Improving User Perceived Load Times Through Saliency Maps

John Murzaku, Khiem Phi, Tanmay Srivastava

Introduction

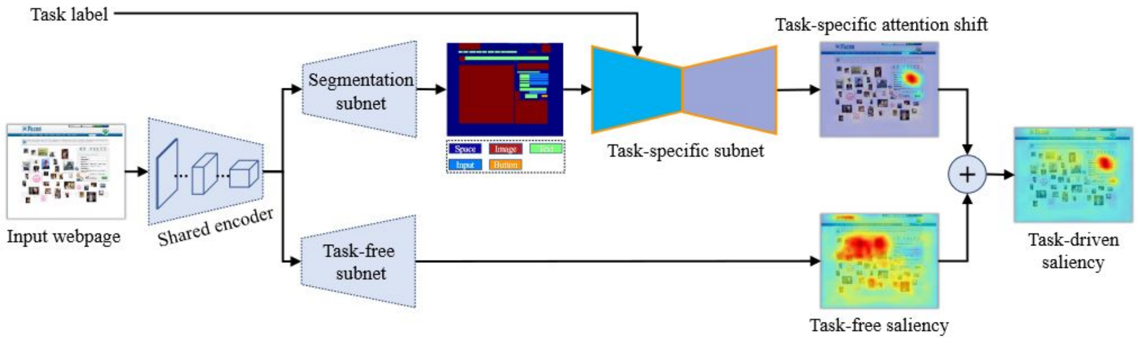
- Server push is an under-utilized functionally of the HTTP protocol.

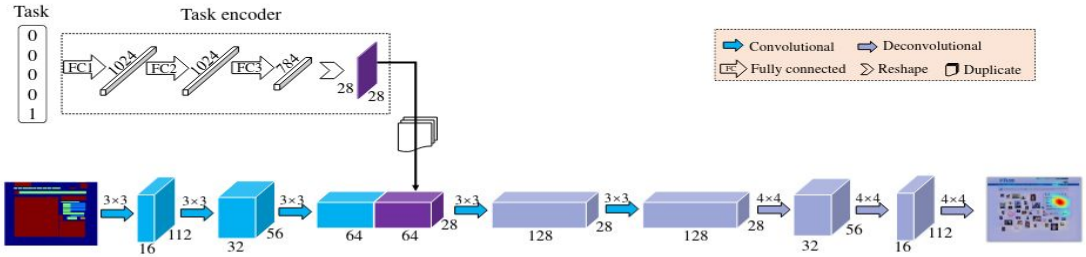
- Previous research has shown that uPLT (user-perceived load time) is highly influenced by the most visually salient regions in web-pages. (Kelton C. 2019)

→ If we can predict salient regions using an ML model, then we can utilize server push to improve uPLT

Salient-Region Prediction Model

- To predict salient regions in a webpage, we utilize a Task-Specific Salient Prediction Model (Zheng et.al 2015)



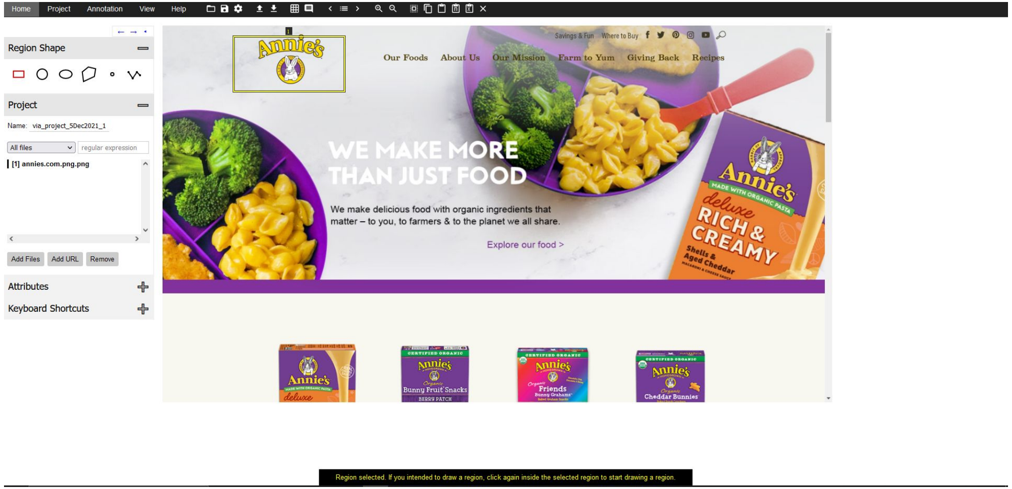


Salient-Region Dataset

- A dataset of 852 webpages was collected

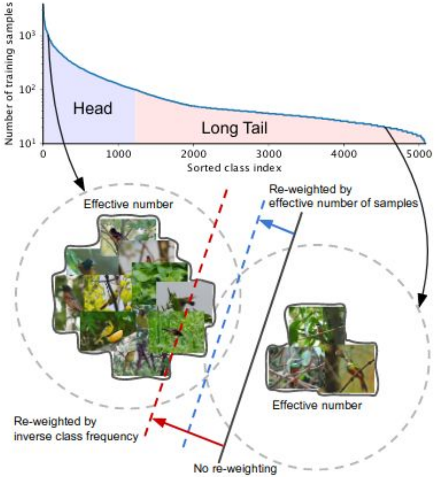
- Developed a custom annotation tool with JQuery based on Oxford-VIA Annotation system (Dutta and Zisserman, 2019)

- Utilized CB-Loss for fast convergence instead of pre-training on SALICON dataset. (Cui et.al 2019)

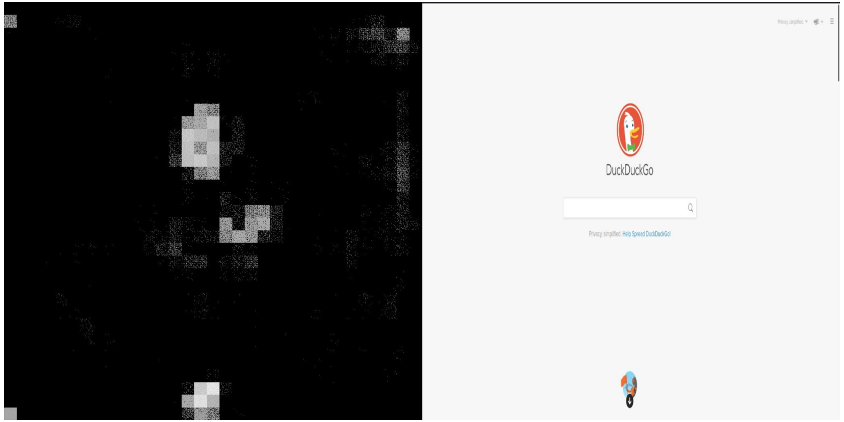


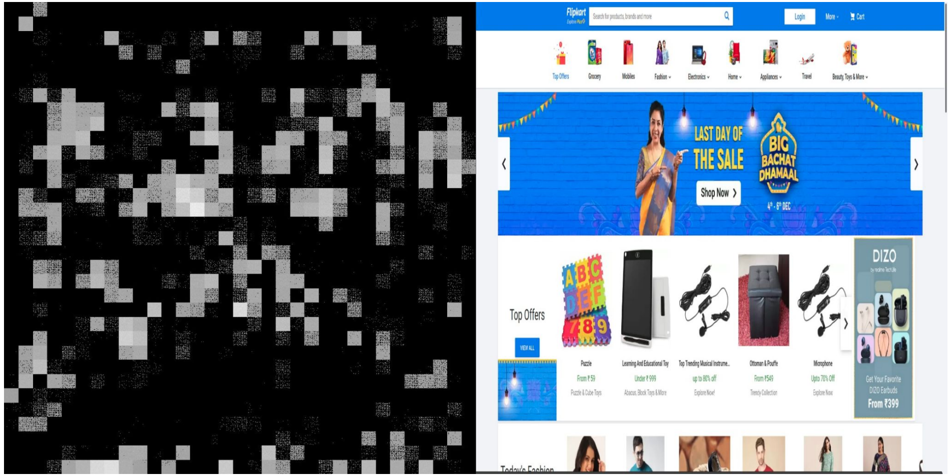
We make delicious food with organic ingredients that matter – to you, to farmers & to the planet we all share.

Region selected. If you intended to draw a region, click again inside the selected region to start drawing a region.

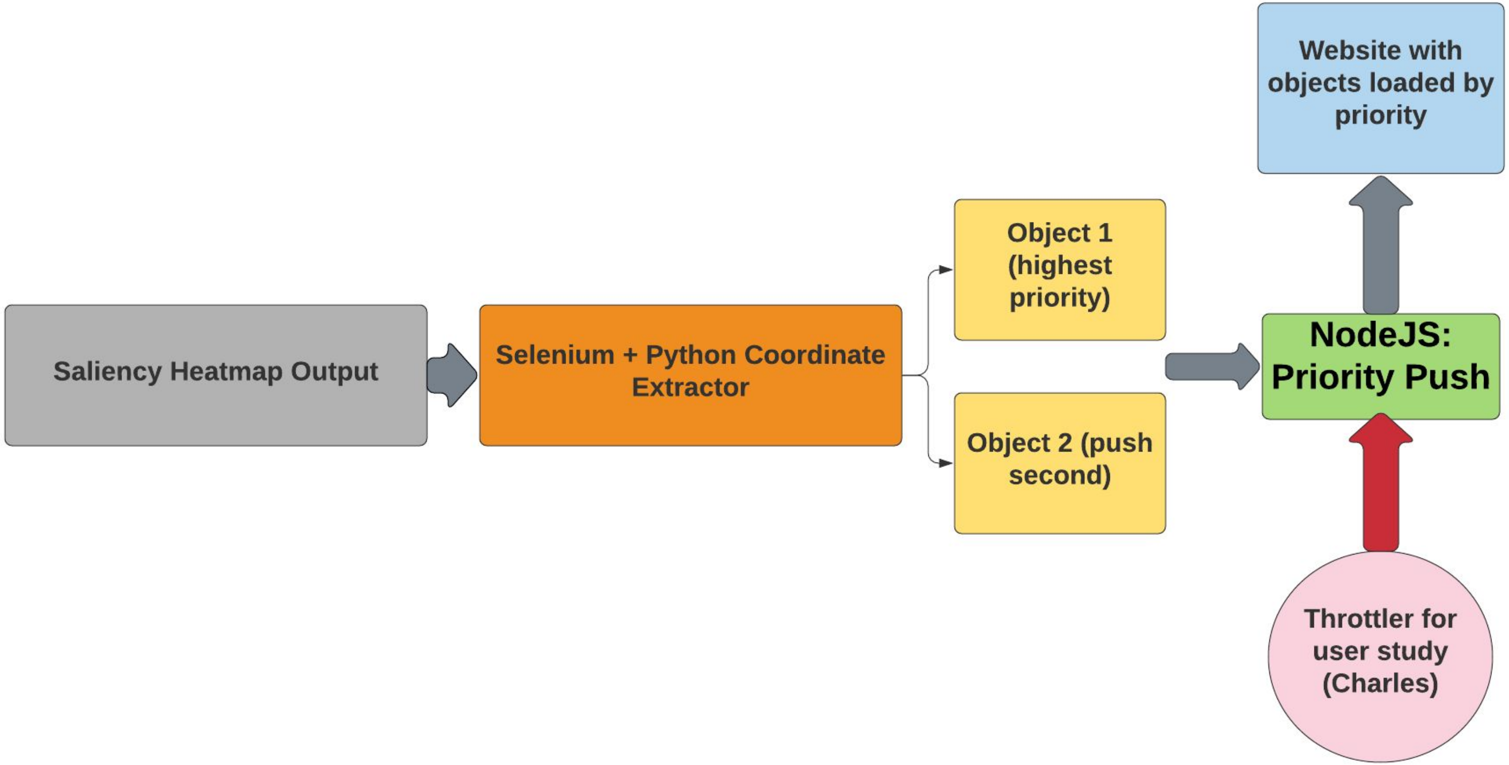


Salient-Region Prediction Models: Qualitative Results





Front End System Architecture



Saliency Heatmap Output

Selenium + Python Coordinate Extractor

Object 1 (highest priority)

Object 2 (push second)

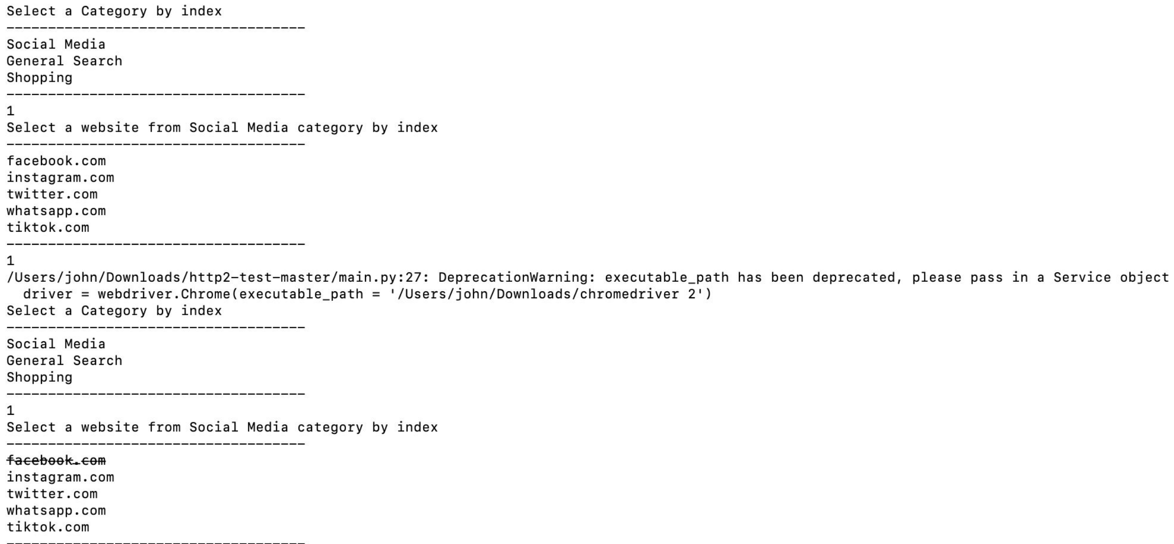
NodeJS: Priority Push

Website with objects loaded by priority

Throttler for user study (Charles)

Selenium and Python Coordinate Extractor

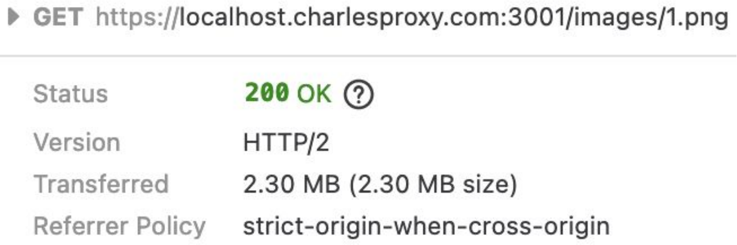
- Given the model saliency heatmap output, prompt user for website category
- Keep memory of the websites that the users have selected
- Use the Selenium library to get website.
  - **Input:** coordinates and rank from model saliency heatmap
  - **Output:** HTML elements inside the coordinates → store objects by rank, send to NodeJS



```
Select a Category by index
------------------------------------
Social Media
General Search
Shopping
------------------------------------
1
Select a website from Social Media category by index
------------------------------------
facebook.com
instagram.com
twitter.com
whatsapp.com
tiktok.com
------------------------------------
1
/Users/john/Downloads/http2-test-master/main.py:27: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
  driver = webdriver.Chrome(executable_path = '/Users/john/Downloads/chromedriver 2')
Select a Category by index
------------------------------------
Social Media
General Search
Shopping
------------------------------------
1
Select a website from Social Media category by index
------------------------------------
~~facebook.com~~
instagram.com
twitter.com
whatsapp.com
tiktok.com
------------------------------------
```

Front End: NodeJS

- **Input:** objects with their priority rankings from the Selenium + Python extractor
- **Output:** Website with objects pushed by priority
- Why NodeJS? Built in HTTP2 module!
- **Code implementation:**
  - Create SSL certificates to use HTTP2 for our local development
  - Open file sync module to prepare pipeline for receiving objects
  - Open HTTP2 stream, push objects in stream by priority ranking



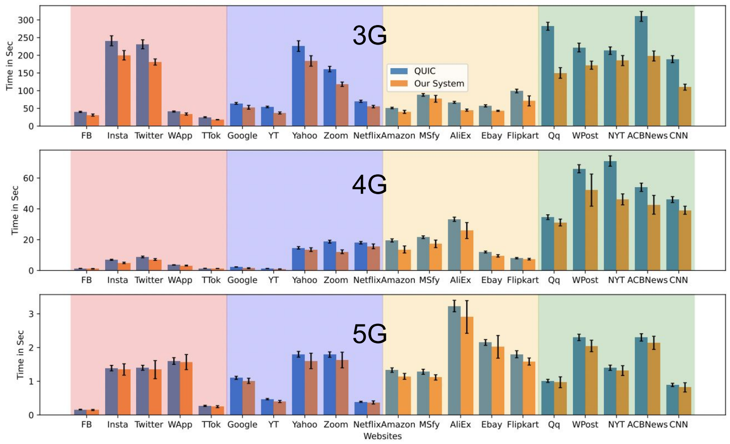
▸ **GET** https://localhost.charlesproxy.com:3001/images/1.png

Status	**200** OK ⓘ
Version	HTTP/2
Transferred	2.30 MB (2.30 MB size)
Referrer Policy	strict-origin-when-cross-origin

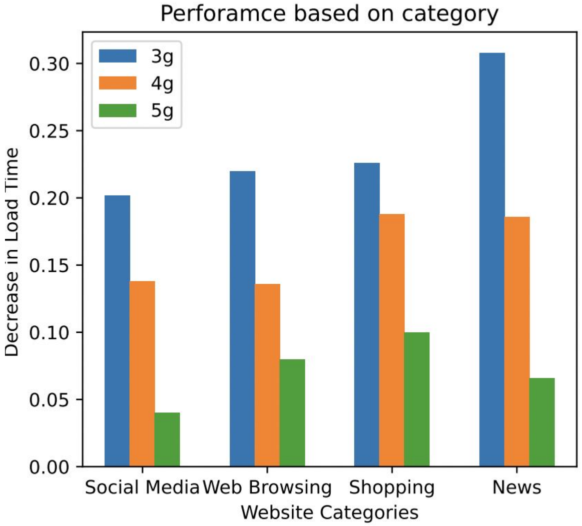
Evaluation

- Conducted user study with 5 users in 3 different network configurations:
  - 3G
  - 4G
  - 5G
- 20 websites across 4 categories: General Search, News, Shopping, Social Media.
- Ground Truth was Page Load Time estimated from Selenium. Mean of loading pages 30 times each websites.
- Made a video of our system and made it of same length of the PL time of respective website.
- User had to click mouse button to indicate when they think page has loaded.

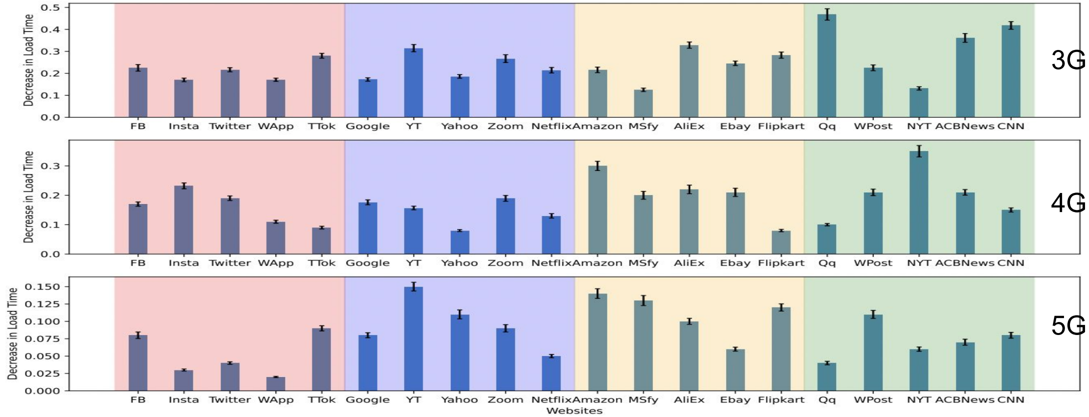
Results



3G
4G
5G
QUIC
Our System
Time in Sec
0
50
100
150
200
250
300
0
20
40
60
0
1
2
3
FB
Insta
Twitter
WApp
TTok
Google
YT
Yahoo
Zoom
Netflix
Amazon
MSfy
AliEx
Ebay
Flipkart
Qq
WPost
NYT
ACBNews
CNN
Websites



Perforamce based on category
3g
4g
5g
Decrease in Load Time
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Social Media
Web Browsing
Shopping
News
Website Categories



3G
4G
5G
Decrease in Load Time
0.0
0.1
0.2
0.3
0.4
0.5
0.000
0.025
0.050
0.075
0.100
0.125
0.150
FB
Insta
Twitter
WApp
TTok
Google
YT
Yahoo
Zoom
Netflix
Amazon
MSfy
AliEx
Ebay
Flipkart
Qq
WPost
NYT
ACBNews
CNN
Websites

Limitations

- Performance on going from one page to the next still untested.
- User study not conducted on mobile phones.
- Model optimization post user study absent.

Demo Video

Our System

Normal browser

Citations

Abhishek Dutta and Andrew Zisserman. 2019. The VIA Annotation Software for Images, Audio and Video. In Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3343031.3350535.

Cui, Y., Jia, M., Lin, T. Y., Song, Y., & Belongie, S. (2019). Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9268-9277).

Kelton, C., Ryoo, J., Balasubramanian, A., & Das, S. R. (2017). Improving user perceived page load times using gaze. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)* (pp. 545-559).

Zheng, Q., Jiao, J., Cao, Y., & Lau, R. W. (2018). Task-driven webpage saliency. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 287-302).