

# Torchlurk

## A neural network visualization tool

- Project summary -

Yann MENTHA, Julien BERGER, Gianni GIUSTO

April, 2020

### 1. Project goal

Due to increasing popularity of the PyTorch framework and the lack of intuitive visual analysis tools, we wish to provide a user-friendly interface targeting 2 main goals:

1. Give insights to the user about what the network is actually learning
  - display most significant images activating a given neuron
  - filter visualization by gradient descent on the input image
  - project back the feature map thanks to deconvolution in order to isolate the pixel(s) responsible for the given activation
2. Allow the user to compute a Torchlurk visualisation from his own trained torch Neural Network.

### 2. Visualization and interactions

#### 2.1 Sketches

You can discover in Appendices an overview of the already implemented visualizations. Note that this is not the final results, each visualization will be improved and incorporated to the main HTML.

#### 2.2 Tools

You will find in the table below a non exhaustive list of tools we used/will use throughout this project (for visualization and interactions) and the Data Visualization course lectures associated with it:

Table 1: List of visualizations.

Visualization	Tool	Course	Description
DL framework popularity	D3.js	D3 (week 4)	Popularity of deep learning frameworks over past year up to now
Layer selection ( <i>swipe</i> )	JavaScript, Swiper.js	Interaction (week 5)	Scroll menu to navigate through layers and select the one we are interested in
Filter/images display	JavaScript	JavaScript (week 2-3) Interaction (week 5)	Dynamic Display of the computed/selected images of each neuron
HTML rendering	HTML, CSS	Intro (week 1) Color perception (week 6)	HTML backbone of the website
Histograms	D3.js	D3 (week 4)	Bar plot displaying mean activations of top 20 classes for each filter

### 2.3 Challenges

#### 2.3.1 Technical challenge

The layer visualisations (images generated by gradient descent) coupled with top5 most activating images ideally allows one to assess which image class a given neuron is most excited by. However, as this characteristic is never exclusive to one class, a new interesting feature would consist in computing a bar plot of the mean activations (mean and/or max) of each class for each filter. That way, the user would have access to an

additional quantitative information about the *filter specificity*. An hypothetical example of such a graph is displayed thereafter.

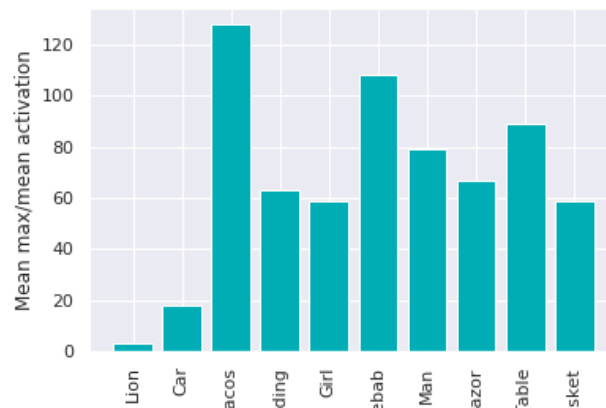


Figure 1: Activation for an arbitrary filter

### 2.3.2 Display challenge

Furthermore, for dense and deep neural networks, the number of parameters (*e.g.* weights, filters) is important. Navigation through the network architecture can be tough and hazardous. The main challenge will be to ease the navigation process by providing a practical interface and displaying relevant information only, while maintaining the navigation experience as fast as possible, using different combinations of display tools.

## 3. Project prototype

You will find the code and a working project prototype on the GitHub repository (team: *ethiopia-26*). To launch the project in the Google Chrome browser, enter the following command in your terminal:

```
$ google-chrome --allow-file-access-from-files main.html
```

## Literature

- Zeiler, Matthew D., and Rob Fergus. *Visualizing and understanding convolutional networks*. European conference on computer vision. Springer, Cham, 2014.
- Yosinski, Jason, et al. *Understanding neural networks through deep visualization*. arXiv preprint arXiv:1506.06579 (2015).
- Yosinski Jason, *Deep visualization toolbox*, (2017), GitHub repository: <https://github.com/yosinski/deep-visualization-toolbox>
- Yosinski Jason, *Understanding Neural Networks Through Deep Visualization*, (2020), Access: <http://yosinski.com/deepvis>

### Used library

- Ozbulak, Utku, *PyTorch CNN visualizations*, (2020), GitHub repository: <https://github.com/utkuozbulak/pytorch-cnn-visualizations>

## Appendices

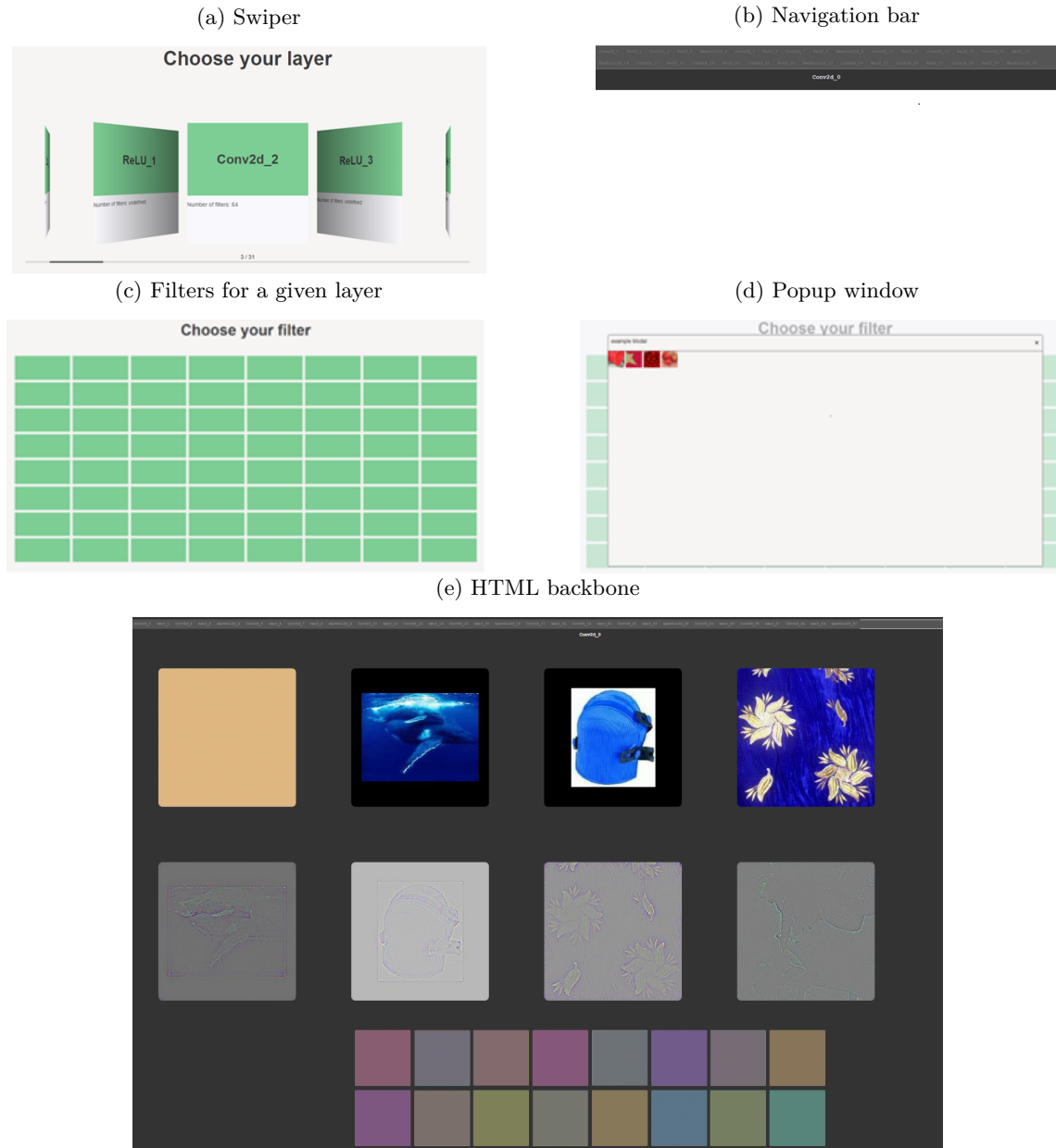


Figure 2: (a) A *swiper* animation to navigate through the network layers. An alternative to (a) could be to adopt a more classical navigation bar (may be more convenient for network with many layers) (b). (c) Filters for the selected layer. (d) Popup window (appearing once a filter is selected) displaying: most significant images activating a given neuron, filters and mean activations of top 20 classes for each filter (bar plot, not implemented yet). (e) Actual HTML backbone: a navigation bar and the list of filters corresponding to the layer of interest.