Lommatzsch, Gebrekirstos Gebremeskel, Frank Hopfgartner, Martha Larson, Jonas Seiler, Davide Malagoli, Andras Sereny, Torben Brodt, and Arjen de Vries. Overview of NewsREEL'16: Multi-dimensional Evaluation of Real-Time Stream-Recommendation Algorithms. In Norbert Fuhr, Paulo Quaresma, Birger Larsen, Teresa Goncalves, Krisztian Balog, Craig Macdonald, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction 7th International Conference of the CLEF Association, CLEF 2016, Evora, Portugal, September 5-8, 2016*. Springer, 2016.
10. Benjamin Kille, Andreas Lommatzsch, Roberto Turrin, András Serény, Martha Larson, Torben Brodt, Jonas Seiler, and Frank Hopfgartner. *Stream-Based Recommendations: Online and Offline Evaluation as a Service*, pages 497–517. Springer International Publishing, 2015.

11. Judith A . Redi and Isabel Povoa. The Role of Visual Attention in the Aesthetic Appeal of Comsumer Images: a Preliminary Study. In *Visual Communications and Image Processing (VCIP)*. Intelligent Systems, Delft University of Technology, The Netherlands, 2013.
12. Paola Ricciardelli, Cristina Iani, Luisa Lugli, Antonello Pellicano, and Roberto Nicoletti. Gaze direction and facial expressions exert combined but different effects on attentional resources. *Cognition and Emotion*, 26(6):1134–1142, 2012.
13. Andreas E. Savakis, Stephen P. Etz, and Alexander C. P. Loui. Evaluation of image appeal in consumer photography. *Proc. SPIE 3959*, 3959:111–120, 2000.
14. P Viola and M Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition (CVPR)*, 1:I—-511—-I—-518, 2001.