UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Identificação de Componentes Usualmente Presentes em Páginas Web

Alon Mota Lourenço

Tese Para Defesa De MBA em Inteligência Atificial e Big-Data

ICMC USP SÃO CARLOS

Alon Mota Lourenço

Identificação de Componentes Usualmente Presentes em Páginas Web

Monografia apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, para o Exame de Qualificação, como parte dos requisitos para obtenção do título de Pós-Graduado em Inteligência Atificial e Big-Data.

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Cristina Dutra de Aguiar

USP – São Carlos
Agosto de 2022

Alon Mota Lourenço

Detecting components that are usually present on web pages

Monograph submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, as part of the qualifying exam requisites of the of the Doctorate Program in Computer Science and Computational Mathematics.

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Cristina Dutra de Aguiar

USP – São Carlos
August 2022

Este trabalho é dedicado à todos os cientistas que
desbravam o desconhecido e incorporam o novo, mudando a vida de todos.
Em especial, ao pesquisadores do Instituto de Ciências Matemáticas e de Computação (ICMC).

AGRADECIMENTOS

Gostaria de agradecer ao Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo por todo o apoio e orientação oferecidos durante o MBA em inteligência artificial e big data. Agradecimentos especiais são direcionados para a orientadora deste projeto, Cristina Dutra de Aguiar, que sempre se mostrou atenta, educada e proativa para sanar quaisquer duvidas encontradas. Adicionalmente também aproveito para agradecer toda a comunidade acadêmica do nosso país por, apesar de todas as adversidades enfrentadas, ainda se mostrar forte e expandir os limites do nosso conhecimento.

Por último porém não menos importante gostaria de agradecer a minha família por ser sempre uma rede de apoio tão confiável e que me motiva a seguir adiante, com agradecimentos em especial direcionados a minha mãe Adriana de Jesus, que é a principal responsável pela base da minha educação e minha ambição para prosperar.

“As invenções são, sobretudo,
o resultado de um trabalho de teimoso.”
(Santos Dumont)

RESUMO

LOURENÇO, A. M. **Identificação de Componentes Usualmente Presentes em Páginas Web**. 2022. 85 p. Monografia (Pós-Graduação em Inteligência Atificial e Big-Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

A web consiste em um conjunto de documentos em hipertexto que são interligados, os quais também são chamados de páginas web. As páginas web e suas funcionalidades, ou componentes, impactam no funcionamento do mundo real. Por exemplo, as páginas que uma empresa disponibiliza bem como os componentes presentes nessas páginas podem ser a diferença entre o sucesso e o fracasso da empresa. Neste sentido, surge a necessidade de se identificar quais componentes uma página web deve englobar, de forma que esses componentes estejam presentes quando uma nova página web for projetada ou atualizada. Neste trabalho de conclusão de curso, aborda-se esse desafio. Para tanto, foi desenvolvida uma metodologia composta das seguintes etapas. Primeiramente, foi identificado um conjunto de páginas web que possuem muitos acessos. Depois, foi desenvolvido um *web-crawler* com o objetivo de obter o código fonte dessas páginas web. Na sequência, o código fonte de cada página foi transformado em uma imagem correspondente. As imagens geradas foram então fragmentadas e manipuladas por uma rede neural convolucional, a qual extraiu os vetores de características dos fragmentos e viabilizou o agrupamento dos fragmentos em diferentes grupos de acordo com o algoritmo *k-means*. Por fim, as características dos agrupamentos gerados foram analisadas. Os resultados obtidos possibilitaram a identificação de componentes usualmente encontrados em páginas web, como barras de pesquisa, menus verticais de navegação, lista de opções e tabelas, dentre outros.

Palavras-chave: Páginas web, componentes web, *web-crawler*, aprendizado de máquina não supervisionado.

ABSTRACT

LOURENÇO, A. M. **Detecting components that are usually present on web pages** . 2022. 85 p. Monografia (Pós-Graduação em Inteligência Atificial e Big-Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

The internet is an aggregation of hypertext documents that are connected. The web pages and their functionalities, a.k.a. components, have an impact in the real world. For example, the web pages a company makes available on the internet, alongside with their functionalities can make a difference between success and failure for that business. Therefore, there is a demand for identifying relevant components that should be considered when developing web page. This paper elaborates further into this necessity. In order to detect components, the following steps were taken. First, a research was done in order to identify the most accessed websites. Second, a web crawler was developed to retrieve the hypertext for the given pages. Then, the hypertext for each page was transformed into an image. Each image was used to create various fragments that were analyzed by a convolutional neural network that extracted the feature arrays from them. The feature arrays were then used to clusters using k-means algorithm. At last the clusters were analyzed and the results enabled identifying a few web components that are usually used on webpages, from witch we can name, search bars, vertical menus, tables and others.

Keywords: Web Pages, Web Components, Web-Crawler, unsupervised machine learning.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE ALGORITMOS

LISTA DE CÓDIGOS-FONTE

LISTA DE TABELAS

LISTA DE SÍMBOLOS

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

Neste capítulo é descrita a introdução deste trabalho de conclusão, cujo objetivo é identificar componentes que usualmente são encontrados em páginas web. Na seção 1.1 é feita a contextualização do trabalho. Na seção 1.2 é aprofundada a motivação para o desenvolvimento do mesmo. Na seção 1.3 são detalhados a questão de pesquisa, o objetivo e a metodologia desenvolvida para alcançar o objetivo proposto. Por fim, na seção 1.4 é descrita a estruturação da monografia.

1.1 Contextualização

A internet é fundamental para um grande número de pessoas e instituições ao redor do mundo, uma vez que ela possibilita a troca de dados de forma rápida e eficiente entre os membros conectados. Ela oferece suporte para a *World Wide Web* (WWW), amplamente conhecida como web. A web consiste em um conjunto de documentos em hipertexto que são interligados, os quais também são chamados de páginas web (MEDEIROS 2020). Esses documentos são usualmente exibidos usando-se um navegador, como Google Chrome e Safari. Quando várias páginas web são agrupadas, elas formam um sítio ("*site*").

A quantidade de páginas web disponíveis tem crescido continuamente desde a criação da web (Onan 2016). Adicionalmente, essas páginas e suas funcionalidades impactam no funcionamento do mundo real. Isso pode ser especialmente observado no âmbito institucional. Nesse âmbito, as páginas que uma empresa disponibiliza bem como as funcionalidades presentes nessas páginas podem ser a diferença entre o sucesso e o fracasso da empresa[1].

Visto que páginas web são um aglomerado de elementos de hipertexto e código (TheFreeDictionary 20 uma estratégia comum na atualidade para desenvolvê-las de forma mais eficiente consiste em separar os elementos de cada página em partes, ou módulos, autônomos o suficiente para

[1] Forbes article: <https://www.forbes.com/sites/theyec/2020/02/03/why-every-business-needs-a-website/?sh=3bdb2d006e75>, último acesso em 17/09/2022

operar isoladamente. Essa estratégia de modularização é conhecida como *Web Components* (Mozila 2021). Neste contexto, um componente, que também pode ser denotado por funcionalidade do inglês *feature*, especifica uma porção (ou seja, pedaço) de uma página, capaz de realizar alguma ação de forma independente. Neste trabalho, adota-se o termo *componente*, o qual é usado ao longo do texto. Um exemplo de componente é um botão de pesquisar. Outro exemplo é a listagem de imagens.

1.2 Justificativa e Motivação

Identificar quais componentes uma páginas web deve englobar é uma tarefa complexa (Onan 2016), principalmente frente ao aumento constante de novas páginas, aos componentes disponíveis e à defasagem de páginas ao longo dos anos. Uma possível solução para suprir a incerteza sobre o que disponibilizar, consiste em produzir páginas web com todos os componentes imagináveis de forma a não faltar nenhum. Entretanto, essa solução não é viável pois dispende de recursos exacerbados. Adicionalmente, sempre surgem inovações e novas necessidades.

Uma solução viável para essa tarefa é a automatização. Como elucidado em (Selamat 2004), a automação deve ser feita por meio da aplicação de técnicas de inteligência artificial capazes de monitorar a web e identificar os requisitos necessários para a criação das páginas e seus componentes. Técnicas utilizadas para esse fim consistem em técnicas de aprendizado de máquina. Essas técnicas podem ser classificadas como supervisionadas, quando as classificações desejadas são conhecidas e os dados usados no treinamento encontram-se rotulados, e não supervisionadas, quando o próprio classificador cria as classes e agrupa os dados de acordo com essas classes. Existe também o aprendizado de reforço, o qual visa treinar o comportamento por meio de mecanismos de recompensa ou penalização (Smola e Vishwanathan 2008).

No desenvolvimento do presente trabalho, são utilizadas técnicas de aprendizagem de máquina não supervisionadas. Dentre as técnicas existentes, são empregados a rede neural convolucional não supervisionada (Lakhani *et al.* 2018; Tian 2020; Jiao e Zhao 2019) e o algoritmo *k-means* (Li e Wu 2012). A escolha das redes neurais se deve ao fato de que elas são próximas à forma na qual os humanos aprendem. Adicionalmente, redes neurais convolucionais incluem abordagens para processamento de imagens que são fundamentais para a manipulação de páginas web. Com relação ao algoritmo *k-means*, ele é um algoritmo amplamente utilizado.

Na literatura, existem estudos que se aprofundam na classificação de páginas da web (Barforoush Hossein Shirazi 2017; Selamat 2004; Leea Wei-Chang Yehb 2015; Li *et al.* 2017; Onan 2016). Também existem estudos que visam criar uma base de dados de páginas web (4) e treinar uma rede neural para detectar padrões nas páginas web (Massaro *et al.* 2021). Entretanto, no melhor do nosso conhecimento e de acordo com as pesquisas feitas até o momento, ainda não existem estudos que investigam sobre a classificação dos componentes que compõem uma página. Entretanto, a classificação desses componentes é importante na identificação de quais

deles devem estar presentes em uma página. Assim, existe uma lacuna que deve ser preenchida com novas pesquisas, motivando o desenvolvimento do presente trabalho.

1.3 Questão de Pesquisa, Objetivo e Contribuições

Ser capaz de identificar quais componentes uma página web deve disponibilizar pode ajudar tanto empresas já consolidadas a identificar novas necessidades de seu público alvo, quanto novas empresas integrantes no mercado a decidir e priorizar componentes de forma a obter resultados melhores.

Para auxiliar a tomada de decisão contextualizada à web, neste trabalho visa-se responder à seguinte pergunta de pesquisa:

"Quais componentes usualmente podem ser encontrados em páginas web?"

A partir dessa pergunta, define-se o objetivo deste trabalho de conclusão, que consiste em identificar componentes que usualmente são encontrados em páginas web, e que devem ser utilizados como base para o desenvolvimento de novas páginas web.

Para atingir esse objetivo, foi desenvolvida uma metodologia composta das seguintes etapas. Primeiramente, foi identificado um conjunto de páginas web que possuem muitos acessos. Considerou-se que, por serem muito acessadas, essas páginas web possuem componentes de interesse. Depois, foi desenvolvido um *web-crawler* com o objetivo de obter o código fonte dessas páginas web. Na sequência, o código fonte de cada página foi transformado em uma imagem correspondente. As imagens geradas foram então fragmentadas e manipuladas por uma rede neural convolucional, a qual extraiu os vetores de características dos fragmentos e viabilizou o agrupamento dos fragmentos em diferentes grupos de acordo com o algoritmo *k-means*. Por fim, as características dos agrupamentos gerados foram analisadas para identificar os componentes usualmente encontrados em páginas web.

Espera-se que o trabalho desenvolvido possa oferecer apoio conceitual e embasar a tomada de decisão no desenvolvimento de páginas web com relação a quais componentes devem estar presentes nessas páginas.

1.4 Estruturação da Monografia

Além deste capítulo introdutório, esta monografia é dividida nos seguintes capítulos. O Capítulo 2 realiza uma revisão bibliográfica com a finalidade de explicar conceitos relacionados a *web-crawlers* e aprendizado de máquina, os quais são necessários para o entendimento do trabalho. Esse capítulo também resume trabalhos relacionados. O Capítulo 3 detalha os aspectos relacionados ao desenvolvimento do trabalho. Ou seja, são descritas todas as etapas referentes à metodologia desenvolvida para atingir os objetivos propostos. No Capítulo 4 são apresentados os

resultados obtidos e lista os principais componentes que devem estar presentes nas páginas web. O trabalho é finalizado no Capítulo 5, o qual descreve as conclusões e lista trabalhos futuros.

CAPÍTULO

2

FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são descritos os principais conceitos teóricos para formulação desta monografia. Na seção 2.1 são definidos aspectos relacionados aos *web-crawlers* e como estes podem ser utilizados para auxiliar a criação de uma base de dados. Na seção 2.2 é feita uma descrição sobre os conceitos gerais do aprendizado de máquina. Na seção 2.3 são resumidos conceitos de redes neurais artificiais, enquanto que na seção 2.3.1 são detalhados aspectos relacionados ao processamento de imagens usando redes neurais convolucionais. Nas seções 2.4 e 2.5 são resumidos outros conceitos usados no desenvolvimento do trabalho, que são a técnica de PCA e o algoritmo *k-means*, respectivamente. Na seção 2.6 são descritos os trabalhos relacionados. O capítulo é finalizado na seção 2.7 com as considerações finais.

2.1 *Web Crawlers*

A web é formada por um agregado de páginas escritas em HTML (*hipertext markdown language* ou, em português, linguagem de marcação de hipertexto), que podem ser programadas para realizar as mais diversas ações e que podem carregar informações relevantes sobre determinado assunto (TheFreeDictionary 2021). Recuperar informações relevantes desse universo é um desafio que possibilita a criação de bases de dados muito ricas. Esse desafio é dificultado por vários aspectos relacionados, como a necessidade de se descartar informações indesejadas e encontrar conteúdos relevantes. Uma das formas mais eficientes e mais estudadas para essa finalidade consiste na utilização de *web-crawlers*.

De maneira geral, um *web-crawler* é um programa que navega pela internet de forma autônoma, procura por conteúdos relevantes e salva esses conteúdos na máquina em que está sendo executado. *Web-crawlers* geralmente atuam da seguinte forma. Eles recebem o endereço de uma ou mais páginas na web, sobre as quais fazem uma varredura e extração de endereços contidos. O processo é então repetido para cada endereço encontrado até que uma condição de parada seja atingida (Bai *et al.* 2014). *Web-crawlers* são muito utilizados em mecanismos de

busca como Google, Yahoo e outros (Du e Li 2007/10).

De acordo com (Bal 2012), os *web-crawlers* podem ser classificados em 6 tipos:

1. *Focused Crawler*: Projetados pare recuperar documentos de um tópico específico.
2. *Collaborative Web Crawler*: Conjunto de *crawlers* que trabalham juntos para recuperar as informações de interesse, sendo cada *crawler* responsável por varrer um escopo específico da web.
3. *Incremental Crawler*: Executa continuamente, frequentemente revisita páginas e as atualiza. Muitas vezes também apaga páginas antigas da memória para liberar espaço para novas páginas.
4. *Parallel Crawler*: Constituído de um *crawler* que roda vários processos em paralelo, permitindo que o seu desempenho seja mais eficiente.
5. *Distributed Crawler*: Vários *crawlers* executando em localizações distintas, de forma que o tráfego gerado por eles seja distribuído da forma mais uniforme possível pela rede, evitando sobrecarregar os nós.
6. *Mobile Crawler*: Capaz de se transferir entre cada um dos servidores antes de coletar as informações desejadas.

Um *web-crawler* é uma ferramenta extremamente útil para a obtenção e criação de bases de dados. Entretanto, a sua má utilização pode gerar sobrecarga em vários servidores, expor dados sigilosos que não deveriam ser expostos, ou criar duplicatas de páginas confiáveis para fins maliciosos. Como resultado, muitos sítios atualmente implementam medidas para detecção e prevenção de *web-crawlers* em seus domínios. Assim, implementar um *web-crawler* que seja funcional e não maléfico é um aspecto importante a ser considerado (Ro e Im 2018).

Neste trabalho, utiliza-se um *focused crawler* com o objetivo de se obter o código fonte de páginas web que possuem muitos acessos.

2.2 Aprendizagem de máquina

Uma comparação utilizada por Janiesch (6) oferece subsídio para entender o cerne do aprendizado de máquina:

> É mais fácil ensinar para uma criança o que diferencia um carro esportivo de um carro normal mostrando exemplos de carros esportivos do que tentando formular regras explícitas que limitam um carro esportivo.

Seguindo a partir desta afirmação, o aprendizado de máquina é uma forma de ensinar às máquinas algum significado do nosso mundo por meio de exemplos, ao invés de programar essas características complexas manualmente.

Os avanços no aprendizado de máquina possibilitaram a criação de sistemas com capacidade cognitiva sobre-humana (6), capazes de tomar decisões com certo nível de precisão de forma rápida e consistente. Combinado com o grande aumento da quantidade de dados disponíveis, é muito provável que essa tecnologia se torne cada vez mais acessível e fundamental para o desenvolvimento (6).

O aprendizado de máquina pode ser classificado em três classes, dependendo do problema que se quer resolver e dos dados disponíveis para solucionar esse problema. Essas classes, denominadas aprendizagem supervisionada, aprendizagem não-supervisionada e aprendizagem por reforço (6), são descritas a seguir:

- Aprendizagem supervisionada: Utilizada quando os escopos de saída são bem definidos e existe um conjunto de dados representativo que já possui classificação. A tarefa da máquina nesse tipo de problema consiste em identificar as características que diferenciam cada conjunto de forma que seja possível atribuir uma classe aos novos elementos (Smola e Vishwanathan 2008).
- Aprendizagem não-supervisionada: Quando existem muitos dados, porém esses não estão previamente rotulados. O trabalho da máquina neste contexto consiste em identificar padrões existentes nos dados e agrupá-los com base em um conjunto de características (Smola e Vishwanathan 2008).
- Aprendizagem por reforço: Quando o estado atual do sistema influencia no resultado esperado da rede, sendo que o objetivo consiste em maximizar algum ganho ao invés de classificar um dado (Xu 2019).

Fazer essa distinção geralmente é interessante, pois muitas vezes problemas parecidos podem ser resolvidos de forma semelhante (Smola e Vishwanathan 2008). Atualmente, existe uma gama muito grande de problemas que podem ser resolvidos utilizando aprendizado de máquina, tais como classificação, regressão, estimação e recomendação (Smola e Vishwanathan 2008). Adicionalmente, para cada um desses problemas, existem diversas variações de algoritmos com especificações diferentes, tais como regressão linear, métodos Bayesianos, árvores de decisão e redes neurais artificiais (6). Portanto, a classificação correta ajuda a identificar as similaridades.

Neste trabalho, empregam-se técnicas de aprendizagem de máquina. Dentre as técnicas existentes, utilizam-se as redes neurais.

2.3 Redes Neurais Artificiais e Profundas

Redes neurais artificiais são particularmente interessantes pois são baseadas em modelos de processamento de informações biológicos, o que faz com que tenham bastante flexibilidade e possam ser adaptadas para atuar em diversos cenários (6). Elas consistem de diversas unidades de processamento, os neurônios, os quais são conectados por meio de pesos. Ao receber um sinal, os neurônios podem intensificá-lo ou reduzí-lo a fim de determinar um resultado. Os pesos

de uma rede neural artificial são continuamente ajustados no processo de treinamento, até que todo o sistema responda aos impulsos de acordo com o esperado (6).

Vale ressaltar que, nessas redes de propagação direta, os sinais captados pelos neurônios tende a seguir sempre em uma única direção, na qual existe: (i) uma camada de entrada, que recebe o sinal; (ii) uma quantia não negativa de camadas intermediárias/ocultas que amplificam as capacidades de aprendizado da rede; e (iii) uma camada de saída, que determina o resultado (6). Por exemplo, a entrada pode ser um conjunto de imagens e a saída pode ser um valor binário indicando se a imagem possui ou não pessoas.

Muitos dos avanços significativos observados na ultima década na área das redes neurais artificiais foi dado no âmbito das redes neurais profundas. Redes neurais profundas diferem de outras redes neurais por geralmente possuírem mais camadas ocultas e pela utilização de neurônios com funções de ativações mais complexas. Essas características melhoram significativamente a capacidade de aprendizado e a versatilidade das redes neurais profundas. O conceito de redes neurais profundas também é conhecido por aprendizagem profunda ou *deep learning* (6).

Devido à dependência de grandes bases de dados para serem treinadas, redes neurais profundas são melhor aplicadas quando utilizadas em aplicações de *big data* ou aplicações com dados complexos como imagem, áudio e vídeo. Essas redes geralmente possuem um desempenho superior ao das redes neurais tradicionais, desde que são muito eficientes em extrair características dos dados analisados e, dessa forma, se adaptam melhor a cada contexto. Uma característica consiste, em geral, de uma propriedade discriminatória adequada para representar determinados contextos desejados, e pode ser obtida por meio dos dados existentes (6).

Atualmente, devido a uma combinação de circunstâncias, o aprendizado de máquina voltado ao processamento de imagens se tornou muito popular em diversos cenários, tais como mídias sociais, imagens médicas, de satélite e de trânsito. Esse é outro fator que contribui para a popularização e aperfeiçoamento de redes neurais profundas (Jiao e Zhao 2019). Adicionalmente, devido à crescente necessidade de processar imagens, uma técnica de rede neural profunda em particular ganhou bastante visibilidade: a rede neural convolucional. Essa técnica é utilizada neste trabalho.

2.3.1 Redes Neurais Convolucionais

A evolução das redes neurais profundas no que tange ao processamento de imagens se deve em grande parte às redes neurais convolucionais. Essas redes foram utilizadas pelos Drs. Krizhevsky and Hinton para ganhar o desafio de reconhecimento de imagens da ImageNet em 2012, que é uma competição internacional de classificação de imagens (Lakhani *et al.* 2018).

A grande vantagem que as redes convolucionais introduzem no processamento de imagens é a utilização de pesos compartilhados. Isso reduz significativamente o número de parâ-

metros, facilitando a aprendizagem e possibilitando a elaboração de redes mais poderosas. O compartilhamento de pesos funciona calculando um valor para cada *pixel* baseado em alguma operação envolvendo um filtro de valores e um conjunto de *pixeis* ao redor dele (Tian 2020).

Redes neurais convolucionais consistem basicamente de três tipos de camadas principais: camadas de convolução, camadas de agrupamento (ou *pooling*) e camada totalmente conectada (Jiao e Zhao 2019), conforme descrito a seguir.

- Camadas de convolução: São o núcleo das redes neurais convolucionais. Nelas, filtros bidimensionais de tamanhos genéricos são aplicados à camada de entrada usando o princípio do compartilhamento de pesos. Cada filtro gera uma nova imagem, na qual alguma informação acerca da entrada é intensificada, enquanto ruídos e redundâncias são amenizados. Ou seja, as imagens resultantes de cada filtro expressam de maneira mais objetiva uma determinada característica da imagem e, por isso, também são chamadas de mapas de características (Tian 2020).
- Camadas de agrupamento: São uma forma de reduzir o tamanho das amostras obtidas na rede, na qual uma imagem é secionada em blocos de tamanhos ajustáveis (2x2, 3x3, 5x2) e é gerada uma nova imagem na qual cada bloco é representado por apenas um valor. Existem várias formas de se selecionar o valor representativo, sendo que, dentre elas, se destacam: selecionar o maior valor, selecionar o valor médio e selecionar um valor aleatório. Camadas de agrupamento são extremamente necessárias devido à seguinte explicação. Uma vez que cada imagem gera inúmeros mapas de características de tamanho semelhante, criar redes convolucionais de várias camadas seria praticamente inviável se não fosse pelo uso das camadas de agrupamento (Tian 2020).
- Camada completamente conectada: A camada completamente conectada geralmente opera no final das redes neurais convolucionais, encarregadas de determinar os resultados. Essa camada é equivalente a uma rede neural artificial convencional, que usa neurônios para processar listas de entrada. Para transformar um conjunto de imagens em uma lista de valores, a camada anterior à camada completamente conectada transforma as imagens resultantes em valores únicos utilizando alguma técnica como agrupamento ou convolução (Tian 2020).

Neste trabalho, é utilizado o modelo EfficientNetB1 (Tan e Le 2019), disponibilizado pela biblioteca Keras[1]. Esse modelo tem como diferencial ser uma arquitetura de rede neural convolucional que permite o redimensionamento uniforme das dimensões de profundidade, largura e resolução usando coeficientes compostos (Tan e Le 2019). Esse redimensionamento uniforme faz com que o modelo de rede neural seja muito versátil, podendo se adaptar a diversos contextos computacionais (Tan e Le 2019). Em especial, a escolha do modelo EfficientNetB1 foi feita motivada pela possibilidade de se criar uma rede neural pequena adaptada aos recursos computacionais disponíveis para execução do trabalho.

[1] Disponível em <https://keras.io>, último acesso em 23/05/2022

Este trabalho utiliza uma rede neural convolucional EfficientNetB1, previamente treinada, para gerar vetores de características sobre as imagens das páginas web coletadas. A dimensionalidade dos vetores gerados é reduzida aplicando-se a técnica de PCA.

2.4 Técnica PCA

A técnica PCA (*Principal Component Analysis*) ou, em português, análise do componente principal, é uma técnica utilizada para reduzir a dimensionalidade de algum dado de interesse, facilitando a interpretação, o processamento e, ao mesmo tempo, minimizando a perda de informações (10). A redução de dimensionalidade é feita criando novas variáveis não correlacionadas sucessivamente de forma a maximizar a variância.

Neste trabalho, a técnica de PCA é aplicada sobre a saída produzida pela rede neural convolucional. Como resultado, são reduzidos a dimensionalidade dos vetores de características e o custo de processamento. Na sequência, os vetores resultantes são agrupados usando-se o algoritmo *k-means*.

2.5 Algoritmo K-Means

O algoritmo *k-means* é um dos algoritmos de aprendizagem não supervisionada mais antigos, que consiste em buscar k agrupamentos em um conjunto de dados (Li e Wu 2012). Suas principais vantagens são a eficiência e a rapidez no cálculo dos agrupamentos.

O algoritmo funciona da seguinte forma. Primeiro, ele escolhe aleatoriamente k elementos entre os dados disponíveis para serem o centro dos grupos. Em seguida, ele efetua diversas iterações para refinar o posicionamento ideal dos centros de cada grupo. O algoritmo termina quando um determinado numero de iterações é efetuado ou os centros se estabilizam.

Neste trabalho, a aplicação do algoritmo *k-means* possibilitou a identificação dos componentes usualmente presentes em páginas web.

2.6 Trabalhos Relacionados

Na literatura, existem estudos que se aprofundam na classificação de páginas da web (Barforoush Hossein Shirazi 2017; Selamat 2004; Leea Wei-Chang Yehb 2015; Li *et al.* 2017; Onan 2016). Como esses estudos não visam identificar quais componentes devem estar presentes em páginas web, eles não são descritos em mais detalhes.

Portanto, dividindo o escopo deste trabalho entre criar uma base de dados de páginas web utilizando um *web-crawler* e empregar uma rede neural convolucinal, foram encontrados os trabalhos relacionados descritos a seguir.

Com relação à criação da base de dados, em Dhanith (4) é proposta uma forma de construção de *focused crawler* de seis camadas, na qual o *web-crawler* é inicializado com o endereço de uma página e um tópico de interesse. Na primeira camada é feito um processamento textual sobre o tópico para que ele seja processado mais rapidamente. Na segunda camada, o *web-crawler* é realizado por meio da navegação entre páginas e da obtenção das páginas encontradas. O conteúdo baixado é encaminhado para uma terceira camada que extrai os textos do HTML (Linguagem de Marcação de Hipertexto). Os textos extraídos são passados para a quarta camada, a qual os processa e os otimiza. A quinta camada então compara os textos obtidos com o tópico informado. Por fim, a sexta camada calcula a relevância do texto baixado com o tópico informado e, caso a página seja relevante, reinicia o processo.

Referente ao treinamento da rede neural para detectar padrões na web, em **(author?)** (Massaro *et al.* 2021) é proposta uma abordagem para ranqueamento de interfaces visuais utilizando redes neurais artificiais e memória de curto e longo prazo. No trabalho em questão são considerados diversos parâmetros para determinar a pontuação para cada página, como tempo de navegação, quantidade de cliques e movimentação do *mouse*. Como resultado, é possível estimar uma pontuação para uma página e intuir o quanto as características aplicadas nesta página satisfazem a um padrão pré-selecionado.

Considerando os estudos descritos anteriormente com os objetivos deste trabalho, propõe-se: (i) a criação de uma base de dados por meio do uso de um *focused web-crawler*, sendo que a base de dados contém as imagens dessas páginas web; e (ii) o uso de uma rede neural convolucional para gerar vetores de características das imagens da base de dados.

2.7 Considerações Finais

Neste capítulo foi descrito o conceito de *web crawler*, que consiste de um robô capaz de navegar a web de forma autônoma e salvar conteúdos encontrados. Associado ao fato de que a web cresce a cada dia e é cada vez mais utilizada, *web crawlers* consistem de uma solução muito relevante na obtenção de bases de dados ricas e diversas.

Neste capítulo também foram detalhados conceitos relacionados ao aprendizado de máquina. Dentro deste contexto, foi destacado o conceito de redes neurais, que possuem modelos de aprendizado baseados nos sistemas biológicos, isto é, utilizam um conceito similar às células nervosas dos serem humanos, o neurônio, e por isso são muito versáteis, se adaptando a diversos contextos. Foi visto também que, dependendo da quantidade de camadas intermediárias e das funções de ativação dos neurônios, redes neurais podem ser consideradas redes neurais profundas, conceito que passou a ser conhecido como *deep learning*.

Adicionalmente, foi destacado o fato de que as redes neurais convolucionais representam um modelo de aprendizado profundo adequado para tratar imagens. Esse tipo de rede neural utiliza do princípio de compartilhamento de pesos e de filtros para considerar relações/redun-

dâncias entre *pixeis* e seus vizinhos. Isso diminui o número de parâmetros necessários para o treinamento, permitindo treinar redes mais rápido e melhor.

Também foram descritos conceitos relacionados à técnica PCA e ao algoritmo *k-means*, os quais possuem como objetivo a redução de dimensionalidade e a geração de agrupamentos, respectivamente.

Finalmente, foram resumidos estudos relacionados ao presente trabalho e foram salientados os diferenciais do trabalho desenvolvido.

CAPÍTULO

3

DETALHAMENTO DO TRABALHO DESENVOLVIDO

Neste capítulo é feita uma descrição da metodologia empregada para atingir o objetivo proposto. Na seção 3.1 o escopo do trabalho é definido, englobando aspectos de tomada de decisão e da estratégia adotada para o desenvolvimento. De forma geral, a metodologia é composta das seguintes etapas. Primeiramente, é identificado um conjunto de páginas web que possuem muitos acessos. Essas páginas web são então processadas usando um *web-crawler* que tem como objetivo obter seus códigos fontes. Na sequências, os códigos fontes são usados para gerar imagens visuais das páginas web coletadas. A criação da base de dados de páginas web é discutida na seção 3.2. Já o processamento dessas páginas é detalhado na seção 3.3. No processamento das imagens, primeiramente foi feita a fragmentação dessas imagens. Os fragmentos gerados foram processados usando uma rede neural convolucional, a qual extraiu os vetores de características dos fragmentos e possibilitou a aplicação da técnica de PCA, com objetivo de reduzir a dimensionalidade dos dados, e acelerando o agrupamento dos fragmentos usando o algoritmo *k-means*. O capítulo é finalizado na seção 3.4, com as considerações finais.

3.1 Contextualização

Considerando o objetivo deste trabalho, que consiste em identificar componentes que usualmente são encontrados em páginas web, foi idealizada uma abordagem exploratória que captura várias páginas web de sítios variados e trabalha com elas de modo a identificar componentes comuns nessas páginas.

O termo *abordagem exploratória* está relacionado ao fato de que o trabalho desenvolvido tem como finalidade identificar componentes comuns, ou seja, componentes que usualmente estão presentes em páginas web. Entretanto, não existem pretensões de se esgotar todos os escopos possíveis de agrupamento ou de extração de características das páginas analisadas.

Adicionalmente, considera-se neste trabalho que a imagem visual de uma página web pode conter mais informações acerca da página propriamente dita que de seus códigos fonte. Isso está relacionado ao fato de que páginas web podem ser descritas de maneiras distintas utilizando HTML, e ainda assim apresentar a mesma imagem visual final. Em outras palavras, diferentes páginas web podem possuir a mesma aparência, ou seja, o mesmo *layout*, porém podem ter sido projetadas utilizando códigos fonte diferentes.

Também considera-se neste trabalho é comum observar tendências nas aparências das páginas web. Isso é decorrente do fato de que a aparência das páginas é projetada para usuários e é de interesse comum que esses usuários se sintam familiarizados ao utilizá-las. Por exemplo, é possível observar que componentes como opções de navegação e fotos, dentre outros, são muito similares em diferentes páginas web.

Como resultado das discussões realizadas nesta seção, o trabalho desenvolvido é baseado na aparência das páginas web, ou seja, é baseado nas imagens dessas páginas.

3.2 Criação da Base de Dados de Páginas Web

Para a criação da base de dados de páginas web, considerou-se a obtenção de páginas que possuem muitos acessos. Neste sentido, considerou-se que, por serem muito acessadas, essas páginas web possuem componentes de interesse, os quais podem ser identificados para atingir o objetivo do trabalho.

Um primeiro desafio enfrentado para a obtenção das páginas web mais acessadas foi descobrir quais são, de fato, as páginas que devem ser consideradas. Uma resposta a esse questionamento foi encontrada por meio do sítio da empresa DataForSEO[1]. Esse sítio disponibiliza publicamente uma base de dados contendo URLs de diversos sítios e suas respectivas quantidades de acesso. O sítio da empresa DataForSEO também oferece suporte para algumas possibilidades relacionadas à filtragem dos dados. Utilizando um filtro para selecionar os sítios mais acessados no mundo, foi obtida uma lista dos 1.000 sítios mais acessados, bem como a quantidade de visualizações de cada um deles no período de análise.

Uma vez identificados e recuperados os 1.000 sítios, o segundo desafio consistiu em identificar quais páginas estavam presentes em cada sítio e em associar cada página à sua respectiva imagem. Para resolver esse desafio, foram realizadas as seguintes atividades: (i) estudo de diferentes formas de se coletar o projeto de uma página como imagem (seção 3.2.1); e (ii) desenvolvimento de uma funcionalidade para percorrer as páginas dos sítios da lista para obter as imagens do projeto de cada uma delas (seção 3.2.2).

[1] Disponível em <https://dataforseo.com>, último acesso em 06/05/2022

3.2.1 Capturando o projeto de páginas web como imagens

A captura de uma página web como imagem pode ser feita por meio da impressão da tela do computador. Porém, esse processo possui falhas quando existem páginas extensas que não cabem por completo na tela e que, portanto, necessitam ser particionadas em várias impressões. Neste cenário, o processo torna-se trabalhoso e sem escalabilidade, principalmente quando se tem o objetivo de analisar várias páginas web. Portanto, a possibilidade de coletar a imagem de uma página web por meio de sua impressão foi descartada.

Outra abordagem estudada foi a utilização de ferramentas de terceiros para lidar com a automação dessa tarefa, tais como *RoboTask*[2] ou *WinAutomation*[3]. Entretanto, essas ferramentas apresentam dificuldade em lidar com as variabilidades do processo de captura. Por exemplo, pode-se citar as diferenças nos tamanhos das páginas e a quantidade de impressões necessárias para capturar páginas extensas que não cabem por completo na tela do computador. Outro aspecto limitante do uso dessas ferramentas se refere ao fato de que funcionalidades mais complexas são disponíveis apenas em versões pagas. Como resultado dessas dificuldades, o uso de ferramentas de terceiros para coletar a imagem de uma página web por meio de sua impressão foi descartado.

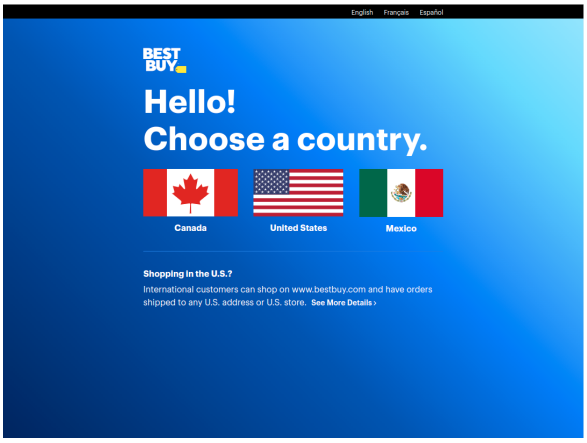
A última opção analisada foi a de gerar imagens para as páginas a partir do código fonte das mesmas. Para tanto, foi utilizada uma biblioteca em *javascript* chamada puppeteer[4]. Essa biblioteca é capaz de simular uma variedade de funcionalidades de um navegador web por meio de programas específicos, sendo uma dessas funcionalidades a capacidade de imprimir a imagem da página a partir de seu código fonte.

Foram realizados vários testes com a biblioteca selecionada. Verificou-se que, em muitos casos, os resultados se demonstraram muito bons, indicando que a imagem gerada pela biblioteca é igual ou muito semelhante à imagem da página original, como ilustrado na Figura 1. Em outros casos, houve discrepâncias com relação ao resultado esperado, ou seja, a imagem gerada pela biblioteca ficou diferente da imagem original, conforme ilustrado na Figura 2.

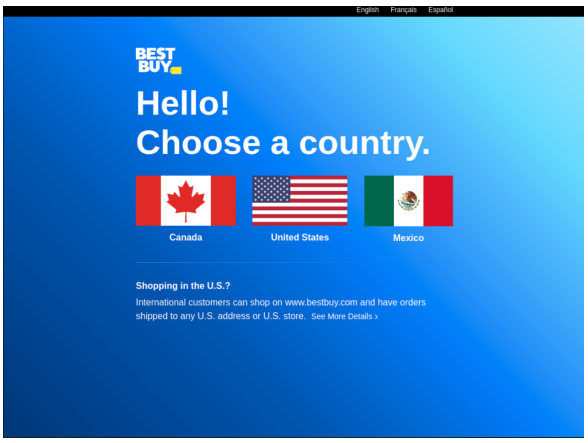
Apesar das diferenças existentes entre a página original e a imagem gerada pela biblioteca, é possível observar que as páginas ilustradas nas Figuras 2a e 2b possuem um *layout* muito semelhante. Por exemplo: (i) no topo das páginas, existe um cabeçalho com várias opções; (ii) depois, é exibido um local no qual os usuários podem digitar *strings* de busca e solicitar a execução da busca; e (iii) existe um texto escrito sobre a imagem e logo abaixo dela, com opções de navegação.

Portanto, dentre as opções estudadas para recuperar páginas como imagens, a geração de imagens a partir do código fonte foi a abordagem escolhida para dar seguimento ao trabalho. E, apesar dessa metodologia implicar na geração de algumas imagens diferentes das esperadas, ela ainda era a ferramenta que melhor atendia às necessidades do projeto. De fato, resultados muito

[2] Disponível em <https://robotask.com>, último acesso em 06/05/2022
[3] Disponível em <https://www.winautomation.com>, último acesso em 08/05/2022
[4] Disponível em <https://github.com/puppeteer/puppeteer>, último acesso em 09/05/2022

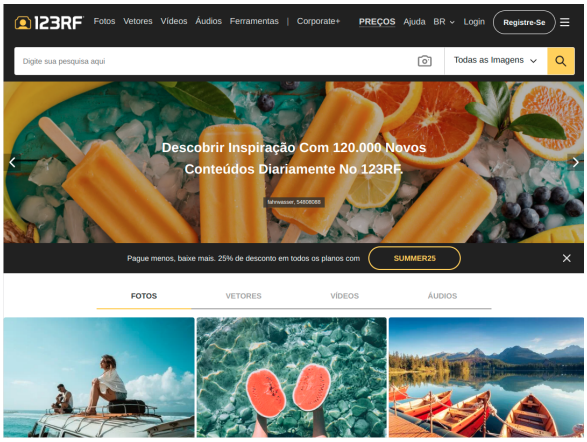


(a) Imagem da página web original (*bestbuy.com)*

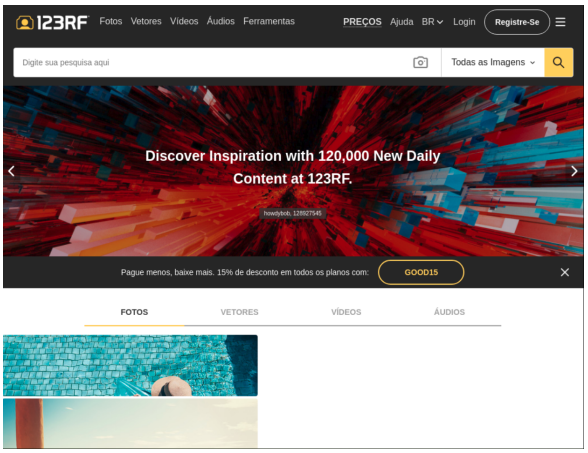


(b) Imagem gerada pela ferramenta

Figura 1 – Exemplo de imagem gerada corretamente pela ferramenta.



(a) Site *123rf.com* original



(b) Imagem gerada para *123rf.com*

Figura 2 – Exemplo de imagem gerada com discrepância pela ferramenta.

discrepantes podem ser descartados antes da análise. Adicionalmente, em resultados levemente diferentes, ainda é possível identificar alguns dos componentes originais da página.

A geração de imagens para as páginas web a partir do código fonte das mesmas introduz duas vantagens. A primeira se refere à facilidade de automatizar a obtenção de códigos fonte utilizando *web crawlers*. A segunda vantagem diz respeito à facilidade em replicar o experimento em diferentes máquinas, uma vez que é possível delimitar dimensões como o tamanho da tela ao se gerar as imagens.

3.2.2 Percorrendo páginas para obtenção das imagens

Visto que a abordagem selecionada para capturar o projeto de páginas web como imagem consiste na geração da mesma a partir do código fonte, o processo de recuperação das imagens foi dividido em duas etapas. Na primeira etapa, é realizada a interação sobre as páginas de cada

um dos sítios mais acessados, sendo salvo o código fonte dessas páginas. Essa etapa é descrita na seção 3.2.2.1. Na segunda etapa, detalhada na seção 3.2.2.2, os códigos fonte recuperados são transformados em imagens.

3.2.2.1 Percorrendo páginas para obtenção dos códigos fonte

Foi desenvolvido um *web crawler* utilizando node.js para percorrer as páginas dos sítios mais acessados com o intuito de obter os seus respectivos códigos fonte. Para cada sítio da lista, o *web crawler* coletou dados referentes ao domínio principal (a própria página) e a um relacionamento adicional (outra página do mesmo sítio referenciada no domínio principal), contabilizando assim a coleta de duas páginas. Foi considerado apenas um relacionamento adicional devido à explosão de relacionamentos presentes em cada página, o que inviabilizaria o desenvolvimento do trabalho devido à necessidade de recursos computacionais apropriados.

No desenvolvimento do *web crawler*, foram encontradas duas situações que motivaram o descarte de páginas. A primeira delas refere-se ao fato de que algumas páginas possuem mecanismos para prevenção contra *crawlers*, causando travamentos na execução do *crawler* desenvolvido. Para resolver esse desafio, temporizadores foram ajustados para continuar a execução do *crawler* em casos de travamento, desconsiderando a página que estava causando o problema.

A segunda situação é relacionada ao fato de que algumas páginas possuem mecanismos de carregamento. Como resultado, apenas uma porção do código fonte era transmitida, a qual era tratada pelo *crawler* como se fosse uma página completa, causando a recuperação de resultados indesejados. Para solucionar esse desafio, foi primeiramente testada uma abordagem de se esperar um período para verificar se algum conteúdo adicional da página era enviado. Porém, essa abordagem não obteve muito sucesso, visto que o tempo de execução do *crawler* cresceu drasticamente. A solução final adotada foi ignorar os casos nos quais esse comportamento ocorreu.

Ao final do processo, foram recuperadas em torno de 1.500 páginas, as quais foram utilizadas como base para o desenvolvimento do trabalho.

3.2.2.2 Transformando os códigos fonte em imagens

Os códigos fontes das páginas recuperadas foram salvos no banco de dados NoSQL MongoDb[5]. Foi desenvolvido um sistema que recuperou esses códigos e gerou suas respectivas imagens. Para a geração, foi considerada uma largura de 1792 *pixeis* para todas as páginas, com a altura se estendendo até o tamanho necessário.

Conforme destacado na seção 3.2.1, pode ocorrer discrepância entre a imagem da página original e a imagem gerada. Para tanto, foi feita uma limpeza manual das imagens coletadas ao

[5] Disponível em <https://www.mongodb.com>, último acesso em 16/05/2022

final do processo. Quando havia muita discrepância entre as imagens, a imagem era removida. Adicionalmente, também foram removidas todas as imagens vazias e todas as imagens geradas sem estilo ou formatação. No final do processo de limpeza, o numero total de imagens geradas na base de dados foi de aproximadamente 1.200, as quais representavam cerca de 650 dos domínios analisados.

As imagens geradas e que não foram descartadas foram salvas no formato png. Para melhor identificação, o título que cada imagem recebeu foi o nome do sítio ao qual a imagem pertence. Por exemplo, para o sítio https://google.com, a imagem gerada foi nomeada como google.com.png.

3.3 Processamento das Imagens das Páginas

O processamento das imagens das páginas que foram coletadas é dividido em duas atividades entrelaçadas: (i) a fragmentação das páginas, descrita na seção 3.3.1; e (ii) o agrupamento dos fragmentos, descrito na seção 3.3.2.

3.3.1 Fragmentação das páginas

Para serem processadas, as imagens foram divididas em fragmentos menores e de tamanhos iguais, considerando também o fato de que os fragmentos resultantes deviam conter informações acerca dos componentes da página. Definir uma abordagem coerente para fazer essa fragmentação foi um dos maiores desafios desse trabalho, uma vez que os componentes comuns variam em tamanho ou forma e que o algoritmo utilizado para o agrupamento necessita de imagens com dimensões idênticas.

A abordagem utilizada na fragmentação foi repartir cada imagem em quadrados menores. Cada imagem foi fragmentada três vezes. Em detalhes, uma mesma imagem foi fragmentada primeiro usando a resolução de 224 x 224, depois a resolução de 448 x 448, e depois a resolução de 896 x 896. Nas Figuras 3 e 4, tem-se dois exemplos de fragmentos de diferentes tamanhos. Nessas duas figuras, a mesma imagem é fragmentada, porém gerando fragmentos menores (Figura 3) ou fragmentos maiores (Figura 4).

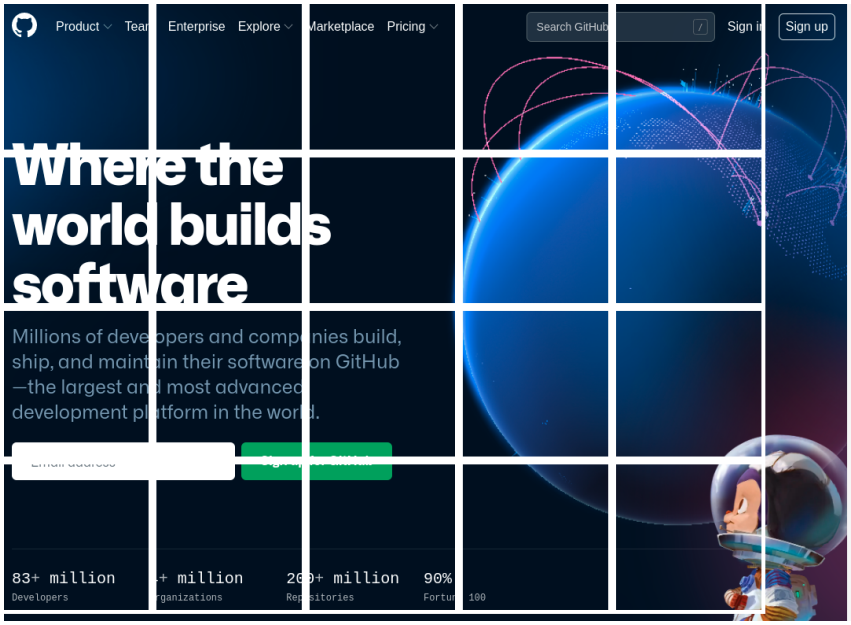


Figura 3 – Fragmentando em blocos menores

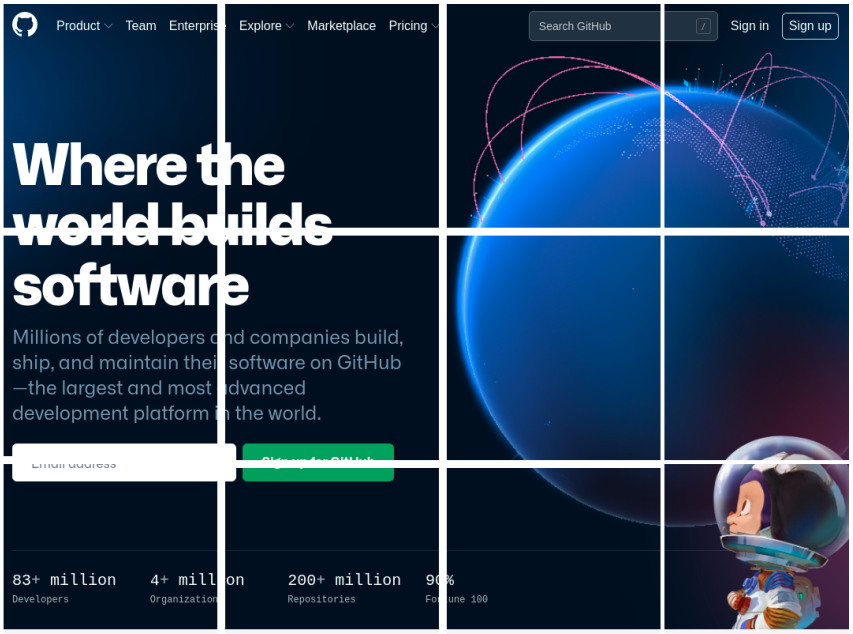


Figura 4 – Fragmentando em blocos intermediários

Ao final do processo, todos os fragmentos resultantes foram redimensionados para um tamanho de 200 x 200, como ilustrado na Figura 5. Desta forma, cria-se a probabilidade de se detectar componentes semelhantes independente das dimensões dos fragmentos, viabilizando assim que surjam grupos nos quais esses componentes estejam presentes. Adicionalmente, garante-se que todas as imagens que são consideradas como entrada para o algoritmo sejam uniformes em relação às suas dimensões.

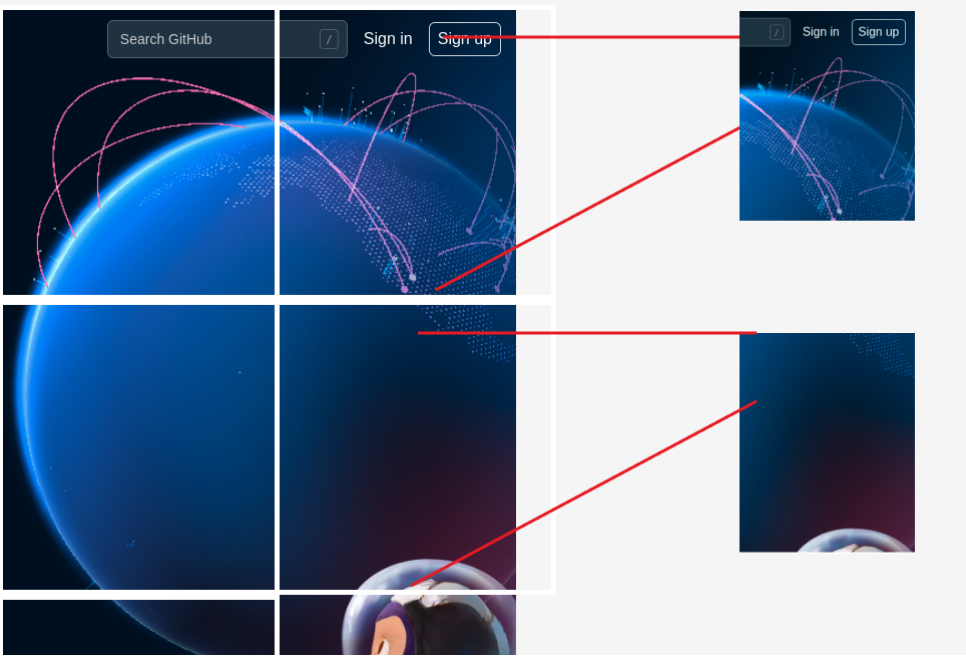


Figura 5 – Redimensionando blocos de tamanhos distintos

3.3.2 Agrupamento dos Fragmentos

Para realizar o agrupamento, foi utilizada a rede neural convolucional EfficientNetB1. A rede foi configurada com uma camada de saída contendo 1.000 nós, ou seja, um vetor com 1000 posições, cada qual representando numericamente alguma característica da imagem, determinada pelos valores pre-treinados da rede, sendo que o valor 1000 foi escolhido por ser a quantidade de classes padrão para este algorítimo e também por ser a quantidade de classes existentes no ImageNet [6]. No contexto desse trabalho, o vetor criado pela rede neural é utilizado como um vetor de características. Um fato relevante é que, mesmo sem um *finetuning* da rede neural em questão e utilizando os valores pré-treinados, quaisquer duas imagens que sejam semelhantes tendem a produzir vetores de características semelhantes ao serem processados pela rede, essa particularidade viabiliza a utilização desse algorítimo para extrair características dos fragmentos obtidos nesse trabalho com intuito de agrupá-los.

Uma vez extraídos os vetores de características das imagens, foi aplicada a técnica PCA para redução de dimensionalidade. O agrupamento das imagens semelhantes foi feito utilizando o algoritmo *k-means*, no qual os grupos são formados considerando os vetores cujas distâncias são as menores possíveis. Uma das características desse algoritmo é a necessidade de delimitar quantos grupos devem ser formados.

Para determinar essa quantidade, foi feita uma exploração sobre a proximidade dos vetores em cada grupo em contraste com a quantidade de grupos formados. O gráfico gerado é

[6] Documentação da biblioteca keras: <https://www.tensorflow.org/api_docs/python/tf/keras/applications/efficientnet/EfficientNetB1>, Último acesso em: 20/09/2022

ilustrado na Figura 6. Com base neste resultado, foi adotada uma quantidade de grupos igual a 50, por estar em um equilíbrio entre distância e número de grupos.

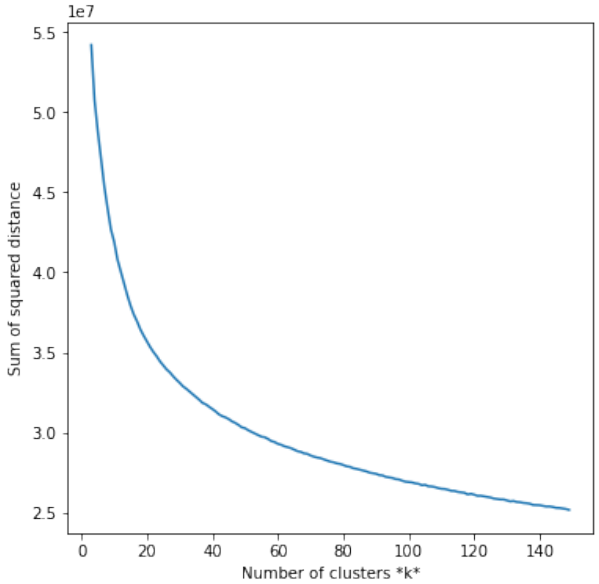


Figura 6 – Gráfico das distâncias entre membros de um grupo pela quantidade de grupos

3.4 Considerações Finais

Neste capítulo foram descritas as atividades realizadas durante o desenvolvimento do trabalho. Inicialmente, foi destacado que a base de dados deveria conter imagens das páginas web mais acessadas, as quais deveriam ser exploradas para a identificação de componentes comuns.

Em seguida, foi descrito que os sítios mais populares foram obtidos usando a plataforma DataForSEO. Também foi detalhado o processo realizado para a obtenção das imagens das páginas vinculadas a esses sítios. Neste processo, foram desenvolvidos um *web crawler* para recuperar o código fonte das páginas e um programa para converter os códigos fonte em imagens png. Ao todo, foram geradas cerca de 1.200 imagens.

Por fim, foi explicado que cada imagem foi fragmentada em imagens menores, e que os fragmentos foram processados pela rede neural convolucional EfficientNetB1. O resultado produzido pela rede neural foi processado usando-se o algoritmo *k-means*. Foram gerados 50 grupos diferentes, os quais são detalhados e analisados no próximo capítulo.

CAPÍTULO

4

RESULTADOS E ANÁLISE

Neste capítulo são mostrados e discutidos os resultados obtidos com o desenvolvimento do trabalho. Na seção 4.1 são detalhados aspectos relacionados aos grupos gerados com a aplicação do algoritmo de agrupamento, bem como alguns exemplos dos agrupamentos gerados. A partir desses agrupamentos, na seção 4.2 são identificados os componentes que usualmente aparecem em páginas web. O capítulo é finalizado na seção 4.3 com as considerações finais.

4.1 Resultados

Conforme detalhado no Capítulo 3, as imagens referentes às páginas web que foram coletadas passaram por um processo de fragmentação. Os fragmentos foram então processados por um algoritmo de extração de características para gerar um vetor de características para cada fragmento. Por fim, os vetores de características foram processados para gerar grupos de imagens semelhantes com base no conteúdo delas.

Foram gerados 50 grupos no total. Na Tabela 1 são mostrados os grupos gerados, bem como suas principais características. Nas Figuras 7 a 12 são ilustrados fragmentos referentes a seis diferentes grupos, de forma que a similaridade entre os componentes presentes nesses fragmentos possa ser observada. Os 50 grupos gerados são ilustrados no Apêndice A.

Tabela 1 – Grupos Gerados e suas Características

Grupo	Principais Característica do Grupo
1	Imagens com fundos de cores variadas com textos em alto contraste por cima.
2	Imagens completamente brancas ou claras.
3	Imagens de fundo claro e uma cor de auto contraste por cima.
4	Imagens em tons amarelados.
5	Imagens contendo quadrados e contrastes entre cores.

Tabela 1 – Grupos Gerados e suas Características

Grupo	Principais Característica do Grupo
6	Imagens em tons azuis e rosa, e poucas letras.
7	Imagens brancas com linhas horizontais bem delineadas.
8	Imagens com linhas verticais fortes com troca de cores.
9	Imagens possuindo elementos com quinas arredondadas.
10	Imagens que continham faces de pessoas.
11	Imagens em tons azuis escuros.
12	Imagens com listas e bandeiras.
13	Imagens com listas e menus de navegação verticais.
14	Imagens claras contendo círculos.
15	Imagens com quadrados e com diversidade de cores.
16	Imagens com fragmentos de fotos e textos.
17	Imagens com linhas horizontais fortes e variedade de cores.
18	Imagens escuras com textos claros e pequenos.
19	Imagens claras com poucos textos e linhas verticais fortes.
20	Imagens carregadas com muito texto em letras pequenas.
21	Imagens com quinas, poucos textos e grandes áreas em branco.
22	Imagens contendo desenhos e áreas em branco.
23	Imagens contendo fotos diversas.
24	Imagens contendo fotos de mulheres.
25	Imagens com fotos de cidades e desastres.
26	Imagens em preto e branco com linhas horizontais fortes.
27	Imagens claras com grande volume de textos.
28	Imagens com linhas horizontais coloridas.
29	Imagens com fotos relacionadas à tecnologia, saúde e música.
30	Imagens contendo fotos e textos em negrito.
31	Imagens contendo fotos de homens e esportes.
32	Imagens com fortes linhas horizontais, muitas cores e textos em letra pequena.
33	Imagens brancas com retângulos em cinza.
34	Imagens com a cor preta.
35	Imagens com hexágonos coloridos.
36	Imagens brancas com poucas letras.
37	Imagens contendo listas e tabelas.
38	Imagens brancas com pouco texto em negrito.
39	Imagens com fragmentos de circunferência.
40	Imagens com cores em tons de cinza e marrom.
41	Imagens com fragmentos de fotos e espaços em branco.

Tabela 1 – Grupos Gerados e suas Características

Grupo	Principais Característica do Grupo
42	Imagens densas em texto.
43	Imagens com sombreamentos e tons de cinza.
44	Imagens contendo parágrafos espaçados.
45	Imagens com cores vivas e poucos textos.
46	Imagens com carros e outros veículos.
47	Imagens com com rostos sorridentes e crianças.
48	Imagens com botões com contornos arredondados.
49	Imagens com quadrados escuros e quinas retas.
50	Imagens com cores vivas e muito texto.

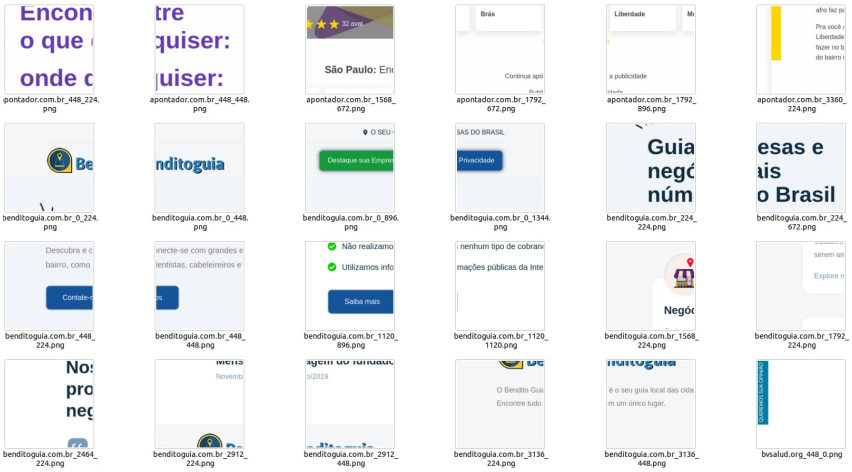


Figura 7 – Grupo 3: Imagens de fundo claro e uma cor de auto contraste por cima.

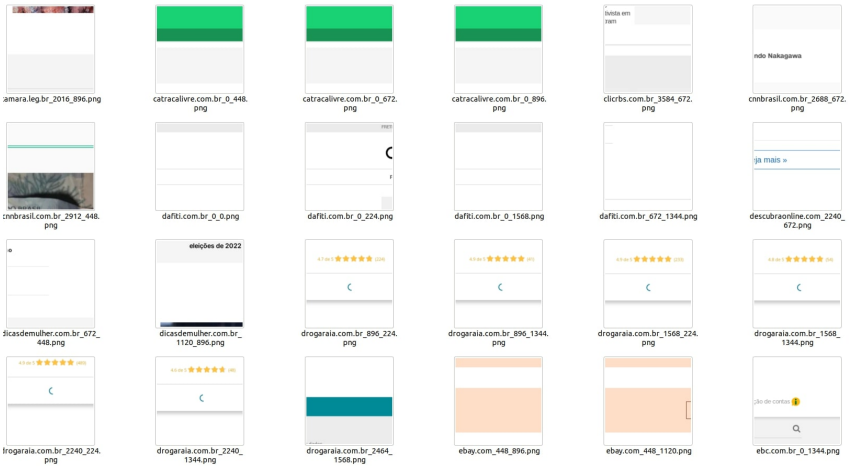


Figura 8 – Grupo 7: Imagens brancas com linhas horizontais bem delineadas.

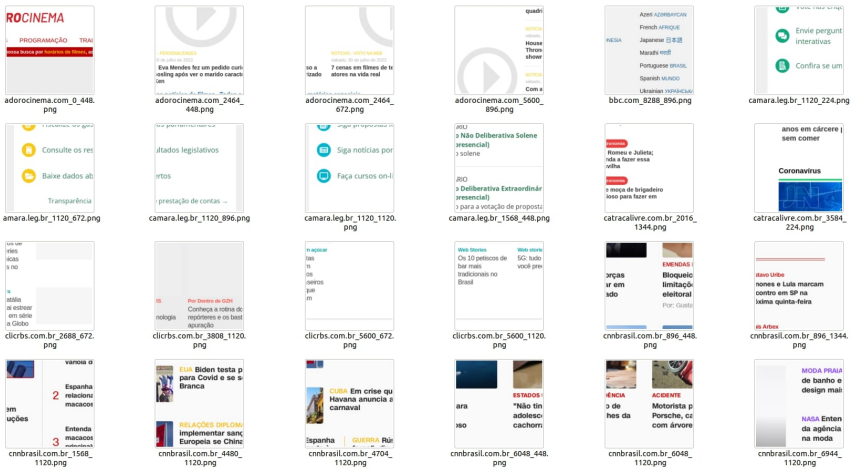


Figura 9 – Grupo 13: Imagens com listas e menus de navegação verticais.

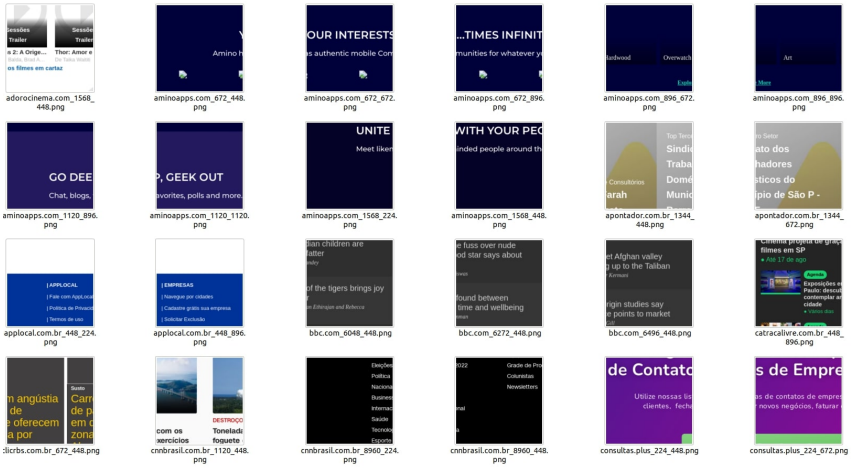


Figura 10 – Grupo 18: Imagens escuras com textos claros e pequenos.

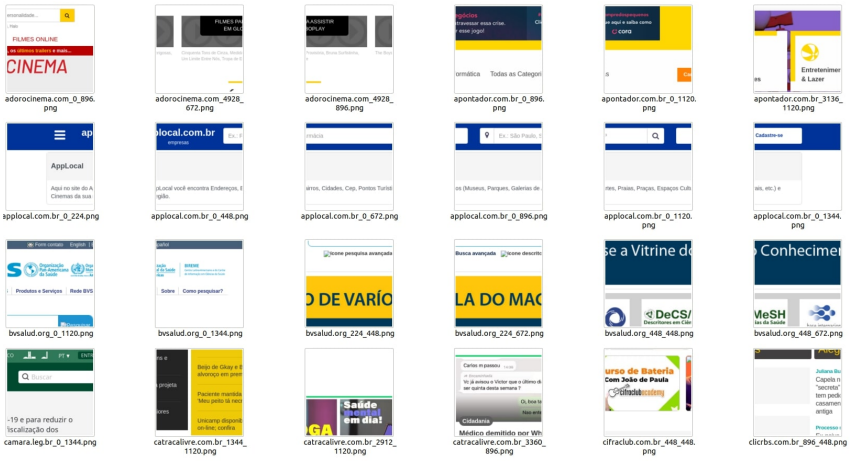


Figura 11 – Grupo 32 - Imagens com fortes linhas horizontais, muitas cores e textos em letra pequena.

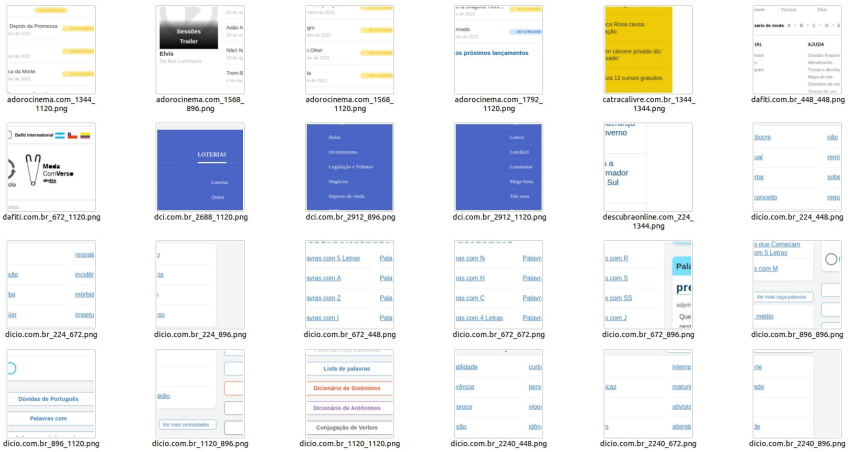


Figura 12 – Grupo 37 - Imagens com listas e tabelas.

4.2 Análise

Com base nos grupos gerados, foram feitas duas análises. A primeira análise referente ao conteúdo de cada grupo, onde foram feitas buscas exploratórias na intenção de identificar se um grupo possuía fragmentos referentes à um componente em particular. Dessa forma foram identificados os grupos que representam componentes comuns bem como o componente que esses grupos representam.

A segunda análise, foi o cálculo da porcentagem de páginas presentes em cada grupo, Figura 13. Note que uma página é considerada presente em grupo quando ela gerou ao menos um dos fragmentos que foi posicionado naquele grupo, por exemplo, uma página que possui espaços em branco e componentes para classificação vai estar presente nos grupos 2 e 7, pois gerou fragmentos que foram posicionados nesses respectivos grupos.

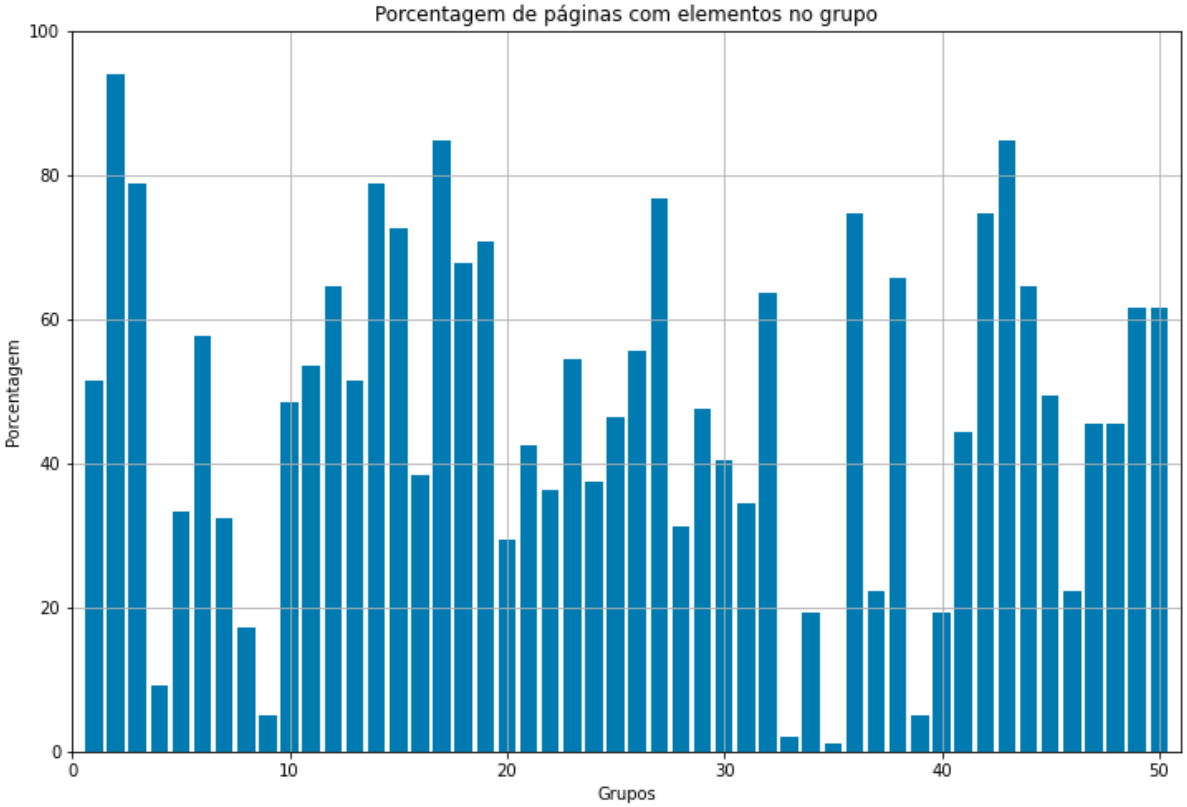


Figura 13 – Porcentagem de páginas presentes em cada grupo.

Por meio da correlação entre as duas análises feitas, é possível nomear quais foram os componentes web identificados e inferir a relevância destes componentes nas páginas estudadas.

Embora tenham sido produzidos 50 grupos diferentes, vários deles agrupam fragmentos de páginas web com aparências similares, porém sem possibilitar a identificação de componentes interessantes para o presente trabalho. Por exemplo, pode-se citar grupos que incluem fragmentos com espaços em branco, imagens de homens e linhas na horizontal. Portanto, nas análises a seguir, são feitos detalhamentos apenas para os grupos que permitem a identificação de componentes comuns, que são os Grupos 3, 7, 13, 18, 32 e 37.

De acordo com as análises realizadas, identificam-se os componentes que devem estar presentes nas páginas web.

- No Grupo 3, Figura 7, existem várias ocorrências de botões em alto contraste sobre fundos claros, contendo chamadas a ações importantes que estavam disponíveis nas páginas web, como cadastrar, comprar ou acessar. Na Figura 13, mostra-se que aproximadamente 80% dos *websites* avaliados contêm uma imagem presente nesse grupo. Pode-se afirmar, então, que esse é um componente muito comum.
- O Grupo 7, ilustrado na Figura 8, apresenta uma variedade de componentes que realizam classificação e avaliação, além de outras imagens em outros contextos. Embora esses componentes estejam presentes em apenas cerca de 30% dos *websites*, conforme mostrado

na Figura 13, esses componentes também devem estar presentes em páginas web.

- No Grupo 13, Figura 9, é possível detectar menus verticais de navegação. Observando-se a Figura 13, nota-se que aproximadamente 50% das páginas avaliadas contêm uma imagem presente nesse grupo, motivando a presença desse componente em páginas web.
- No Grupo 18, Figura 10, são identificados como padrão frequentemente utilizado em rodapés uma lista de *links* em fonte pequena apresentados sobre um fundo escuro. A popularidade desse componente dentro dos *websites* analisados pode ser visto devido a sua presença em mais de 80% dos casos na Figura 13, indicando que esse é um componente muito comum.
- No Grupo 32, Figura 11, são identificadas várias barras de pesquisa. Observando-se a Figura 13, pode-se notar que 60% dos *websites* analisados continham elementos que se enquadravam nesse contexto, motivando a presença desse componente em páginas web.
- No Grupo 37, Figura 12, observa-se que os fragmentos contêm alguma forma de lista ou de tabela. Embora esses componentes estejam presentes em apenas cerca de 25% dos *websites*, conforme mostrado na Figura 13, esses componentes também devem estar presentes em páginas web.

Uma observação adicional refere-se ao componente denominado esqueleto de carregamento, identificado no Grupo 33. Esse componente ocorre quando se usam formas em tons de cinza para preencher dados que ainda não foram obtidos. Esse componente é relevante para indicar aos usuários que os dados esperados estão sendo baixados. No caso do presente trabalho, esse componente se encontra presente em um número muito pequeno de páginas. Isso pode ser observado na Figura 13, que mostra uma porcentagem próxima a zero, desde que os fragmentos analisados no trabalho foram gerados com dados já carregados. Todavia, espera-se que este componente esteja presente em diversas páginas web durante o carregamento.

Além dos componentes identificados anteriormente, também foram observadas outras características sobre as páginas web, que são listadas a seguir. Entretanto, essas características não são consideradas como componentes de interesse no nosso trabalho, dado que elas não possuem uma ligação direta com os componentes web.

- A web contem muitas páginas com espaços em branco, como pode ser observado pela quantidade de paginas presentes no Grupo 2.
- Muitas páginas utilizam projetos com linhas e cores variadas como indica a porcentagem de páginas no Grupo 17 (Figura 13).
- A maioria das páginas também possui espaços claros e com grandes volume de texto, como mostra a quantidade de páginas presentes no Grupo 27 na Figura 13.
- Efeitos de sombreamento são muito comuns como mostra a porcentagem de páginas do Grupo 43 na Figura 13.

4.3 Considerações Finais

Neste capítulo foram apresentados os resultados obtidos com o desenvolvimento do trabalho. Em seguida, foram discutidas características relacionadas aos agrupamentos gerados, as quais contribuíram para a identificação de componentes que devem estar presentes em páginas web. Esses componentes são:

- Botões em alto contraste sobre fundos claros, contendo chamadas a ações importantes como cadastrar, comprar ou acessar.
- Opções de classificação e avaliação.
- Menus verticais de navegação.
- Rodapés contendo uma lista de links em fontes pequena apresentados sobre um fundo escuro.
- Barras de pesquisa.
- Lista de opções ou tabela.
- Esqueleto de carregamento.

CAPÍTULO

5

CONCLUSÕES

Neste capítulo são descritas as conclusões sobre o trabalho desenvolvido. Na seção 5.1 são detalhadas a metodologia empregada para desenvolver o trabalho e as contribuições. Na seção 5.2 são destacadas as limitações do trabalho desenvolvido, juntamente com possibilidades de trabalhos futuros.

5.1 Trabalho Desenvolvido

No contexto atual da web, centenas de novos sítios são disponibilizados todos os dias e uma quantidade ainda maior de sítios é atualizada ou melhorada periodicamente. Neste contexto, identificar componentes (ou funcionalidades) que deveriam estar presentes em páginas web é uma necessidade para que as empresas mantenham-se competitivas no mercado. Assim, ser capaz de analisar páginas web e identificar componentes comuns é muito valioso. Neste trabalho, foi desenvolvida uma metodologia voltada para esse fim.

A metodologia foi aplicada da seguinte forma. Primeiramente, foi feita a criação de uma base de dados contendo páginas web a serem analisadas. Baseado na hipótese de que a aparência visual das páginas contém mais informações acerca das páginas quando comparado ao código fonte das mesmas, optou-se por montar uma base de dados contendo a aparência visual das páginas, ou seja, as suas imagens. Para criar essa base de dados, foram identificadas quais páginas web deveriam ser consideradas. Nessa escolha, foram considerados as páginas mais acessadas ao redor do mundo. Também foi utilizado um *web-crawler* em conjunto com a biblioteca *puppeteer* do *javascript*, sendo que o *crawler* assumiu a responsabilidade de obter os códigos fontes das páginas e a biblioteca assumiu o papel de transformar esses códigos em imagens.

Uma vez criada a base de dados com as imagens das páginas web, essas imagens foram padronizadas. O padronização foi alcançada por meio da fragmentação das imagens em

quadrados menores, sendo que cada página foi fragmentada várias vezes, cada uma considerando quadrados de tamanhos diferentes. No final, todos os quadrados gerados foram redimensionados em para um único tamanho padrão. Essa estratégia de fragmentação teve como objetivo criar fragmentos de componentes que poderiam estar em tamanhos diferentes nas diferentes páginas analisadas.

Na sequência, os fragmentos semelhantes foram agrupados. Para criar os agrupamentos com base no conteúdo das imagens, o modelo de redes neural convolucional EfficientNetB1 foi utilizado para gerar um vetor de características para cada fragmento. Os fragmentos gerados foram manipulados por meio da técnica de redução de dimensionalidade PCA para facilitar o processamento computacional. Por fim, foi aplicado o algoritmo *k-means* e foram gerados 50 agrupamentos.

A partir da análise dos 50 agrupamentos, foram identificados os seguintes componentes comuns nas páginas analisadas, os quais devem ser utilizados quando do desenvolvimento de novas páginas web:

- Botões em alto contraste sobre fundos claros, contendo chamadas a ações importantes como cadastrar, comprar ou acessar.
- Opções de classificação e avaliação.
- Menus verticais de navegação.
- Rodapés contendo vários *links* em fontes pequenas.
- Barras de pesquisa, contendo por exemplo um campo de texto e um botão para pesquisar.
- Listas ou tabelas para mostrar informações.
- Esqueleto de carregamento, o qual é relevante para indicar aos usuários que os dados da página solicitada está sendo baixado.

5.2 Limitações e Trabalhos Futuros

Da forma como foram conduzidos os agrupamentos, observou-se que os padrões das aparências das páginas web tiveram maior peso na geração dos grupos do que os componentes dessas páginas. Por exemplo, alguns grupos foram caracterizados unicamente por estilos, como paleta de corres, suavização de curvas ou conteúdo de fotos.

Portanto, pode-se inferir que a abordagem utilizada neste trabalho para isolar os componentes e agrupá-los precisa ser melhorada. Para minimizar esse aspecto, pode-se propor como um trabalho futuro a coleta manual de dados visando evitar imagens contendo apenas estilos da página web que ela representa.

Outro trabalho futuro consiste na aplicação de outros algoritmos de extração de características com foco maior na forma. Esses algoritmos poderiam aproximar componentes iguais que possuem paletas de cores diferentes. Outra abordagem de agrupamento que não o k-means

também poderia ser aplicada para fins de comparação, tal como *consensus clustering*.

Também é possível fazer um trabalho para, a partir da identificação de componentes que devem existir em uma página web, investigar quais páginas de um sítio utilizam cada componente e quais a correlação entre essas páginas em termos dos componentes.

No presente trabalho, foram analisadas páginas que possuem muitos acessos. Outro trabalho futuro poderia ser realizado considerando páginas que possuem poucas visualizações. Dessa forma, seria possível comparar os resultados obtidos e determinar componentes diferenciais que possibilitam que uma página web seja muito acessada.

Por fim, estudos correlacionando o país onde a página web é mais acessada com as características do mesmo poderiam indicar aspectos culturais interessantes acerca do uso da web em cada cultura.

REFERÊNCIAS

BAI, Q.; XIONG, G.; ZHAO, Y.; HE, L. Analysis and detection of bogus behavior in web crawler measurement. **Procedia Computer Science**, v. 31, p. 1084–1091, 2014. ISSN 1877-0509. 2nd International Conference on Information Technology and Quantitative Management, ITQM 2014. Citado na página 35.

BAL, S. The issues and challenges with the web crawlers. **International Journal of Information Technology Systems**, v. 1, p. 1–10, 02 2012. Citado na página 36.

BARFOROUSH HOSSEIN SHIRAZI, H. E. A. A. 39a new classification framework to evaluate the entity profiling on the web: Past, present and future. **ACM Computing Surveys**, v. 50, n. 39, 2017. Citado nas páginas 32 e 40.

DHANITH, P. R. J.; SURENDIRAN, B.; RAJA, S. P. A word embedding based approach for focused web crawling using the recurrent neural network. **INTERNATIONAL JOURNAL OF INTERACTIVE MULTIMEDIA AND ARTIFICIAL INTELLIGENCE**, v. 6, n. 6, p. 122–132, JUN 2021. ISSN 1989-1660. Citado nas páginas 32 e 41.

DU, Y.; LI, X. An semantic rank for web crawler based on formal concept analysis. In: **Proceedings of the 2007 International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2007)**. [S.l.]: Atlantis Press, 2007/10. p. 1447–1453. ISBN 978-90-78677-04-8. ISSN 1951-6851. Citado na página 36.

JANIESCH, C.; ZSCHECH, P.; HEINRICH, K. Machine learning and deep learning. **Electronic Markets**, v. 31, n. 3, p. 685–695, Sep 2021. ISSN 1422-8890. Citado nas páginas 36, 37 e 38.

JIAO, L.; ZHAO, J. A survey on the new generation of deep learning in image processing. **IEEE Access**, v. 7, p. 172231–172263, 2019. Citado nas páginas 32, 38 e 39.

LAKHANI, P.; GRAY, D. L.; PETT, C. R.; NAGY, P.; SHIH, G. Hello world deep learning in medical imaging. **Journal of Digital Imaging**, v. 31, n. 3, p. 283–289, Jun 2018. ISSN 1618-727X. Citado nas páginas 32 e 38.

LEEA WEI-CHANG YEHB, M.-C. C. J.-H. Web page classification based on a simplified swarm optimization. **Applied Mathematics and Computation**, Elsevier, v. 270, p. 13—-24, 2015. Citado nas páginas 32 e 40.

LEVER, J.; KRZYWINSKI, M.; ALTMAN, N. Principal component analysis. **Nature Methods**, v. 14, n. 7, p. 641–642, Jul 2017. ISSN 1548-7105. Citado na página 40.

LI, H.; XU, Z.; LI, T.; SUN, G.; CHOO, K.-K. R. An optimized approach for massive web page classification using entity similarity based on semantic network. **Future Gener. Comput. Syst.**, v. 76, p. 510–518, 2017. Citado nas páginas 32 e 40.

LI, Y.; WU, H. A clustering method based on k-means algorithm. **Physics Procedia**, v. 25, p. 1104–1109, 2012. ISSN 1875-3892. International Conference on Solid State Devices and Materials Science, April 1-2, 2012, Macao. Citado nas páginas 32 e 40.

MASSARO, A.; GIANNONE, D.; BIRARDI, V.; GALIANO, A. M. An innovative approach for the evaluation of the web page impact combining user experience and neural network score. **Future Internet**, v. 13, n. 6, 2021. ISSN 1999-5903. Citado nas páginas 32 e 41.

MEDEIROS, O. V. F. e Moacir MORAIS e V. The access to the internet as a fundamental right to strike effectiveness in digital society. **Revista Juridica**, v. 1, n. 58, p. 88–115, 2020. ISSN 2316-753X. Citado na página 31.

MOZILA. **Developer Mozila**. [S.l.: s.n.], 2021. Citado na página 32.

ONAN, A. Classifier and feature set ensembles for web page classification. **Information Sciences**, SAGE, v. 42, p. 150—-165, 2016. Citado nas páginas 31, 32 e 40.

RO, J. S. H. I.; IM, E. G. Detection method for distributed web-crawlers: A long-tail threshold model. **Security and Communication Networks**, v. 2018, 2018. ISSN 1939-0114. Citado na página 36.

SELAMAT, S. O. A. Web page feature selectionand classification using neural networks. **Information Sciences**, Elsevier, v. 158, p. 69—-88, 2004. Citado nas páginas 32 e 40.

SMOLA, A.; VISHWANATHAN, S. **Introduction to Machine Learning**. [S.l.]: Cambridge university press, 2008. Citado na página 32.

____. **Introduction to Machine Learning**. The Pitt Building, Trumpington Street, Cambridge, United Kingdom: the press syndicate of the university of cambridge, 2008. ISBN 0 521 82583 0. Citado na página 37.

TAN, M.; LE, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv, 2019. Citado na página 39.

THEFREEDICTIONARY. **For the American Heritage Dictionary definition**. [S.l.: s.n.], 2021. Citado nas páginas 31 e 35.

TIAN, Y. Artificial intelligence image recognition method based on convolutional neural network algorithm. **IEEE Access**, v. 8, p. 125731–125744, 2020. Citado nas páginas 32 e 39.

XU, N. Understanding the reinforcement learning. **Journal of Physics: Conference Series**, IOP Publishing, v. 1207, p. 012014, apr 2019. Citado na página 37.

GLOSSÁRIO

Framework: é uma abstração que une códigos comuns entre vários projetos de *software* provendo uma funcionalidade genérica. *Frameworks* são projetados com a intenção de facilitar o desenvolvimento de *software*, habilitando designers e programadores a gastarem mais tempo determinando as exigências do *software* do que com detalhes de baixo nível do sistema.

HTML: Sinônimo mais conhecido de *World Wide Web* (WWW). É a interface gráfica da Internet que torna os serviços disponíveis totalmente transparentes para o usuário e ainda possibilita a manipulação multimídia da informação.

Padrões de projeto: ou *Design Pattern*, descreve uma solução geral reutilizável para um problema recorrente no desenvolvimento de sistemas de *software* orientados a objetos. Não é um código final, é uma descrição ou modelo de como resolver o problema do qual trata, que pode ser usada em muitas situações diferentes.

Template: é um documento sem conteúdo, com apenas a apresentação visual (apenas cabeçalhos por exemplo) e instruções sobre onde e qual tipo de conteúdo deve entrar a cada parcela da apresentação.

Web: Sinônimo mais conhecido de *World Wide Web* (WWW). É a interface gráfica da Internet que torna os serviços disponíveis totalmente transparentes para o usuário e ainda possibilita a manipulação multimídia da informação.

WYSIWYG: "What You See Is What You Get" ou "O que você vê é o que você obtém". Recurso tem por objetivo permitir que um documento, enquanto manipulado na tela, tenha a mesma aparência de sua utilização, usualmente sendo considerada final. Isso facilita para o desenvolvedor que pode trabalhar visualizando a aparência do documento sem precisar salvar em vários momentos e abrir em um *software* separado de visualização.

APÊNDICE

A

GRUPOS

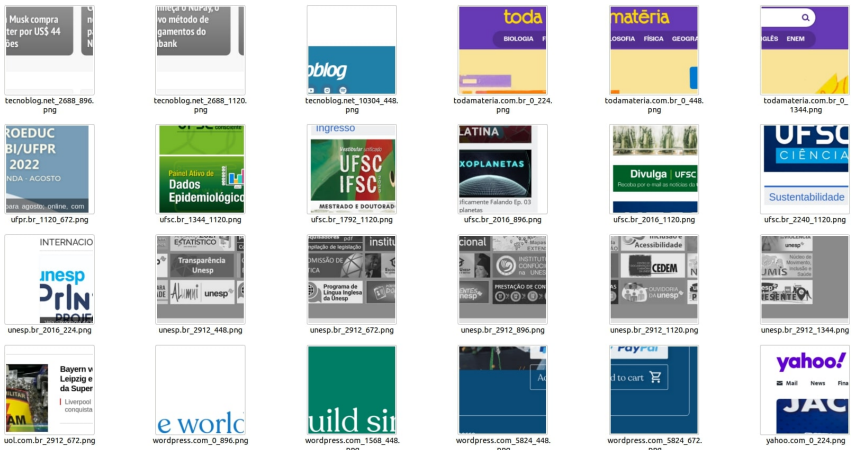


Figura 14 – Grupo 1 - Agrupou em grande parte de imagens com cores sólidas que apresentavam textos em alto contraste por cima.

Figura 15 – Grupo 2 - Agrupou imagens completamente brancas ou claras.

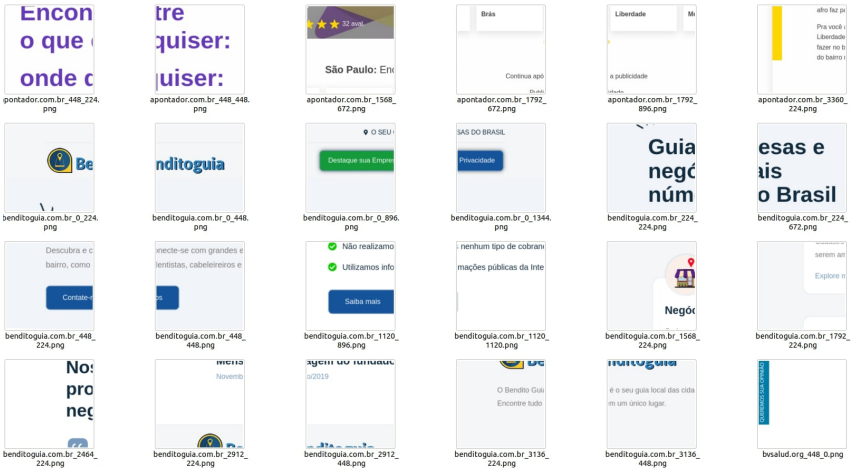


Figura 16 – Grupo 3 - Agrupou imagens de fundo claro e uma cor de auto contraste por cima.

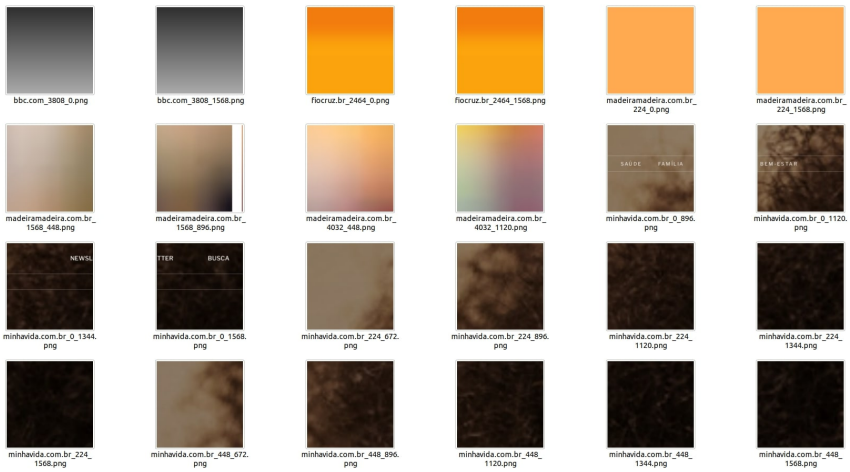


Figura 17 – Grupo 4 - Agrupou imagens em tons amarelados.

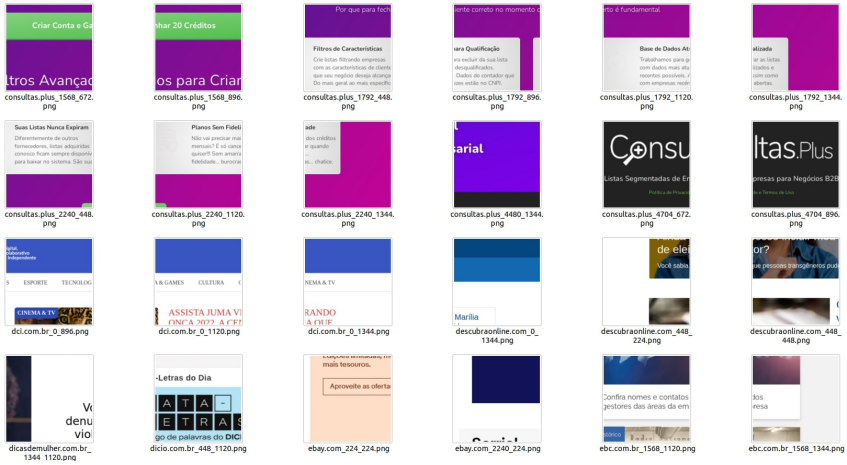


Figura 18 – Grupo 5 - Agrupou imagens contendo quadrados e contrastes entre cores.

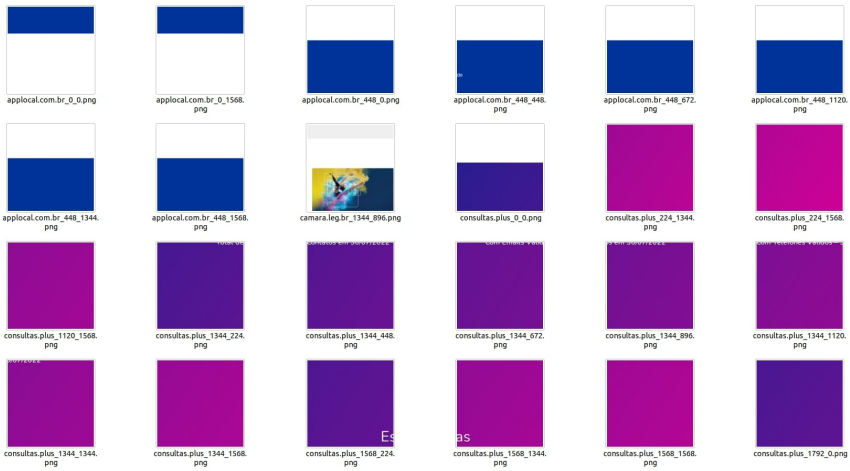


Figura 19 – Grupo 6 - Agrupou imagens em tons azuis e rosa, e poucas letras.

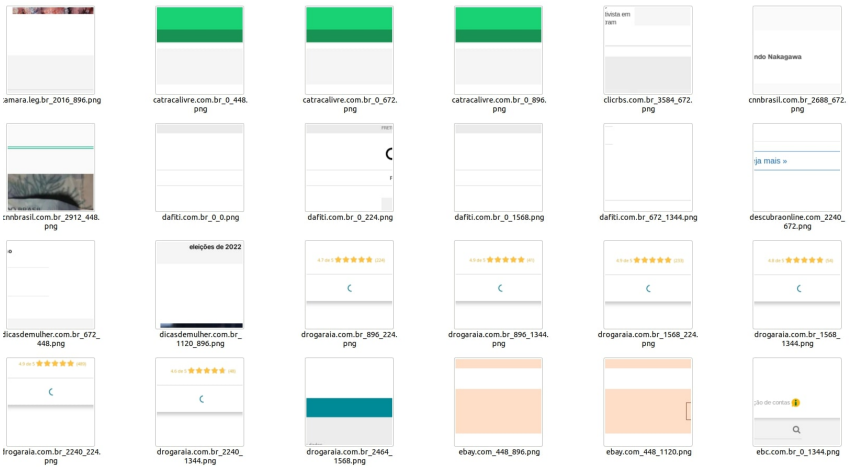


Figura 20 – Grupo 7 - Agrupou imagens brancas com linhas horizontais bem delineadas.

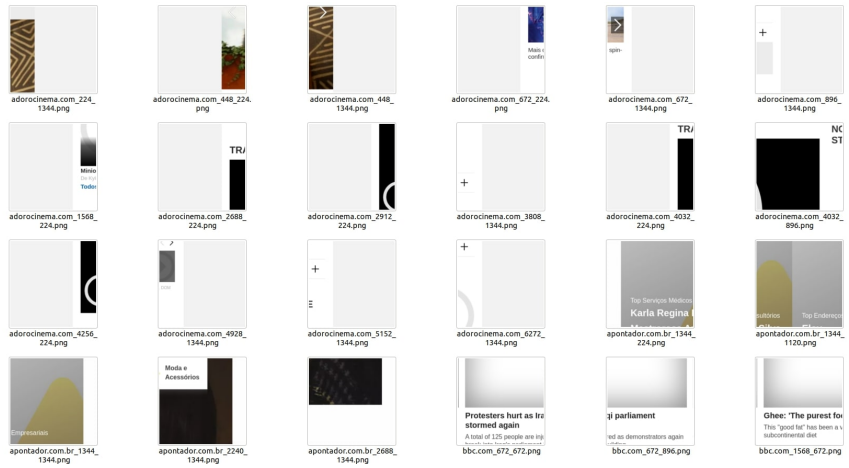


Figura 21 – Grupo 8 - Agrupou imagens com linhas verticais fortes com troca de cores.

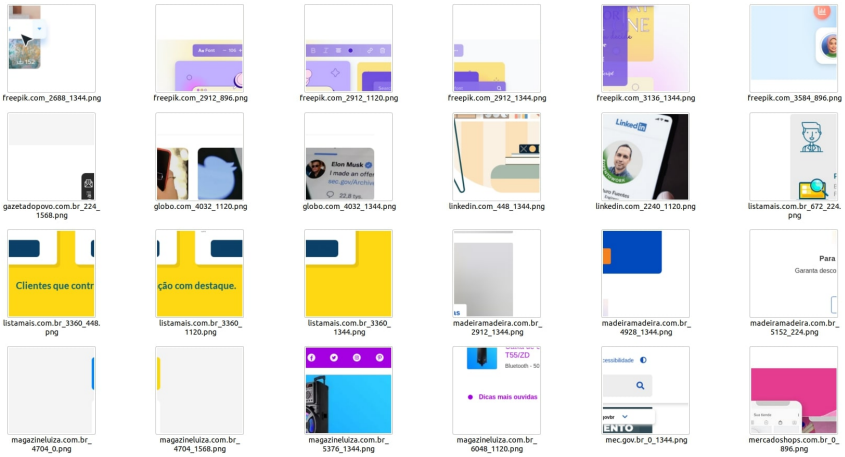


Figura 22 – Grupo 9 - Agrupou elementos com quinas aredondadas.

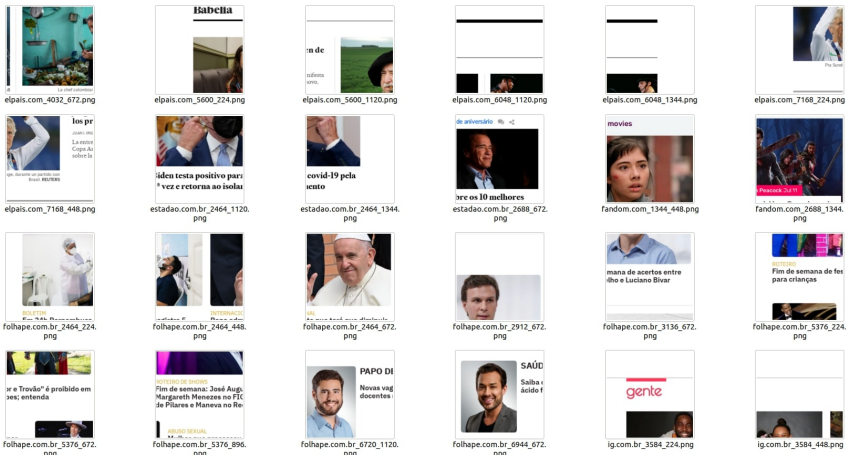


Figura 23 – Grupo 10 - Agrupou componentes que continham faces de pessoas.

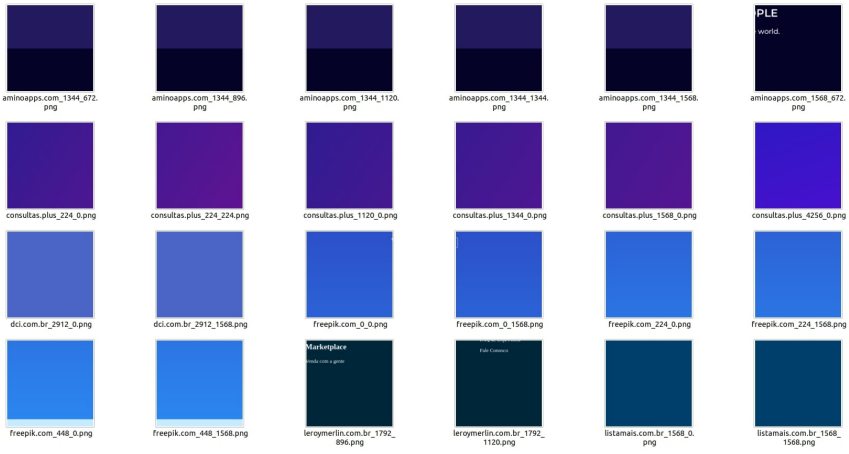


Figura 24 – Grupo 11 - Agrupou imagens em tons azuis escuros.

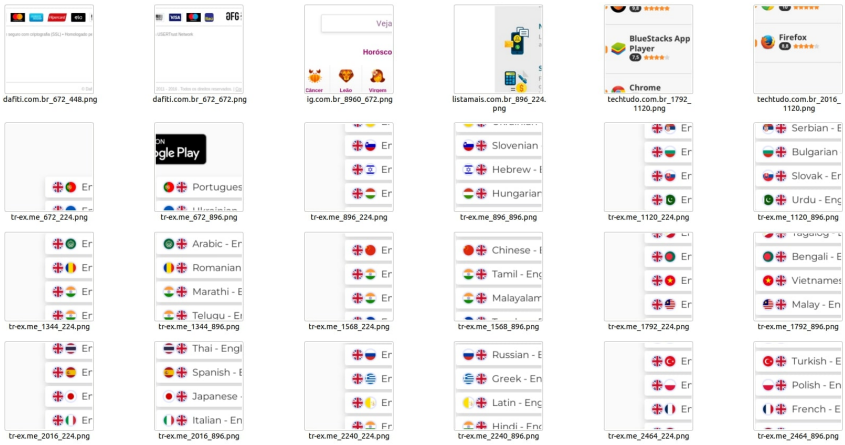


Figura 25 – Grupo 12 - Agrupou imagens com listas e bandeiras.

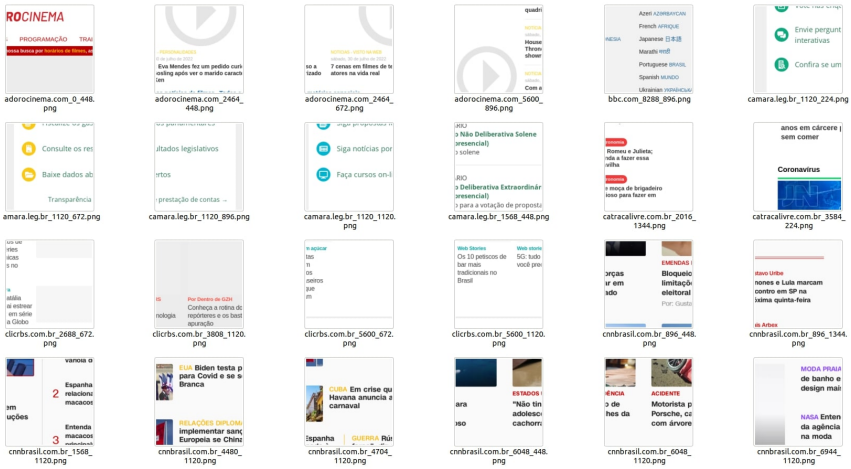


Figura 26 – Grupo 13 - Agrupou imagens com listas e menus de navegação verticais.

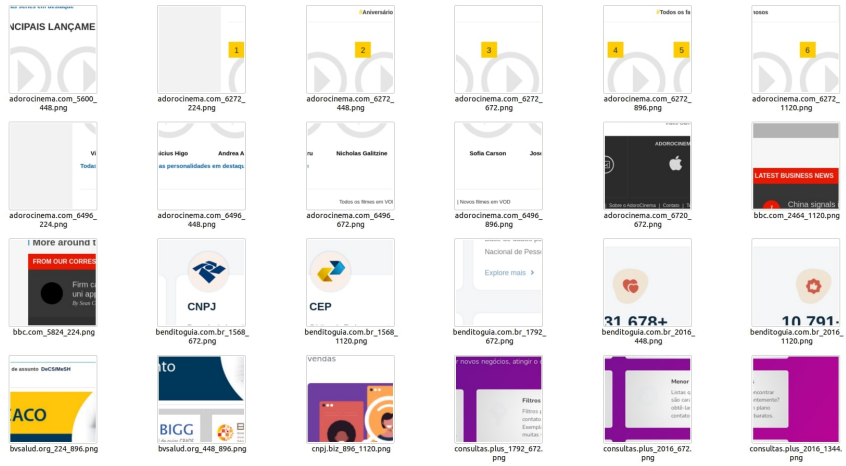


Figura 27 – Grupo 14 - Agrupou imagens claras contendo círculos.

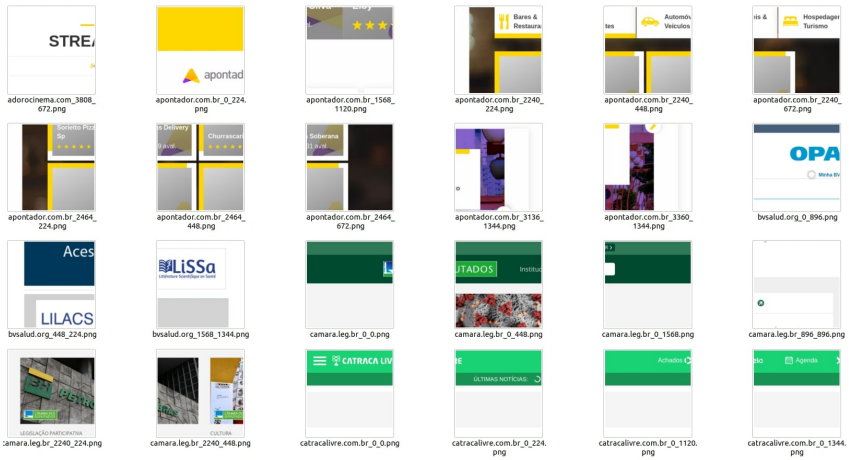


Figura 28 – Grupo 15 - Agrupou imagens com quadrados e com diversidade de cores.

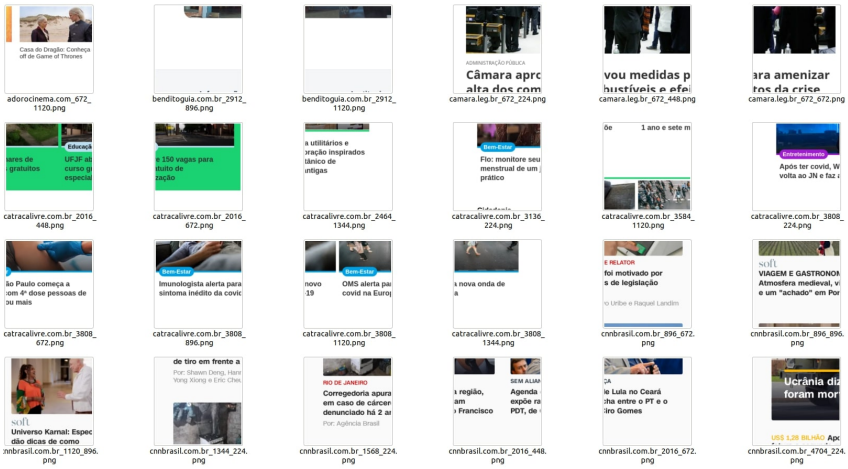


Figura 29 – Grupo 16 - Agrupou imagens com fragmentos de fotos e textos.

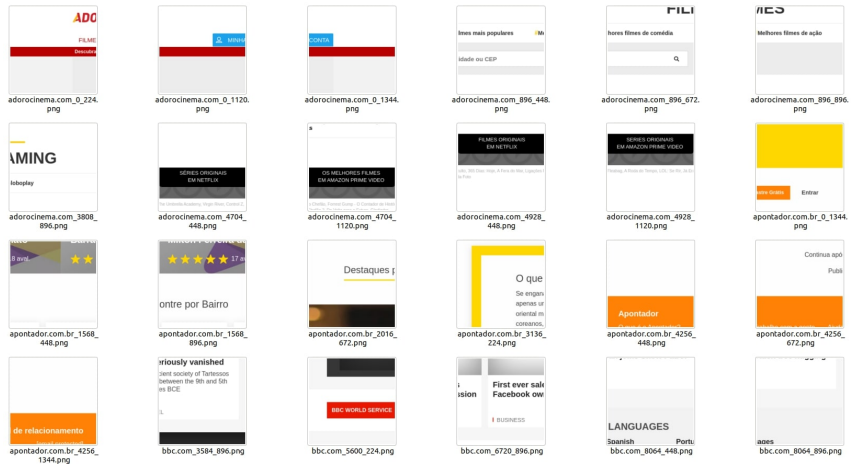


Figura 30 – Grupo 17 - Agrupou imagens com linhas horizontais fortes e variedade de cores.

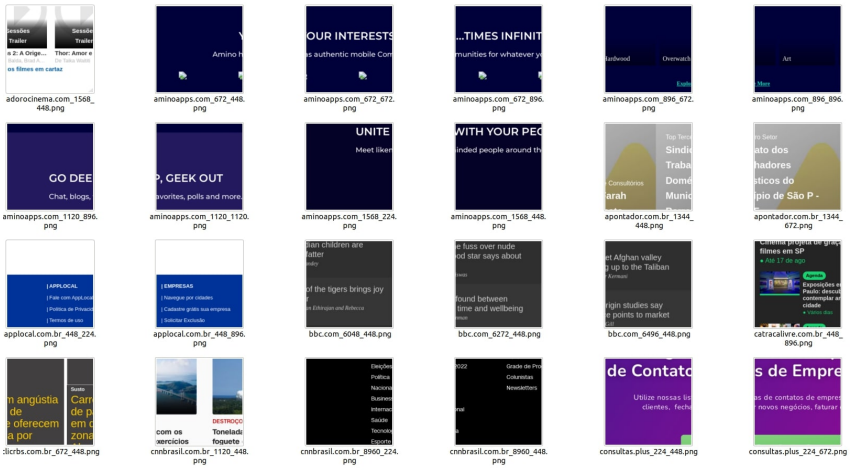


Figura 31 – Grupo 18 - Agrupou imagens escuras com textos claros e pequenos.

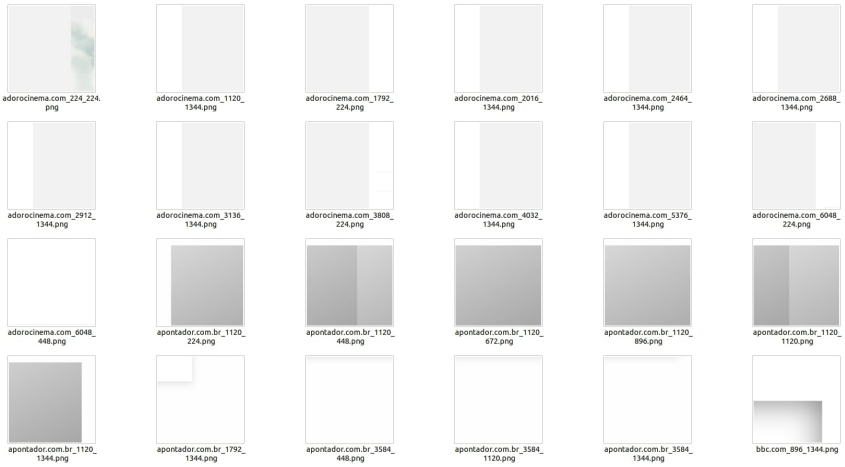


Figura 32 – Grupo 19 - Agrupou imagens claras com poucos textos e linhas verticais fortes.

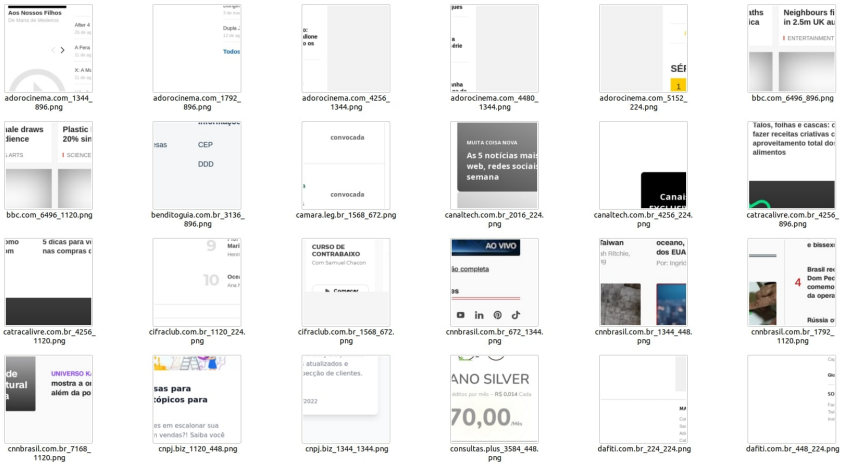


Figura 33 – Grupo 20 - Agrupou imagens carregadas com muito texto em letras pequenas.

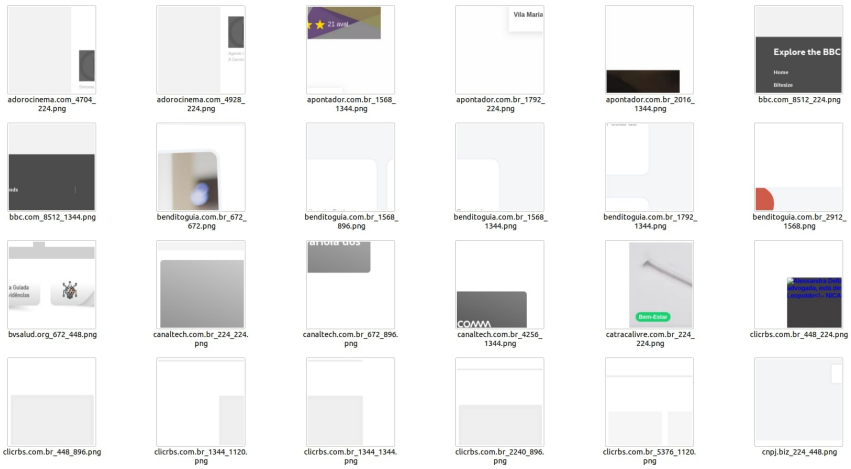


Figura 34 – Grupo 21 - Agrupou imagens com quinas, poucos textos e grandes áreas em branco.

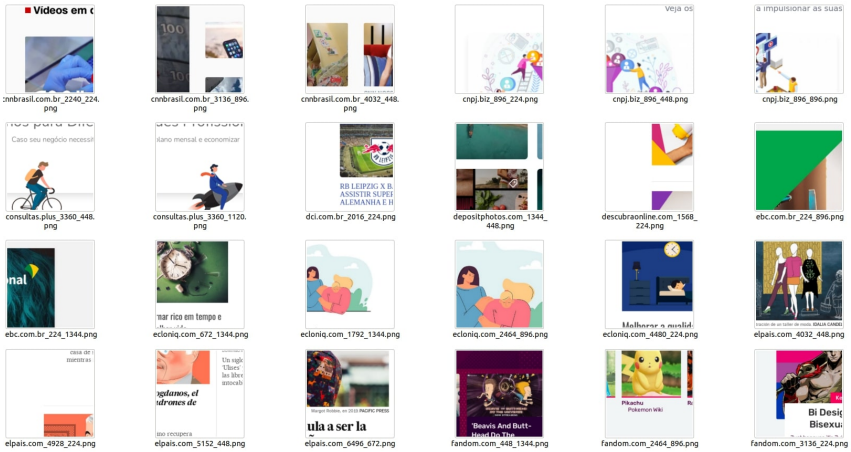


Figura 35 – Grupo 22 - Agrupou imagens contendo desenhos e áreas em branco.

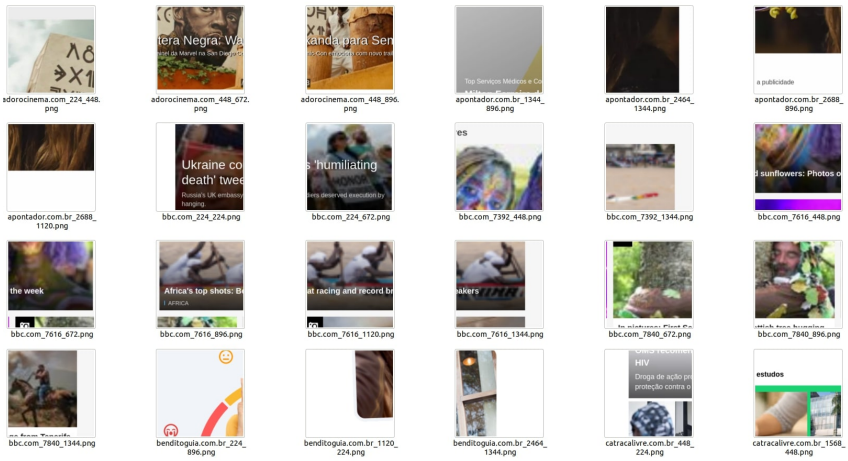


Figura 36 – Grupo 23 - Agrupou imagens contendo fotos diversas.

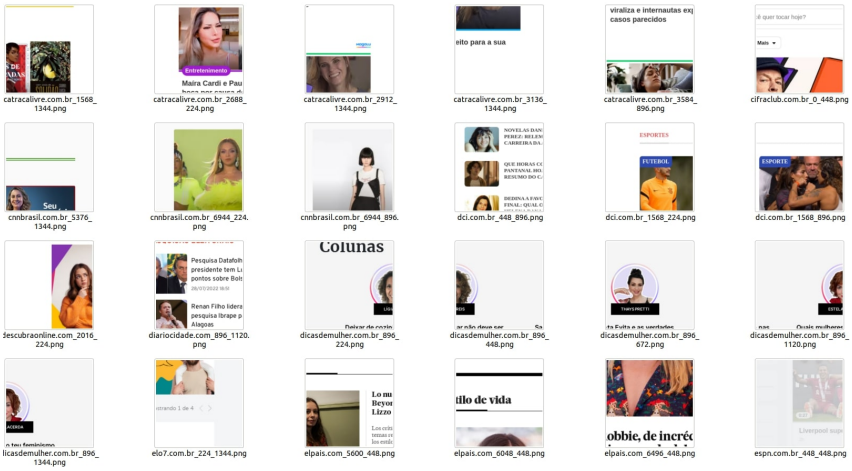


Figura 37 – Grupo 24 - Agrupou imagens contendo fotos de mulheres.

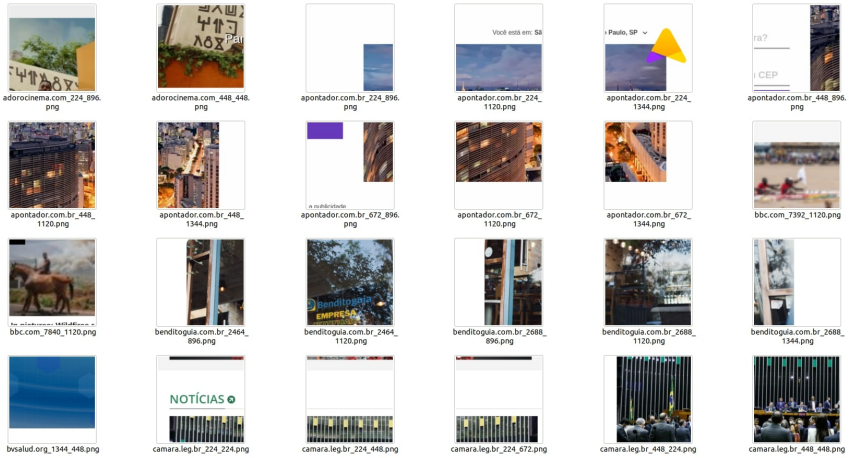


Figura 38 – Grupo 25 - Agrupou imagens com fotos de cidades e desastres.

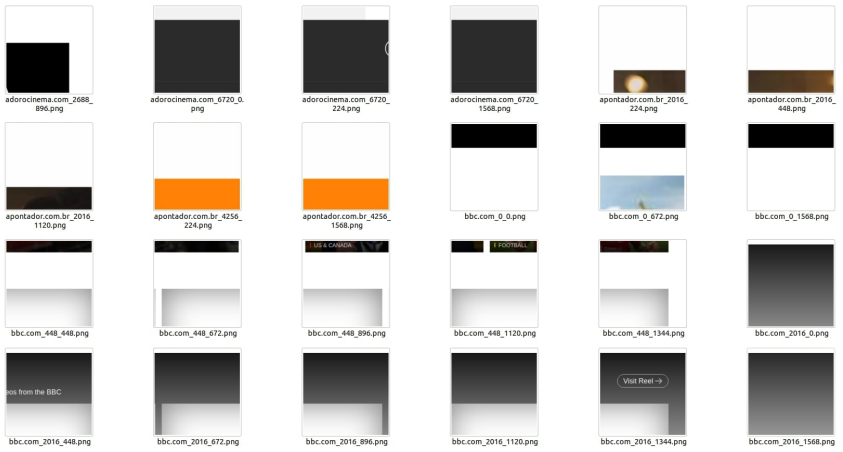


Figura 39 – Grupo 26 - Agrupou em grande parte imagens em preto e branco com linhas horizontais fortes.

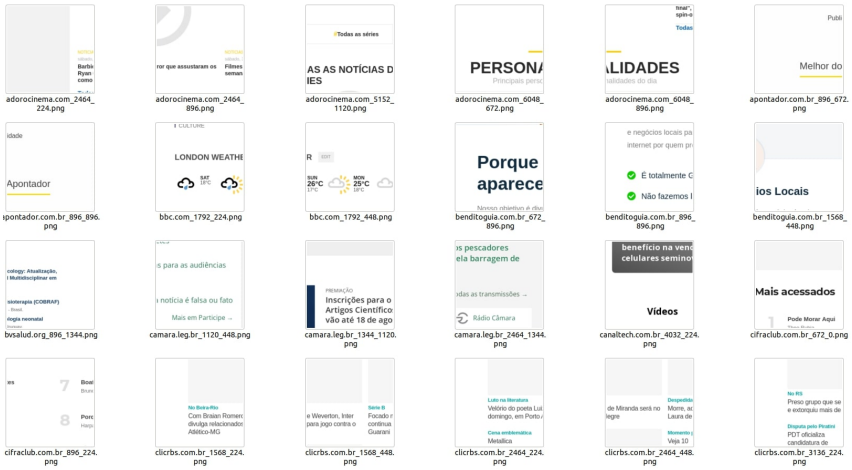


Figura 40 – Grupo 27 - Agrupou imagens claras com grande volume de textos.

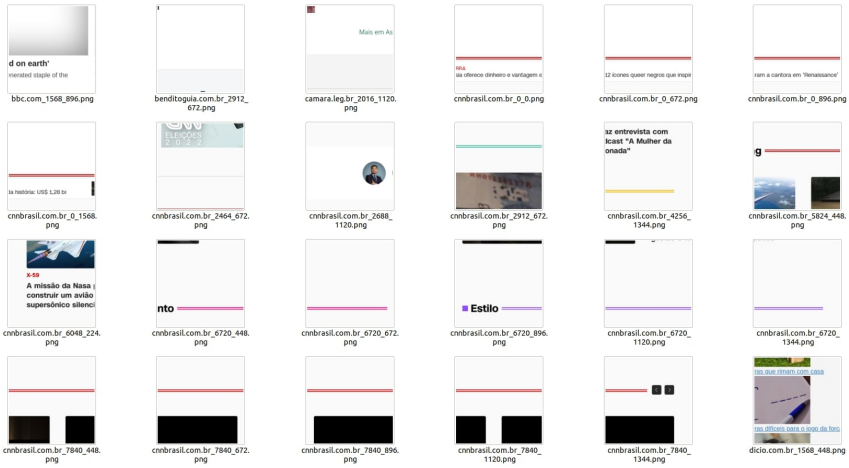


Figura 41 – Grupo 28 - Agrupou imagens com linhas horizontais coloridas.

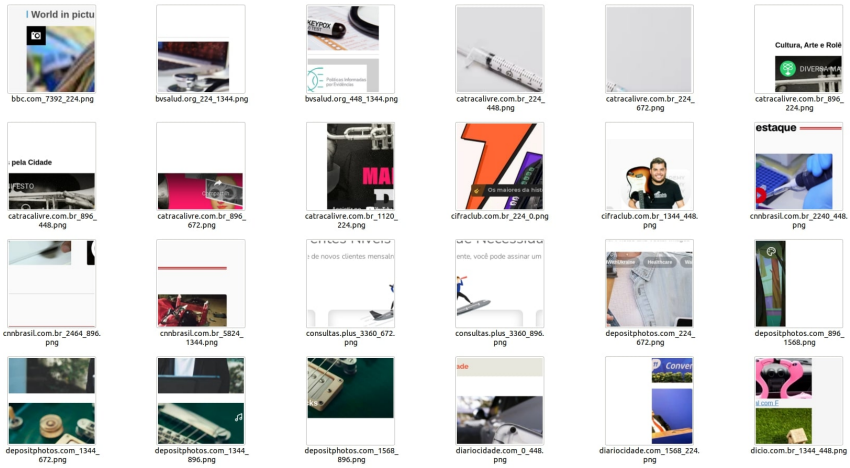


Figura 42 – Grupo 29 - Agrupou imagens com fotos relacionadas à tecnologia, saúde e música.

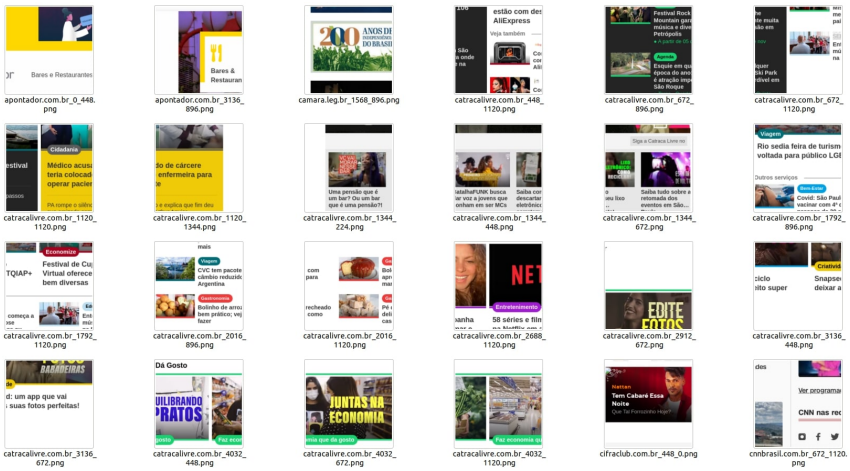


Figura 43 – Grupo 30 - Agrupou imagens contendo fotos e textos em negrito.

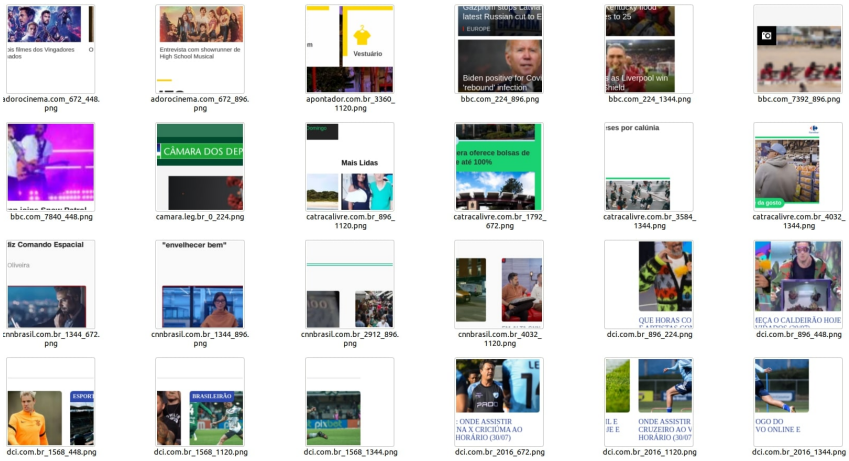


Figura 44 – Grupo 31 - Agrupou imagens contendo fotos de homens e esportes.

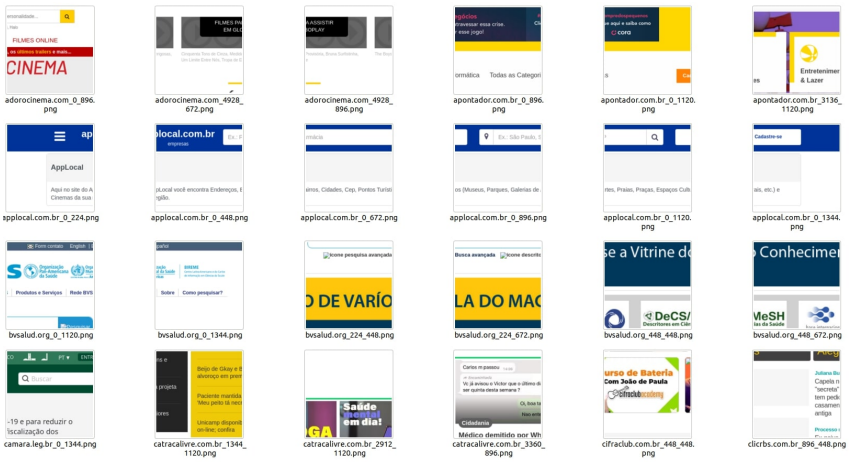


Figura 45 – Grupo 32 - Agrupou imagens com fortes linhas horizontais, muitas cores e textos em letra pequena.

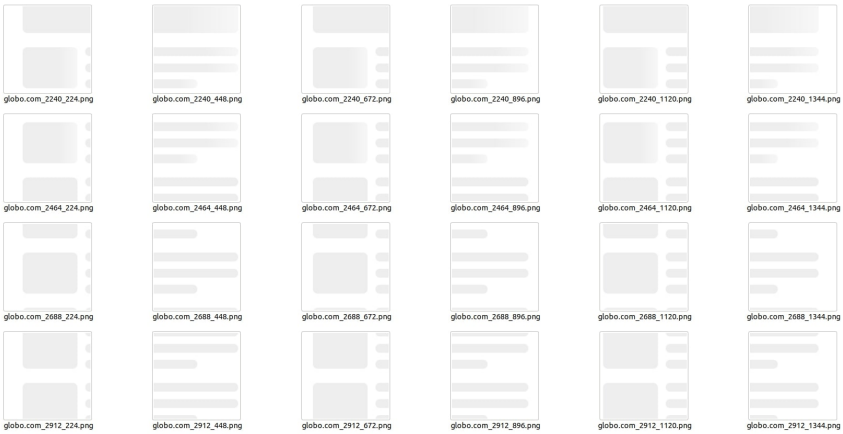


Figura 46 – Grupo 33 - Agrupou brancas com retângulos em cinza.

Figura 47 – Grupo 34 - Agrupou imagens com a cor preta.

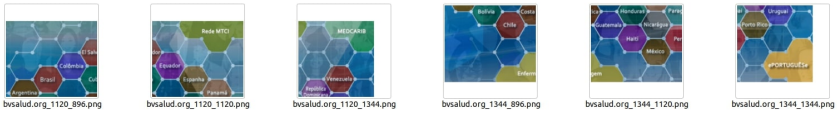


Figura 48 – Grupo 35 - Agrupou imagens com hexágonos coloridos.

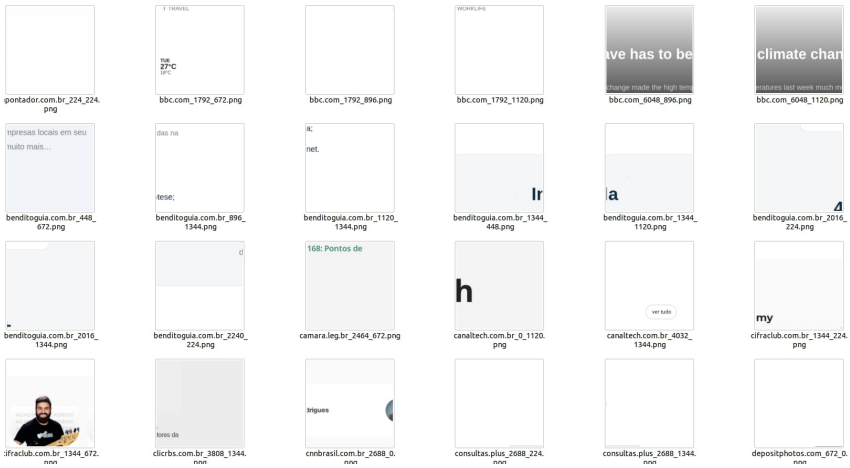


Figura 49 – Grupo 36 - Agrupou imagens brancas com poucas letras.

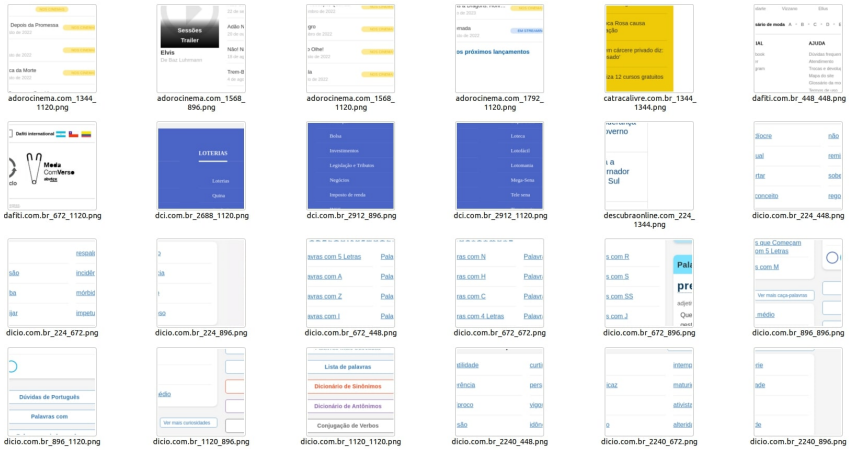


Figura 50 – Grupo 37 - Agrupou listas e tabelas.

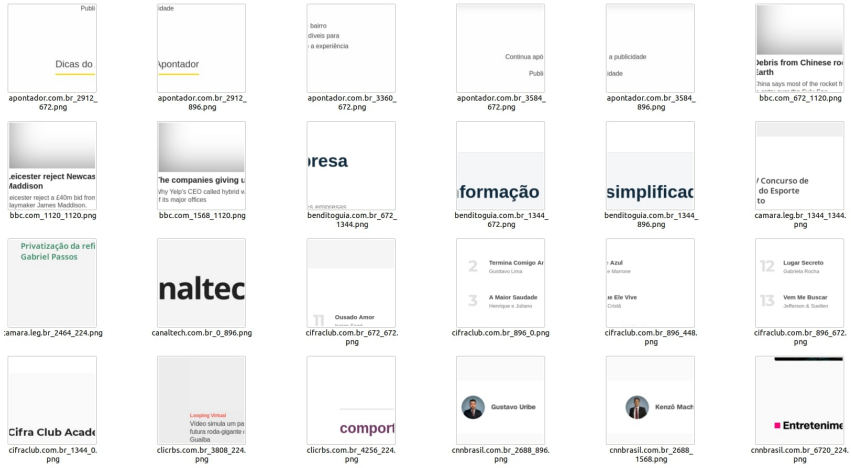


Figura 51 – Grupo 38 - Agrupou imagens brancas com pouco texto em negrito.

Figura 52 – Grupo 39 - Agrupou imagens com fragmentos de circunferência.

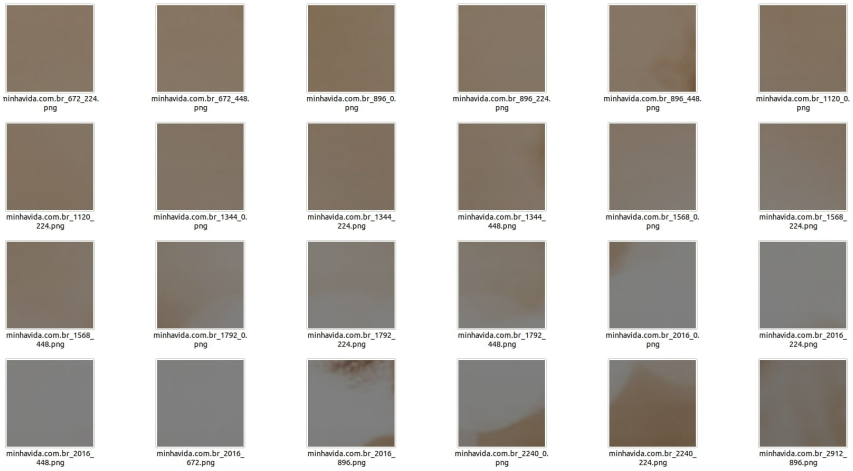


Figura 53 – Grupo 40 - Agrupou imagens com cores em tons de cinza e marrom.

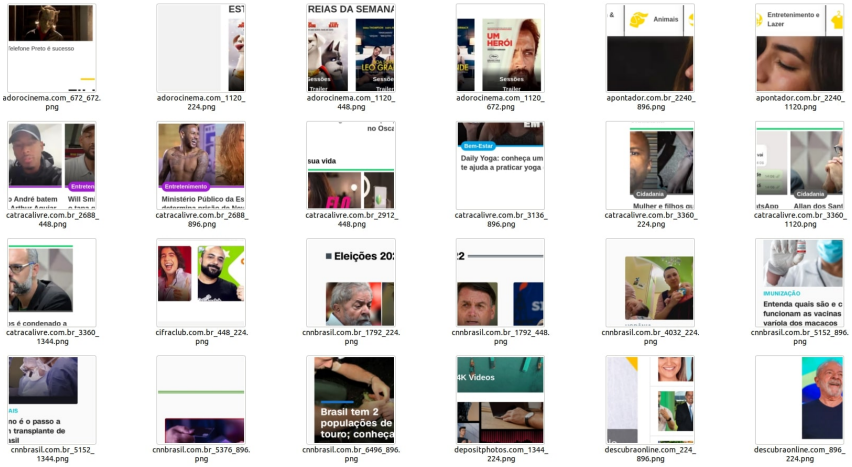


Figura 54 – Grupo 41 - Agrupou imagens com fragmentos de fotos e espaços em branco.

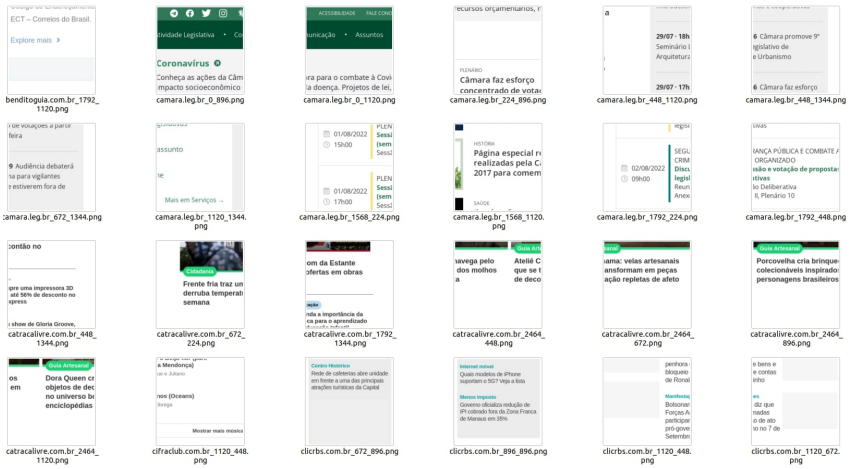


Figura 55 – Grupo 42 - Agrupou imagens densas em texto.

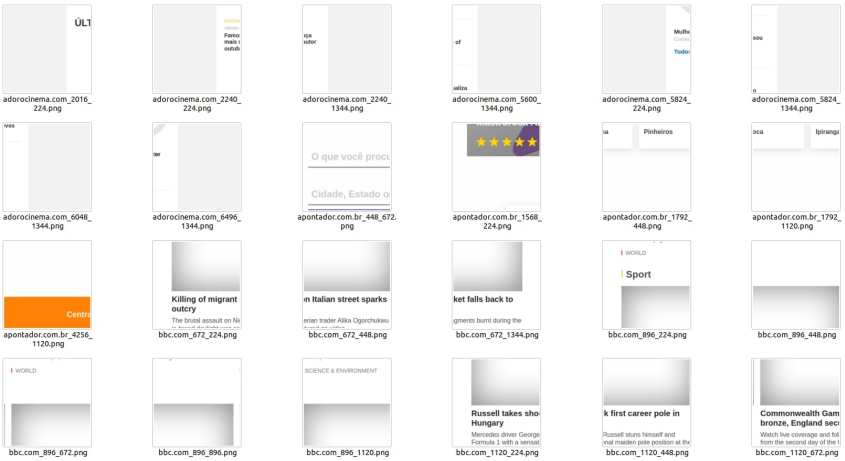


Figura 56 – Grupo 43 - Agrupou imagens com sombreamentos e tons de cinza.

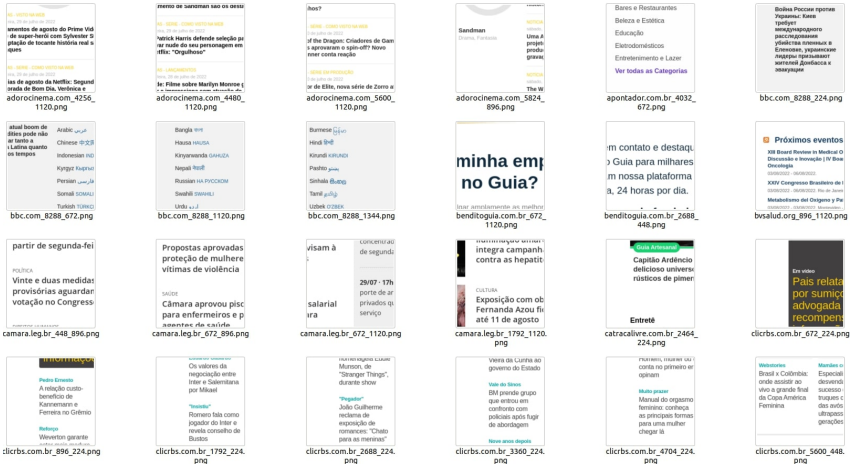


Figura 57 – Grupo 44 - Agrupou imagens contendo parágrafos espaçados.

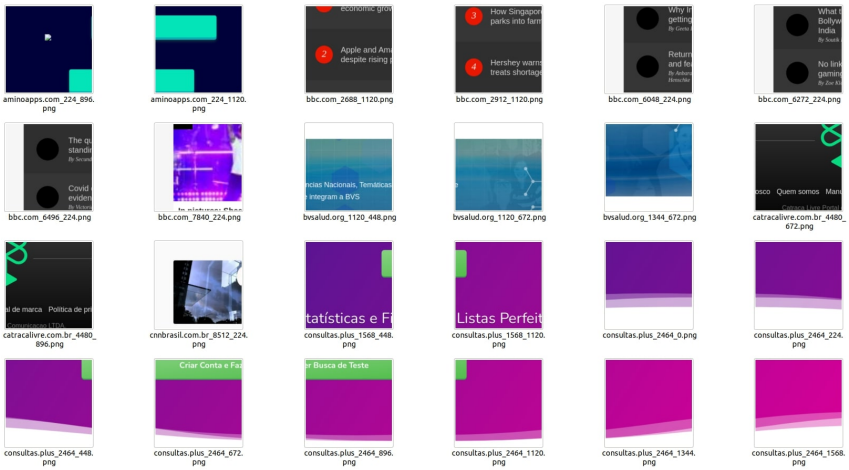


Figura 58 – Grupo 45 - Agrupou imagens com cores vivas e poucos textos.

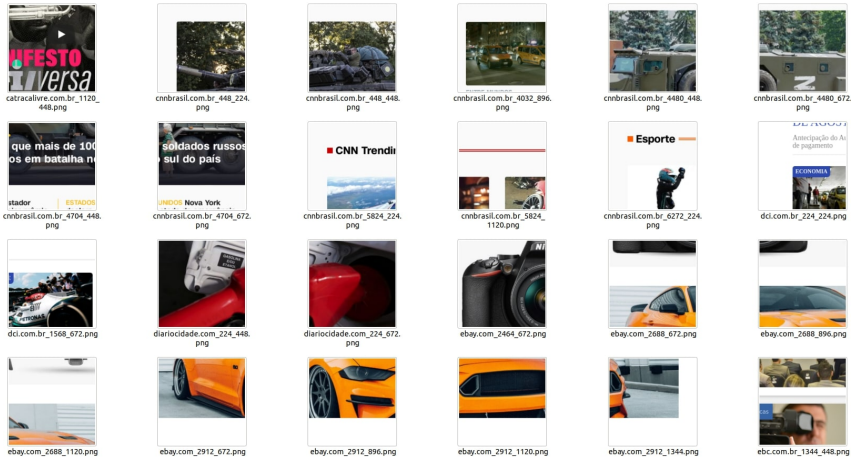


Figura 59 – Grupo 46 - Agrupou imagens com carros e outros veículos.

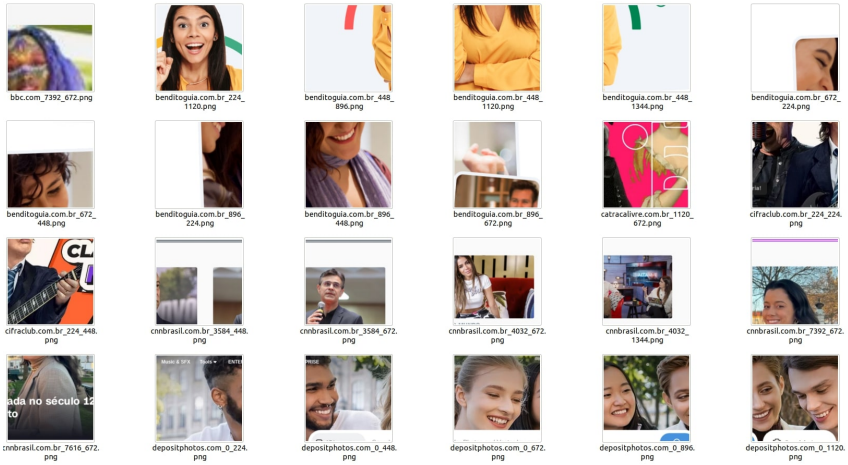


Figura 60 – Grupo 47 - Agrupou imagens com com rostos sorridentes e crianças.

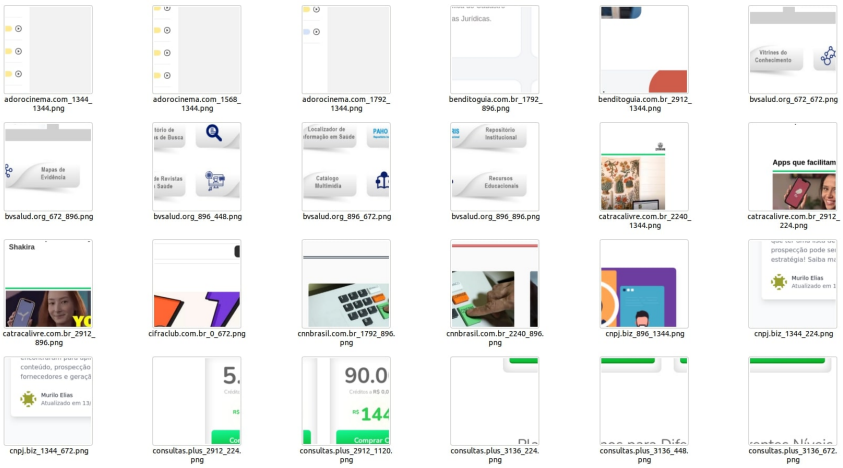


Figura 61 – Grupo 48 - Agrupou imagens com botões com contornos arredondados.

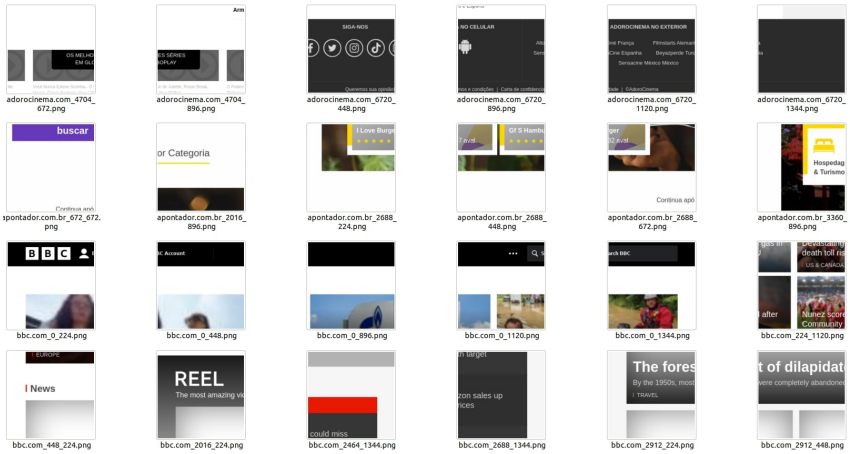


Figura 62 – Grupo 49 - Agrupou imagens com quadrados escuros e quinas retas.

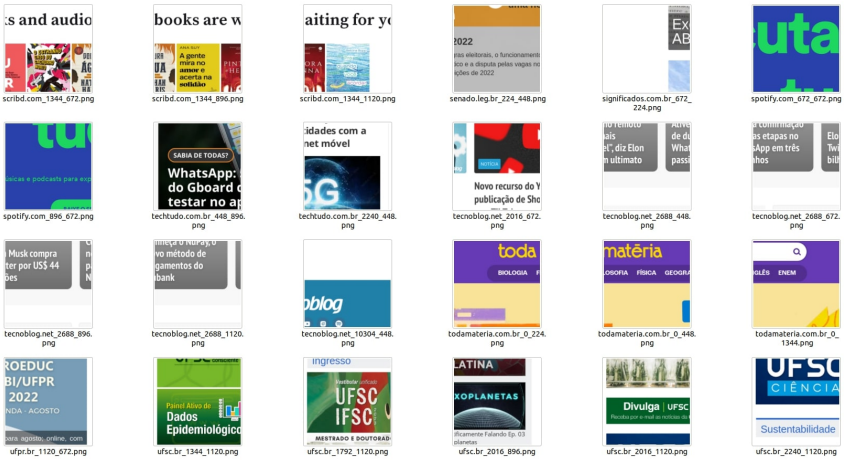


Figura 63 – Grupo 50 - Agrupou imagens com cores vivas e muito texto.


ICMC
USP
SÃO CARLOS