Fine-tuning SalGAN and PathGAN for extending saliency map and gaze path prediction from natural images to websites

Fine-tuning SalGAN and PathGAN for extending saliency map and gaze path prediction from natural images to websites

Enrico Corradini[1], Gianluca Porcino[2], Alessandro Scopelliti[1], Domenico Ursino[1*], Luca Virgili[1]

[1] DII, Polytechnic University of Marche, Italy
[2] Data Lab, Daimler AG, Germany
[*] Corresponding author
{e.corradini, l.virgili}@pm.univpm.it; gianluca.porcino@daimler.com;
alessandro.scopelliti94@gmail.com; d.ursino@univpm.it

Abstract

In recent years, researches dealing with the study of visual attention have become very popular thanks to the enormous increase of Artificial Intelligence. Machine Learning and, in particular, Deep Learning allowed researchers to propose new predictive models operating on natural images. In the meantime, an increasing number of websites has been made available on the Internet. However, few approaches, aiming at extending the results obtained on natural images to web pages, have been proposed. In this paper, we provide a contribution in this setting by applying fine-tuning and other refinements to two existing GAN-based approaches (i.e., SalGAN and PathGAN) originally proposed to predict the saliency maps and gaze paths on natural images. Our ultimate goal is defining some variants of them able to deal with websites. In particular, our SalGAN variant represents one of the first attempts to employ GANs for saliency map prediction on web pages, whereas our PathGAN variant is the first attempt to adopt GANs for gaze path prediction on websites. Here, we present our proposals, highlight their main novelties, describe the tests done and the results obtained. We also highlight two further contributions of this paper, namely: *(i)* a new dataset, more complete than the existing ones, supporting the analysis of visual attention on websites, and *(ii)* a tool supporting a web page designer in her attempt to increase the visitor interest and curiosity.

Keywords: Saliency Maps; Gaze Paths; Generative Adversarial Networks; SalGAN fine-tuning; PathGAN fine-tuning; FiWI enrichment

1 Introduction

Year by year, more and more contents are available for people on the Web. For instance, during 2018, it is estimated that 1,500,000,000 websites, along with their related information, products, services, etc. were online[1]. In this "jungle", the capability of capturing the attention of a user when she

[1] `https://www.internetlivestats.com/total-number-of-websites/`

visits a website is crucial [7]. Indeed, the design of a website capable of effectively conveying the desired message could lead to an increase of popularity and, possibly, of returns, for the corresponding company.

However, the evaluation of the attention paid by a person while watching a picture is not trivial and depends on several factors. Thankfully, it is possible to rely on two powerful tools to reach this goal. They are saliency maps [3] and visual scanpaths [13], which represent a formal definition of the areas where a user poses her eyes and the path made by her gaze, respectively. In the past, the first application scenario of these concepts was the one of natural images. However, with the increase of the number of websites available on the Internet, the interest on evaluating saliency maps and visual scanpaths also on websites has enormously increased. In fact, this last scenario is very valuable for a company, because it could increase its earnings if the website is able to capture user attention, in particular on the products/services it offers.

Scientific literature provides many approaches, belonging to different categories, to achieve this goal. In particular, in recent years, we have witnessed an important development of deep learning, which has impacted many research issues, including the prediction of saliency maps and visual scanpaths. One of the first proofs of the effectiveness of deep learning-based techniques in this setting is reported in [39]. After this attempt, several approaches involving neural networks have been proposed, and most of them achieved important results. In particular, an architecture that has recently gained a lot of attention and has several sophisticated applications is Generative Adversarial Networks (hereafter, GANs) [40, 17, 30, 9]. It is well-known that this architecture can be employed to address different issues and, thanks to it, satisfactory results have been obtained in many fields. Even in the prediction of saliency maps and visual scanpaths, GANs provided satisfying outcomes in the evaluation of user attention [31, 26, 1]. As a matter of fact, they achieved the state-of-the-art results in this field.

Actually, the vast majority of approaches involving GAN-based architectures for the prediction of saliency maps and visual scanpaths has been developed only for operating on natural images and not on websites. Indeed, to the best of our knowledge, as far as the web domain is concerned, few GAN-based approaches are able to evaluate saliency maps [26], and none of them can compute visual scanpaths. In this paper, we aim at filling this gap by proposing some GAN-based approaches to predict the saliency map and the gaze path of a user accessing a web page.

The approaches we propose here are variants of GAN-based approaches presented in the past literature and are specifically designed to work on websites. As will be clear below, the starting approaches (i.e., SalGAN [31], for saliency map prediction, and PathGAN [1], for gaze path prediction) were originally designed to operate in the context of natural images. Actually, the context of web pages is much more complex because more natural images, along with texts, logos and animations, can be simultaneously present in a single web page. The peculiarities of web pages make traditional computer vision saliency detection methods, such as the one described in [47], much less effective when applied to them than to natural images. The reason is that a web page presents several salient stimuli and competitions, which make it hard to accurately predict eye fixation [37]. We defined three variants of SalGAN, for saliency map prediction, and two variants of PathGAN, for gaze path prediction. As we will see below, the best variant of SalGAN and the two variants of PathGAN are fine-tuned. In addition, they present several other refinements taking into account various observations we made

during some experiments conducted "on the field". As we will see below, at the end of all these activities, we managed to achieve: *(i)* a SalGAN-based approach for website saliency map prediction that has a better performance than existing approaches carrying out the same task; *(ii)* two PathGAN-based approaches that, to the best of our knowledge, are the first ones proposed in the literature for gaze path prediction on websites.

In order to provide deep and accurate training, testing and evaluation of our approaches, we preliminary strived to create a new complete dataset, which could represent all the interface heterogeneities currently found in the web. In fact, existing datasets have some limitations in this aspect (see below). In order to construct such a dataset, we started from a popular existing one called FiWI (Fixations in Webpage Images)[37], and enriched it with new web pages, more in line with the current graphical standards, and new people involved. As we will see below, this much more complete dataset increased the quality of training, testing and evaluation of our approaches significantly.

Using our dataset, we tested: SalGAN and all our variants, in order to verify if one of them has a better performance than the others and several related approaches proposed in the past; *(ii)* PathGAN and its two variants, in order to verify if one of them has a better performance than the original PathGAN. In both cases, we obtained a positive result.

We implemented our approach in a user-friendly web application. In this way, a designer can easily upload her web page and, then, know in advance the behavior of future visitors when accessing it. Indeed, our web application returns both the saliency map and the gaze path of visitors accessing the uploaded web page. A designer can leverage the information returned to improve the user interface by moving its objects accordingly and verifying again the visitors reaction. The adoption of our tools allows web designers to reduce the number of meetings with the final users for evaluation purposes, which leads to save a huge amount of time and money.

Summarizing, the main contributions of this paper are the following:

- We propose some fine-tuned variants of SalGAN (and, then, select one of them), along with two fine-tuned variants of PathGAN, conceived for extending saliency map and gaze path prediction from natural images to web sites.
- We present a new dataset supporting training, testing and evaluation of approaches for predicting saliency maps and gaze paths of users accessing websites.
- We illustrate a tool supporting a web page designer to organize the graphical layout of the page in order to increase the visitor interest and curiosity.

The outline of our paper is as follows: In Section 2, we examine related literature. In Section 3, we provide a technical description of our approaches. In Section 4, we illustrate the experiments performed and evaluate the results obtained. Finally, in Section 5, we draw our conclusions and have a look at possible future developments.

2 Related work

Saliency map and visual scanpath (also called gaze path) have attracted a lot of interest from researchers in recent years. These concepts are really close to each other; as a matter of fact, the past

literature offers some approaches exploiting both of them [27, 33]. We can classify related approaches proposed in the past literature according to the type of architecture (in which case, we can distinguish traditional approaches and deep learning-based ones), and the domain they are tested in (which could be natural images or websites).

Before examining the state of the art of saliency map generation, it is fundamental to understand the definition of this concept. In [3], the authors say that saliency

> "intuitively characterizes some parts of a scene - which could be objects or regions - that appear to an observer to stand out relative to their neighboring parts".

As stated before, in the saliency map generation literature, two different kinds of approach can be identified, namely traditional ones [16, 35, 46, 42, 19, 41, 25] and deep learning-based ones [39, 28, 31, 24, 14, 26, 36].

As for traditional approaches applied on natural images, the authors of [16] propose the generation of saliency maps through a graph-based methodology using Markov chains. In particular, the corresponding approach first creates activation maps for each feature channel and then normalizes these maps to highlight important areas. Furthermore, the authors of [35] introduce a bottom-up framework for both static and space-time saliency detection. This framework computes the local regression kernels from a given image, which measure the likeness of a pixel to its surroundings. The result consists of a saliency map, where each pixel indicates the statistical likelihood of saliency of a feature matrix, given its surrounding feature matrices.

Shifting the focus to the evaluation of saliency maps in web pages, the authors of [19] introduce a model to predict both the locations of the most attended information and the corresponding attention sequence on a web page. This model considers three features for each element of a web page, i.e., chromatic contrast, size and position. Then, it computes a parameter, called attention factor, which summarizes the attention to the page paid by a user. The authors of [41] extend a web page saliency model by including the history of the previous interactions. They show that adding spatial conventions to a saliency map can help the prediction of the attention deployment within web page interfaces. In [25], the authors propose a framework to predict visual attention on web pages through the extraction of multi-features and a machine learning algorithm. This framework considers both bottom-up and top-down factors, such as color, orientation and intensity contrast, subband features, position bias, and so on. These factors are given as inputs to a Support Vector Machine algorithm that generates a saliency map. The overall approach was tested on the FiWI dataset [37]. Although the framework of [25] is really interesting and has given an important contribution to the saliency map prediction, the new deep learning-based approaches have proved to obtain more accurate results than the previous ones [26].

Deep learning-based approaches have been developed during the last years thanks to the recent growth of research efforts in this field. Actually, deep learning has contributed to the design of increasingly sophisticated frameworks to create saliency maps. One of the first examples is reported in [39], where the authors show that deep learning-based techniques can be employed in this scenario. Again, for a better understanding of these approaches, we can classify them according to the context they operate in; in particular, we have approaches working on natural images [39, 28, 31, 24] and others operating in web page layouts [37, 14, 26].

Regarding natural images, the authors of [28] obtain remarkable results using a recurring Convolutional Neural Network (hereafter, CNN), which extracts features and takes the spatial Long Short-Term Memory (hereafter, LSTM [18]) into account. The authors of [24] introduce two new models employing a unique architecture but different feature spaces. The former predicts human fixations-based on deep neural network features, trained on object recognition. The latter uses purely low-level features (e.g., the isotropic contrast). The authors compare the two models to highlight the relevance of low- versus high-level features in predicting fixation locations.

Unlike natural images, websites have been rarely investigated in the past. One of the first approaches generating saliency maps for websites is reported in [37]. It proposes the usage of multiple kernel learning to integrate several feature maps. In [36], the authors present a framework that combines low-level features and high-level representations from deep neural networks of images. It also exploits a filter on early features to inhibit the noise of text, pictures, logos and animations on the web pages, as well as a PCA to reduce the dimension of the output of the deep neural networks. This approach was tested on the FiWI dataset, and obtained interesting results in terms of standard metrics.

During the last years, a class of deep learning architectures, i.e., GANs, has achieved outstanding results in the saliency prediction for both natural images and web pages. For instance, the authors of [31] propose a GAN architecture, called SalGAN, consisting of two different networks. The former predicts saliency maps from the raw pixels of an input image, while the latter takes these maps and, for each of them, tries to determine whether it is a predicted one or a ground truth. In this way, SalGAN is expected to generate saliency maps resembling the ground truth. As it is reported in Section 4.3.1, the approaches employing SalGAN-based architectures achieve better results than the ones proposed in [25, 36]. TSGAN (Two-Stage Generative Adversarial Network) [26] is a GAN-based architecture employing an autoencoder to generate saliency maps. Its strength is the usage of a two-stage generator, which creates a coarse saliency map during the first stage, and refines it during the second one.

Apart from saliency maps, visual scanpaths have obtained the interest of researchers as well. Indeed, the past literature provides several approaches to perform this task. They are based on traditional techniques [8, 44, 27, 22, 13, 10] or deep learning [6, 1, 21, 43, 38]. Also in this case, traditional approaches can be classified according to the context in which they are applied; again, contexts could be natural images [45, 27, 8] or websites [22, 13, 10].

As far as natural images are concerned, the authors of [27] model scanpaths through three main factors influencing human attention; these are low-level feature saliency, spatial position and semantic content. Low-level feature saliency is represented by means of transition probabilities between different image regions, while spatial positions and gaze shifts are modeled through a Levy flight and a 2D Cauchy distribution. A Hidden Markov Model (hereafter, HMM) with a Bag-of-Visual-Words descriptor of image regions is used for taking semantic content into account. The authors of [8] propose a method for scanpath modeling and classification. It relies on variational HMMs and discriminant analysis. Specifically, HMMs encapsulate the dynamic and individualistic dimensions of gaze behavior, allowing discriminant analysis to capture systematic patterns diagnostic of a given class of observers. Interestingly, this approach is released as a Matlab toolbox freely available.

Regarding the web page layouts, the authors of [13] present several compact visual scanpath repre-

sentations. They introduce three categories of representations, namely scaled traces, time expansions and radial plots. Then, they employ all these representations through heuristic algorithms to find the better strategy to apply for scanning. Furthermore, in [10], the authors introduce a scanpath clustering algorithm called Scanpath Trend Analysis. It considers not only the visual elements visited by all users, but also those ones visited by the majority of users in any order. It first analyzes the most visited visual elements in given scanpaths; then, it clusters scanpaths by arranging these visual elements based on their overall positions in them; finally, it constructs a trending scanpath starting from these visual elements.

Besides traditional approaches, a lot of deep learning-based techniques have been employed to evaluate visual scanpaths [6, 1, 21, 43, 38]. All of them have been developed for natural images. Instead, to the best of our knowledge, no approaches for websites have been proposed in past literature.

The authors of [6] propose an approach based on regions of interest and inhibition of return. This last feature avoids predicting fixations already observed in the path. Detecting regions of interest, instead, ensures that fixations are only in salient regions of the image. In this approach, LSTM layers are used to consider the time dimension of the problem. The authors of [1] present PathGAN, a conditional GAN architecture to predict gaze paths. The generator receives the image of a web page layout and generates the corresponding path. The original image and the generated data are, then, fed to the discriminator, which evaluates its equality. The authors of [43] propose an approach employing both CNNs and HMMs. In this approach, a LSTM neural network learns the mapping of image features to eye fixations by modeling the sequential dependencies of the fixations in a scanpath. Then, it leverages a new data augmentation technique based on HMM, which increases the number of available images and trains the LSTM appropriately.

3 Description of the proposed approaches

3.1 Improving SalGAN to derive saliency maps for web pages

As pointed out in the Introduction, in order to derive saliency maps for web pages, we started from SalGAN [31], because this approach has proven to be the most accurate in the prediction of saliency maps for natural images. Then, we performed several adjustments to make it more suitable to operate on websites and to return accurate results.

Here, we feel important pointing out that, during our research, we also started from TSGAN [26] and tried various refinements on it, in order to improve its performance. As pointed out in Section 2, to the best of our knowledge, TSGAN is the only already existing GAN-based approach to predict saliency maps of users on websites. Actually, all our attempts to obtain improved versions of TSGAN have been unsuccessful. Therefore, as we will see in Section 4.3.1, in conducting the test campaign for evaluating the performance of our SalGAN variants when they are applied on websites, we compared them to the original TSGAN and not to our proposed variants.

We started by investigating how SalGAN behaves when it is directly applied on websites. The architecture of SalGAN is reported in Figure 1. The generator is a simple single stage autoencoder that generates saliency maps from input images. The discriminator receives both generated and real images and must identify which of them are coming from the real data distribution. The SalGAN

loss is built around the standard GAN loss function, customized to obtain better results on images. Authors have also published pre-trained weights of their network on a dataset made of natural images.

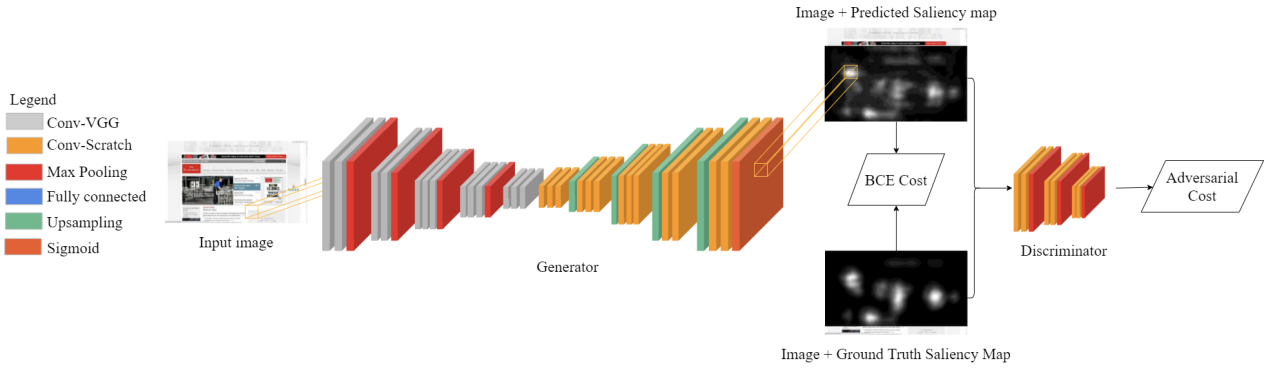


Figure 1: The architecture of SALGAN

In order to apply SalGAN to web pages, we focused on fine-tuning this model in the best possible way, leaving most of the network structure unchanged.

We did not consider necessary the change of the architecture, as it already performs very well on real images. Furthermore, we left the structure of the loss unchanged, with one part measuring the content loss and the other one measuring the adversarial loss [31]. The main fixed point we had was to keep frozen the layers referring to the encoder inside the generator. In fact, in this part of the network, the authors of [31] use the pre-trained VGG network structure because it has been shown to accelerate the convergence of the model. TSGAN also uses this approach, which greatly reduces the amount of time required to obtain a good training. Instead, we have considered necessary a complete re-training of the second part of the generator, i.e., the decoder, where the saliency map is generated and adapted as much as possible to the web page domain.

The reasoning underlying our choice of freezing only one part of the generator concerns the different goals of the two parts. The first part is responsible for recognizing objects in the input image. It already obtains very satisfactory results with the pre-trained configuration. Therefore, we decided to keep it unchanged. The second part has the goal to create the saliency map. Since the generator of [31] is trained only on natural images, it can create saliency maps suitable for them, while it makes several faults in performing predictions in an artificial domain, such as the one regarding websites. For this reason, the second part of the generator needs to be trained again so that it can learn how to create saliency maps for both natural images and web pages.

In addition to the first part of the generator, we also decided to freeze the first four convolutional layers of the discriminator. The reason for this choice is similar to the previous one. In particular, we need to freeze the first layers of both the generator and the discriminator in order to maintain the right level of competitiveness between these two neural networks, which is crucial to get fine results from a GAN architecture. For example, training the discriminator from scratch implies that all the weights obtained from the natural image dataset must be recomputed, which would lead the discriminator to overfit on the web page domain, where it would train very quickly, being this set small and very specific.

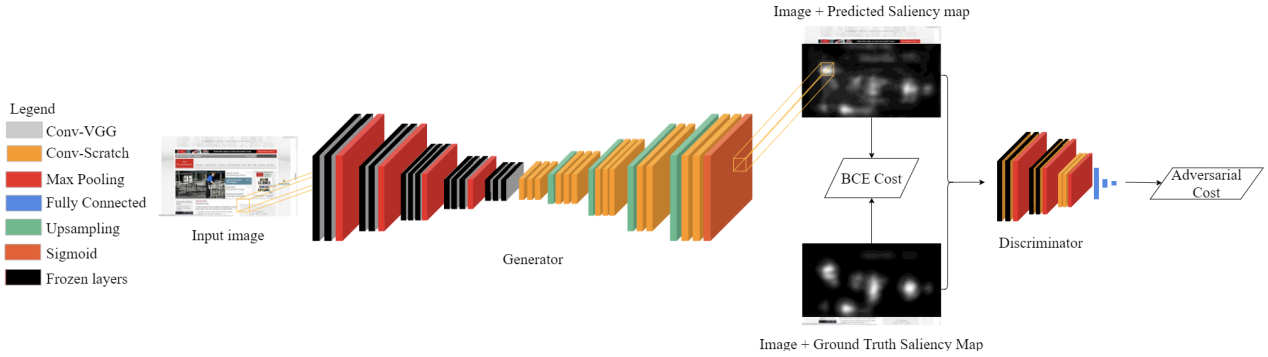


Figure 2: SalGAN frozen layers during training

Instead, the choice to freeze the first four layers of the discriminator keeps its ability to distinguish between real and fake saliency maps almost completely intact. In fact, if we compare two saliency maps, one coming from the natural image domain and one coming from the web page domain, it should be difficult to determine which comes from one domain instead of from the other. In their own right, saliency maps from these two different domains can be considered similar because their structure does not present remarkable differences. The features that the discriminator has learned as determinant for asserting the quality of a saliency map are common in both domains. This is the reason for which it is important to preserve what the network has learned previously, avoiding the training of these first levels. With this choice, training improves the quality of the saliency maps produced thanks to a discriminator with a lot of "experience".

In Figure 2, we can see the layers of the network that have been frozen. We obtained the optimal number of layers to keep out of training after making preliminary observations on the quality of generated images. Deriving the optimal number of layers to freeze implies a trade-off on how many layers we should train on our dataset and how many layers we should keep with the same weights provided by [31]. For the sake of space, we are not reporting all the experiments we made, but the reasoning underlying our choice. Indeed, as we pointed out before, the first layers of both generator and discriminator are devoted to extract features from the input image, while the next ones are used for creating a saliency map, and detecting if the input saliency map is real or fake, respectively. In this perspective, if we freeze more layers than the optimal solution we found, the resulting SalGAN would not be able to adapt to the web domain, since there are few layers to train on our dataset. On the other hand, if we train more layers than the optimal solution, we both lose the training weights of [31] and overfit the resulting SalGAN to the web pages layout, thereby taking away the capability to perform well with natural images.

Furthermore, we decided to lower the learning rate of the neural network from $3 \cdot 10^{-4}$ to $1 \cdot 10^{-4}$. This allows a more gradual, but smoother and more stable, convergence to the optimal solution. A higher learning rate could allow a faster convergence to the optimal solution, but this would be done with the presence of "ups and downs" of the loss function before it reaches the possible convergence.

As a final remark, we point out that both the two refinements mentioned above are necessary to adapt SalGAN to the web layout domain. In fact, assume that we change only the learning rate parameter, without freezing any layers. The network weights provided in [31] must be recomputed. To

perform this task, we should train the whole SalGAN from scratch, which is a huge time-consuming task. Actually, we need to preserve the SalGAN's capability of working with both natural images and web layouts, because web pages could contain several natural images. Therefore, we must freeze the first layers of both generator and discriminator. This implies that we should keep the weights of the pre-trained networks.

On the other hand, assume that we keep frozen the first layers of both generator and discriminator and do not modify the learning rate parameter. This leads to an unsuitable scenario. In fact, maintaining the previous learning rate means training SalGAN too quickly, which makes the weight tuning unstable for many epochs, eventually resulting in mode collapse or unstable training.

As a conclusion to our reasoning regarding fine-tuning operations, in Figure 3, we report a qualitative visualization of the predictions returned by the original and fine-tuned SalGAN. Specifically, we report in this figure some examples of the results returned by these two models. From the analysis of it we can see that the saliency areas returned by the fine-tuned SalGAN are more in line with the ground truths. In fact, the original SalGAN identifies saliency areas that are not present in the ground truths, and does not report some saliency areas present in the ground truths. This is much less the case for the fine-tuned SalGAN, whose predictions are much closer to the ground truths than the original SalGAN.

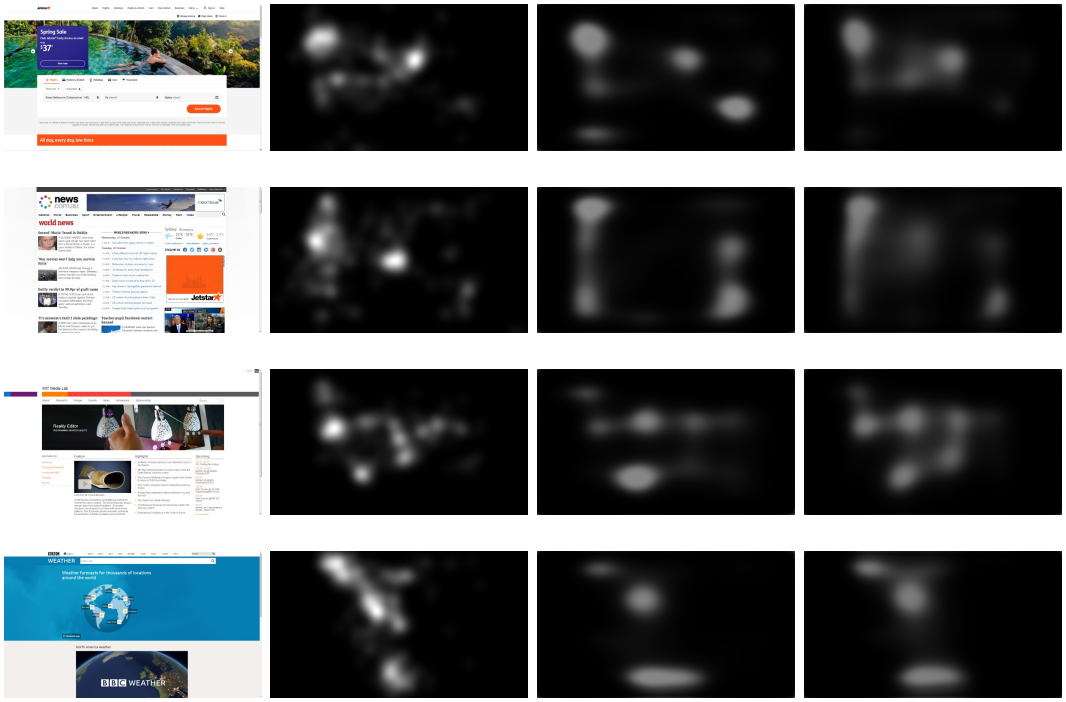


Figure 3: Qualitative comparison between the predictions of the original and fine-tuned SalGAN

The images shown in Figure 3 represent only qualitative examples of the potential of fine-tuned SalGAN. Beside a qualitative evaluation, it is important to quantitatively verify the possible benefits brought by the fine-tuning procedure. To this end, in Section 4, we present the results of several tests

showing that fine-tuned SalGAN achieves better results than other models.

3.2 Improving PathGAN to derive gaze path predictions for web pages

In the previous sections, we defined an approach to derive saliency maps for web pages. However, saliency maps are not sufficient to understand the order in which the elements are seen by a user. In fact, unlike natural images, the layout of elements in a web page affects its ability to capture the user's attention. This is the main reason why we have defined an approach for estimating the gaze path in a web page. Indeed, there are several practical applications, which are really difficult (or even impossible) to perform with saliency maps alone. Some examples of such applications are:

- understanding how user behavior is affected by different web page layouts;
- finding the best priority order for the elements of a web page;
- performing automatic A/B testing on different layouts.

In the field of eye movement prediction, scientific research did not achieve many important results yet. The lack of annotated datasets makes the creation of new models not easy: no data very often leads to the impossibility of performing a successful training. All the solutions proposed so far apply to the prediction of gaze on natural images. Among the limited studies in this field, a Generative Adversarial Network, called PathGAN [1], stands out for its results. PathGAN predicts the visual scanpath of people observing images, both in normal and in 360 degrees format. The quality of its results places it as one of the best performing models in this domain. In Figure 4, we report the architecture of PathGAN. The first part of the network is a generator; an input image is fed to obtain a gaze path, which represents the route of the eyes of a potential user observing that specific image. The generated path is a sequence of 63 fixations, each consisting of a tuple of four elements: a x-coordinate, a y-coordinate, a timestamp and an end of path probability. This last element allows the generation of paths of variable length; a threshold is set on it to determine which fixations should not be included in the final prediction. As expected, the first three values of each tuple include information on both position and duration of every fixations.

The first tests with the network did not give encouraging results in the GUI domain. We identified several problems that needed to be resolved to improve performances. First of all, the generator and discriminator weights are not updated with the same frequency. The choice to make more updates on the discriminator, rather than on the generator, did not lead to performance improvements. Moreover, assigning a very low weight to the content loss ($\alpha = 0.05$) makes the discriminator very strong, compared to the generator. Training the generator for the first 5 epochs alone was still not sufficient to prevent this phenomenon. In addition, the number of values to be predicted complicated the problem too much. In order to allow the prediction of paths of variable length, the last element of each fixation tuple is an end of path probability. We noticed that the training did not manage this extra variable adequately, resulting in either very short or very long paths. Overall we experienced completely wrong predictions that, in the long run, during the training, led to mode collapse. In Figure 5, we reported two examples of mode collapse. In this figure, blue lines and squares represent

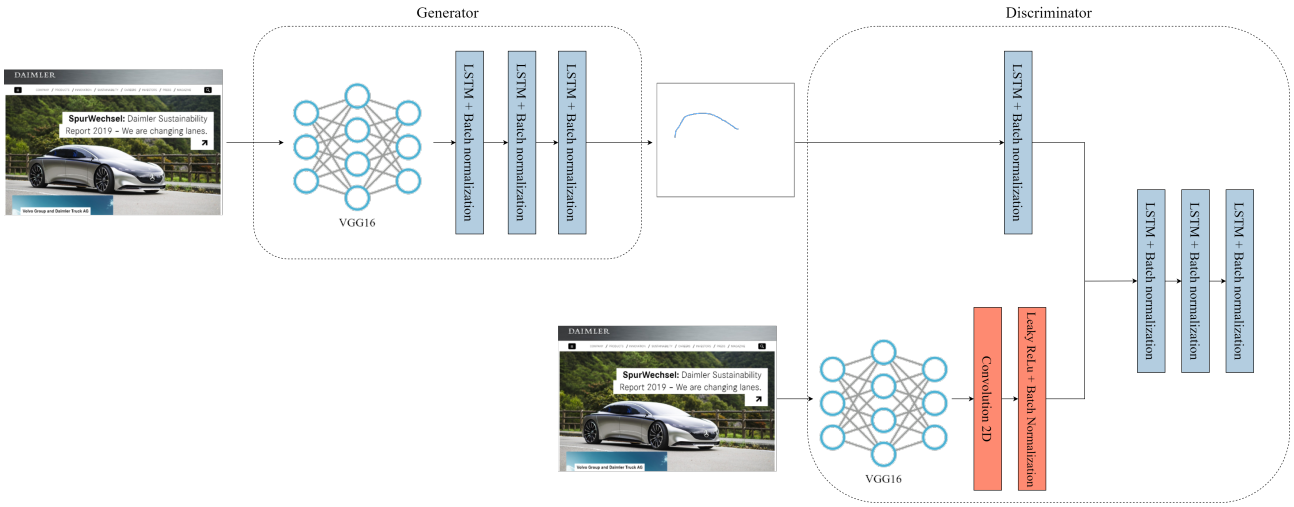


Figure 4: The architecture of the original PathGAN

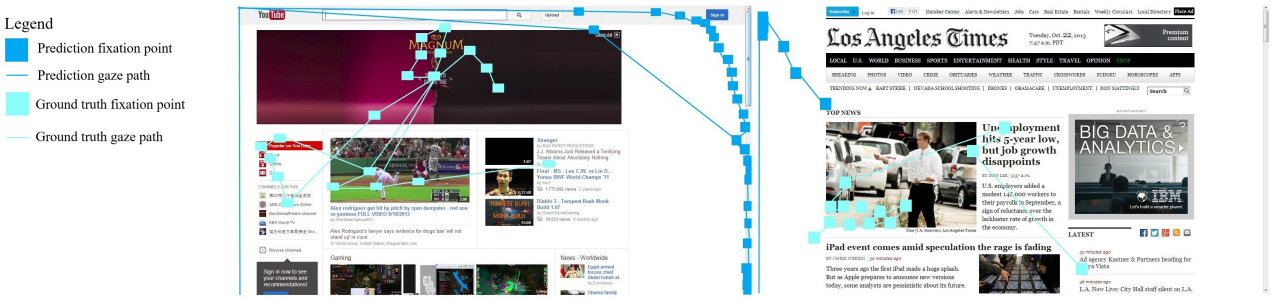


Figure 5: Two examples of mode collapse

the gaze path and the fixation points of the PathGAN prediction, respectively. Instead, the light blue lines and squares denote the gaze path and the fixation points of the ground truth.

During mode collapse the generator tends to predict outputs that match the edge of the image, completely ignoring the original ground truth. The triggering cause of this phenomenon was identified in the combination of several elements. The discriminator becomes too strong, compared to the generator, which can no longer make realistic predictions. The lack of data does not help in this regard. The discriminator clearly overfits on the training data after several epochs. Changing the number of times the generator and discriminator weights are updated does not prevent this.

An explanatory graph of this phenomenon is visible in Figure 6. As shown in this figure, the generator (blue line) is very strong in the early training stages because it was trained alone for the first five epochs. In the first steps, the discriminator loss starts to decrease gradually. It suddenly experiences a huge drop that matches with a degradation of the generator's loss score. From that point onward, the predictions start to be totally wrong. It is clear that the network needs some adjustments before being applied to our domain.

All the improvements we have introduced to the network strive to mitigate the problems described above. Since we did not have a dataset as large as the one used by the authors of PathGAN, we

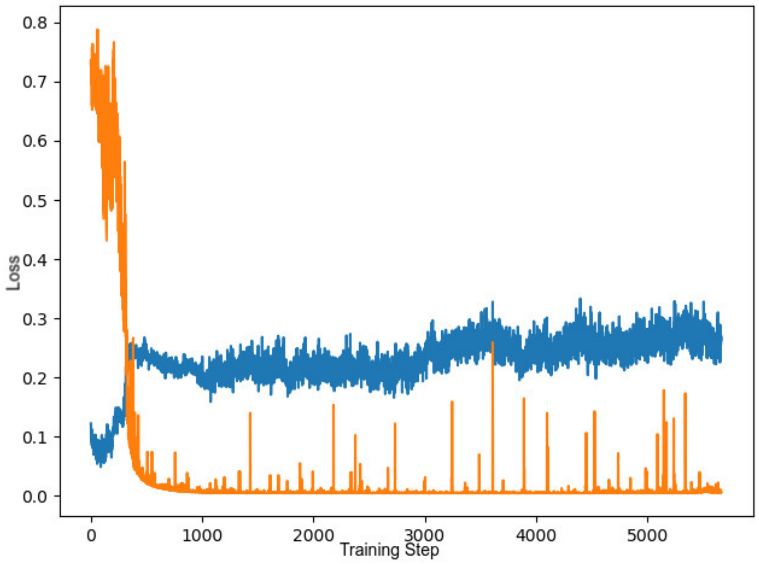


Figure 6: Generator (blue) and discriminator (orange) loss after that the discriminator overfits the training set

chose to reduce the complexity of the problem. Therefore, instead of predicting a sequence of tuples of four elements (x-coordinate, y-coordinate, timestamp, end of path probability), we chose to remove a variable. Our goal was to predict paths of different lengths with only three variables, instead of four. We removed the end of path probability to force the network to predict paths of the same length. However, since not all the paths of our dataset are of the same length, we opted to introduce dummy nodes on the arcs connecting two fixations. This solution makes sure that also our dataset has 63 fixation long paths. The dummy nodes have been added on the arcs by applying linear interpolation. Since, in reality, they do not correspond to a fixation, the timestamp is increased by a negligible constant. This allows us to distinguish which are the real fixations and the fictitious ones. After post-processing predictions, we are able to generate a sequence of real fixations. Each timestamp is transformed into a duration, allowing us to better distinguish the real fixations from the dummy ones. The predicted fixations with a duration below a threshold are considered dummy and, therefore, removed from the path. This procedure preserves the generation of paths of variable length without the need of the end of path probability.

As a result of this modification, we obtained a PathGAN model having a different output structure from the original one. In fact, this new version can generate a gaze path of variable length, in which the duration of human fixation points is in line with the literature [23], and there is no need for the end of path probability. This was already an important improvement because the output path is consistent with a real scenario. However, we believed that this is not enough because it produces changes only in the structure of the model outcome.

The next step was to introduce improvements in the way the network is trained. At the same time, we also changed the weight assigned to content and adversarial loss. The only way to make the discriminator weaker is to update the generator weights more often.

At first we tried to keep the weights assigned to the various parts of the loss unchanged. Initially, we tried to tune the number of weight updates for every training step of both the generator and the discriminator. After several attempts, we realized that we were just postponing the moment when the discriminator would overfit. Therefore, it proved essential to also modify the weights of the various parts of the loss. We saw positive effects when we decreased the weight given to the adversarial loss within the objective function. A higher content loss weight prevented the discriminator from taking over. The quality of the samples generated increased dramatically, allowing network training to be completed successfully. In conclusion, we decided to multiply the adversarial loss by a constant equal to 0.35, and the content loss by a constant equal to 1. Moreover, we found the right number of weight updates for every part of the network; specifically, we decided to update 16 times the generator weights every step, limiting the discriminator to only 4 times.

We also introduced other modifications aimed at avoiding overfitting. We took our cue from saliency prediction models and added noise to the images passed to the discriminator. Also in this case, the qualitative evaluations of the output, together with the loss trend, were fundamental; in fact, we could immediately detect any overfitting and mode collapse. We undertook further attempts to improve the results, but without success. For example, we tried to modify the network to receive a saliency map input. Both the generator and the discriminator should have benefited from this modification, because there is a match between saliency map and path. Instead, we noticed that there were no tangible benefits; on the other side, the complexity of the architecture increased. We also tried to modify the path preprocessing; in particular, instead of adding dummy nodes on the arcs, we tried to superimpose them on existing nodes. This should have brought more precision in predicting fixations. Again, we noticed no improvement. We believe that further attempts can be made using analogous techniques in the future. After all these changes and improvements to the original PathGAN, we obtained a new version of it, which we called NormalGAN. This has the same architecture as PathGAN. However, thanks to the changes explained above, it is able to deal with the web page scenario.

Beside this first version of improved and fine-tuned PathGAN, we designed a second one. In this new version, we started from the considerations that had led us to define NormalGAN and flanked them with additional considerations that prompted us to make further changes. In particular, we modified the network making it to follow the structure of a conditional Wasserstein GAN [15] (we call it WGAN in the following). We also modified the training process to respect the characteristics of a WGAN. Moreover, we updated the weights of the generator and the discriminator with different frequencies. In particular, the discriminator was updated more often because weight clipping was introduced. The discriminator was updated 5 times, while the generator was updated only once. We also reset the weights of the various terms of the loss to their original values; in particular, we set content loss to 0.05 and adversarial loss to 1.

Setting the update rate of the weights and the constant that multiplies the loss function allowed us to achieve a balance between the strength of the generator and the discriminator. In fact, increasing the update rate of the weights leads to a scenario where the generator and/or the discriminator learn too much from our dataset, which causes overfitting. On the other hand, decreasing the update rate of the weights implies that the training process takes longer or that, for the same duration, the generator and/or the discriminator cannot learn enough from the dataset. The constants that multiply the

loss functions are even more important because they tune the balance between the generator and the discriminator. Recall that, in a GAN scenario, both the generator and the discriminator learn from the other's errors. This process requires the right amount of time. For example, if we increase the constant that multiplies adversarial loss, the discriminator will have much more power than the generator, which means it would be able to discriminate real paths from fake ones without giving the generator the time necessary to acquire enough information from that and react appropriately. By contrast, decreasing the weight of the adversarial loss leads to a weak discriminator, which is fooled by the generator. A similar reasoning can be made for the constant that multiplies the content loss. In Table 1, we summarize these considerations, along with the corresponding ones related to the setting of all the other parameters involved in the two variants of PathGAN.

	Parameter	*Our solution*	*Lower values*	*Higher values*
NormalGAN	Constant multiplying the adversarial loss	0.35	The Generator overcomes the Discriminator	The Discriminator overcomes the Generator
	Constant multiplying the content loss	1	The Discriminator overcomes the Generator	Not feasible
	Update frequency of the generator weights	16	Generator underfitted	Generator overfitted
	Update frequency of the discriminator weights	4	Discriminator underfitted	Discriminator overfitted
WGAN	Constant multiplying the adversarial loss	1	The Generator overcomes the Discriminator	Not feasible
	Constant multiplying the content loss	0.05	Not feasible	The Generator overcomes the Discriminator
	Update frequency of the generator weights	1	Generator underfitted	Generator overfitted
	Update frequency of the discriminator weights	5	Discriminator underfitted	Discriminator overfitted

Table 1: Overview of the parameters of our PathGAN versions

The improvements we made to the original PathGAN can also be observed by analyzing the loss values of the generator and discriminator of WGAN, shown in Figure 7. From the analysis of this figure, we can see that the discriminator does not overfit, as previously happened in Figure 6, because the loss values do not increase after some training steps. Furthermore, the loss values of the generator decrease rapidly and, therefore, reach an equilibrium point. This means that it has the time to learn how to create good saliency maps.

We report the WGAN architecture in Figure 8. As we can see, it is similar to the PathGAN architecture (and also to the NormalGAN one, because PathGAN and NormalGAN share the same architecture). However, unlike the latter, it does not include the batch normalization components.

As a first qualitative result of our work, we tested the obtained WGAN on the images that led the original PathGAN to collapse. In Figure 9, we report the results obtained. In it, colored lines

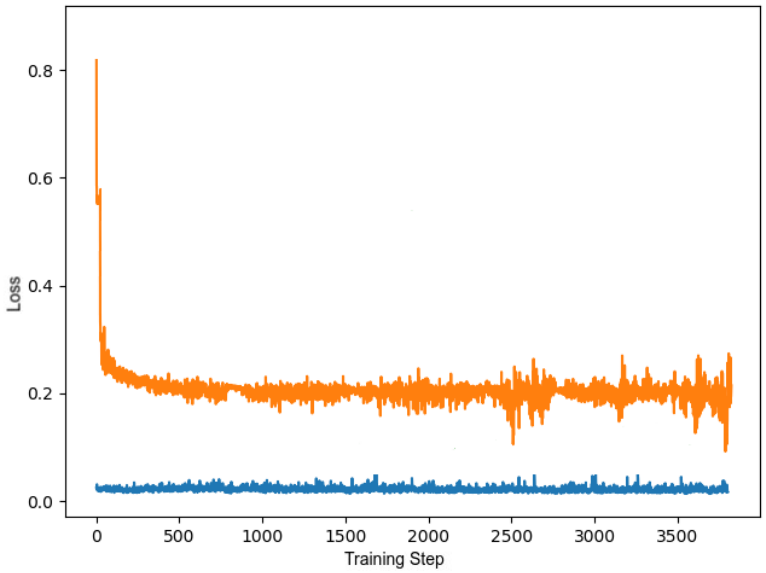


Figure 7: Loss values of the generator (blue) and discriminator (orange) of WGAN

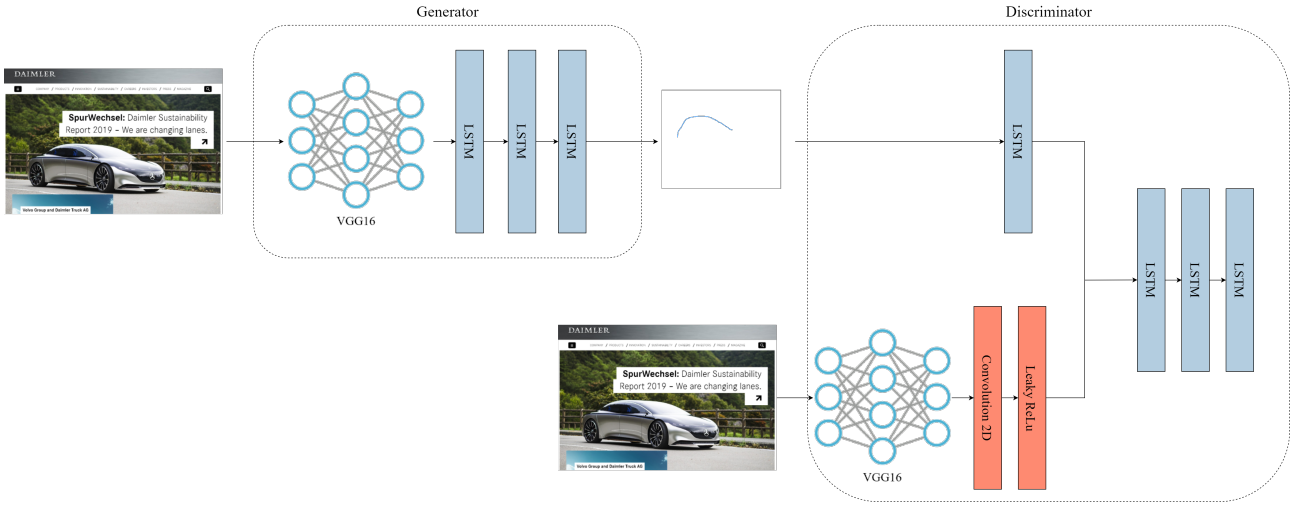


Figure 8: The architecture of WGAN

represent the gaze paths. The image on the left shows the ground truth, while the image on the right shows the prediction of WGAN. As can be easily seen, our model does not suffer from mode collapse and predicts gaze paths comparable with those of the ground truths.

Finally, in Table 2, we provide a summarization of the differences between the original PathGAN and the two modified versions that we are proposing in this paper.

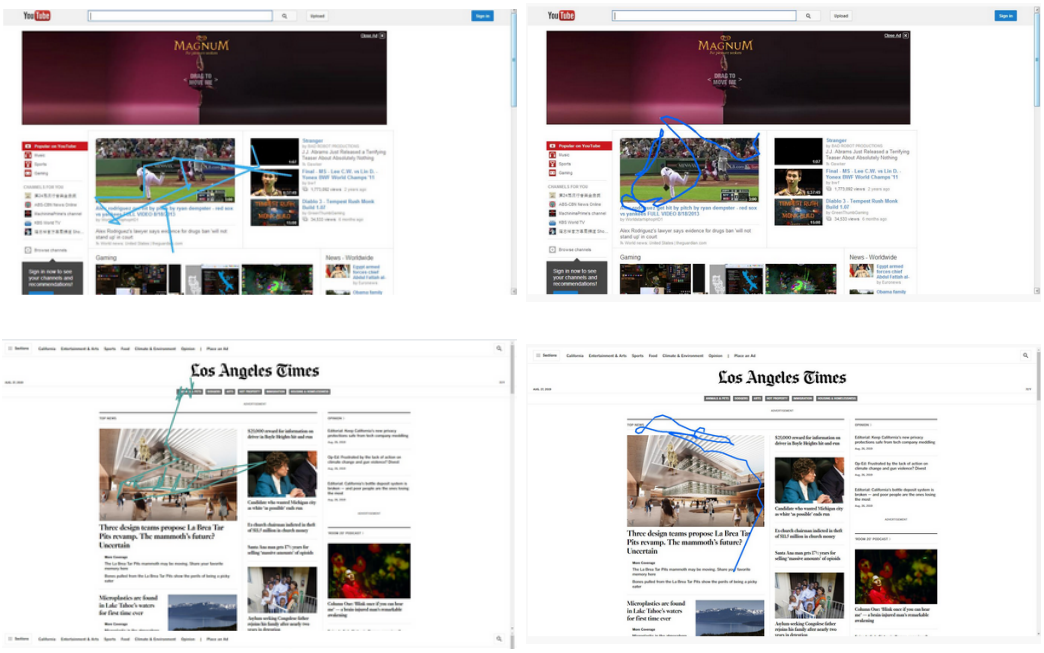


Figure 9: Ground truth (on the left) and WGAN prediction (on the right) of images that had led the original PathGAN to mode collapse

Original PathGAN	*NormalGAN*	*WGAN*
Best results with natural images	Fine-tuned for the GUI domain	Fine-tuned for the GUI domain
End of path probability	Fixed path length	Fixed path length
Content loss weight equal to 1	Content loss weight equal to 1	Content loss weight equal to 0.05
Adversarial loss equal to 0.2	Adversarial loss equal to 0.35	Adversarial loss equal to 1
Conditional GAN	Conditional GAN	Conditional Wasserstein GAN

Table 2: Differences between the original PathGAN and our proposed variants (i.e., NormalGAN and WGAN)

4 Experiments

4.1 Saliency map and gaze path prediction tool

In this section, we illustrate the tool implementing our approaches for the generation of saliency maps and gaze paths of users accessing websites. We strived to build a usable and efficient tool, which can be easily employed by any user (for instance, a designer), who wants to evaluate the effectiveness of a web interface. Our tool consists of a web application. We developed it in Python; in particular, we implemented the neural network-based algorithms representing the core of our approach using the well-known Keras and Tensorflow Python libraries. Moreover, we used Django as the core of our web application.

Having in mind the need to guarantee the best possible User Experience, we created a home page where a user can upload an image and specify if she desires to evaluate the saliency map or the gaze

path for that image.

In Figure 10 (resp., 11), we report an example of the output provided by our tool for saliency map (resp., gaze path) prediction.

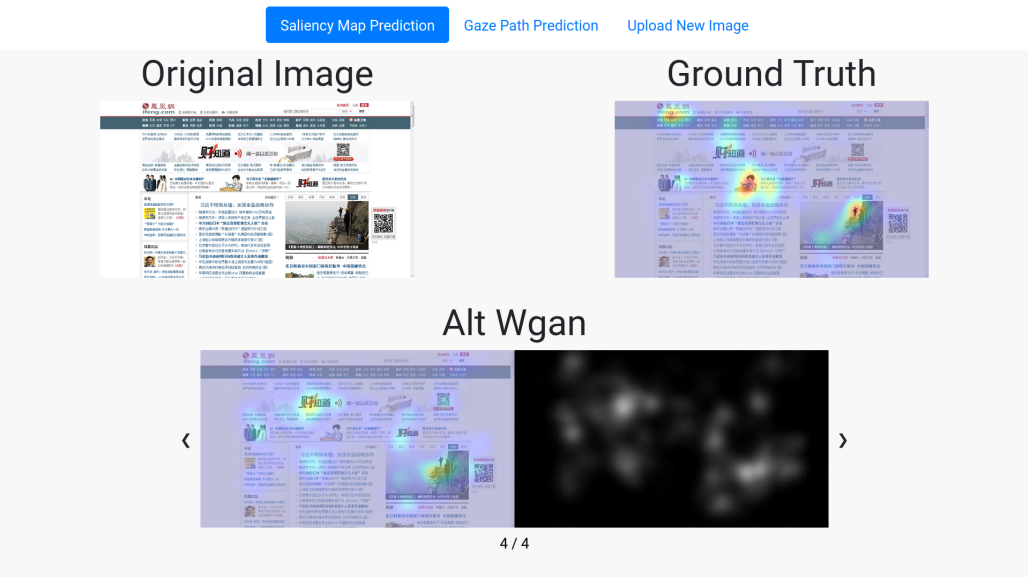


Figure 10: An example of a saliency map prediction returned by our tool

Figure 11: An example of a gaze path prediction returned by our tool

4.2 Dataset description

To the best of our knowledge, only one dataset containing both web page layout images and gaze data is available in scientific literature. This dataset, called FiWI (Fixations in Webpage Images) [37] is used in all researches concerning saliency map and gaze path prediction in the GUI domain. It was built by collecting data from 11 volunteers, each observing 149 websites. The limited number of volunteers involved in data collection makes FiWI incapable of completely enclosing all the ways in which human beings observe images. Furthermore, the user gender is not balanced in it, because 7 volunteers were women and 4 volunteers were men; this fact could introduce a gender bias. The data gathering procedure used for this dataset was very intensive because each volunteer was required to observe numerous datasets, each for 5 seconds, generating a considerable amount of stress to her/him. Furthermore, data was collected in an unnatural way, which could prevent the creation of a realistic model. In fact, volunteers were placed with their chin on a head rest, in a dark room, 60 cm away from the screen. Finally, the set of images of the dataset comes from the same time period. With regard to this aspect, we observe that web pages are subject to changes of style guidelines over time; for this reason, today's web pages are very different from the layouts present in the original FiWI dataset. This fact introduces a bias related to the evolution of the page design techniques over time.

All these considerations led us to build a new dataset aiming at avoiding, or at least mitigating, these biases.

Since the beginning, we thought it was necessary to increase the total number of images present in the dataset. We added new layouts to the ones already considered in FiWI, as they are useful to make the results as general as possible. The websites composing FiWI belong to three different classes, namely: *(i) Pictorial*, in which case the pages are occupied by a dominant picture, or several thumbnail pictures and little text (e.g., photo sharing websites); *(ii) Text*, whose pages contain high-density informational text (e.g., Wikipedia); and *(iii) Mixed*, whose pages present a mix of thumbnail pictures and text (e.g., social network sites). In our dataset, these classes have been extended while keeping balanced the fraction of websites belonging to each of them.

Furthermore, in our opinion, the three classes of websites were not fully representative of the whole variety of the World Wide Web. For this reason, we added a fourth class consisting of a set of *Business* websites, presenting analytical layouts (in particular, dashboards) and layouts of the Daimler intranet. Analytical dashboards and web pages of an intranet are very different from traditional websites, because they are not designed for ordinary web surfers.

Finally, we considered that the design principles used in the creation of websites change over time. For this reason, we decided to add in the dataset the updated layouts of the pages already present in FiWI. As a last task, we dropped from the final dataset the FiWI websites without an updated version available (e.g., web pages of companies that no longer exist) in order to obtain a balanced set of old and new layouts. Starting from this original core of 149 images, we arrived at a total of 262 web layouts.

Furthermore, it was necessary to consider more people than the FiWI dataset; for this reason, we collected data from 100 volunteers to include more nuances of how different people look at images. The need to have more testers made us focus also on who are the potential users of the websites under consideration. We felt that collected data should come from both an enterprise and a more general

environment. For this reason, 25% of the data collected come from employees of the Daimler AG, who are used to work with websites. Also the gender of enrolled volunteers was perfectly balanced. Due to time limitations and the difficulty to recruit old people, we were not able to balance the dataset by age groups, making it slightly unbalanced towards younger people. However, compared to FiWI, we have a better representation of all age groups. In particular, the age of volunteers ranges from 15 to 70 years old.

Since most of the time people navigate the web in uncontrolled environments, we chose to respect this principle also during data collection. Our data gathering activity allowed volunteers to stay in a comfortable position and made them more willing to participate to the test. This also granted us to collect more natural data, compared to a "laboratory" situation, like the one used in FiWI. In fact, we argue that a controlled environment can cause a different user behavior. We employed a laptop connected to an eyetracker fixed to the base of the display and placed on a horizontal plane (e.g., a desk or a table) in front of the volunteer. The screen distance was variable according to the eyetracker's ability to correctly detect the eyes of the volunteer. After explaining the task to the volunteer, we carried out a quick calibration of the device, assuring a high quality of gathered data. Then, we started the data collection procedure, with an estimated duration of about 3 minutes. Each image appeared on the volunteer's screen for 4 seconds. Images were interspersed with a black screen with a central white dot to allow the volunteer to rest her/his eyes. When ready, she/he could press the space bar to continue with the next image. In total, every volunteer observed 30 web page layouts extracted from the image dataset. Our algorithm selected the images to display in such a way as to keep balanced the number of volunteers who observed each page.

In our dataset, each path consists of a sequence of fixation points, associated with a volunteer and an observed image. For each captured fixation point, the x and y coordinates, along with the timestamp it was observed, were recorded. Each of the 262 images was seen by 11 or 12 volunteers. Since each volunteer observed 30 images, our dataset stores a total of 3000 gaze paths. If compared with the FiWI dataset, the number of available paths is more than twice, providing us with a solid base for training the path prediction model. In Table 3, we report several information allowing a comparison between our dataset and FiWI.

	FiWI [37]	*Our dataset*
Number of subjects	11 (4 males, 7 females)	100 (50 males, 50 females)
Age range of subjects	21 - 25	15 - 70
Number of web pages	149	262
Time necessary to display a web page	5 seconds	5 seconds
Screen resolution	1360×768	1920×1080
Number of gaze paths	1,639	3,000

Table 3: Comparison of several characteristics of FiWI and our dataset

4.3 Experiment Results

4.3.1 Saliency map prediction

As for the saliency map prediction, we adopted several metrics, which have been largely employed in the past literature. They are:

- *Normalized Scanpath Saliency* (hereafter, NSS) [32]; it ranges in the real interval $[0, +\infty)$.
- *AUC-Judd* [34]; it ranges in the real interval $[0, 1]$.
- *AUC-Borji* [4]; it ranges in the real interval $[0, 1]$.
- *Pearson Correlation Coefficient* (hereafter, CC) [29]; it ranges in the real interval $[-1, 1]$.
- *Kullback-Leibler divergence* (hereafter, KL) [11]; it ranges in the real interval $[0, +\infty)$.

For the first four metrics, the higher their value, the better the quality of the approach into evaluation. Instead, as for KL, the lower its value, the better the approximation of the ground truth by the saliency map.

We compared the different SalGAN variants we have proposed in Section 3.1 to verify if at least one of them provided better results than the original SalGAN. In particular, we evaluated four SalGAN models. The first (hereafter, Reference) is the original SalGAN using pre-trained weights on natural images. The second (hereafter, FineTuned) is the SalGAN that we fine-tuned by ourselves. The third (hereafter, KeepTrain) is the SalGAN that we kept trained, using our dataset, without any fine-tuning. The fourth (hereafter, FromScratch) is the SalGAN that we completely re-trained with our dataset. In Table 4, we show the metric values for the four models. We also report the values of the same metrics for the original TSGAN.

	NSS	*AUC-Judd*	*AUC-Borji*	*CC*	*KL*
TSGAN	1.43	0.82	0.76	0.66	0.63
Reference SalGAN	1.25	0.80	0.76	0.56	0.90
FineTuned SalGAN	1.61	0.85	0.82	0.74	0.52
KeepTrain SalGAN	1.58	0.84	0.83	0.73	0.52
FromScratch SalGAN	1.49	0.83	0.80	0.68	0.65

Table 4: Values of the adopted evaluation metrics obtained for the original SalGAN, the three variants of this network proposed in this paper and TSGAN

This table highlights that the worst performing model is the original SalGAN. This can be easily explained considering that, as SalGAN was previously trained only on natural images, the domain change causes a significant performance drop. Instead, our three SalGAN variants prove to have a great ability to predict saliency maps. All of them have similar or higher metric values than TSGAN. Overall, we observe a superiority of the model that has undergone fine-tuning. For some metrics, the performance is also superior to the one achieved by SalGAN for natural images in the MIT300 dataset

[31]. We also observe that both the models previously trained on natural images benefit a lot in terms of performance. In fact, websites are often very rich of natural images; this fact give both FineTuned and KeepTrain a big advantage.

To better evaluate the characteristics of the three variants of SalGAN proposed in this paper, we decided to analyze their behavior for the different website classes composing our dataset. More specifically, in Section 4.2, we saw that our dataset consists of 262 images grouped into four classes, namely *Pictorial* (58 images), *Text* (65 images), *Mixed* (76 images) and *Business* (63 images).

The results of the saliency map prediction metrics obtained by our three variants of SalGAN for the four website classes are shown in Table 5.

		NSS	*AUC-Judd*	*AUC-Borji*	*CC*	*KL*
FineTuned Salgan	Overall	1.61	0.85	0.82	0.74	0.52
	Pictorial	1.72	0.91	0.89	0.85	0.41
	Text	1.51	0.75	0.70	0.64	0.62
	Mixed	1.63	0.88	0.88	0.76	0.51
	Business	1.59	0.87	0.83	0.71	0.53
KeepTrain Salgan	Overall	1.58	0.84	0.83	0.73	0.52
	Pictorial	1.50	0.77	0.74	0.62	0.61
	Text	1.67	0.90	0.87	0.83	0.43
	Mixed	1.56	0.85	0.84	0.71	0.52
	Business	1.59	0.86	0.85	0.76	0.52
FromScratch Salgan	Overall	1.49	0.83	0.80	0.68	0.65
	Pictorial	1.58	0.88	0.87	0.79	0.61
	Text	1.43	0.75	0.68	0.58	0.68
	Mixed	1.50	0.85	0.86	0.70	0.67
	Business	1.45	0.84	0.81	0.65	0.64

Table 5: Values of the adopted evaluation metrics obtained by the three variants of SalGAN for the four website classes considered in this paper

The analysis of this table allows us to make much more refined considerations than those made analyzing Table 4. In fact, we can observe that FineTuned has a much better performance than KeepTrain for pictorial websites, while KeepTrain has a much better performance than FineTuned for text websites. As for mixed websites, FineTuned shows a slight better performance than KeepTrain, while the results are inverted in the case of business websites. As expected, FromScratch shows a lower performance than FineTuned in all classes and a lower performance than KeepTrain in all classes except for pictorial websistes.

The results in this table can be explained taking into account that, during the training phase, FineTuned freezes the first four convolutional layers of the generator and discriminator. The reason for this choice is that the first levels of the generator and discriminator are dedicated to the extraction of features from images, while the last levels of them are devoted to image classification. Because of this, freezing the first part of the generator and discriminator preserves the ability of the original SalGAN to extract features from natural images. This is important since SalGAN is very accurate on natural images, having been trained on a large set of such images. Instead, the last layers of the generator and discriminator are trained on website layouts in such a way that both of them can learn to

handle this kind of image. In fact, as we said in the previous sections, website layouts have substantial differences from natural images because of the presence of texts and advertisements, the coexistence of different pictures in the same page, and so on. If we had frozen all the layers of the generator and discriminator, we would not have any training on this kind of image. Our choice preserves the positive aspects of the original SalGAN while extending these features to other contexts on which it had not been trained. As a consequence of this, we can see from Table 5 that FineTuned has a greater ability to discriminate natural images than KeepTrain, and in this it is similar to FromScratch. On the other hand, all the other features included in the realization of FineTuned make it perform better than FromScratch. In the case of text websites, the best behavior is obtained by KeepTrain. This can be explained taking into account that the whole training of KeepTrain is done on websites. Furthermore, in it, there is no freezing of the convolutional levels that could allow it to maintain the training on natural images, at least partially. As a consequence, KeepTrain is especially trained to handle those pages that least of all contain natural images, i.e., text websites.

After this in-depth analysis of the performance of FineTuned, KeepTrain and FromScratch on the website domain, we can state that FineTuned has the best performance. From the analysis of Table 5, we can deduce that the values of the performance metrics obtained by FineTuned are higher than those of KeepTrain. This result must be coupled with the examination of the different types of website layouts on which the two approaches show the best performance. In fact, KeepTrain performs better than FineTuned on text websites, which are not very popular currently because they are text-heavy web pages (like Wikipedia). Instead, FineTuned performs better than KeepTrain in all the other web page layouts, which currently characterize the vast majority of web pages found online because they contain two or more images at a time, possibly along with some text.

The differences between FineTuned and KeepTrain in terms of performance can be explained by taking into account the training procedures of the two approaches. In fact, as seen above, FineTuned freezes the first layers of the generator and discriminator while KeepTrain continuously updates the weights of all layers. In this way, KeepTrain loses some of its ability to extract features from natural images and, at the same time, overfits to the web page domain only.

This last consideration, the previous one about the current level of popularity of the various types of web pages, as well as the results shown in Table 5, lead us to conclude that FineTuned is preferred to KeepTrain as the best saliency map construction approach for websites.

In Table 6, we report the results obtained by FineTuned SalGAN, TSGAN and some of the state-of-the-art saliency map prediction approaches, as reported by the authors of [25], when they are applied on the FiWI dataset. As we can see from this table, our fine-tuned variant of SalGAN returns better results than all the other approaches for all the metrics considered.

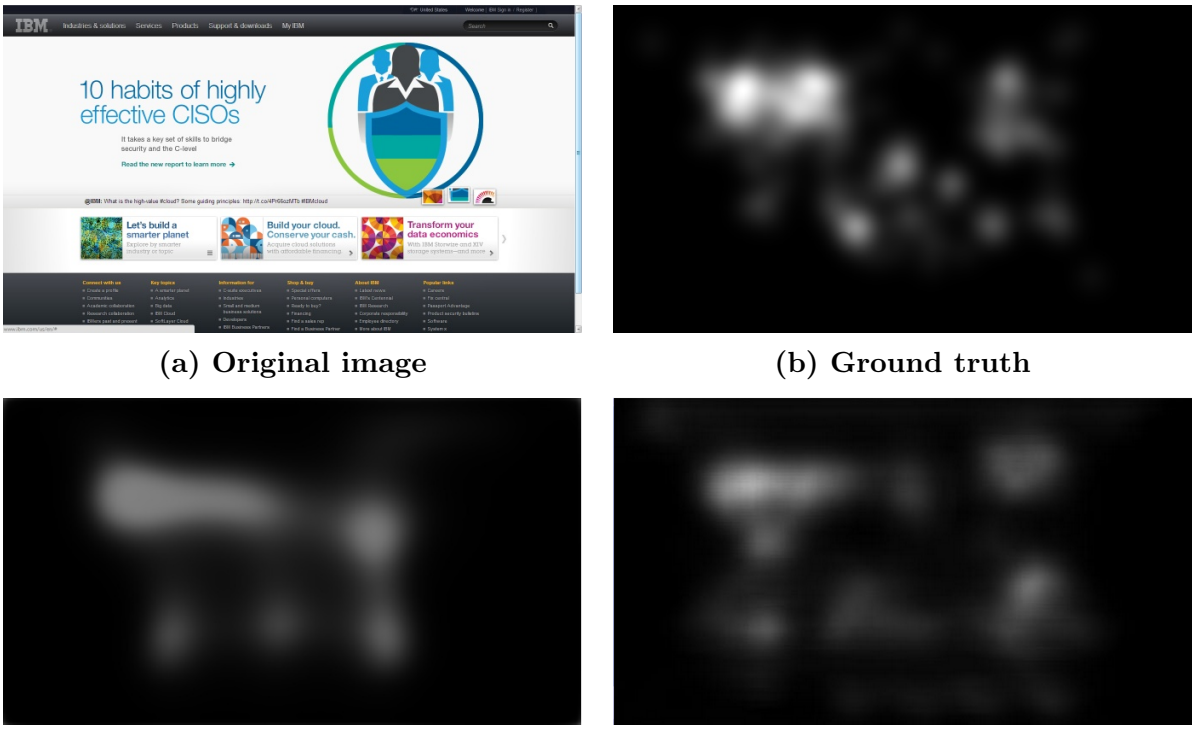
We end this section with a qualitative evaluation of the approaches into consideration. In particular, Figure 12 reports a representation of how the fine-tuned variant of SalGAN, on one hand, and TSGAN, on the other hand, behave on a layout rich of images, where text is not predominant. From this figure, we can see that our fine-tuned variant of SalGAN is actually performing slightly worse than TSGAN. Indeed, salient areas are less detailed, highlighting wider portions of the image. A less specific prediction introduces many false positives. Instead, TSGAN achieves a better performance than our SalGAN variant, as it minimizes the defects found in this last approach.

From the examination of this figure, we might think that TSGAN performs better than our SalGAN

Model	*NSS*	*AUC-Judd*	*CC*
TSGAN Reference	1.43	0.82	0.66
FineTuned SalGAN	1.61	0.85	0.74
Li et. al [25]	0.91	0.73	0.44
Shen and Zhao [37]	0.88	0.72	0.43
Garcia-Diaz et al [12]	0.82	0.68	0.41

Table 6: Values of the adopted evaluation metrics obtained for our fine-tuned variant of SalGAN and some other saliency map prediction approaches proposed in the past literature

variant. Actually, this is not the case. In fact, the situation changes dramatically in presence of layouts with rich textual information and images. We identify this configuration as a weak point of TSGAN.



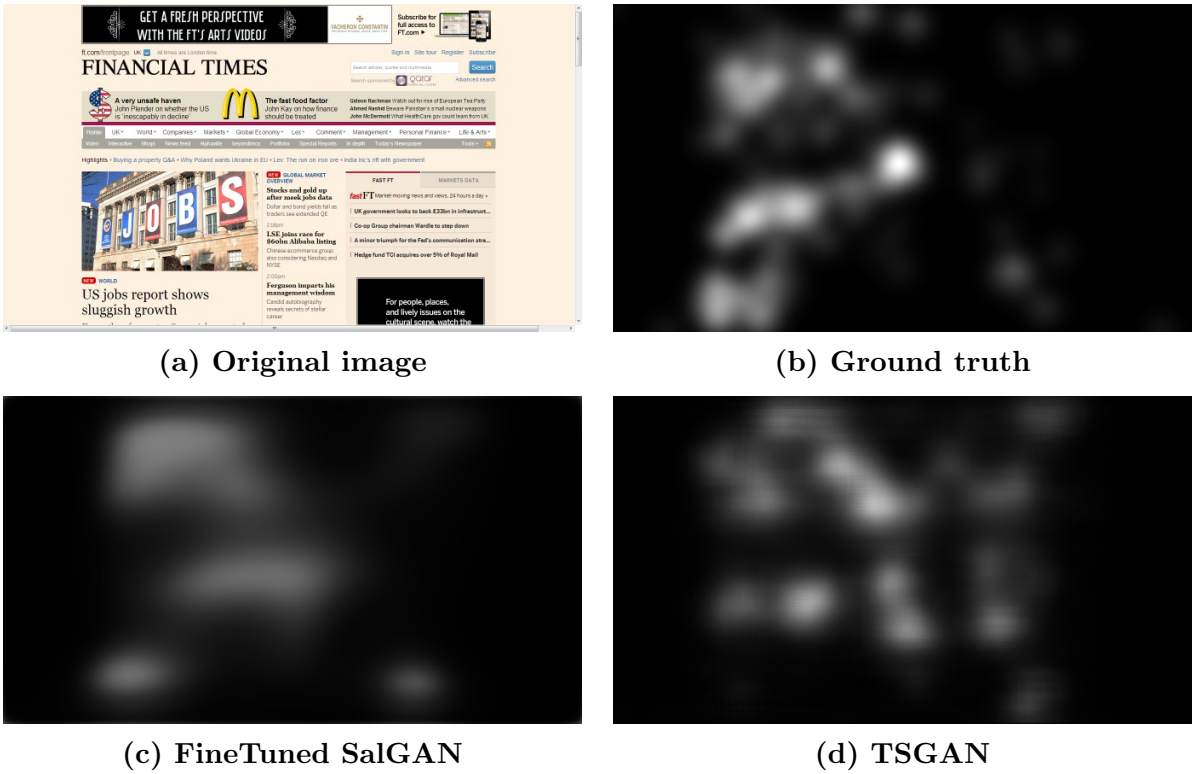
(a) Original image **(b) Ground truth**

(c) FineTuned SalGAN **(d) TSGAN**

Figure 12: Comparison of the predictions returned by our fine-tuned variant of SalGAN and TSGAN on a layout rich of images

Figure 13 shows how our fine-tuned variant of SalGAN is able to return a much better prediction in this case. In fact, TSGAN fails to identify the salient parts and produces an almost completely wrong map. If we compare the metric values computed on this image, we obtain that our SalGAN variant performs much better than TSGAN. The metrics that most highlight this difference are *NSS* and *AUC-Borji*; here, TSGAN scores 1.02 and 0.72, respectively, while SalGAN scores 1.25 and 0.81.

The difference in the values of *AUC-Borji* confirms once again how TSGAN struggles to find the salient areas, introducing many false positives.



(a) Original image **(b) Ground truth**

(c) FineTuned SalGAN **(d) TSGAN**

Figure 13: Comparison of the predictions returned by our fine-tuned variant of SalGAN and TSGAN on a layout dense of images and texts

Actually, it is possible to show that our variant of SalGAN performs generally better than TSGAN on a wider variety of layouts. TSGAN suffers when working with pages rich of information, where every single element could be a highlight. On the other hand, SalGAN is generally not able to provide too detailed information about salient areas, merely identifying large areas that are equally likely. TSGAN generally proves to be better in all those layouts where there is an information scattering, ensuring better detail. As SalGAN is already trained on a large dataset, it is able to generalize better than TSGAN.

All the previous reasonings allow us to conclude that our fine-tuned variant of SalGAN is the preferred model in most situations requiring the saliency map prediction on web pages.

4.3.2 Gaze path prediction

As for gaze path prediction, first of all we decided to verify if the original PathGAN, which was explicitly conceived to operate on natural images, showed an acceptable performance on web pages, in order to compare our variants for gaze path prediction (i.e., NormalGAN and WGAN, described in Section 3.2) with it.

We recall that NormalGAN uses a content loss weight equal to 1.0, whereas WGAN sets the same parameter to 0.05. The adversarial loss of NormalGAN is set to 0.2, while the one of WGAN is set to 1. Each variant advantages one term of the objective function over the other. This is obtained thanks to the different combination of weight updates between generator and discriminator, which requires a different parameter tuning.

We performed both a quantitative and a qualitative comparison of PathGAN and our two variants. In order to carry out their quantitative evaluation, we leveraged Jarodzka's metrics [20], which define scanpaths as a series of geometric vectors (also called saccade vectors) and compare them across the following dimensions:

- *Vector shape*: it denotes the difference in shape between saccade vectors.
- *Vector direction*: it indicates the difference in direction (i.e., angle) between saccade vectors.
- *Vector length*: it represents the difference in amplitude between saccade vectors.
- *Vector position*: it denotes the distance between fixations.
- *Fixation duration*: it indicates the difference in duration between fixations.

All the measures above range in the real interval $[0, 1]$. In fact, the first three measures are normalized by the screen diagonal, vector direction is normalized by π, whereas each fixation duration is normalized against the maximum value of the two durations being compared. The reasoning underlying these measures is the same: the higher the value, the closer saccade vectors.

Recall that each web page in our dataset was observed by 11 or 12 different users (see Section 4.2). As a consequence, given a web page, there is no single truth, but every path corresponding to a user who observed it was considered as a ground truth. Based on this choice, the prediction returned by the approach into evaluation was compared with each ground truth and, then, the average performance was computed. Finally, the performances associated with every image were averaged to obtain the evaluation of the approach on the whole dataset. In Table 7, we report the results returned by the original PathGAN, NormalGAN and WGAN, when no threshold was set on the fixation duration.

	Shape	*Direction*	*Length*	*Position*	*Duration*
Original PathGAN	0.652	0.421	0.850	0.435	0.295
NormalGAN	0.992	0.693	0.991	0.836	0.290
WGAN	0.993	0.699	0.992	0.840	0.310

Table 7: Performance of the original PathGAN, NormalGAN and WGAN when no threshold was set on the duration of fixations

This figure shows that, in the web page domain, the original PathGAN achieves much lower results in all benchmark metrics than NormalGAN and WGAN. As we know, this is due to the fact that the website domain is complex because it can contain several natural images simultaneously, along with text. This makes the direct application of PathGAN (designed for only one natural image at a time)

not effective. Looking at NormalGAN and WGAN, it is possible to conclude that the vector shape and the path length are predicted very well by both approaches. In fact, the corresponding values are very high for both variants. The position of fixations is also high, compared to ground truths. On the other hand, direction similarity decreases significantly, even if it remains within an acceptable range. Both variants struggle to determine the duration of each fixation. Duration is by far the metric with the worst performance for both approaches, highlighting a common weakness of them. The values in Table 7 also say that WGAN is the best performing model. Indeed, it achieves the best score in all the five metrics. NormalGAN also performs well, being behind WGAN for just some decimal points in every metric. Table 7 also highlights that the two approaches have the same strengths and weaknesses because they perform well and poorly in the same metrics.

We performed a second quantitative evaluation by setting a threshold on the duration of fixations with the goal of improving results. The idea motivating this attempt was to eliminate the dummy fixations in the prediction generated by the network to return a path 63 fixations long, which is the same output length of the original PathGAN. We set a threshold on the duration of fixations equal to 0.0027; this means that all fixations with shorter duration were not considered as such. Since duration is normalized between 0 and 1, and the predicted path has a total duration of 4 seconds, we can compute the corresponding threshold expressed in seconds. In particular, a threshold of 0.0027 corresponds to about 10 milliseconds, i.e., one order of magnitude smaller than the average fixation duration [5]. This threshold has been conceived in such a way as to avoid losing important fixations in the final prediction. In Table 8, we show the results obtained.

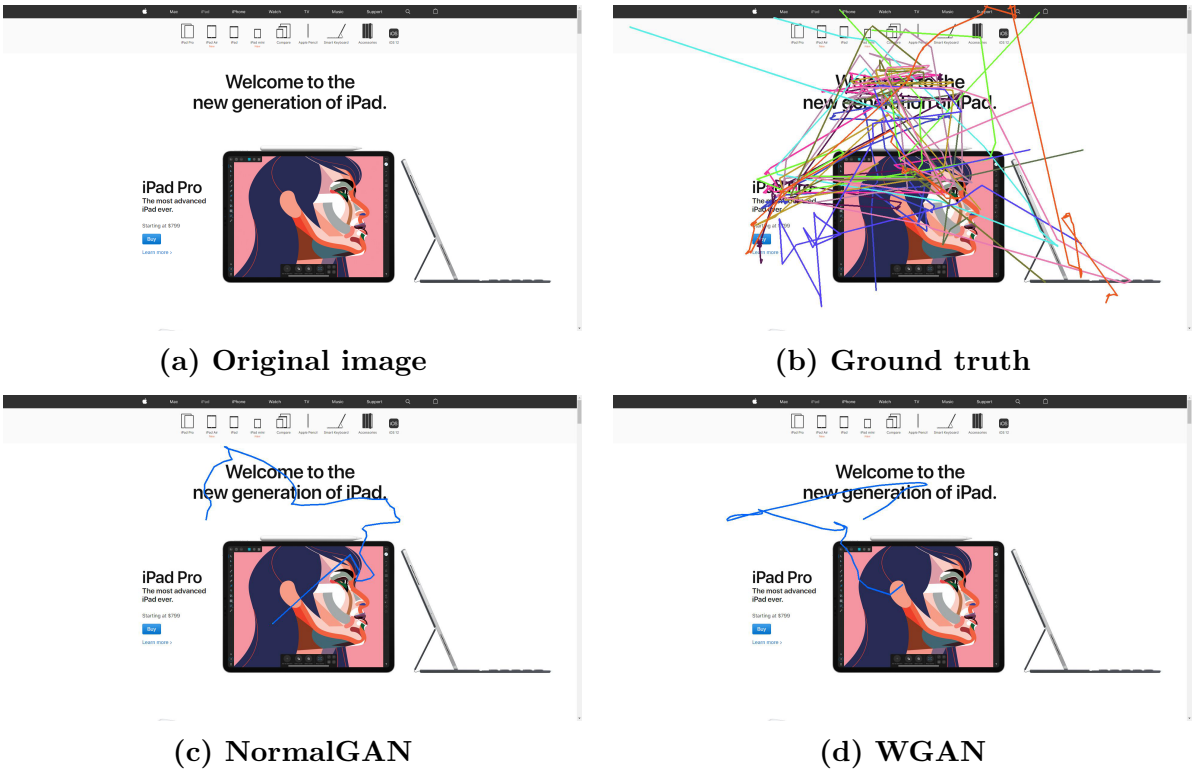
	Shape	*Direction*	*Length*	*Position*	*Duration*
Original PathGAN	0.645	0.424	0.852	0.437	0.310
NormalGAN	0.992	0.699	0.991	0.838	0.308
WGAN	0.993	0.698	0.992	0.840	0.326

Table 8: Performance of the original PathGAN, NormalGAN and WGAN when a threshold equal to 0.0027 has been set on the duration of fixations

Similarly to Table 7, this table shows that the performance of the original PathGAN is much lower than that of NormalGAN and WGAN. This represents a further confirmation that the original PathGAN is not adequate to predict the gaze path of a user while she is surfing web pages. NormalGAN and WGAN represent two ways to increase its effectiveness. Clearly, we are not saying that these variants are the only ways to obtain this result. However, we can certainly say that they result in significant improvements over the original PathGAN whatever evaluation metrics are considered. As far as NormalGAN and WGAN are concerned, we can see that adding a threshold on the duration of fixations introduces only a slight improvement of results. In fact, the performance value regarding duration remains modest, even if a very small improvement is visible. Once again, WGAN confirms as the best approach, even if NormalGAN manages to shorten the distance from WGAN in most metrics.

After the quantitative evaluation, we proceeded with the qualitative one. In Figure 14, we report an example of prediction provided by NormalGAN and WGAN models.

From the analysis of this figure, we can easily observe that WGAN performs better in this case.



(a) Original image **(b) Ground truth**

(c) NormalGAN **(d) WGAN**

Figure 14: Comparison of the predictions returned by NormalGAN and WGAN on one of the web pages of our dataset

This qualitative conclusion can be drawn considering that:

- Most of the gaze paths of the ground truth pass through the left part of the area between the text and the image. This is because the eye is also caught by the text placed on the image left. The gaze path predicted by WGAN crosses the area between the text and the image in the same way, going to the left. The gaze path predicted by NormalGAN crosses the area between the text and the image from the right side. Therefore, it does not take into account all the gazes that, on the left, are captured by the text close to the figure.
- Most of the gaze paths of the ground truth cross the screen going overall from left to right. The gaze path of WGAN behaves in the same way. Instead, the gaze path of NormalGAN goes immediately from left to right and then back to left making almost a clockwise rotation. This behavior is not found in almost any of the gaze paths of the ground truth.

The metric values computed on this web page confirm again our qualitative conclusions. The direction similarity in WGAN is higher than in NormalGAN, with a score of 0.74 against 0.70. The same trend between WGAN and NormalGAN can be also observed for all the other metrics. The values of position similarity are very good; it reaches 0.87 in both cases. Also in this experiment, duration does not return satisfactory values. Overall, both approaches show the same strengths and weaknesses.

In every image analyzed we found analogous trends and results. Therefore, we can conclude that WGAN behaves generally better than NormalGAN. This probably happens because Wasserstein training present in WGAN improves the final results in almost every aspect.

After having determined that WGAN is the best gaze path prediction approach for web pages, we must now verify if its absolute performance is anyway acceptable. In fact, it could happen that WGAN, although better than NormalGAN, has very low (and, therefore, unacceptable) performances. Unfortunately, past literature lacks GAN-based approaches to predict gaze paths on websites.

To do this verification, we used an approach that considers humans, their behavior and their evaluation. In particular, we leveraged the well-known One human baseline technique [2]. This technique, given N observers, tells us how well a fixation map of one of them, represented as a saliency map, predicts the fixation of the other $N-1$ observers. This verification task is performed for each of the N observers, and the results thus obtained are averaged. In this way, in turn, each individual is used to predict the behavior of all the others. In order to make an as accurate and complete as possible evaluation, which takes into account not only the average behavior of observers, but also the full range of their possible behaviors, we decided to specify more values for the prediction scores associated with humans. In particular, we considered the maximum, minimum and mean values.

In Table 9, we report the values of the evaluation metrics for One human baseline and WGAN. Since, with One human baseline, the evaluation of the gaze paths is transformed into an evaluation of the corresponding saliency map (see [2] for all details), the metrics we use are those related to saliency map prediction.

	NSS	*AUC-Judd*	*AUC-Borji*	*CC*	*KL*
One human baseline	Min: 0.55 Mean: 0.99 Max: 1.61	Min: 0.20 Mean: 0.22 Max: 0.26	Min: 0.49 Mean: 0.51 Max: 0.54	Min: 0.32 Mean: 0.44 Max: 0.59	Min: 4.29 Mean: 6.06 Max: 7.88
WGAN	0.71	0.66	0.61	0.34	7.67

Table 9: Comparison between One human baseline and WGAN

Table 9 shows that WGAN is able to achieve satisfactory results. For example, consider the *AUC-Judd* and *AUC-Borji* metrics. For them, the mean value reached through One human baseline is 0.22 and 0.51, respectively; instead, WGAN reaches 0.66 and 0.61, respectively. Interestingly, for these two metrics, WGAN achieves an even better performance than the maximum values reached by One human baseline.

On the other hand, WGAN performs below the mean, but still above the minimum for *KL*, *NSS* and *CC*. This means that the prediction of the length and duration of the gaze path made by WGAN is quite different from the values of the ground truth, even if predicted values are still acceptable.

A final contribution on this evaluation process is obtained by considering the qualitative evaluation of WGAN compared to One human baseline. In this case, the gaze path generated by WGAN has a shape close to ground truth. This clearly represents a very encouraging result for the research we have described.

5 Conclusion

In this paper, we have proposed one fine-tuned variant of SalGAN and two fine-tuned variants of PathGAN conceived for extending saliency map and gaze path prediction from natural images to websites. First of all, we have seen the motivations underlying our work and, in particular, why this is an important issue to address from both theoretical and application perspectives. Then, we have examined related literature and pointed out the small number of approaches for the evaluation of visual attention on website layouts. Afterwards, we have proposed our variants and described the underlying architecture. Next, we have presented our dataset, which is specifically built for the web domain. After this, we have shown our experiments, carried out for saliency map and gaze path predictions, and we have compared our variants to the other already existing approaches. Finally, we have presented our prototype that implements all the functionalities discussed in this paper and allows the designer to access them easily.

Our work should not be considered as an ending point. Indeed, several further improvements can be designed to boost the results of this research. For instance, it could be possible to fuse both saliency map and gaze path prediction and create a unique pipeline. In this way, we could exploit the saliency map prediction to generate the corresponding visual scanpath that, we argue, could be more accurate. Finally, it would be also interesting to evaluate the possibility of applying reinforcement learning in this scenario. Here, the challenge would involve the definition of a reward function able to highlight the correct aspects of web interfaces and, therefore, to ensure an appropriate training to the model.

Acknowledgments

The authors thank the Editor and the anonymous Reviewers whose competent, relevant, thorough and constructive suggestions enabled them to significantly improve the quality of this paper.

This work was partially funded by the Department of Information Engineering at the Polytechnic University of Marche under the project "A network-based approach to uniformly extract knowledge and support decision making in heterogeneous application contexts" (RSAB 2018), and by the Marche Region under the project "Human Digital Flexible Factory of the Future Laboratory (HDSFIab) - POR MARCHE FESR 2014-2020 - CUP B16H18000050007". The authors thank the Daimler AG for supporting them in all their testing activities by providing its laboratories and encouraging its employees to participate as volunteers.

References

[1] M. Assens, X. Giro i Nieto, K. McGuinness, and N.E. O'Connor. PathGAN: visual scanpath prediction with generative adversarial networks. In *Proc. of the European Conference on Computer Vision (ECCV'18)*, pages 406–422, Munich, Germany, 2018.

[2] A. Borji. Saliency prediction in the deep learning era: Successes and limitations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. IEEE.

[3] A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):185–207, 2012. IEEE.

[4] A. Borji, H.R. Tavakoli, D.N. Sihite, and L. Itti. Analysis of scores, datasets, and models in visual saliency prediction. In *Proc. of the International Conference on Computer Vision (ICCV'13)*, pages 921–928, Sidney, Australia, 2013. IEEE.

[5] M.S. Castelhano and K. Rayner. Eye movements during reading, visual search, and scene perception: An overview. *Cognitive and cultural influences on eye movements*, 2175:3–33, 2008.

[6] Z. Chen and W. Sun. Scanpath Prediction for Visual Attention using IOR-ROI LSTM. In *Proc. of the International Joint Conferences on Artificial Intelligence (IJCAI'18)*, pages 642–648, Stockholm, Sweden, 2018.

[7] S. Cirillo, D. Desiato, and B. Breve. Chravat-chronology awareness visual analytic tool. In *Proc. of the International Conference Information Visualisation (IV'19)*, pages 255–260, Paris, France, 2019. IEEE.

[8] A. Coutrot, J.H. Hsiao, and A.B. Chan. Scanpath modeling and classification with hidden Markov Models. *Behavior research methods*, 50(1):362–379, 2018. Springer.

[9] G. Douzas and F. Bacao. Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Systems with applications*, 91:464–471, 2018. Elsevier.

[10] S. Eraslan, Y. Yesilada, and S. Harper. Scanpath trend analysis on web pages: Clustering eye tracking scanpaths. *ACM Transactions on the Web*, 10(4):1–35, 2016.

[11] T. Van Erven and P. Harremos. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014. IEEE.

[12] A. Garcia-Diaz, V. Leboran, X.R. Fdez-Vidal, and X.M. Pardo. On the relationship between optical variability, visual saliency, and eye fixations: A computational approach. *Journal of vision*, 12(6):17–17, 2012. The Association for Research in Vision and Ophthalmology.

[13] J.H. Goldberg and J.I. Helfman. Visual scanpath representation. In *Proc. of the Symposium on Eye-Tracking Research & Applications (ETMA'10)*, pages 203–210, Austin, Texas, USA, 2010. ACM.

[14] Y. Gu, J. Chang, Y. Zhang, and Y. Wang. An element sensitive saliency model with position prior learning for web pages. In *Proc. of the International Conference on Innovation in Artificial Intelligence (ICIAI'19)*, pages 157–161, London, England, 2019.

[15] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved Training of Wasserstein GANs, 2017.

[16] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Proc. of the International Conference on Neural Information Processing Systems (NIPS'07)*, pages 545–552, Cambridge, MA, USA, 2007. MIT Press.

[17] R. He, X. Li, G. Chen, G. Chen, and Y. Liu. Generative adversarial network-based semi-supervised learning for real-time risk warning of process industries. *Expert Systems with Applications*, 150:113244, 2020. Elsevier.

[18] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. MIT Press.

[19] A. Jana and S. Bhattacharya. Design and validation of an attention model of web page users. *Advances in Human-Computer Interaction*, 2015:1–14, 2015. Hindawi.

[20] H. Jarodzka, K. Holmqvist, and M. Nyström. A vector-based, multidimensional scanpath similarity measure. In *Proc. of the Symposium on Eye Tracking Research & Applications (ETRA'10)*, pages 211–218, Austin, TX, USA, 2010. ACM.

[21] M. Jiang, X. Boix, G. Roig, J. Xu, L. Van Gool, and Q. Zhao. Learning to predict sequences of human visual fixations. *IEEE Transactions on Neural Networks and Learning Systems*, 27(6):1241–1252, 2016. IEEE.

[22] S. Josephson and M.E. Holmes. Visual attention to repeated internet images: testing the scanpath theory on the world wide web. In *Proc. of the Symposium on Eye tracking Research & Applications (ETMA'02)*, pages 43–49, New Orleans, LA, USA, 2002. ACM.

[23] M.A. Just and P.A. Carpenter. Eye fixations and cognitive processes. *Cognitive psychology*, 8(4):441–480, 1976. Elsevier.

[24] M. Kummerer, T.S. Wallis, L.A. Gatys, and M. Bethge. Understanding low-and high-level contributions to fixation prediction. In *Proc. of the IEEE International Conference on Computer Vision (ICCV'17)*, pages 4789–4798, Venezia, Italy, 2017.

[25] J. Li, L. Su, B. Wu, J. Pang, C. Wang, Z. Wu, and Q. Huang. Webpage saliency prediction with multi-features fusion. In *Proc. of the IEEE International Conference on Image Processing (ICIP'16)*, pages 674–678, Phoenix, Arizona, USA, 2016. IEEE.

[26] Y. Li and Y. Zhang. Webpage Saliency Prediction with Two-Stage Generative Adversarial Networks. *arXiv preprint arXiv:1805.11374*, 2018.

[27] H. Liu, D. Xu, Q. Huang, W. Li, M. Xu, and S. Lin. Semantically-based human scanpath estimation with HMMs. In *Proc. of the IEEE International Conference on Computer Vision (ICCV'13)*, pages 3232–3239, Sydney, NSW, Australia, 2013.

[28] N. Liu and J. Han. A deep spatial contextual long-term recurrent convolutional network for saliency detection. *IEEE Transactions on Image Processing*, 27(7):3264–3274, 2018. IEEE.

[29] O. Le Meur and T. Baccino. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior research methods*, 45(1):251–266, 2013. Springer.

[30] J.H. Oh, J.Y. Hong, and J.G. Baek. Oversampling method using outlier detectable generative adversarial network. *Expert Systems with Applications*, 133:1–8, 2019. Elsevier.

[31] J. Pan, C. Canton Ferrer, K. McGuinness, N.E. O'Connor, J. Torres, E. Sayrol, and X. Giro i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1701.01081*, 2017.

[32] R.J. Peters, A. Iyer, L. Itti, and C. Koch. Components of bottom-up gaze allocation in natural images. *Vision research*, 45(18):2397–2416, 2005. Elsevier.

[33] M. Assens Rein, X. Giro i Nieto, K. McGuinness, and N.E. O'Connor. Saltinet: Scan-path prediction on 360 degree images using saliency volumes. In *Proc. of the IEEE International Conference on Computer Vision Workshops (ICCV'17)*, pages 2331–2338, Venezia, Italy, 2017. IEEE.

[34] N. Riche, M. Duvinage, M. Mancas, B. Gosselin, and T. Dutoit. Saliency and human fixations: State-of-the-art and study of comparison metrics. In *Proc. of the IEEE International Conference on Computer Vision (ICCV'13)*, pages 1153–1160, Sidney, Australia, 2013. IEEE.

[35] H.J. Seo and P. Milanfar. Static and space-time visual saliency detection by self-resemblance. *Journal of vision*, 9(12):15–15, 2009. The Association for Research in Vision and Ophthalmology.

[36] C. Shen, X. Huang, and Q. Zhao. Predicting eye fixations on webpage with an ensemble of early features and high-level representations from deep network. *IEEE Transactions on Multimedia*, 17(11):2084–2093, 2015. IEEE.

[37] C. Shen and Q. Zhao. Webpage saliency. In *Proc. of the European Conference on Computer Vision (ECCV'14)*, pages 33–46, Zurich, Switzerland, 2014. Springer.

[38] D. Simon, S. Sridharan, S. Sah, R. Ptucha, C. Kanan, and R. Bailey. Automatic scanpath generation with deep recurrent neural networks. In *Proc. of the Symposium on Applied Perception (SAP'16)*, pages 130–130, Anaheim, USA, 2016.

[39] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *Proc. of the International Conference on Learning Representations (ICLR'14)*, 2013. ICLR Press.

[40] V.K. Singh, H.A. Rashwan, S. Romani, F. Akram, N. Pandey, M.M.K. Sarker, A. Saleh, M. Arenas, M. Arquez, and D. Puig. Breast tumor segmentation and shape classification in mammograms using generative adversarial and convolutional neural network. *Expert Systems with Applications*, 139:112855, 2020. Elsevier.

[41] J.D. Still. Web page attentional priority model. *Cognition, Technology & Work*, 19(2-3):363–374, 2017. Springer.

[42] J.D. Still and C.M. Masciocchi. A saliency model predicts fixations in web interfaces. In *Proc. of the International Workshop on Model Driven Development of Advanced User Interfaces (MDDAUI'10)*, page 25, Atlanta, GA, USA, 2010.

[43] A. Verma and D. Sen. HMM-based Convolutional LSTM for Visual Scanpath Prediction. In *Proc. of the European Signal Processing Conference (EUSIPCO'19)*, pages 1–5, La Coruna, Spain, 2019. IEEE.

[44] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.L. Barabási. Human mobility, social ties, and link prediction. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*, pages 1100–1108, San Diego, California, USA, 2011. ACM.

[45] W. Wang, C. Chen, Y. Wang, T. Jiang, F. Fang, and Y. Yao. Simulating human saccadic scanpaths on natural images. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*, pages 441–448, Colorado Springs, CO, USA, 2011. IEEE.

[46] J. Yan, M. Zhu, H. Liu, and Y. Liu. Visual saliency detection via sparsity pursuit. *IEEE Signal Processing Letters*, 17(8):739–742, 2010. IEEE.

[47] J. Zhang and S. Sclaroff. Exploiting surroundedness for saliency detection: a boolean map approach. *IEEE transactions on pattern analysis and machine intelligence*, 38(5):889–902, 2015. IEEE.