

UNIVERSIDADE DE LISBOA  
FACULDADE DE BELAS-ARTES



## **MEMÓRIA E REMEDIAÇÃO**

**Reproduções Descolonizadoras, através do Speculative Design, em  
Monumentos Comemorativos do Colonialismo**

### **ANEXOS**

Aldo Medina Zerega

Trabalho de Projeto

Mestrado em Design de Comunicação

Trabalho de Projeto orientado pela Professora Doutora

Sónia Isabel Ferreira dos Santos Rafael

2021

## i. Lista de objectos detectáveis pelo modelo coco-sdd mobilenet v2

```
/**
 * @license
 * Copyright 2019 Google LLC. All Rights Reserved.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 * =====
 */

export interface ObjectDetectionClass {
  name: string;
  id: number;
  displayName: string;
}

export const CLASSES: { [key: string]: ObjectDetectionClass } = {
  1: {
    name: "/m/01g317",
    id: 1,
    displayName: "person",
  },
  2: {
    name: "/m/0199g",
    id: 2,
    displayName: "bicycle",
  },
  3: {
    name: "/m/0k4j",
    id: 3,
    displayName: "car",
  },
  4: {
    name: "/m/04_sv",

```

```
id: 4,  
  displayName: "motorcycle",  
},  
5: {  
  name: "/m/05czz6l",  
  id: 5,  
  displayName: "airplane",  
},  
6: {  
  name: "/m/01bjv",  
  id: 6,  
  displayName: "bus",  
},  
7: {  
  name: "/m/07jdr",  
  id: 7,  
  displayName: "train",  
},  
8: {  
  name: "/m/07r04",  
  id: 8,  
  displayName: "truck",  
},  
9: {  
  name: "/m/019jd",  
  id: 9,  
  displayName: "boat",  
},  
10: {  
  name: "/m/015qff",  
  id: 10,  
  displayName: "traffic light",  
},  
11: {  
  name: "/m/01pns0",  
  id: 11,  
  displayName: "fire hydrant",  
},  
13: {  
  name: "/m/02pv19",  
  id: 13,  
  displayName: "stop sign",
```

```
},
14: {
  name: "/m/015qbp",
  id: 14,
  displayName: "parking meter",
},
15: {
  name: "/m/0cvnqh",
  id: 15,
  displayName: "bench",
},
16: {
  name: "/m/015p6",
  id: 16,
  displayName: "bird",
},
17: {
  name: "/m/01yrx",
  id: 17,
  displayName: "cat",
},
18: {
  name: "/m/0bt9lr",
  id: 18,
  displayName: "dog",
},
19: {
  name: "/m/03k3r",
  id: 19,
  displayName: "horse",
},
20: {
  name: "/m/07bgp",
  id: 20,
  displayName: "sheep",
},
21: {
  name: "/m/01xq0k1",
  id: 21,
  displayName: "cow",
},
22: {
```

```
name: "/m/0bwd_0j",
id: 22,
displayName: "elephant",
},
23: {
name: "/m/01dws",
id: 23,
displayName: "bear",
},
24: {
name: "/m/0898b",
id: 24,
displayName: "zebra",
},
25: {
name: "/m/03bk1",
id: 25,
displayName: "giraffe",
},
27: {
name: "/m/01940j",
id: 27,
displayName: "backpack",
},
28: {
name: "/m/0hnnb",
id: 28,
displayName: "umbrella",
},
31: {
name: "/m/080hkjn",
id: 31,
displayName: "handbag",
},
32: {
name: "/m/01rkbr",
id: 32,
displayName: "tie",
},
33: {
name: "/m/01s55n",
id: 33,
```

```
    displayName: "suitcase",
  },
  34: {
    name: "/m/02wmf",
    id: 34,
    displayName: "frisbee",
  },
  35: {
    name: "/m/071p9",
    id: 35,
    displayName: "skis",
  },
  36: {
    name: "/m/06__v",
    id: 36,
    displayName: "snowboard",
  },
  37: {
    name: "/m/018xm",
    id: 37,
    displayName: "sports ball",
  },
  38: {
    name: "/m/02zt3",
    id: 38,
    displayName: "kite",
  },
  39: {
    name: "/m/03g8mr",
    id: 39,
    displayName: "baseball bat",
  },
  40: {
    name: "/m/03grz1",
    id: 40,
    displayName: "baseball glove",
  },
  41: {
    name: "/m/06_fw",
    id: 41,
    displayName: "skateboard",
  },
},
```

```
42: {
  name: "/m/019w40",
  id: 42,
  displayName: "surfboard",
},
43: {
  name: "/m/0dv9c",
  id: 43,
  displayName: "tennis racket",
},
44: {
  name: "/m/04dr76w",
  id: 44,
  displayName: "bottle",
},
46: {
  name: "/m/09tvcd",
  id: 46,
  displayName: "wine glass",
},
47: {
  name: "/m/08gqpm",
  id: 47,
  displayName: "cup",
},
48: {
  name: "/m/0dt3t",
  id: 48,
  displayName: "fork",
},
49: {
  name: "/m/04ctx",
  id: 49,
  displayName: "knife",
},
50: {
  name: "/m/0cmx8",
  id: 50,
  displayName: "spoon",
},
51: {
  name: "/m/04kkgm",
```

```
    id: 51,  
    displayName: "bowl",  
  },  
  52: {  
    name: "/m/09qck",  
    id: 52,  
    displayName: "banana",  
  },  
  53: {  
    name: "/m/014j1m",  
    id: 53,  
    displayName: "apple",  
  },  
  54: {  
    name: "/m/01515",  
    id: 54,  
    displayName: "sandwich",  
  },  
  55: {  
    name: "/m/0cyhj_",  
    id: 55,  
    displayName: "orange",  
  },  
  56: {  
    name: "/m/0hkxq",  
    id: 56,  
    displayName: "broccoli",  
  },  
  57: {  
    name: "/m/0fj52s",  
    id: 57,  
    displayName: "carrot",  
  },  
  58: {  
    name: "/m/01b9xk",  
    id: 58,  
    displayName: "hot dog",  
  },  
  59: {  
    name: "/m/0663v",  
    id: 59,  
    displayName: "pizza",
```

```
},
60: {
  name: "/m/0jy4k",
  id: 60,
  displayName: "donut",
},
61: {
  name: "/m/0fszt",
  id: 61,
  displayName: "cake",
},
62: {
  name: "/m/01mzpv",
  id: 62,
  displayName: "chair",
},
63: {
  name: "/m/02crq1",
  id: 63,
  displayName: "couch",
},
64: {
  name: "/m/03fp41",
  id: 64,
  displayName: "potted plant",
},
65: {
  name: "/m/03ssj5",
  id: 65,
  displayName: "bed",
},
67: {
  name: "/m/04bcr3",
  id: 67,
  displayName: "dining table",
},
70: {
  name: "/m/09glw",
  id: 70,
  displayName: "toilet",
},
72: {
```

```
name: "/m/07c52",
id: 72,
displayName: "tv",
},
73: {
name: "/m/01c648",
id: 73,
displayName: "laptop",
},
74: {
name: "/m/0201f",
id: 74,
displayName: "mouse",
},
75: {
name: "/m/0qjjc",
id: 75,
displayName: "remote",
},
76: {
name: "/m/01m2v",
id: 76,
displayName: "keyboard",
},
77: {
name: "/m/050k8",
id: 77,
displayName: "cell phone",
},
78: {
name: "/m/0fx9l",
id: 78,
displayName: "microwave",
},
79: {
name: "/m/029bxz",
id: 79,
displayName: "oven",
},
80: {
name: "/m/01k6s3",
id: 80,
```

```
    displayName: "toaster",
  },
  81: {
    name: "/m/0130jx",
    id: 81,
    displayName: "sink",
  },
  82: {
    name: "/m/040b_t",
    id: 82,
    displayName: "refrigerator",
  },
  84: {
    name: "/m/0bt_c3",
    id: 84,
    displayName: "book",
  },
  85: {
    name: "/m/01x3z",
    id: 85,
    displayName: "clock",
  },
  86: {
    name: "/m/02s195",
    id: 86,
    displayName: "vase",
  },
  87: {
    name: "/m/01lsmm",
    id: 87,
    displayName: "scissors",
  },
  88: {
    name: "/m/0kmg4",
    id: 88,
    displayName: "teddy bear",
  },
  89: {
    name: "/m/03wvsk",
    id: 89,
    displayName: "hair drier",
  },
}
```

```
90: {  
  name: "/m/012xff",  
  id: 90,  
  displayName: "toothbrush",  
},  
};
```

ii. Protótipo de guião semi-dinâmico para as descrições fornecidas por Colombot:

```
const descriptionsPerson = [  
  "Potential slave",  
  "Hmm... Gives me mercancy vibes",  
  "Could be a good exportation",  
  "Cute savage, maybe with a soul, though",  
  "Meh... Skin problems",  
  "No soul, no love, welcome to the inventory",  
  "Looks like a good and talented servant",  
  "I will call you... indian",  
  "He could become a Christian very easily",  
  "We need one of those for the zoo"  
];  
  
const descriptionsThings = [  
  "Shall be delivered to the King",  
  "How splendid! new exotic piece for my museum",  
  "LOOOOOOOT!!!!",  
  "Kind of shine, might be gold? mine!",  
  "From now own that's mine",  
];
```

### iii. Função utilizada para processar imagens utilizando o algoritmo dithering

```
function bayerFilter(imageData, threshold, color) {
  var bayerThresholdMap = [
    [15, 135, 45, 165],
    [195, 75, 225, 105],
    [60, 180, 30, 150],
    [240, 120, 210, 90],
  ];
  var imageDataLength = imageData.data.length;

  var w = imageData.width;
  for (
    var currentPixel = 0;
    currentPixel <= imageDataLength;
    currentPixel += 4
  ) {
    // 4x4 Bayer ordered dithering algorithm
    var x = (currentPixel / 4) % w;
    var y = Math.floor(currentPixel / 4 / w);
    var map = Math.floor(
      (imageData.data[currentPixel] + bayerThresholdMap[x % 4][y % 4]) / 2
    );
    const show = map < threshold;

    imageData.data[currentPixel] = show ? 255 : color.r;
    imageData.data[currentPixel + 1] = show ? 255 : color.g;
    imageData.data[currentPixel + 2] = show ? 255 : color.b;
    imageData.data[currentPixel + 3] = show ? 0 : 250;
  }

  return imageData;
}
```

### iv. Componente responsável pela fragmentação e renderização da imagem obtida através da câmara de um dispositivo móvel:

```
import { useRef, useEffect, useState } from "react";
import { shuffleArray } from "../utils";
```

```

import useAnimationFrame from "../Hooks/useAnimationFrame";
import styled from "styled-components";

const SCanvas = styled.canvas`
  transform: ${({ modifier }) =>
    `scale(${modifier.aspectRatio > 1 ? modifier.width : modifier.height})`;
`;

const VideoKolarCanvas = ({
  videoRef,
  screenH,
  screenW,
  areMonuments,
  videoConstraints,
}) => {
  const canvasRef = useRef(null);
  const [pieces, setPieces] = useState([]);
  const [settings, setSettings] = useState({});
  const [firstOrder, setFirstOrder] = useState([]);
  const [sizeModifier, setSizeModifier] = useState(1);

  useEffect(() => {
    // 1. initialize settings
    if (!canvasRef || !screenW || !screenH || !videoConstraints.width) return;
    const canvas = canvasRef.current;
    const { width: canvasW, height: canvasH, aspectRatio } = videoConstraints;
    canvas.width = canvasW;
    canvas.height = canvasH;
    aspectRatio > 1
      ? setSizeModifier({
          width: screenW / canvasW,
          height: screenH / canvasH,
          aspectRatio: innerWidth / screenW,
        })
      : setSizeModifier({
          width: screenW / canvasW,
          height: screenH / canvasH,
          aspectRatio: innerHeight / screenH,
        });

    const xRows = screenW < 768 ? 10 : 20;
    const pieceW = canvasW / xRows;
  });
};

```

```

const yRows = canvasH / pieceW;
const pieceH = canvasH / yRows;
const puzzleW = pieceW * xRows;
let piece,
    xPos = 0,
    yPos = 0;
const copyPieces = [];

// 2. create pieces
for (let i = 0; i < xRows * yRows; i++) {
    piece = {};
    piece.sx = xPos;
    piece.sy = yPos;
    copyPieces.push(piece);
    xPos += pieceW;
    if (xPos >= canvasW) {
        xPos = 0;
        yPos += pieceH;
    }
}
setFirstOrder(copyPieces);
setPieces(copyPieces);
setSettings({
    canvasW,
    canvasH,
    pieceW,
    pieceH,
    puzzleW,
});
}, []);

// 3. Kolarize
useEffect(() => {
    if (firstOrder.length && pieces.length) {
        areMonuments ? setPieces(shuffleArray(pieces)) : setPieces(firstOrder);
    }
}, [areMonuments]);

// 4. Draw in canvas
useAnimationFrame(() => grabFrame());
const grabFrame = () => {
    if (!canvasRef || !videoRef) return;

```

```

const videoInput = videoRef.current;
const canvas = canvasRef.current;
if (!canvas || !videoInput) return;
const ctx = canvas.getContext("2d");
const { pieceW, pieceH, puzzleW } = settings;
let piece = {},
    xPos = 0,
    yPos = 0;

for (let i = 0; i < pieces.length; i++) {
  piece = pieces[i];
  piece.xPos = xPos;
  piece.yPos = yPos;
  ctx.drawImage(
    videoInput,
    piece.sx,
    piece.sy,
    pieceW,
    pieceH,
    xPos,
    yPos,
    pieceW,
    pieceH
  );
  xPos += pieceW;
  if (xPos >= puzzleW) {
    xPos = 0;
    yPos += pieceH;
  }
}
};

return (
  <div
    className="absolute left-5 top-5 flex justify-center items-center
overflow-hidden"
    style={{ height: screenH - 10, width: screenW - 10 }}
  >
    <SCanvas className="-z-10" ref={canvasRef} modifier={sizeModifier} />
  </div>
);
};

```

```
export default VideoKolarCanvas;
```

v. Link do repositório Github com o código base da web-app *Colombot: Bring Your Favorite Colonialist to Life!*

<https://github.com/aldomedina/ArteColonialistaFazMalAVista>.

vi. Link da aplicação web *Colombot: Bring Your Favorite Colonialist to Life!*

<https://arte-colonialista-faz-mal-av-ista.vercel.app/>

