

INSPIRATION HUNTER: A SOFTWARE PLATFORM TO SUPPORT IDEATION
IN PICTURE DESIGN

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GÜLCE BAL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2019

Approval of the thesis:

**INSPIRATION HUNTER: A SOFTWARE PLATFORM TO SUPPORT
IDEATION IN PICTURE DESIGN**

submitted by **GÜLCE BAL** in partial fulfillment of the requirements for the degree
of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering**

Prof. Dr. Sibel Tari
Supervisor, **Computer Engineering, METU**

Examining Committee Members:

Prof. Dr. Volkan Atalay
Computer Engineering Department, METU

Prof. Dr. Sibel Tari
Computer Engineering Department, METU

Assist. Prof. Dr. Venera Adanova
Computer Engineering Department, TED University

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Gülce Bal

Signature :

ABSTRACT

INSPIRATION HUNTER: A SOFTWARE PLATFORM TO SUPPORT IDEATION IN PICTURE DESIGN

Bal, Gülce

M.S., Department of Computer Engineering

Supervisor: Prof. Dr. Sibel Tari

September 2019, 41 pages

Visual materials are commonly used in creative design process as source of inspiration. Nowadays, many of these materials became accessible on the web by means of the image search engines. In this thesis, we developed a platform that adds image hunt and style adaptation features to Adobe Photoshop which is popular among designers as a computer aided design tool. Designers can search via a keyword or an image without changing their working environment and reuse the search results by using the facilities of Adobe Photoshop. This platform also enables to search with designer's unfinished designs, so it can be used not only in the idea generation step but also in the implementation step. An experimental setup was used to determine the success of the platform and verbal data were collected from the users.

Keywords: Creative Design, Computer-facilitated Creativity, Ideation

ÖZ

İLHAM AVCISI: RESİM TASARIMINDA FİKİR ÜRETİMİNİ DESTEKLEYEN BİR YAZILIM PLATFORMU

Bal, Gülce

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Sibel Tarı

Eylül 2019, 41 sayfa

Görsel malzemeler genellikle yaratıcı tasarım sürecinde ilham kaynağı olarak kullanılırlar. Günümüzde, bu malzemelerin birçoğu, görüntü arama motorları aracılığıyla web üzerinde erişilebilir hale geldi. Bu tezde, bilgisayar destekli tasarım aracı olarak tasarımcılar arasında popüler olan Adobe Photoshop'a görüntü avı ve stil uyarlama özellikleri ekleyen bir platform geliştirdik. Tasarımcılar, çalışma ortamlarını değiştirmeden bir anahtar kelime veya resim aracılığıyla arama yapabilir ve Adobe Photoshop'un olanaklarını kullanarak arama sonuçlarını yeniden kullanabilirler. Bu platform aynı zamanda tasarımcının tamamlanmamış tasarımları ile de arama yapabilmeyi sağlar, böylece sadece fikir üretme aşamasında değil, uygulama aşamasında da kullanılabilir. Platformun başarısını belirlemek için deneysel bir kurulum kullanılmış ve kullanıcılardan sözlü veriler toplanmıştır.

Anahtar Kelimeler: Yaratıcı Tasarım, Bilgisayar İmkanlarıyla Yaratıcılık, Fikir Üretimi

To my family

ACKNOWLEDGMENTS

I would like to thank my advisor Prof. Dr. Sibel Tari for her guidance and encouragement.

I would like to give special thanks to Caner Bozkurt, Nagihan Tuna, and Çağkan Uludağlı for their efforts during the experiments.

Last of all, I would like to thank my family, my spouse and my friends. It would not have been possible to complete this thesis without their endless support.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vi
ACKNOWLEDGMENTS	viii
TABLE OF CONTENTS	ix
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS	xiii
CHAPTERS	
1 INTRODUCTION	1
2 METHOD	5
2.1 Technologies and Tools	5
2.1.1 Adobe Photoshop	5
2.1.2 Google Images	6
2.1.3 Neural Style Transfer	6
2.1.4 Adobe Common Extensibility Platform	7
2.1.5 Representational State Transfer - REST	8
2.1.6 JavaScript Object Notation - JSON	10
2.1.7 Waffle	10
2.1.8 Jsoup - Java HTML Parser Library	10

2.1.9	Apache Tomcat	10
2.1.10	Node.js	10
2.2	Implementation Details	11
2.2.1	Image Hunt	13
2.2.2	Style Transfer	14
2.3	Performance Results	14
3	EXPERIMENTS	19
3.1	Experiment 1	19
3.2	Experiment 2	22
3.3	Experiment 3	22
3.4	Experiment 4	24
3.5	Experiment 5	25
3.6	Experiment 6	27
3.7	Experiment 7	28
4	SUMMARY AND DISCUSSION	31
	REFERENCES	35
	APPENDICES	
A	INSPIRATION HUNTER - USER INTERFACES	37

LIST OF FIGURES

FIGURES

Figure 1.1	Creativity Model - Adapted from Warr and O'Neill [1]	1
Figure 2.1	Google Images (a) Query Image (b) Related Images (c) Unrelated Images	6
Figure 2.2	Neural Style Transfer (a) Content Image (b) Style Image (c) Result	7
Figure 2.3	CEP Component Architecture	8
Figure 2.4	Communication diagram	12
Figure 2.5	Image-based Hunt Process	15
Figure 2.6	Text-based Hunt Process	16
Figure 2.7	Selected Image Style Transfer Process	17
Figure 2.8	Existing Style Transfer Process	18
Figure 3.1	Experiment 1	20
Figure 3.2	Experiment 2	21
Figure 3.3	Experiment 3	23
Figure 3.4	Experiment 4	25
Figure 3.5	Experiment 5	26
Figure 3.6	Experiment 6	27

Figure 3.7	Experiment 7	29
Figure A.1	Search by image - User Interface	38
Figure A.2	Search by image - User Interface	38
Figure A.3	Search by text - User Interface	39
Figure A.4	Search by image and text - User Interface	39
Figure A.5	Apply the style of the selected image - User Interface	40
Figure A.6	Apply the style of the selected image result - User Interface	40
Figure A.7	Apply the styles of known images - User Interface	41

LIST OF ABBREVIATIONS

API	Application Programming Interface
HTTP	Hyper-Text Transfer Protocol
JSOUP	Java HTML Parser
HTML	Hypertext Markup Language
CEP	Common Extensibility Platform
REST	Representational State Transfer
JSON	JavaScript Object Notation
PNG	Portable Network Graphics
BPMN	Business Process Model and Notation
DOM	Document Object Model
CC	Creative Cloud
CEF	Chromium Embedded Framework

CHAPTER 1

INTRODUCTION

Design process is a delicate example of the creativity of human mind. Nowadays, human computer interaction researchers engage in developing tools for supporting the creativity of designers. Since creativity could not be measured in traditional ways, it is important to model creativity for better understanding how the developed tools can aid this process [1, 2, 3]. Figure 1.1 demonstrates a sample creativity model which divides the creative process into three main stages: Problem Preparation, Idea Generation and Idea Evaluation. Problem Preparation is a stage in which the designer analyzes the problem and develops potential solutions. At the Idea Generation stage, the designer produces new ideas with the combination of old and existing ideas. The Idea Evaluation stage is to assess the suitability of ideas created in the Idea Generation stage and, if so, whether they are considered creative [1].

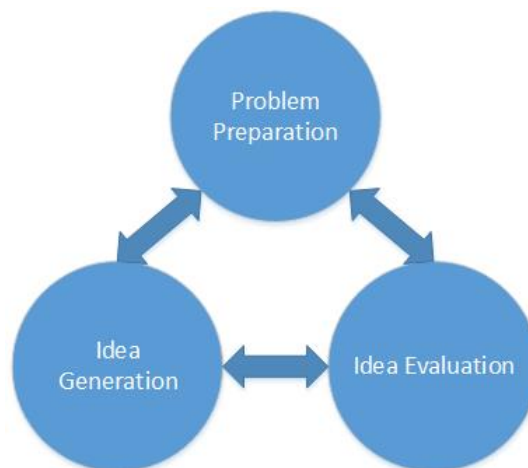


Figure 1.1: Creativity Model - Adapted from Warr and O'Neill [1]

Recent research focuses on the relationship between the provided materials to the de-

signer and their creativity. Designers are getting inspired from these provided materials through their senses and visual ones are the strongest stimulant among those. According to the designers, materials which include images such as magazines, books, and web sites are the most used sources since the design process is image-oriented rather than text-oriented. The creative design process involves exploring foreign domains, retrieving related results and incorporating them in designs. Designers like viewing examples not only from similar domains but also diverse sources efficiently support their creativity [4, 5, 6, 7, 8].

According to Herring et al. [9] designers perceive the web useful for accessing visual sources via search interfaces which make the process of finding examples easier. Image search engines provide both text-based and image-based querying interfaces. Designers can enter keywords or upload images to these engines then resulted images are shown in a table, most relevant images are shown in the first place. Herring et al. [9] argue that although these engines consider that there is one exact result for the designer and this result is the target in the designer's mind, designers often do not have any target in their mind. They also emphasized that if designers search only similar images, they can stick to the resulted examples and this increases fixation.

Although the traditional design process does not involve the implementation stage, nowadays the implementation stage is seen by design researchers as an important part of the design process [10]. Sass and Oxman [11] define the design process as situated between conceptual design and real-world construction.

Computers can support designers' creativity by providing visual materials while they proceed designing. Do and Gross [5] proposed a sketch-based visual reference software which is an integrated computer aided design tool. They argued that since CAD tools do not provide any visual references, they are lack of inspiring designers. They developed a tool which enables free-hand sketching and provides visually similar images to the designer's sketch. They also created some databases which store the provided images.

The digitalized world creates new opportunities with the rich content of the world wide web. The web includes a lot of prior work and many user interfaces are developed for accessing them easily. Mougnot et al. [12] proposed a software Adaptive

Idea web page builder for web page designers which automatically generates examples from the web. Finding useful references from the web is difficult since the web has rich content. This software provides examples of existing pages through employing an algorithm which measures usefulness by using designer's requests then from these useful examples generates web page templates.

In this thesis, we developed Inspiration Hunter which is a software platform integrated with very common computer-aided design tools such as Adobe Photoshop and Illustrator. Inspiration Hunter provides image hunt and style adaptation features to these tools. Source of the visual references is Google Images. Inspiration Hunter communicates with Google Images at the background and this process provides seamless interaction with crowd-generated big data on the web. Hence, its users can integrate the retrieved images to their designs via cut-and-paste or style adaptation operations without having to leave their content creation environment. Even though the majority still believe that creativity is a rare gift given to rare individuals, increasing number of design researchers invite everybody to design [13, 14]. Therefore, we conceive Inspiration Hunter as a platform that encourages everyday people to design by reusing existing material.

CHAPTER 2

METHOD

Inspiration Hunter provides image hunt and style adaptation features to Adobe Photoshop and Illustrator. Section 2.1 includes detailed information about the technologies and tools that we used while implementing Inspiration Hunter. Section 2.2 is about implementation details of Inspiration Hunter.

2.1 Technologies and Tools

Adobe Photoshop, Google Images, and Neural Style Transfer sections give detailed information about the tools that Inspiration Hunter is built on. Adobe Common Extensibility Platform section is about how to extend the functionality of Adobe applications. Representational State Transfer - REST, JavaScript Object Notation - JSON, Waffle sections explain the technologies which deal with communication needs. Jsoup section gives information about extracting information from web pages like Google Images. Apache Tomcat section gives brief information about the technology which is used for containing Java applications.

2.1.1 Adobe Photoshop

Adobe Photoshop is a computer aided design tool which is developed by Adobe Systems Inc. This software is a member of Adobe Creative Suite which is a software suite of graphic design, video editing and web development applications [15]. At 2011, Adobe Systems proposed Adobe Creative Cloud (CC) which is a service that allows users to access the softwares in the creative suite.

This software allows users to manipulate images by provided tools. These tools are classified into seven categories, such as selection, crop, and slice, measuring, retouching, painting, drawing and type, navigation.

2.1.2 Google Images

Google Images is a user interface which provides users to search visual references from the web. At the beginning, this interface allowed only text-based querying, but Google has employed searching via images mechanism since 2010. Google indexed over 10 billion images to feed the search engine. When users query the engine with an image first of all uploaded image is analyzed and some features such as colors, points, lines and textures are extracted. Then by using these features a search query is generated. Google’s matching algorithms return similar results and these results are ordered by relevance. Figure 2.1 shows the search results when the query image is a colored bird (a). In the search results, there are other colored bird images (b), but also unrelated images (c) which have similar color to the query image. Google’s success depends on not only the size of the indexed images but also the used computer vision algorithms [16, 17].

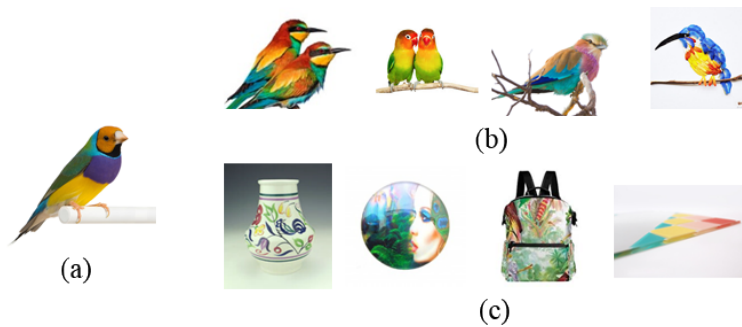


Figure 2.1: Google Images (a) Query Image (b) Related Images (c) Unrelated Images

2.1.3 Neural Style Transfer

Neural style transfer is a method introduced in Gatys et al. [18] to transferring the style from a source image to the target image. Each layer of convolutional neural networks does filtering or pooling operations to extract features of images. After these

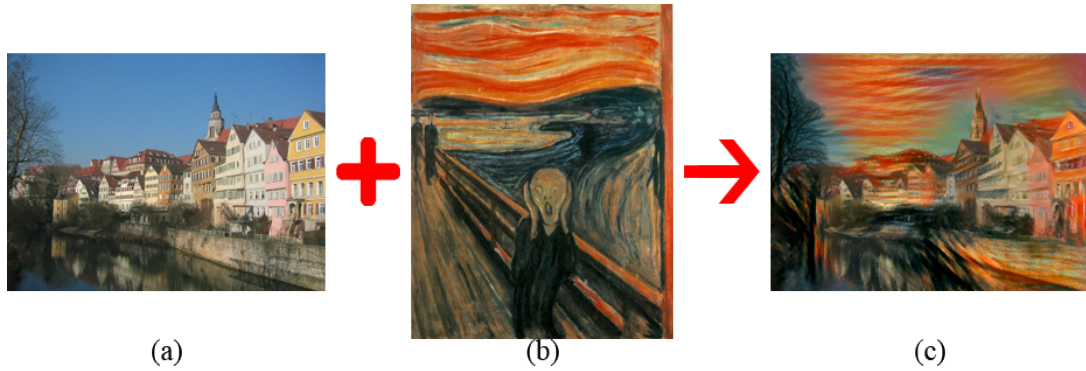


Figure 2.2: Neural Style Transfer (a) Content Image (b) Style Image (c) Result

operations, at each layer, a new representation of the input image is produced. Gatys et al. [18] reconstructed images by using those layers and they observed that while reconstructed images from higher layers reproduce the high level content such as objects and their positions, lower layers simply reproduce the exact pixel values of the original image. Hence, they used feature responses from higher layers of the network to generate content representation. For the style representation, they used correlations between the feature responses from the lower layers. Figure 2.2 demonstrates a sample neural style transfer result. In this case, (a) is the content image. After transferring the style of (b), (c) is obtained.

2.1.4 Adobe Common Extensibility Platform

Adobe Common Extensibility Platform (CEP) is used in Adobe CC desktop applications and it enables developers to extend the functionality of applications. Since CEP is a custom implementation of Google Chromium Embedded Framework (CEF), it also includes support for V8 JavaScript engine which means including a Google Chrome browser in the host application. CEP runs in CC applications and connects to the host application's scripting document object model (DOM) by using the Plug-Plug architecture which is an Adobe shared technology component.

CEP extensions include two components: HTML/JavaScript and ExtendScript. While HTML/JavaScript component is used for developing user interface, ExtendScript component is used for manipulating the host application's DOM. Figure 2.3 demon-

strates the architecture of these components and a sample communication scenario between them. In this scenario, the host application is Adobe Photoshop and the user wants to open a document and sends a request about it. HTML/JavaScript component manipulates this request then call evalScript. PlugPlug architecture sends this request to the host application's DOM. After processing the request, ExtendScript component open the new file then returns the result. PlugPlug architecture returns this result to the HTML/JavaScript component.

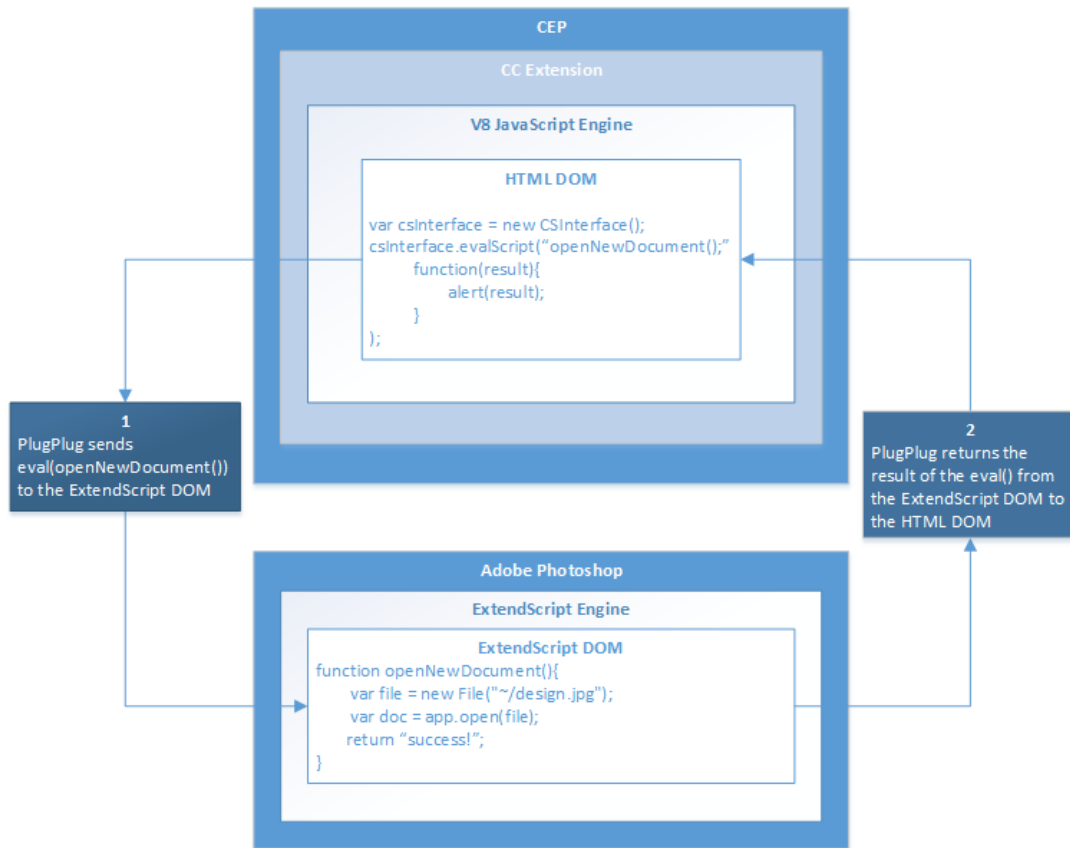


Figure 2.3: CEP Component Architecture

2.1.5 Representational State Transfer - REST

REST is an architectural style for developing web services. It is introduced in 2000 by Fielding [19]. REST architectural style has six constraints and web services that satisfy these constraints called RESTful Web services. These constraints are listed below:

Client–server architecture: This principle separates the user interface concerns from the data storage concerns. It improves the platform-independency and simplifies the server components.

Stateless: Stateless principle enforces keeping the state management on the client. All client-server operations should be stateless.

Cacheable: The data in a response to a request has to be labeled as cacheable or non-cacheable. If this data is labeled as cacheable then, the client cache can reuse the data for the same requests.

Uniform interface: This constraint is defined by four parts. First of all, any request to the server has to include a resource identifier which is called as URI. Secondly, the server has to return a response which includes enough information to allow changes from the client. Also, all the requests that are sent from the client have to contain all the information the server needs. Likewise, all the responses that are returned from the server have to contain all the information the client needs, too. These are the principles of the third part of the uniform interface constraint. Lastly, the client has to be able to use dynamically the hyperlinks which are provided from the server. The client does not need to have the hard-coded information about the server structure or dynamics.

Layered system: This constraint allows the middle layers between the client and the server. These middle layers are also server layers which might provide a security layer, a caching layer, a load-balancing layer, or other functionality. Each layer should be agnostic to beyond the immediate layer with which it is interacting. Lastly, the middle layers should not affect the request or the response.

Code on demand: This is an optional constraint. The client can request an executable code from the server, and then the server will return the code. The client then can execute that code.

Commonly, HTTP protocol is used for the REST architectural style. Hence, the HTTP methods such as GET, HEAD, POST, PUT, PATCH, DELETE, CONNECT, OPTIONS and TRACE are available in RESTful systems.

2.1.6 JavaScript Object Notation - JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format [20]. It is easy for people to read and write, and machines also parse and generate it easily. It is language independent but it has conventions that are familiar to programmers of the C, Java, etc. JSON has two main structures, one is a collection which consists of name and value pairs. This structure is known in different languages as an object, struct or hash table. The other structure is an ordered list of values which is realized as an array, vector or list in most languages.

2.1.7 Waffle

Waffle is a fast, asynchronous web framework for Lua/Torch to generate the RESTful web services [21].

2.1.8 Jsoup - Java HTML Parser Library

Jsoup is a Java library which is used for extracting and manipulating data from Hypertext Markup Language (HTML) pages. It parses HTML pages as same as modern browsers do [22].

2.1.9 Apache Tomcat

Apache Tomcat is an open-source software which provides HTTP web server environment for Java web applications. It implements Java EE specifications such as: Java Servlet, JavaServer Pages (JSP), Java EL, and WebSocket [23].

2.1.10 Node.js

Node.js is an open-source runtime environment used for executing JavaScript code outside of a browser. It can be used for input/output and data streaming on the server side [24].

2.2 Implementation Details

Inspiration Hunter implements client-server architecture. While the client layer runs on the user's PC, the server layer runs on remote server. The client layer consists of multiple CEP extensions and its main purpose is to manage user's requests. The server layer contains applications for image hunt and style transfer. Our motivation for the client-server separation is threefold. First of all, we want to ensure platform independence; this means that our system should work on the personal computers of designers without any system requirements. For example, designers are able to use our system regardless of the operating system they use. Besides, high-cost services that require a GPU (e.g. neural style transfer) can work regardless of whether the designer's personal computer has a GPU. Second, we want designers to not need to download any patches or code when there are updates on the server side, such as any bug fixes or adding new features. Third, we desire reusability: with client server separation, our services can also be used by systems other than Inspiration Hunter.

Figure 2.4 shows the communication diagram of our system. We used REST architectural style for client-server communication. We developed RESTful web services for image hunt by using Java's REST libraries. Likewise, we used Waffle for communicating to style transfer services. The client layer sends POST requests to upload images or keywords to services. When a service finish its processes, it returns a JSON list containing the result set. Then the client layer parse this list and show the results to the designer. Image hunt services are Java web applications and run in Apache Tomcat application server. The style transfer services do not need any application server to run because Waffle also provide an HTTP web server environment. The image hunt services use HTTP requests and responses to communicate with Google Images. Unfortunately, Google does not provide any application programming interface (API) for image-based search and, the text-based search API is no longer supported. Google returns a response containing an HTML context which consists of components that are understandable to people when they are interpreted by a web browser. Actually, this process is equivalent to searching Google Images by using a regular web browser. Therefore, the image hunt services should be able to parse the response of the Google Images. We used Jsoup for parsing the return HTML context

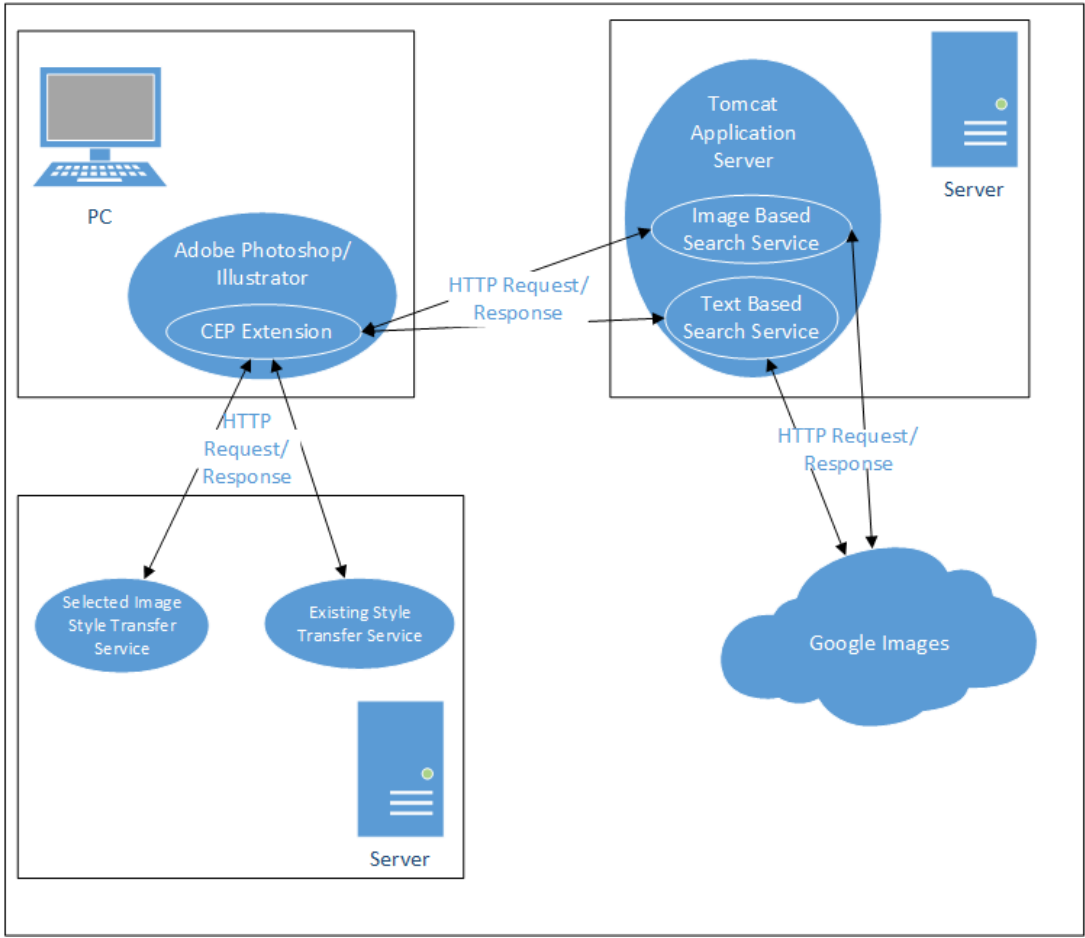


Figure 2.4: Communication diagram

and selecting the returned image links.

The client layer consists of CEP extensions which we developed for Adobe Photoshop and Illustrator. The designers do not need to leave the design environment while using our extensions. These extensions are used for saving active document, uploading images or keywords to the servers, showing the result set and downloading any image then opening it as a new document or layer. CEP provides built-in functions to perform the exporting active document to PNG image and opening a downloaded image as a new layer/document operations. Also, we used Node.js to perform the input/output operations such as saving the image to the disk, downloading the image from the web.

2.2.1 Image Hunt

There are two types of image hunt, one is image-based hunt and the other is text-based hunt. Figure 2.5 depicts the process of image-based hunt. Once the designer decides to image hunt by using the active document, he/she clicks the search button on the user interface. Then, the client layer exports the active document to PNG and sends it to the image-based hunt web service in a binary string format. Google Images does not accept POST requests, so our service is not allowed to POST the image which is uploaded from the client layer. Therefore, the image-based hunt service need to store the uploaded image on a publicly accessible server in order to give permission to Google Images to access it. After saving the uploaded image our service generate an image link then, send it to Google Images. When Google Images is queried with the generated image link, it return an HTML context which includes the result set. Our service is responsible for parsing the Google Images' response and generating a JSON array with image links. After returning the resulted JSON array, the client layer displays the search results. If the designer selects any image from the result set, the client layer opens it as a new document/layer.

Whenever an image is downloaded, the system generate a new image by copying only the content of the downloaded image. Therefore, all the associated metadata and filename of the image are automatically removed. This is important because if the designer uses downloaded image as a probe for an image hunt, the retrievals will be

purely visual based.

Figure 2.6 shows the process of text-based hunt. Text-based hunt has a similar process to image-based hunt except that the client layer takes keywords from the designer then uploads them to the text-based hunt web service.

2.2.2 Style Transfer

There are two options for the style transfer, one is applying the style of a selected image to the content image and the other is applying known artistic styles to the content image. Figure 2.7 shows the process of the selected image style transfer process. First of all, the designer should select a style image from image-based/text-based hunt results. Then, the client layer exports the active document to PNG image and sends the selected image and the exported image to the selected image style web service. We developed the web service by using Torch framework. This web service employ a publicly available Neural Style Transfer code [25]. Figure 2.8 shows the existing style transfer process. In this process, the designer does not need to select a style image.

2.3 Performance Results

Image hunt performance depends on the network speed. It takes 3-5 secs for text-based image hunt and 5-7 secs for image-based image hunt when the download speed is 18 Mbps and upload speed is 2 Mbps. For style adaptation operations, we used RTX 2080 Ti graphics card. When the size of the content image is 512x512 px, applying known artistic styles to the content image service takes 3 secs and applying the style of a selected image to the content image service takes 20 secs.

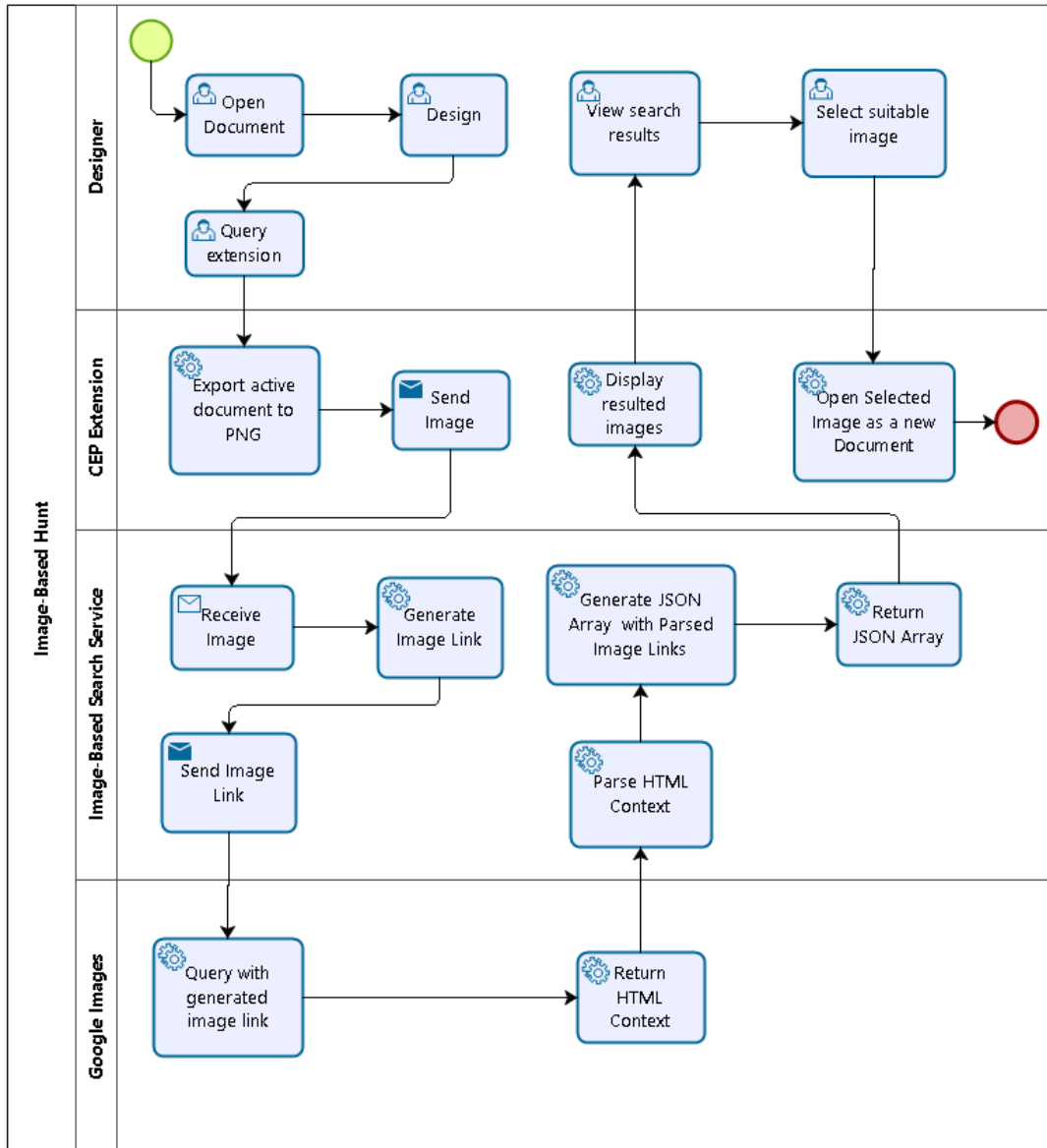


Figure 2.5: Image-based Hunt Process

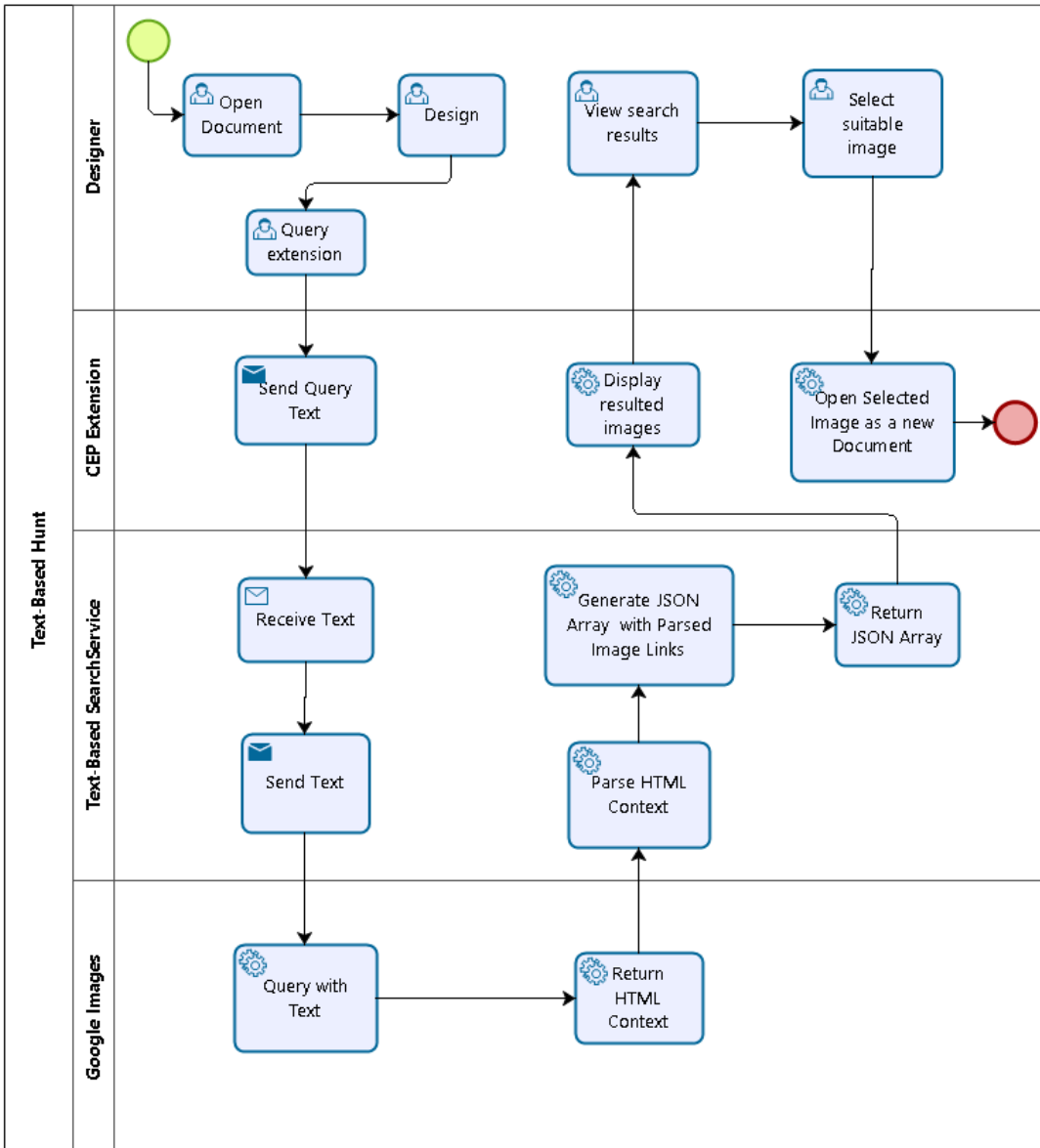


Figure 2.6: Text-based Hunt Process

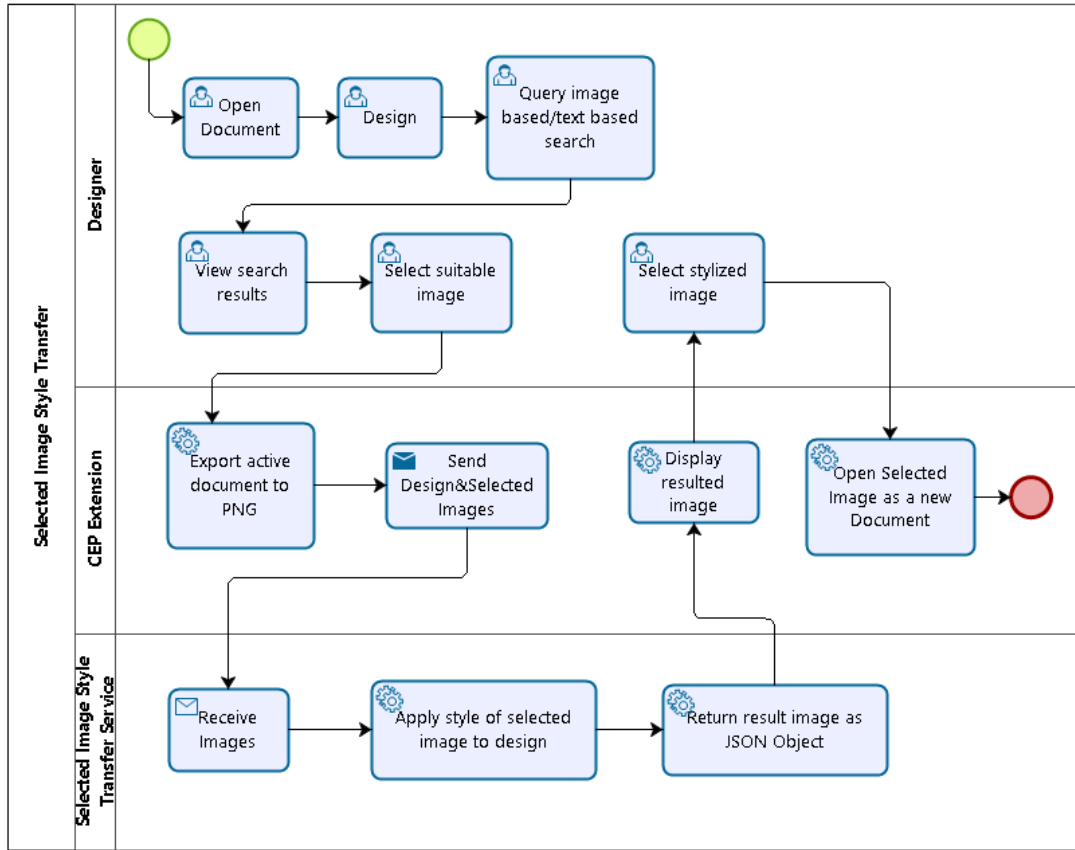


Figure 2.7: Selected Image Style Transfer Process

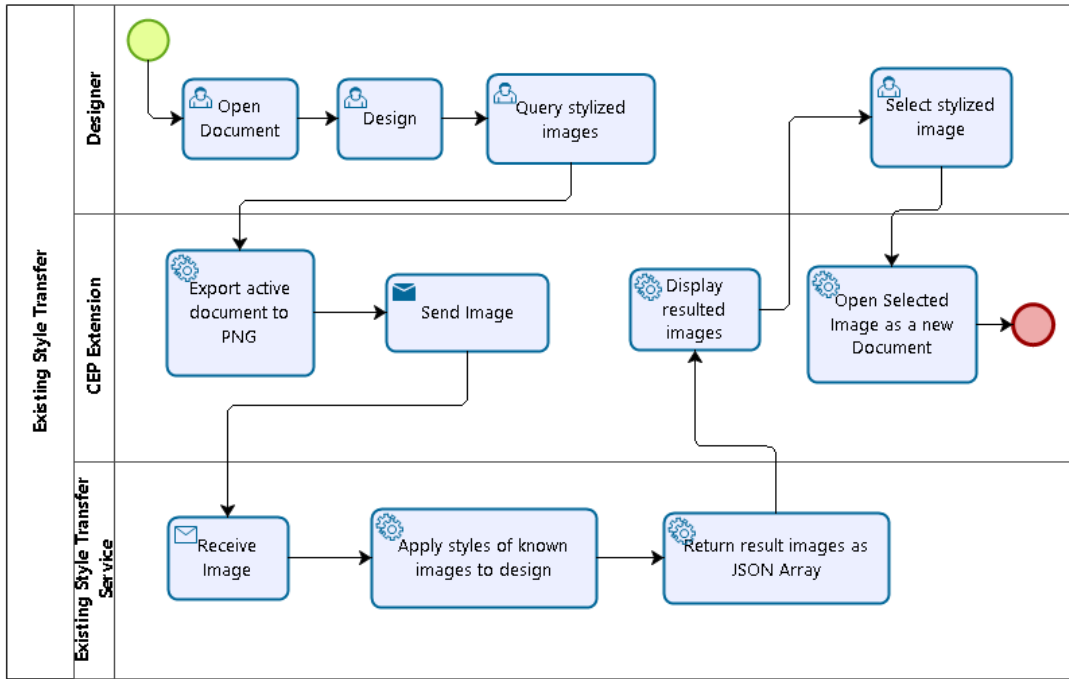


Figure 2.8: Existing Style Transfer Process

CHAPTER 3

EXPERIMENTS

We organized seven experiments with five volunteer participants. One of the participants is a graduate student in Industrial Design, and the others are computer engineers who have not studied art or design. Two of the participants have no experience in using Photoshop or design. One of the experienced participants is a game developer and he uses Photoshop while designing graphical part of the games. The other experienced one uses Photoshop for designing web sites, banners, and posters.

We requested participants to design something after giving brief information about the software. Except for the last two experiments, we did not give any subjects, ideas or templates to the participants. In the last two experiments, we gave the participants an image obtained in the fourth experiment and asked them to design the content by reusing the given image. Every design step was recorded in the background. Designers only used Inspiration Hunter for exploring materials.

3.1 Experiment 1

The designer is a 22-year-old computer engineer and, he loves designing banners, photoshopping digital images turn them into funny pictures. The evolution of the design is depicted in Figure 3.1. He started the design by drawing a Sponge Bob character (1). In the next step, he used the Sponge Bob as a query image. Among the retrievals, he picked a boxing glove (2) to turn the character into a boxer (3). After painting the character to yellow (4), he did another image-based search. Among the retrievals, he picked a yellow tooth image (5). He duplicated the yellow tooth image and added them to the character as its two front teeth (6). He used (6) for image-

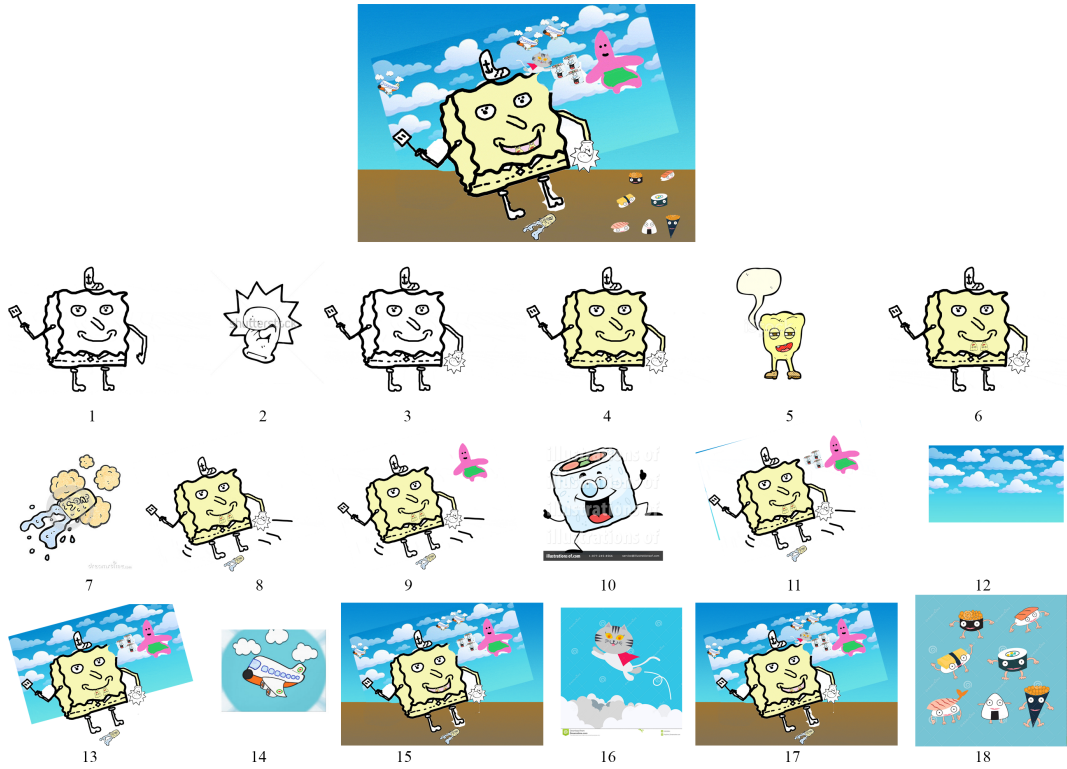


Figure 3.1: Experiment 1

based search. Among the retrievals, he found a soap image (7) then thought that the character stepped on the soap and it slipped (8). In the next step, he drew Patrick the Starfish, who is the friend of the Sponge Bob (9). After searching with (9), he picked a sushi roll image (10). He thought that his characters were in a sushi party, so he pasted multiple copies of sushi rolls (11). He wanted this party was an open-air party and searched for some cloud images by using a text-based search. Among the retrievals, he picked the image shown in (12). He pasted it on the collage (13) and used the new collage as the next query. The query results were related to clouds. Among the retrievals, he picked an airplane that gliding on the air (14). Then he pasted multiple copies of the plane (15). He drew the ground (15). The query with the active collage retrieved a flying cat (16) then he decided to paste this cat to his collage (17). After another image-based search, he noticed different kind of sushi (16) among the retrievals then added (16) to the collage.



1



2



3



4



5



6



7



8



9



10



11



12

Figure 3.2: Experiment 2

3.2 Experiment 2

The designer is the same person in the previous experiment. The evolution of the design is depicted in Figure 3.2. The initial idea of the designer about his design is putting Lara Croft character in a desert, so he started with making a text-based search using "Lara Croft" and "desert" keywords then chose two of them from search results (1,2). After pasting Lara Croft character on the desert image, he queried with the current collage but, he was not satisfied with the search results. So he decided to use only Lara Croft as a query image (4). Among the retrievals, he picked a bush-like tree image (5). He thought that Lara Croft could use this tree for hiding from people on camels (6). Searching with hiding Lara Croft image returned a woman image (7) and, he decided to paste this woman in a sitting position on one of the camels (8). Then he did another image-based search. Among the retrievals, he picked a rose image (9). Then he pasted multiple copies of this rose arrows (10). He thought that in his design, Lara Croft wanted to befriend with the people on camels, so she used roses as a sign of friendship. Since he did not like any images from next search results, he decided to search with using only the Lara Croft with rose arrows and the woman (11). He picked a squid image among the retrievals (12). He thought that he could use this image as a monster that appeared from the sand.

3.3 Experiment 3

The designer is a 26-year-old computer engineer and, he defines himself as a game developer. He designs game graphics by using Adobe Photoshop. The evolution of the design is depicted in Figure 3.3. He decided to start with a butterfly so, he did a text-based image search with the keyword butterfly. He picked a colorful butterfly image (1) among the retrievals. He continued with a background image which is colorful like the butterfly (2). He used his current collage as a query image and, among the retrievals, he picked a butterfly landed on a purple flower image (3). He liked the flower idea and cut-and-paste the flower and butterfly to the current collage (4). He did another image-based search and, among the retrievals, he picked a daisy garden image (5). He merged the colorful background and daisy garden images as a new

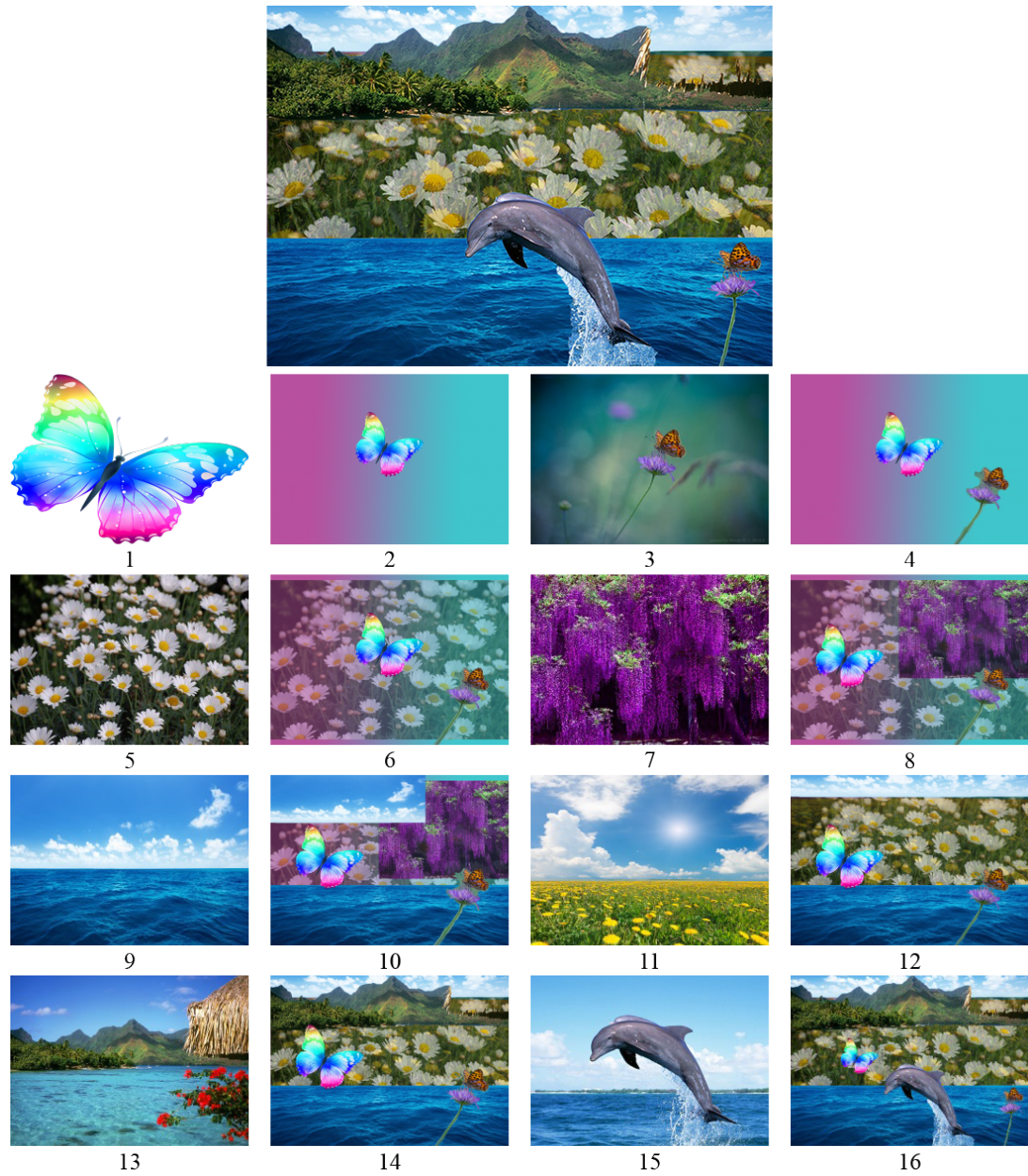


Figure 3.3: Experiment 3

background image (6). Searching with flowers resulted in new flower images and, he picked a purple flower tree forest image (7). Then he pasted it on the current collage (8). In the next step, he used the current collage as a query image but, results did not satisfy him, so he decided to search with sea keyword. Among the retrievals, he picked a wavy sea image (9). After adding the blue sea, he wanted to add something blue again so he did a text-based search with air keyword but, none of the retrievals pleased him. So he decided to cut (9) into two parts, which are sea and sky and paste them separately on the current collage (10). Next image-based search returned a garden of yellow dandelions image (11). He merged the daisy garden layer and dandelions to form a new background layer. After removing the purple tree (12), he did another image-based search. Among the retrievals, he picked a landscape image (13). He decided to cut-and-paste the mountain region (14). The next image-based search returned a dolphin image. After pasting dolphin image (16), he removed the butterfly, which added in the first step.

3.4 Experiment 4

The designer is a 27-year-old computer engineer who had no design and Photoshop experience. The evolution of the design is depicted in Figure 3.4. The designer did not have any initial idea. She started by drawing four different colored rectangles(1) then use the current image as a query image. Among the retrievals, she picked a frame image. She pasted frame on her collage then did another image-based search. She picked a hands image (4) then pasted it on her collage (5). The next image-based search with current collage returned a colored collaborated people image (6). After pasting the colored people image to her collage (7), she did another image-based search. She picked an image which includes human figures (8) then cut a figure from the image to paste it on her collage. She used the current collage as a query image but, search results were not satisfied her. So, she changed background color to yellow (9) then did another image-based search. The search results still did not pleased her, so she removed the colored collaborated people figure from the collage (10). The next image-based search retrieved a Clockwork Orange image (11). She cut the hat figure then pasted it on the human figure's face as mustache (12). In the next step



Figure 3.4: Experiment 4

she did another image-based search with the current design. Among the retrievals, she picked an artist image (13) and cut the head part and paste it on the collage (14). The next image-based search with the current collage returned a photographer image (15). She decided to combine the artist and the photographer, so she pasted the artist image on the photographer image and replaced the woman portrait with the head of the photographer.

3.5 Experiment 5

The designer is an Industrial Design graduate student. The evolution of the design is depicted in 3.5. She started by drawing a purple square (1) then she added a yellow disk on top of it (2). In the next step, she did an image-based search with the current design. Among the retrievals, she picked an image of concentric circles (3). The next image search with (3) returned a green circle image (4). After pasting the image (5), she changed its color to yellow. Then she replaced the initial yellow disk

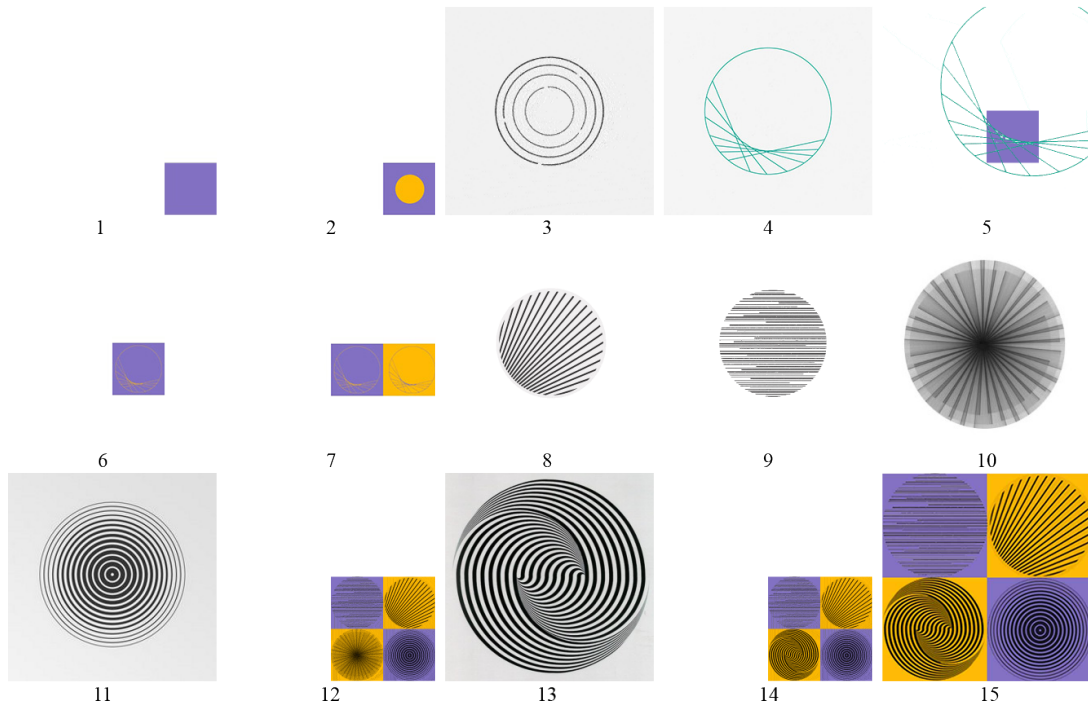
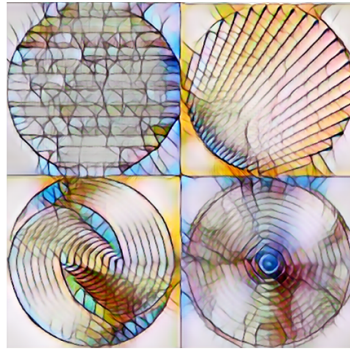


Figure 3.5: Experiment 5

with this circle (6). In the next step, she duplicated the square and the circle. Then she painted the second square and the second circle reverse colors (7). She used the current collage for the next image-based search. She picked (8), among the retrievals. She liked the form of the retrieved image (8) and she searched for similar ones by using (8). Among the similar images, she picked (9), (10) and (11). She placed the forms that she liked on the squares (12). In the next step, she did another image-based search with the current collage. She picked (13) and replaced one of the forms that she had already pasted with the new arrival (14). After cutting the blank area(15), she applied neural style transfer and finalize her design.

3.6 Experiment 6

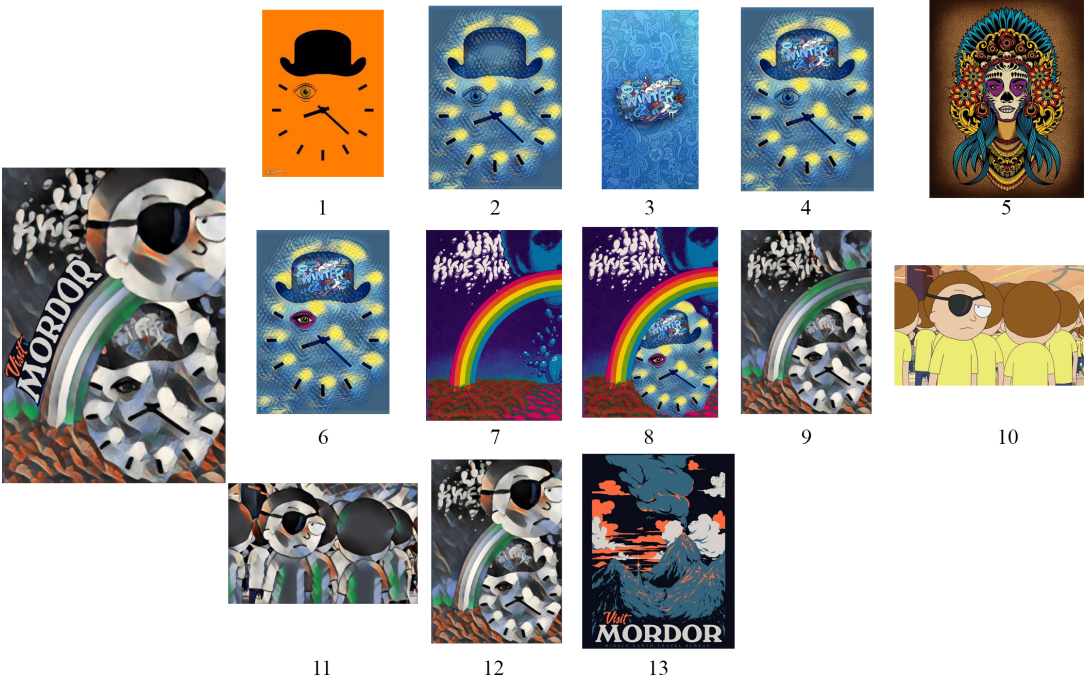


Figure 3.6: Experiment 6

The designer is the same person as the designer of experiments 1 and 2. The evolution of the design is depicted in 3.6. We passed the Clockwork Orange image which is retrieved from Experiment 4 and instructed him to start the design by reusing this image (1). He started by applying style transfer to (1) and obtained (2). He did an image-based search by using (2). Among the retrievals, he picked (3). He cut the

central content of the image, then he placed it on the hat (4). The next image-based search with the current collage returned a death goddess image (5). He cut the eye of the goddess then paste it as the eye of the figure of his collage (6). The next image hunt with the current collage returned Jim Kweskin poster (7). He cut the evolved clockwork orange figure from the active collage then pasted it under the rainbow of the Kweskin poster (8). After applying style transfer (9), he realized that the evolved clockwork orange figure looked liked Rick character from the Rick Morty. Then, he did a text-based search by using "Rick and Morty" keywords. Among the retrievals, he picked a one-eyed Morty face image (10). After applying style transfer to (10), he obtained (11) then he cut the face of one-eyed Morty and pasted it on the top of the evolved clockwork orange. In the next step, he did another image-based search with the current collage. Among the retrievals, he picked "Visit Mordor" poster (13). He cut the "Visit Mordor" text then pasted it on the top of the rainbow.

3.7 Experiment 7

The designer is the same person as the designer of experiment 4. The evolution of the design is depicted in 3.7. We passed the Clockwork Orange image which is retrieved from Experiment 4 and instructed her to start the design by reusing this image (1). She started by changing the square frame to circle frame (2). Then she did text-based image search by using "hat icon" keywords. Among the retrievals, she picked (3) then she replaced the hat of the clockwork orange with (3). After applying style transfer to (4), she obtained (5). At this step, she decided to design a coin. Then, she cut the central part of the current design and used it (6) for image-based search. The search results did not include any coin-like similar images so she gave up the coin design idea. She noticed the image that Alex holding a glass of milk (12) in the search results. Then, she decided to design a milk brand and use (6) as this new brand's logo. Then she queried with "milk in bottle" keywords. Among the retrievals, she picked (7) then cut one of the bottles and pasted the logo (6) in front of it (8). She did not satisfied with the current bottle figure so, she did another text-based search bu using "milk in bottle" keywords. Then she obtained (9) which is more aesthetic then she had before. She did three queries with "milk in glass", "table", and "clockwork orange"

keywords and, she picked (10), (11) and (12,13) respectively. Then she pasted all of the query results on her collage and wrote the text "Milk for adults" (14). She liked the font of the text of (13), so she downloaded this font from the Internet. Then she changed the text and its font (15). She decided to change the background layer (16) then, she applied the style of (17) to (16). She obtained (18) then, she decreased the opacity value of (18) and used it as a new background layer (19). In the next step, she applied style transfer (20) then she changed the font of the text (21). Finally, she decreased the opacity value of her current design.



Figure 3.7: Experiment 7

CHAPTER 4

SUMMARY AND DISCUSSION

In this thesis, we presented Inspiration Hunter which supports idea generation and evaluation stages of the creative design process by supplying crowd-generated big data from the world wide web. Current web content is quite rich in terms of visual materials. Search engines, on the other hand, provide suitable user interfaces to access these materials. Inspiration Hunter extends the capability of a content creation tool by providing services that link the tool to a web-based search engine. As the search engine, we chose Google Images because it has the richest content among the image search engines. Since we do not know which criteria used in Google Images' search algorithm, any results may pop up in the search results. In our experiments, we observed that the search results contain not only similar images but also many other unrelated images. The unrelated ones may contribute to the generation of new and interesting ideas.

We used Photoshop and Illustrator applications of Adobe Creative Suite as content creation tools because thanks to CEP it is easy to develop extensions to these tools. Also, they are widely used by both amateurs and professionals. We implemented client-server architecture while developing Inspiration Hunter. The client-side handles only the user requests and the manipulation of the content creation tool. The core functionality places in the server-side of the system. The server-side is not dependent on the client-side because of the separation of the concerns. So, any content creation tool which satisfies REST constraints can work with this server-side functionality. Therefore, the client-server separation promotes reusability and platform-independency. We are planning to integrate GIMP - GNU Image Manipulation Program which is an open-source content creation tool to our platform. Like Adobe,

GIMP provides a platform for extending its functionality. Thanks to our client-server separation, it is easy to integrate such extensible content creation tools to our platform.

Inspiration Hunter brings crowd-generated images to the designer and enables reusing the existing work by using cut/paste operations. We observed that even a person who did not have any design experience or skill could complete a design by using iterative cut-and-paste operations from the provided materials. Experiments show that, even without any initial idea, Inspiration Hunter facilitates the reuse of existing material in new and unexpected ways. Recall Experiment 7, Alex holding a glass of milk gave the idea of creating an advertisement for a milk brand to the designer.

The style transfer feature was turned off in the first four experiments. In the other experiments, the style transfer feature was used at least once. Designers usually used this option for changing the style of their final work. Also, application of the style transfer to a query image may vary the search results. As the search results diversify, the likelihood of coming up with new ideas is increasing.

Although we did not give any initial idea, starting topic, etc. the designers successfully produced meaningful and complete works. Only in the experiments 6 and 7, the designers were constrained with starting their design with the Clockwork Orange image. The designer of the Experiment 6 has never seen the film. Because of his clean mind, the designer conducted design experiment without no emotional attachment to the film. In Experiment 7, the film was no stranger to the designer so, the provided image of the film channeled the designer into film centric design ideas. Although the search results did not fixate to only film-related content, the designer stayed in the circle of ideas that related only the film itself.

We did not evaluate the completed designs in terms of aesthetics and usefulness. We conducted experiments to observe Inspiration Hunter's contribution to the design process.

Inspiration Hunter was developed to work on both Adobe Photoshop and Illustrator. One of the designers provided the information that only the vector image results would be useful in specific to Adobe Illustrator; hence, we did not conduct any exper-

iment using Illustrator. To ensure Inspiration Hunter work properly on Adobe Illustrator, we are planning to select only vector images from the search engine as a future improvement. Another future improvement is to move the server side operations to cloud for increasing reusability. Also, server side maintenance and management will be assigned to cloud to reduce the effort to maintain the system. We want to provide better and appealing user experience to the designers with more professional user interfaces.

As a future improvement, we want to reorganize the search results to increase diversity. We are planning to cluster the search results and choose the best-representing images from each cluster. Search results will include images that are similar to both the first query image and a random representing image.

REFERENCES

- [1] A. Warr and E. O'Neill, "Understanding design as a social creative process," *Proceedings of the 5th conference on Creativity & cognition*, pp. 118–127, 2005.
- [2] S. R. Herring, B. R. Jones, and B. P. Bailey, "Idea generation techniques among creative professionals," *Proceedings of the 42nd Hawaii International Conference on System Sciences*, 2009.
- [3] B. Shneiderman, "Creating creativity: User interfaces for supporting innovation," *ACM Transactions on Computer-Human Interaction*, vol. 7, pp. 114–138, 03 2000.
- [4] N. Cross, "Descriptive models of creative design: application to an example," in *Design Studies*, vol. 18, pp. 427–440, 1997.
- [5] E. Y.-L. Do and M. D. Gross, "Drawing analogies - supporting creative architectural design with visual references," in *3d International Conference on Computational Models of Creative Design*, pp. 37–58, 1995.
- [6] S. Lim, M. Prats, I. Jowers, S. Chase, S. Garner, and A. McKay, "Shape exploration in design: formalising and supporting a transformational process," in *International Journal of Architectural Computing*, vol. 6, p. shap, 2008.
- [7] C. Mougnot, C. Bouchard, A. Aoussat, and S. Westerman, "Inspiration, images and design : an investigation of designers' information gathering strategies," in *Journal of Design Research*, vol. 7, pp. 331–351, 2008.
- [8] N. Bonnardel, "Creativity in design activities: the role of analogies in a constrained cognitive environment," *Proceedings of the 3rd conference on Creativity & cognition*, pp. 158–165, 1999.
- [9] S. R. Herring, C.-C. Chang, J. Krantzler, and B. P. Bailey, "Getting inspired! understanding how and why examples are used in creative design practice," *CHI 2009 - Design Methods*, 04 2009.

- [10] E. B. Sanders and P. J. Stappers, “Probes, toolkits and prototypes: Three approaches to making in codesigning,” *CoDesign*, vol. 10, no. 1, pp. 5–14, 2014.
- [11] L. Sass and R. Oxman, “Materializing design: The implications of rapid prototyping in digital design,” *Design Studies*, vol. 27, no. 3, pp. 325–355, 2006.
- [12] B. Lee, S. Klemmer, S. Srivastava, and R. Brafman, “Adaptive interfaces for supporting design by example,” 2007.
- [13] N. Cross, *Here Comes Everyman*. 1972.
- [14] C. Shirky, *Here Comes Everybody: The Power of Organizing Without Organizations*. Penguin Group, 2009.
- [15] “Adobe creative suite.” https://en.wikipedia.org/wiki/Adobe_Creative_Suite, 2016.
- [16] N. Smith, “Ooh! ahh! google images presents a nicer way to surf the visual web.” <https://googleblog.blogspot.com.tr/2010/07/ooh-ahh-google-images-presents-nicer.html>, 2010.
- [17] “Google images.” https://en.wikipedia.org/wiki/Google_Images, 2016.
- [18] L. A. Gatys, A. S. Ecker, and M. Bethge, “A neural algorithm of artistic style,” *CoRR*, vol. abs/1508.06576, 2015.
- [19] R. T. Fielding, *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.
- [20] “Introducing json.” <https://www.json.org/>.
- [21] B. Englard, “waffle.” <https://github.com/benglard/waffle>, 2015.
- [22] “jsoup: Java html parser.” <https://jsoup.org/>.
- [23] “Apache tomcat.” <http://tomcat.apache.org/>.
- [24] “Introduction to node.js.” <https://nodejs.dev/>.
- [25] J. Johnson, “neural-style.” <https://github.com/jcjohnson/neural-style>, 2015.

APPENDIX A

INSPIRATION HUNTER - USER INTERFACES

Figure A.1 and A.2 show the image-based search user interfaces of Inspiration Hunter. Users can explore similar materials just by clicking the blue refresh button in figure A.1. They do not need to upload or save their current design, the system handles these processes in the background. By clicking an image in the search results, users can download the selected image and this downloaded image is opened as new document automatically. Figure A.3 shows the text based search user interface.

For text-based searching, users should enter any keyword to text-box and click the blue search button. Users can download the selected images same as the image-based search. Figure A.4 demonstrates combination of image and text-based searches.

Figure A.5 and A.6 demonstrate how users can select an image from the search results and apply its style to the current design. After clicking "Apply Style" button, the system applies the style of the selected image to the current design and the resulted image is opened as a new document. Figure A.7 shows the user interface of applying styles of known images to the current design. After using the blue refresh button, stylized images are shown in the panel. Users can select a stylized image and just by clicking this image it is opened as a new document.

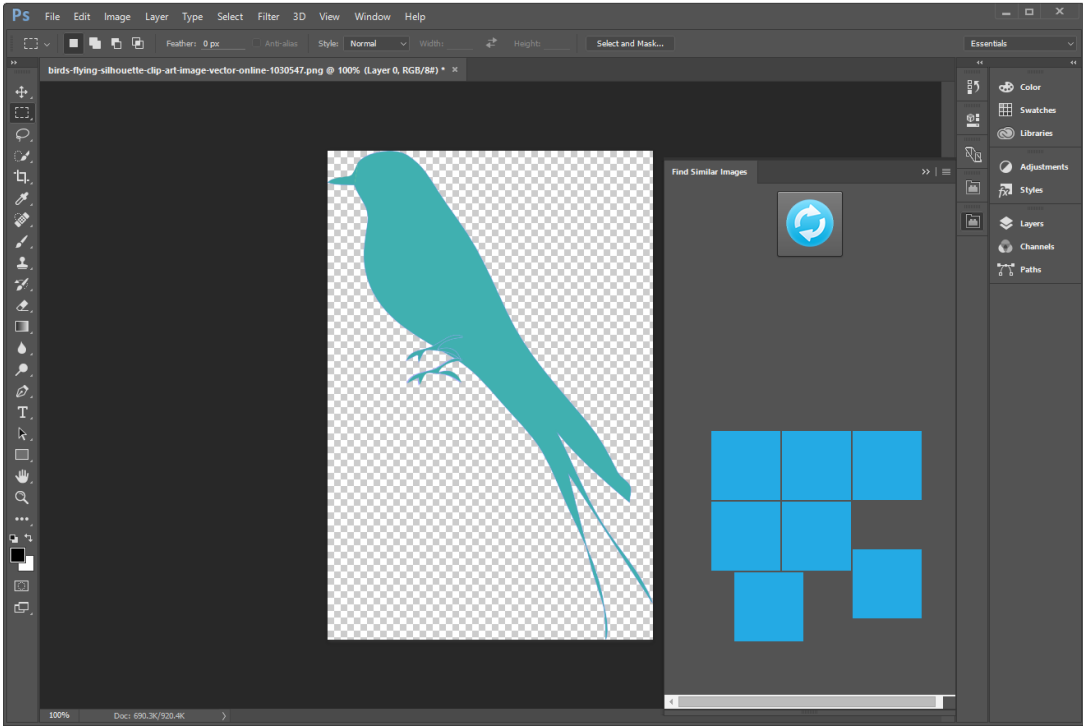


Figure A.1: Search by image - User Interface

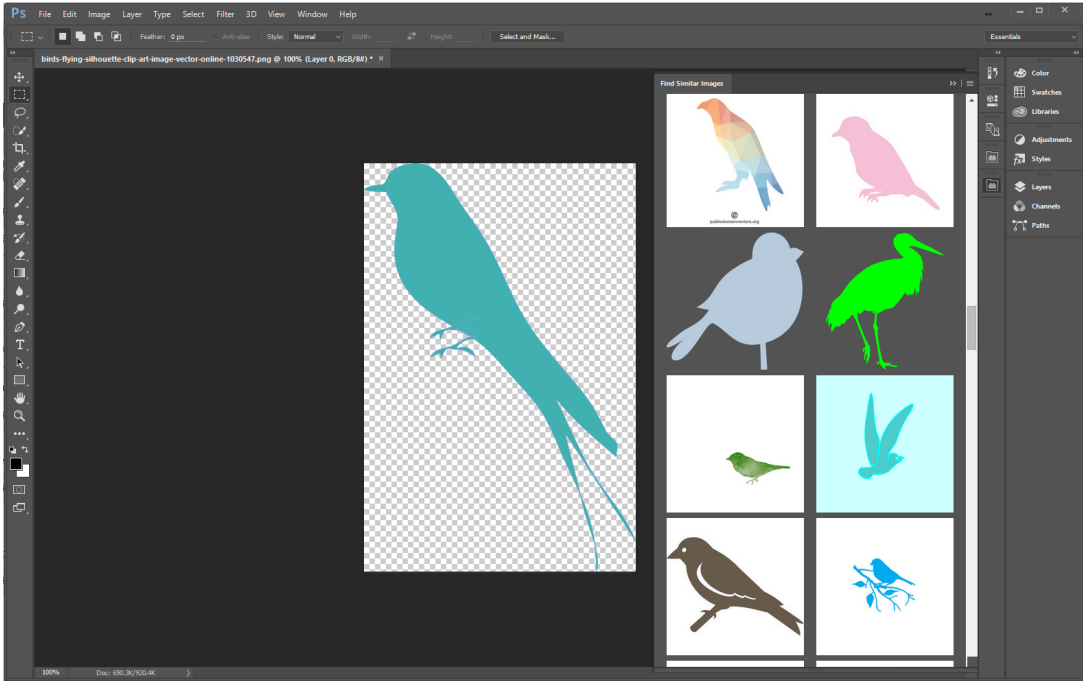


Figure A.2: Search by image - User Interface

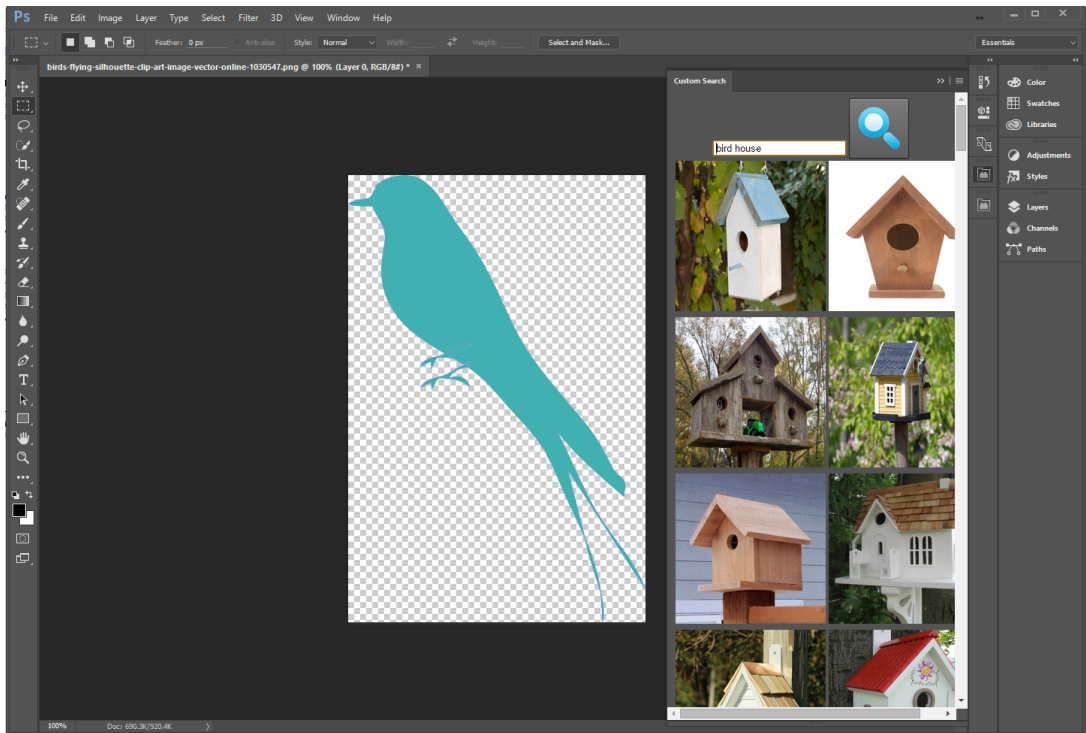


Figure A.3: Search by text - User Interface

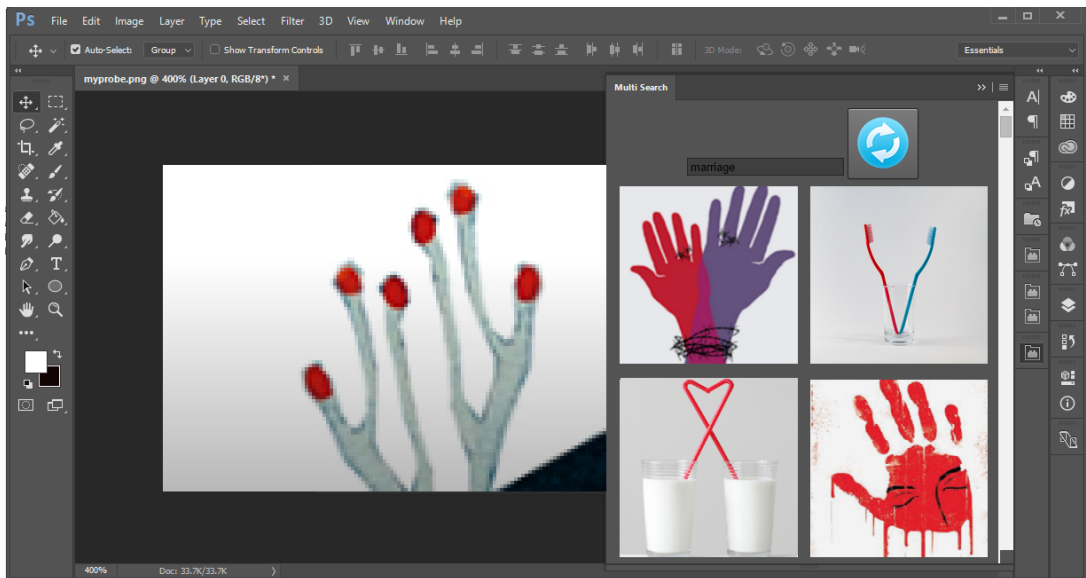


Figure A.4: Search by image and text - User Interface



Figure A.5: Apply the style of the selected image - User Interface

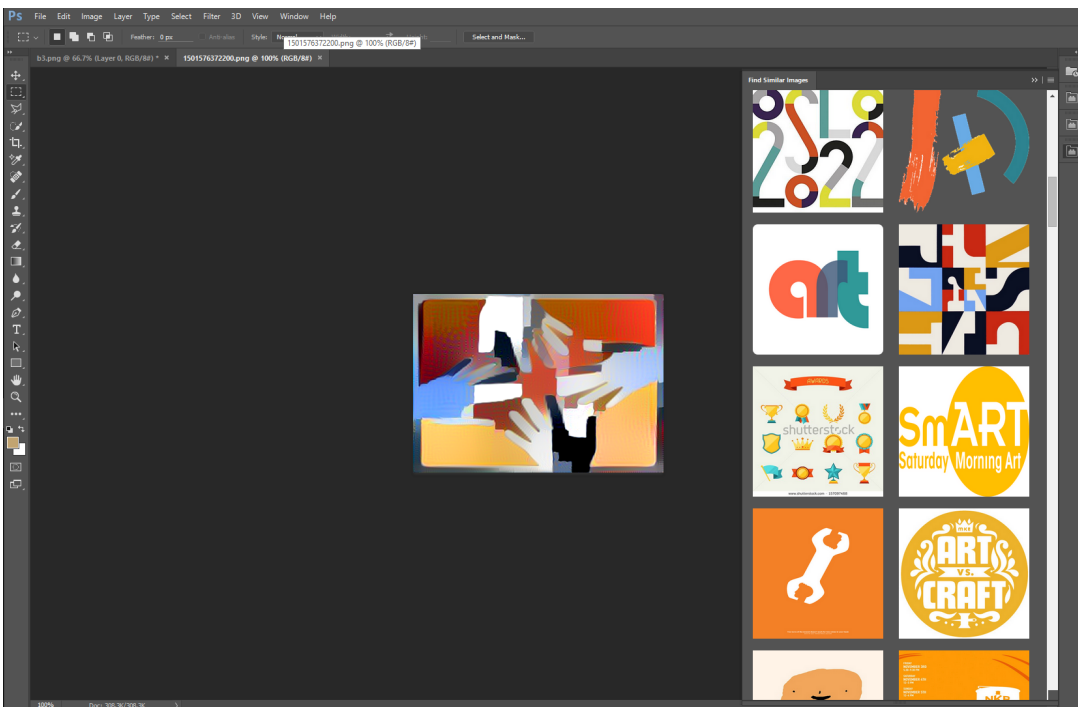


Figure A.6: Apply the style of the selected image result - User Interface

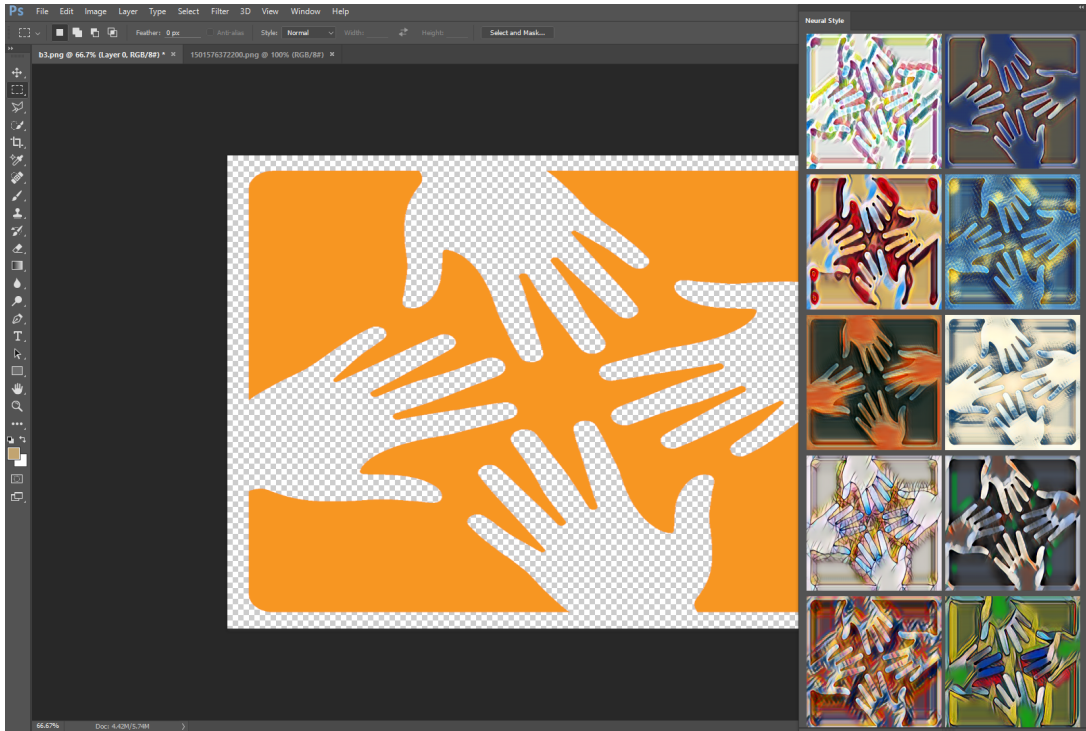


Figure A.7: Apply the styles of known images - User Interface