NANDA KISHORE nkk263@nyu.edu +1 (718) 679-7382

CLARIFAI FACE DETECTION

Codebase: https://github.com/nandukalidindi/clarifai-face-detection

NOTE: PLEASE SPARE SOME TIME IN CHECKING OUT THE VIDEO (Clarifai-face-detection-demo) THAT IS ATTACHED ALONG WITH THIS ZIP

Minimal client application **Javascript** and **Clarifai Predict API** to detect and highlight faces on the provided set of images.

TOOLS:

- React, Webpack and various loaders, presets. All the explanations for using those are mentioned below.

```
"babel-core": “^6.26.3",
"babel-loader": “^7.1.4",      Actual babel loader that is used by web pack during bundling
"babel-plugin-transform-class-properties": “^6.24.1",  Ability to use arrow functions inside classes
"babel-plugin-transform-object-rest-spread": “^6.26.0", Spread operator usage […x, …y]
"babel-preset-env": “^1.7.0",
"babel-preset-es2015": “^6.24.1",  ES6 syntax transpiler
"babel-preset-react": “^6.24.1",   JSX syntax transpileer
"css-loader": “^0.28.11",          CSS loader
"file-loader": “^1.1.11",          File loader for loading images
"style-loader": “^0.21.0"          CSS loader
```

- HTML5, localStorage API, CSS with latest features like **flex, css-grid**, etc.

- Javascript(ES6)

- Webpack with babel loader is used to build the application and to use ES6 features to make it cross browser compatible.

- JSDoc for documentation

IMPLEMENTED FEATURES

- Ability to upload images via Drag N Drop, Manual Upload and Inserting URL

- Calculating and rendering borders for returned bounded_regions from the Predict API

- Flash messages to show the status of the processing

- Modal popup to render specific components like Uploader and FaceDetectionPreview

- localStorage API is used to persist the images through the session to avoid an additional backed server.

- **fbemitter** to communicate across components that are not in the same hierarchy, this could have been avoided, just wanted to showcase the emitter functionality

- Portals to render Modal popups in the form of FlashMessages, although the actual Modal popups that are used to render Uploader components can also use the same Portal component it’s not refactored due to time constraints.

TO BE IMPLEMENTED

- Consolidated error handling and a refined UX

- Portals for Modal popups that render Uploader component

CONVENTIONS

Variable and method names are deliberately named a little relaxed and out of the usual manner to make the code more relatable to application. Please ignore if they are way too weird.

Tried to put in as much ES6 sugar as possible.

All methods are documented with as much explanation as possible.

BUILD INSTRUCTIONS:

The index.html can be opened without any npm install because the bundle is already built beforehand. Since this is not HMR I have access to the static bundle files that get generated during webpack watch.

If the app needs to be modified then **npm install** on the project root and **npm run webpack** to start webpack watch so that new bundle.js is built on every change an open index.html on a browser page. HMR is deliberately not used due to time constraints and also to ease the build process.

- npm install
- npm run webpack
- Open index.html on browser

POSSIBLE BUGS:

- Edge case scenarios that I might have missed

- Closing the Uploader popup before the processing completes will result in nothing because once the popup is closed the component that is responsible for setting the localStorage is destroyed.

- localStorage has an upper limit on the number of string characters that can be assigned for a particular item. Too many images could lead to an error while persisting into the localStorage API

- CSS organisation might be a little

TIME MANAGEMENT:

- Like suggested I spent the first **3** hours to get a complete working application in place however the styling, transitioning, comments, JSDoc, code optimisation took another **3-4** hours. I pushed everything to Github so, please feel free to track my commits and the code progress.

EPILOGUE!

Development of the app is done on Chrome Version 61.0.3163.100 (Official Build) (64-bit)

Please let me know anything is ambiguous.

I really had fun working on this assignment, this is very fresh compared to the usual boring hacker rank stuff. Thank you!