FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Análise percetual automática de imagens de fotojornalismo

Tiago Lima Dias Lavarinhas

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Pedro Miguel Machado Soares Carvalho

Coorientador: Luís António Pereira de Meneses Corte-Real

26 de Julho de 2018

Resumo

Atualmente as imagens possuem cada vez mais importância em vários domínios, nomeadamente no fotojornalismo, publicidade e entretenimento. Com a crescente relevância das imagens nos dias que correm torna-se crucial conseguir extrair o máximo de informação de uma imagem, facilitando o arquivamento e a reutilização de imagens.

No fotojornalismo as imagens têm como objetivo transmitir uma mensagem clara e objetiva e como tal torna-se importante conhecer quais as regiões percetualmente mais relevantes neste tipo de imagens. Deste modo, neste projeto desenvolveu-se um sistema de análise contextual de imagem que permite efetuar a classificação semântica da imagem, detetar objetos e cores dominantes. Por outro lado, desenvolveu-se um detetor automático de regiões percetualmente mais relevantes em imagens de fotojornalismo.

Para efetuar o detetor de zonas de interesse em imagens de fotojornalismo foi necessário criar um *dataset* constituído por imagens de fotojornalismo e, como tal, desenvolvou-se uma ferramenta de anotação para obter as regiões percetualmente mais relevantes nas imagens que constituem o *dataset*.

Abstract

Nowadays, images are becoming more important in several areas like photojournalism, advertising and entertainment. With the increasing relevance of images, it is crucial to extract the maximum information possible, facilitating the categorization and reusability of images.

In photojournalism the aim is to send a clear and objective message through the photography, for this reason, it becomes important to know which regions are the most relevant in this type of images. Therefore, a system of contextual image analysis was created, performing the semantic classification of an image detecting objects and dominant colors. On the other hand, an automatic detector of the most relevant regions in photojournalism images was also developed.

For the development of the previously stated method, the regions of interest detector, it was necessary to create a dataset composed of photojournalism images. To collect information about the most relevant regions in the dataset images, it was necessary to develop an annotation tool.

Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus pais por fazerem de mim a pessoa que sou hoje. Obrigado por todo o apoio ao longo dos anos e por isso dedico-lhes este trabalho.

Aos meus orientadores Pedro Carvalho e Luís Corte-Real por estarem sempre disponíveis para tirar dúvidas e para mostrar qual o melhor caminho a seguir.

Ao Américo Pereira e à Inês Teixeira, do INESC, por estarem sempre disponíveis para ajudar em caso de necessidade.

À minha namorada, Carolina, por me conseguir sempre incentivar e motivar nos momentos mais difíceis.

Por último, aos meus amigos pelo apoio e amizade e por me acompanharem ao longo deste percurso.

"I have not failed.
I've just found 10,000 ways that won't work."

Thomas A. Edison

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

API	*Application programming interface*
AP	*Average Precision*
ASM	*Angular Second Moment*
CNN	*Convolutional Neural Networks*
GLCM	*Gray Level Co-occurrence Matrix*
GLD	*Gray-Level Difference*
LBP	*Local Binary Patterns*
mAP	*mean Average Precision*
MCDNN	*Multi-Column Deep Convolutional Neural Networks*
MLRBM	*Multi-Layer Restricted Boltzmann Machines*
NMS	*non-maximum supression*
RCNN	*Region-Based CNN*

Capítulo 1

Introdução

1.1 Contexto

Presentemente, as imagens são imprescindíveis em diversos domínios, nomeadamente jornalismo, publicidade, entretenimento e no arquivo de conteúdo áudio-visual. Tendo em consideração o valor atual das imagens torna-se crucial descrever e armazenar o conteúdo presente nas imagens, facilitando a sua reutilização.

A extração de metadados de uma imagem é uma tarefa demorada, cara e intensiva, quando efetuada manualmente. Nos casos em que a anotação é realizada manualmente está dependente de variados fatores, por exemplo, o propósito da anotação, o domínio de aplicação, o estado de espírito e a personalidade do anotador. Por sua vez, a anotação automática ou semi-automática revelou ser mais económica e rápida para a obtenção de *ground truth*, sobretudo na presença de *datasets* extensos.

Inicialmente, as imagens eram anotadas através de textos descritivos. Através destes textos, as imagens poderiam ser organizadas por tópicos semânticos de modo a facilitar a pesquisa por imagens em bases de dados. Mas, as anotações recorrendo a texto requerem anotação manual o que é um processo exaustivo e dispendiosos em *datasets* de grande dimensão. Devido ao aumento do número de imagens, as dificuldades em anotar recorrendo a texto aumentaram, portanto, foi necessário encontrar uma alternativa, surgindo assim as anotações baseadas em caraterísticas de baixo nível, por exemplo, cor, textura, formas e conteúdos visuais.

Em Redmon et al. (2016) foi proposto utilizar uma *Convolutional Neural Network* (CNN) para detetar objetos presentes numa imagem. Do mesmo modo, outros algoritmos para a deteção de imagens foram identificados por Borji et al. (2015).

Com a crescente importância da deteção e classificação de imagens, surgiram aplicações comerciais para o efeito, nomeadamente a Google Vision API, Microsoft Vision API, Amazon Rekognition e Clarifai. Estas *Application Program Interfaces* (*API*) fornecem informação contextual sobre a imagem, por exemplo, objetos presentes; deteção de faces (podendo retornar idade, género e expressão facial); cores dominantes e descrições semânticas sobre imagem. Outras funcionalidades normalmente presentes nestas *API* são a deteção de celebridades; deteção de estruturas

populares (naturais ou não); presença de conteúdo para adultos ou violento.

1.2 Motivação

As *APIs* de classificação de imagem não apresentam tudo o que encontram numa imagem, apenas as etiquetas que apresentam mais confiança. Por outro lado, os detetores de objetos apresentam os objetos que fazem parte das suas classes e podem não detetar o objeto mais relevante numa imagem. Portanto, um classificador percetual seria importante, pois identificaria as zonas que chamam mais à atenção duma pessoa que visualiza a imagem. Se as *APIs* de classificação de imagem fossem constituídas por um classificador percetual só era necessário identificar os objetos presentes nas zonas de interesse melhorando a etiquetagem das imagens.

1.3 Objetivos

O objetivo desta dissertação foi desenvolver um sistema capaz analisar uma imagem, retornando objetos nela presentes, faces, caraterísticas de baixo nível e etiquetas descritivas da imagem. Por outro lado pretendeu-se desenvolver um detetor de regiões de interesse orientado para imagens de fotojornalismo.

1.4 Estrutura da dissertação

No próximo capítulo será apresentada a revisão de literatura, onde são abordados os algoritmos e ferramentas relacionadas com o problema a tratar. No capítulo 3 é explicitado o funcionamento e estrutura do sistema de análise contextual de imagem desenvolvido. De seguida, no capítulo 4, é abordado a caraterização do *dataset* e a forma como foi obtido, bem como as respetivas anotações. No caítulo 5 é efetuada uma análise das caraterísticas e objetos percetualmente mais relevantes. Além disso, no capítulo 6,são apresentados os resultados da identificação automática de regiões de interesse. Por fim, no capítulo 7 ,são apresentadas as conclusões do trabalho desenvolvido e dos respetivos resultados.

1.5 Contribuições

Deste projeto surge um *dataset* constituído por imagens de fotojornalismo, eventos e moda com anotações das regiões percetualmente mais relevantes nessas mesmas imagens. Surge também uma ferramenta para a anotação de imagens que permite a um utilizador selecionar retângulos nas regiões mais relevantes da imagem e guarda essas anotações num ficheiro XML.

Capítulo 2

Revisão bibliográfica

Neste capítulo, serão apresentadas diferentes plataformas de classificação de imagem, detetores de objetos e faces essenciais para retirar informação duma imagem. De seguida, são caraterizados vários *datasets* orientados para algoritmos de aprendizagem máquina. Depois, são descritos alguns algoritmos usados para a extração de caraterísticas de baixo nível da imagem. Além disso, são abordados alguns algoritmos de classificação de imagem. Por último, é descrito o funcionamento da atenção visual humana, bem como a aplicação de modelos baseados na atenção visual.

2.1 Plataformas de classificação de imagem

A principal função das plataformas de classificação de imagem é de identificar objetos e atividades presentes na imagem, por forma a descreve-la. Normalmente, estas *Application programming interface* (*APIs*) apresentam etiquetas que descrevem a imagem, ou seja, efetuam a descrição semântica da imagem. Para tal, estas *APIs* recorrem a algoritmos aprendizagem máquina para etiquetar a imagem de forma rápida. Para o utilizador saber se pode aceitar as etiquetas pode visualizar uma percentagem de confiança na etiqueta exposta.

Figura 2.1: Imagem de teste retirada de Pexels

2.1.1 Google Vision API

As etiquetas consistem nos elementos presentes na imagem, mas algumas das etiquetas são mais genéricas permitindo caraterizar a imagem. A plataforma de classificação da Google® (segundo a página da *API* (Google)) apresenta um conjunto de etiquetas que permitem caraterizar a imagem. Para cada etiqueta é retornado o grau de confiança do algoritmo. Caso a imagem contenha faces de pessoas, a *API* apresenta o tipo de expressão facial (e a percentagem de confiança associada à expressão facial) e os ângulos *roll*, *tilt*, *pan*, permitindo conhecer a inclinação da cabeça. A plataforma apresenta ainda um conjunto de cores dominantes (e as respetivas percentagens) e algumas proporções adequadas para o corte da imagem. Caso na imagem esteja presente um edifício/local ou até mesmo um logótipo popular, este é detetado pela plataforma. O algoritmo consegue detetar e extrair texto da imagem e ainda tem outras funcionalidades como verificar se o conteúdo é violento ou para adultos.

Para verificar a eficácia da *Google Vision API* testou-se a plataforma com imagem da Figura 2.1 e obtiveram-se os resultados apresentados na Figura 2.2. Podemos reparar na Figura 2.1 que o que sobressai na imagem é a mulher, principalmente a face e a camisola, mas nos resultados obtidos as primeiras três etiquetas, sendo elas "beleza" (o que é um pouco subjetivo), "lábio" e "fotografia", não são o que mais se destaca na imagem. Pela Figura 2.2 pode-se confirmar algumas das funcionalidades anteriormente explicitadas, nomeadamente a *API* realizou a etiquetagem da foto, apresentou a expressão facial da face identificada e um conjunto de cores dominantes da imagem. Pode-se concluir, quando comparando a imagem com os resultados obtidos, que a plataforma consegue identificar os objetos e caraterísticas na imagem corretamente.

As etiquetas apresentadas pela plataforma são genéricas mas cliente pode precisar de uma classificação num contexto mais específico. Para tal, são disponibilizadas bibliotecas em várias linguagens de programação e oferecendo a possibilidade de o cliente desenvolver código que satisfaça as suas necessidades.

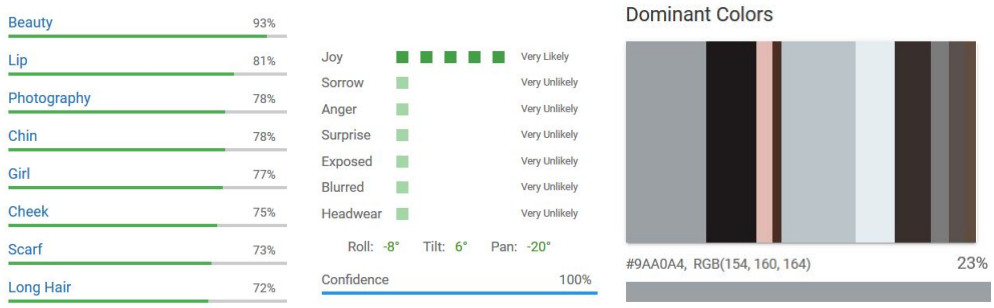


Figura 2.2: Resultados apresentados pela *API* da Google® quando testada com a Figura 2.1

Apesar dos bons resultados apresentados pela plataforma ainda existe espaço para melhoramento. No que diz respeito a imagens que contêm ruído o algoritmo não consegue identificar o objeto, apesar de ser facilmente identificado por uma pessoa. Este facto foi comprovado por Hosseini et al. (2017), que iludiu o *software* adicionando ruído *salt and pepper* de tal forma que

os resultados apresentados pela *API* sejam incorretos, apesar do objeto ser identificável por um humano.

2.1.2 Computer Vision API

A *API* da Microsoft® (Microsoft), contrariamente à plataforma da Google®, consegue realizar a classificação em vídeo (em tempo quase real), além da classificação de imagem. Além de efetuar a classificação da imagem, a *API* insere a imagem numa das 86 categorias presentes na Figura 2.3 e descreve a imagem com uma frase em inglês. Caso esteja presente uma face na imagem, a *API* retorna a idade e o género da pessoa e as coordenadas da face na imagem. Contrariamente à Google® que apresenta um conjunto de cores dominantes, a plataforma da Microsoft® apresenta apenas uma cor dominante para o fundo, para o primeiro plano e a cor que fornece realce à imagem. 'É ainda verificado se a imagem é um *clip-art* ou um desenho de linha. Além das funcionalidades mencionadas ainda existem outras, nomeadamente: deteção de conteúdo violento ou para adultos; deteção de celebridades locais/edifícios populares; e deteção e extração de texto em imagens.

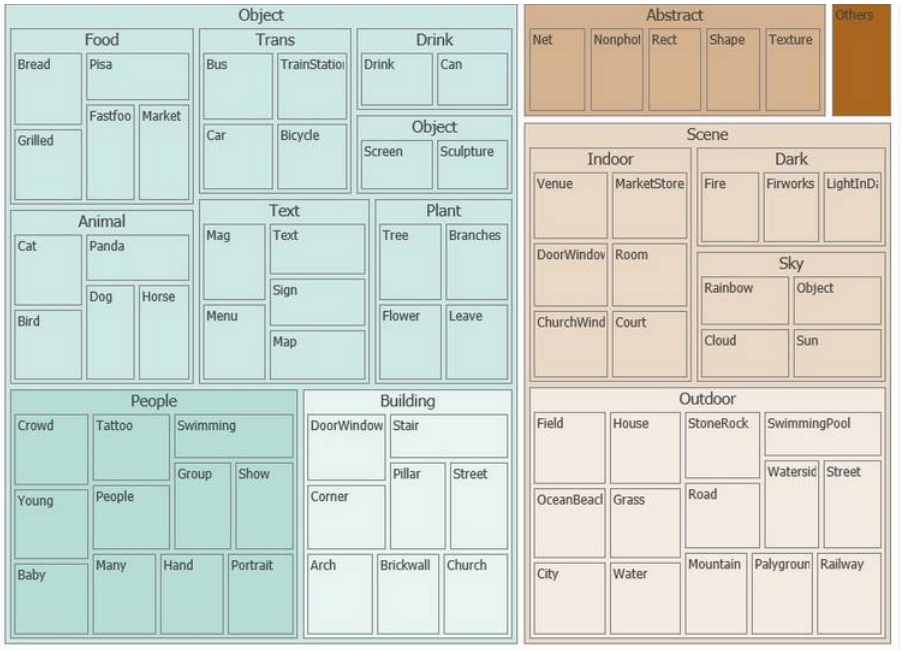


Figura 2.3: Conjunto de categorias usadas pela *Vision API* da Microsoft®
(retirado de Microsoft)

Recorreu-se à imagem da figura 2.1 para visualizar os resultados apresentados pela *API*, resultados que podem ser observados nas imagens da figura 2.3. Constatou-se que apesar dos *labels* apresentados diferirem um pouco dos exibidos pela *API* da Google continuam coerentes com a imagem de teste. A *API* apresenta, além das etiquetas, uma frase para descrever a imagem. Neste caso, a plataforma retornou "*a woman smiling for the camera*"o que descreve de a imagem 2.1 forma precisa. A *vision API* da Google apresenta um conjunto de cores dominantes, por sua vez,

a *API* da Microsoft apenas apresenta apenas a cor dominante do primeiro plano, do fundo e a cor de realce.

FEATURE NAME:	VALUE
Description	{ "tags": ["clothing", "person", "outdoor", "woman", "smiling", "wearing", "standing", "sitting", "young", "posing", "holding", "white", "phone", "street", "girl", "table", "umbrella", "suit", "city", "shirt"], "captions": [{ "text": "a woman smiling for the camera", "confidence": 0.9530815 }] }
Tags	[{ "name": "clothing", "confidence": 0.993031859 }, { "name": "person", "confidence": 0.97089684 }]
Image format	"Jpeg"
Image dimensions	183 x 275
Clip art type	0
Line drawing type	0
Black and white	false
Adult content	false
Adult score	0.0132834669
Racy	false
Racy score	0.0163911469
Categories	[{ "name": "people_portrait", "score": 0.796875 }]
Faces	[{ "age": 21, "gender": "Female", "faceRectangle": { "top": 14, "left": 99, "width": 86, "height": 86 } }]
Dominant color background	"Grey"
Dominant color foreground	"Grey"
Accent Color	#5C686F

Figura 2.4: Resultados apresentados pela plataforma para a imagem da figura 2.1

2.1.3 Amazon Rekognition

A *API* da Amazon® não só faz a classificação de imagens com também classifica vídeo conseguindo efetuar seguimento de pessoas e até permite fazer a sequência temporal das pessoas ao longo dum vídeo. As caraterísticas da plataforma são descritas de seguida de acordo com a informação fornecida na página da *API* (Amazon). Esta plataforma identifica pessoas em fotos ou vídeos, desde que as pessoas façam parte do repositório do utilizador. A partir de uma face detetada a *API* permite saber algumas caraterísticas da pessoa nomeadamente, o género, a gama de idades, a expressão facial, se os olhos estão abertos. É possível ainda, construir a sequência das emoções da pessoa através das caraterísticas apresentadas ao longo do tempo. A *API* tem algumas funcionalidades referentes à análise de vídeo, sendo uma delas seguir o movimento das pessoas, mesmo que não lhes esteja a observar a cara. Tal como as *APIs* anteriores esta extrai as cores dominantes; deteta conteúdo violento ou para adultos; deteta celebridades na imagem; e deteta e extrai de texto de imagens. Não foi possível testar a *Amazon Rekognition API* uma vez que esta só o permite a quem já é utilizador.

2.1.4 Clarifai

De acordo com (Clarifai), a plataforma efetua a classificação de imagem e vídeo; analisa faces (género, idade e origem cultural); deteta conteúdo violento ou para adultos; deteta e extrai texto em imagens; e deteta de logótipos populares na imagem. Por outro lado, a Clarifai também tem modelos que permitem dar uma classificação especifica num determinado cenário, por exemplo, casamentos, viagens e comida.

Os resultados desta plataforma são idênticos aos apresentados pelas outras plataforma e portanto de acordo com a imagem. Tal pode ser verificado pelas imagens da figura 2.5.

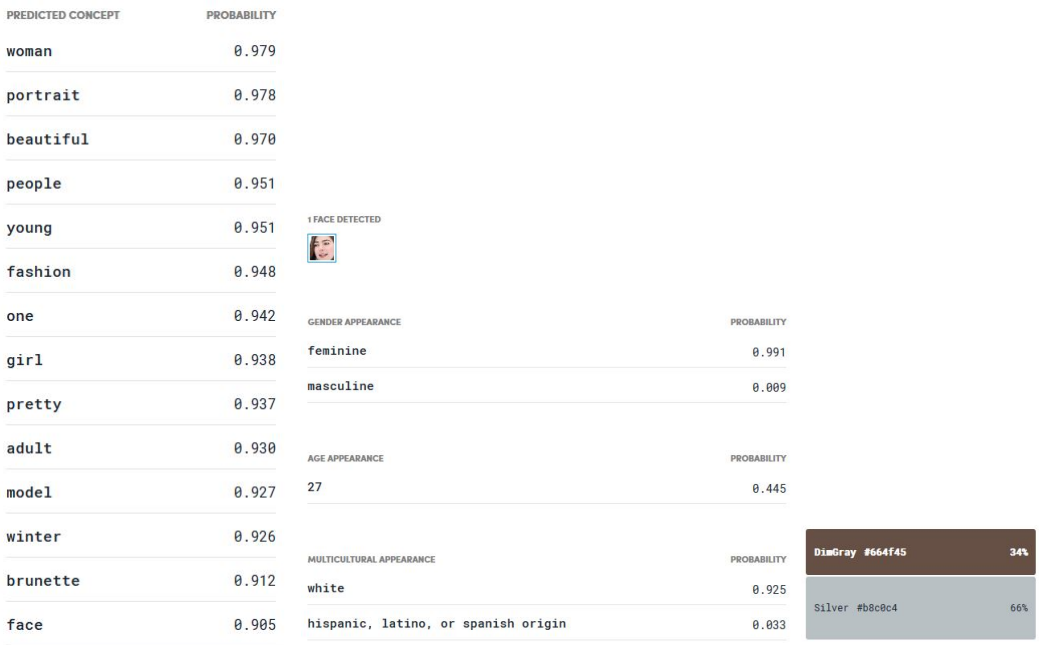


Figura 2.5: Resultados apresentados pela plataforma para a imagem da figura 2.1

2.1.5 Comparação entre plataformas de classificação

Na classificação de imagem as plataformas comportam-se de forma idêntica, identificando os elementos presentes, apresentando bons resultados quando comparando com a imagem. De salientar, que a *API* da Microsoft® que descreve a imagem numa frase além de a etiquetar. A deteção e análise de faces é um aspeto importante na caraterização da imagem e como tal é uma funcionalidade presente nas plataformas analisadas. Na análise da face, a plataforma da Google® é a única a apresentar o ângulo em que se encontra a cabeça, mas não apresenta informação sobre a idade da pessoa. As caraterísticas de baixo nível, como a cor dominante também contribuem para caraterizar uma imagem e como tal é algo apresentado por todas as *APIs*. Na tabela 2.1 encontra-se uma síntese das funcionalidades de cada plataforma que anteriormente foram mencionadas.

Tabela 2.1: Resumo das caraterísticas das plataformas de classificação

	Google®	**Microsoft®**	**Amazon®**	**Clarifai®**
Classificação de imagem	Sim	Sim	Sim	Sim
Classificação de vídeo	Não	Sim	Sim	Sim
Deteção de faces	Sim	Sim	Sim	Sim
Cor dominante	Sim	Sim	Sim	Sim
Conteúdo explicito	Sim	Sim	Sim	Sim
Pontos de referência	Sim	Sim	Não	Sim
Deteção de texto	Sim	Sim	Sim	Sim
Deteção de logótipos	Sim	Sim	Não	Sim
Deteção de celebridades	Não	Sim	Sim	Não
Seguimento de pessoas	Não	Não	Sim	Não

2.2 Ferramentas de deteção de objetos e classificação de imagens

2.2.1 YOLO

YOLO (Redmon and Farhadi, 2018) é uma biblioteca para deteção de objetos em imagens ou vídeo. A maioria dos algoritmos de deteção de objetos aplicam o seu modelo a várias regiões da imagem, em que se assume a presença de um objeto nas regiões que apresentam maior pontuação. O YOLO por outro lado, aplica uma simples rede neuronal a toda a imagem. Esta rede decompõe a imagem em várias regiões com uma probabilidade associada. Portanto o YOLO retorna as coordenadas das caixas delimitadoras de cada região, a categoria em que o objeto foi inserido e a respetiva probabilidade. É ainda importante evidenciar que o YOLO só apresenta os objetos cuja probabilidade seja superior a um *threshold* a definir pelo utilizador. O facto de ser aplicada uma rede neuronal à imagem completa faz com que a deteção seja rápida quando comprando com outros algoritmos aprendizagem máquina, nomeadamente o *R-CNN* e *fast R-CNN* . O YOLO possui uma rede pré treinada, mas também permite treinar uma rede caso o utilizador queira utilizar o YOLO num cenário mais especifico.

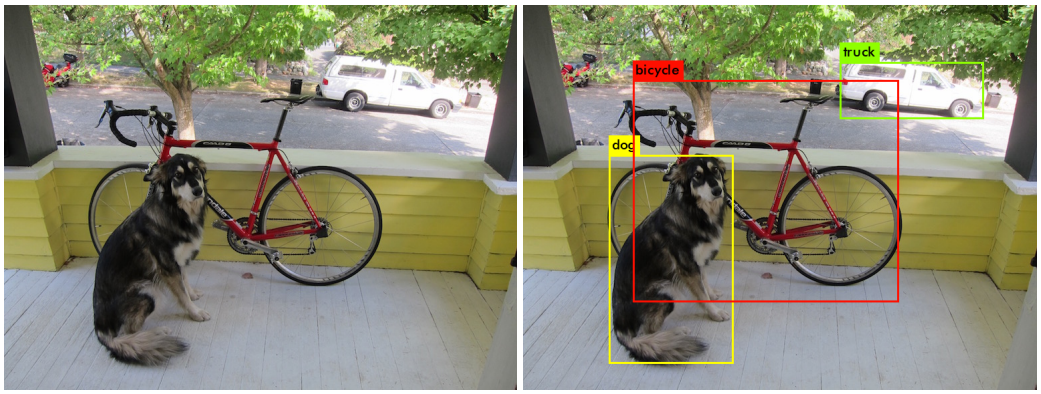


Figura 2.6: Resultados da aplicação do YOLO (imagem retirada de Redmon and Farhadi (2018))

Na Figura 2.6 pode ser visualizados resultados apresentados pelo YOLO. Para este caso, o YOLO detetou o cão, a bicicleta e a carrinha com confianças de 99%, 99% e 92% respetivamente.

2.2.2 Dlib

Dlib (King, 2009) é uma biblioteca de C++ contendo implementações de algoritmos de aprendizagem máquina com vista a resolver problemas reais. É uma biblioteca com várias áreas de aplicação, nomeadamente robótica, sistemas embebidos e telemóveis, sendo utilizada quer na industria quer para fins de investigação. Dos exemplos disponíveis, os de deteção de faces são os mais relevantes para o objetivo pretendido. A partir destes exemplos é possível obter as coordenadas dos retângulos que delimitam as faces presentes numa imagem, recorrendo a uma *CNN*. Além desta função, é possível obter as coordenadas de 5 ou 68 pontos de interesse numa face, como se pode verificar à esquerda e à direita na Figura 2.7, respetivamente.

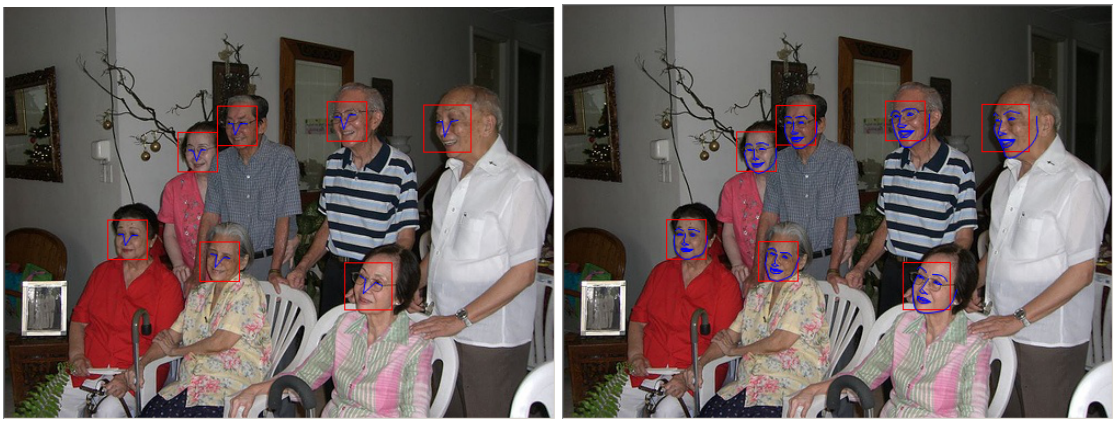


Figura 2.7: Deteção de faces e 5 (esquerda) ou 68 (direita) pontos de interesse

Figura 2.8: Imagem de panda, retirada de TensorFlow (2008)

2.2.3 TensorFlow

TensorFlow TensorFlow (2008) é uma biblioteca orientada para o uso de algoritmos aprendizagem máquina. O TensorFlow possui uma rede pré treinada para a classificação de imagem, com o *dataset* ImageNet. Na Tabela 2.2 podemos analisar os resultados do teste do TensorFlow à Figura 2.8. A etiqueta com maior confiança é "panda", que é a única coisa presente na imagem, portanto o resultado apresentado é correto.

Tabela 2.2: Classificação da Figura 2.8 pelo TensorFlow

giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca (*score* = 0.88493)
indri, indris, Indri indri, Indri brevicaudatus (*score* = 0.00878)
lesser panda, red panda, panda, bear cat, cat bear, Ailurus fulgens (*score* = 0.00317)
custard apple (*score* = 0.00149)
earthstar (*score* = 0.00127)

2.3 *Datasets*

Tendo em conta o facto de os algoritmos aprendizagem maquina necessitarem de muita informação para conseguirem obter bons resultados é importante proceder ao levantamento de *datasets*

existentes nesta área. Como tal, de seguida serão descritos alguns *datasets* normalmente usados para classificação de imagem.

O *dataset* ImageNET (Russakovsky et al., 2015) está organizado de acordo com a hierarquia WordNet (Miller, 1995). WordNet é um *dataset* léxico orientado para a computação que organiza nomes, verbos, adjetivos e advérbios em conjuntos de sinónimos, representando um conceito. Existem mais de 100,000 conceitos no WordNet e o ImageNet tem com objetivo obter uma média de 1000 imagens para ilustrar cada conceito. O ImageNET não possui direitos sobre as imagens e como tal fornece o *URL* de cada imagem, mas é possível efetuar *download* das imagens para uso não comercial ou educacional. As imagens encontram-se anotadas em ficheiros *XML* de acordo com o formato PASCAL VOC (Everingham et al., 2010). Apesar de algumas imagens estarem anotadas apenas cerca de 3000 categorias/conceitos têm imagens anotadas sendo que por cada categoria apenas existem cerca de 150 imagens anotadas.

O projeto CAVIAR (CAVIAR, 2006) gravou pessoas em alguns cenários de interesse, formando um *dataset*. Foram gravadas pessoas a caminhar e encontando-se com outras, entrando e saindo de lojas e a deixar um pacote em público . O vídeo contém 25 frames por segundo com uma resolução de 384 x 288 pixeis e foi comprimindo usando MPEG2. O *dataset* contém anotações relativas ao vídeo em ficheiros XML.

O *dataset* Microsoft COCO (Lin et al., 2014) contém imagens de objetos comuns num determinado contexto. O *dataset* está dividido em 80 categorias de objetos e 1,5 milhões de instâncias desses objetos. As imagens foram segmentadas e as anotações gravadas em ficheiros JSON. Cada imagem tem, pelo menos, 5 frases que descrevem o seu conteúdo.

Fashion-MNIST (Zalando) é um *dataset* composto por 60,000 imagens para treino e 10,000 para teste. Cada imagem tem uma resolução de 28x28 e encontra-se em tons de cinzento. Os ficheiros de treino e teste são constituídos por 785 colunas, em que cada linha corresponde a uma imagem. O valor da primeira coluna corresponde a um de 10 categorias (t-shirt/top; claças; camisola; vestido; casaco; sandália; camisa; ténis; mala e bota) enquanto as restantes 784 (28x28) colunas correspondem à intensidade de cada pixel na imagem.

DAVIS (Caelles et al., 2018; Pont-Tuset et al., 2017; Perazzi et al., 2016) é um *dataset* constituído por frames de vídeo. A resolução das imagens é 4k ou 1080p, mas também existe a possibilidade de se efetuar *download* em 480p. Este *dataset* também contém anotações das imagens do vídeo que consistem em imagens binárias em que o objeto segmentado se encontra a branco. Existe um total de 90 categorias e apenas um vídeo por categoria.

O projeto Pascal VOC (Everingham et al., 2015) criou várias competições de pesquisa na área de reconhecimento de objetos em imagens. Para tal, foi criado o Pascal VOC *dataset* que na ultima versão é constituído por 20 categorias; 11530 imagens com 27450 regiões de interesse anotadas e 6929 segmentações.

LabelMe *dataset* (Russell et al., 2008) é constituído por um conjunto de imagens para treino e para teste. O conjunto de treino contém cerca de 1000 imagens completamente anotadas e cerca de 2000 imagens parcialmente anotadas. Este conjunto é constituído por 2920 imagens; 32164 objetos; 4441 carros; 2524 pessoas; 3004 edifícios; 1321 estradas; 1272 passeios; 1009 céu e 2652

árvores. Quanto ao conjunto de teste é constituído por 1133 imagens (completamente anotadas); 32853 objetos; 2265 carros; 2119 pessoas; 2117 edifícios; 739 estradas; 1107 passeios; 823 céu e 1652 árvores.

MIT *Indoor Scene Recognition* é um *dataset* orientado para algoritmos aprendizagem máquina que surge devido ao facto da maioria dos algoritmos de reconhecimento de cenários funcionar corretamente ao ar livre mas falha em espaços fechados. Este *dataset* é constituído por 67 categorias e um total de 15620 imagens. O número de imagens por categoria é variante, mas sempre superior a 100. Um subconjunto dessa imagens estão anotadas e segmentadas com os objetos nelas contidos (as anotações estão no formato LabelMe).

Uma vez que o cenário de aplicação é o fotojornalismo estes *datasets* não puderam ser utilizados uma vez que não possuem imagens desse contexto.

2.4 Extração de caraterísticas de baixo nível

2.4.1 Cor dominante

A cor dominante, apesar de ser uma caraterística de baixo nível da imagem, pode fornecer alguma informação sobre a imagem. Segundo o algoritmo de Yu et al. (2010), de modo extrair as cores dominantes pode-se efetuar através de um conjunto de etapas. Em primeiro lugar, seleciona-se um conjunto de n cores dominantes (C_1, C_2, ..., C_n), no espaço de cores *RBG*. Naturalmente, as imagens são constituídas por uma elevada quantidade de cores tendo interesse conhecer apenas as Q cores dominantes que representam a maioria da imagem. O algoritmo é um processo iterativo, em que na iteração j a cor C_j é agrupada com a primeira cor C_{j+1}, C_{j+2} ,...,C_n tal que a distância euclidiana (mas sem efetuar a raiz quadrada) das componentes R, G e B das duas cores sejam inferiores a um *threshold* previamente definido. Deste modo, a cor C_j será atualizada, em que as novas componentes são calculadas através da média ponderada das componentes tendo em conta a sua probabilidade de ocorrência, e a nova probabilidade da cor agrupada será a soma das duas probabilidades. Caso não seja agrupada nenhuma cor o *threshold* é incrementado em 100. O processo é repetido até que as primeiras Q do vetor correspondam a 70% do número de pixeis da imagem.

O método de extração da cor dominante proposto em Shao et al. (2008) é efetuado no espaço de cores *HSV* contrariamente ao algoritmo anterior que o realiza em *RGB*. Posteriormente à conversão para *HSV* realiza-se uma quantização de acordo com as equações apresentadas de seguida.

$$H = \begin{cases} 0 & if \quad h \in [316, 20[\\ 1 & if \quad h \in [20, 40[\\ 2 & if \quad h \in [40, 75[\\ 3 & if \quad h \in [75, 155[\\ 4 & if \quad h \in [155, 190[\\ 5 & if \quad h \in [190, 270[\\ 6 & if \quad h \in [270, 295[\\ 7 & if \quad h \in [295, 316[\end{cases} S = \begin{cases} 0 & if \quad s \in [0, 0.2] \\ 1 & if \quad s \in]0.2, 0.\gamma] \\ 2 & if \quad s \in]0.\gamma, 1] \end{cases} V = \begin{cases} 0 & if \quad v \in [0, 0.2] \\ 1 & if \quad v \in]0.2, 0.7] \\ 2 & if \quad v \in]0.7, 1] \end{cases}$$

Como se observa pelas equações, depois de efetuada a quantização a imagem passa a ser representada apenas por 72 cores. Posteriormente, calcula-se o histograma da imagem para se conhecer a probabilidade de cada pixel correspondente a cada uma das 72 cores. De seguida, são normalizadas as probabilidades para as primeiras M cores dominantes.

2.4.2 Textura

A extração de textura é uma abordagem comum para caraterizar a distribuição espacial das intensidades na imagem (Dixit and Hegde, 2013). A definição de textura não é consensual, mas pode ser definida como repetições ou padrões contidos na imagem.

2.4.2.1 Métodos estatísticos

Os métodos estatísticos calculam um conjunto de estatísticas através da análise da distribuição de caraterísticas na imagem. Estes métodos podem ser classificados consoante o número de pixeis utilizados na estimação das propriedades estatísticas (Dixit and Hegde, 2013). Deste tipo de métodos, alguns são usados com mais recorrência, nomeadamente *GLCM* (*Gray Level Co-occurrence Matrix*), *GLD* (*Gray-Level Difference*) e *LBP* (*Local Binary Patterns*).

GLCM

O algoritmo *GLCM* é um dos métodos estatísticos mais usados, sendo uma das razões a sua fácil implementação (Saroja and Sulochana, 2013; Dixit and Hegde, 2013). Este método contém informação que permite conhecer a posição dos pixeis com intensidades similares (em tons de cinzento), permitindo reconhecer padrões e assim extrair a textura. Por norma, costuma-se obter quatro matrizes, assumindo que a textura terá uma das orientações 0^o, 45^o, 90^o, 135^o . Existe um conjunto de caraterísticas das medições, nomeadamente *Angular Second Moment* (*ASM*), contraste, correlação e energia, que a partir das matrizes *GLC* permitem definir as caraterísticas da textura.

GLD

O método *GLD*, a partir da diferença de intensidades numa determinada direção, calcula a probabilidade de cada nível de cinzento. Este algoritmo, apresenta resultados similares para diferentes tamanho da imagem e tal pode ser verificado em experiências de Zhang et al. (2011) que para três imagens com uma textura de pele animal, com diferentes tamanhos, apresentaram as mesmas curvas de *GLD*. Apesar de ser um método bastante aplicado, não apresenta bons resultados quando a imagem tem presente ruído gaussiano.

LBP

O algoritmo *LBP*, de acordo com Li et al. (2016), baseia-se na comparação da intensidade, em tons de cinzento, do pixel central, da região a analisar, com as intensidades de *N* pixeis presentes numa vizinhança de raio R. Esta comparação tem como resultado 1 ou 0 caso a intensidade do pixel da vizinhança seja superior ou inferior à intensidade do pixel central, respetivamente. Concluindo, o valor resultante do *LBP*, para um determinado pixel central, é o somatório da conversão para decimal dos zeros e uns, anteriormente referidos, resultando num total de 2^P valores possíveis para o *LBP*.

2.4.2.2 Outros modelos

Os métodos geométricos assumem que a textura é constituída por texturas mais simples, descrevendo a sua organização espacial (Dixit and Hegde, 2013). Dentro destes métodos destacam-se *Laplacian-of-Gaussion* ou *difference of-Gaussian filter*.

Os métodos de processamento de sinal recorrem à análise da frequência da imagem para conhecer a textura bem como a várias máscaras para detetar contornos (Dixit and Hegde, 2013).

2.5 Classificação de imagem

Tendo em conta que para *datatsets* de grandes dimensões a classificação manual iria ser exaustiva e demorada, o uso de classificadores automáticos permite uma classificação rápida e eficaz. Um dos algoritmos aprendizagem máquina que tem obtido bons resultados é a *Convolutional Neural Networks* (*CNN*) (Guo et al., 2017).

(Guo et al., 2017) propôs uma simples *CNN* para a classificação de imagem e obteve um erro de apenas 0.66% nos seus resultados. Apesar deste resultado ser pior quando comparado com outros algoritmos, o facto de ser uma simples *CNN* permite efetuar a classificação em menor tempo apresentando resultados relativamente bons.

Devido à elevada precisão apresentada pelas *CNN*, surgem novos algoritmos, baseados nestas, de modo a obter melhores resultados. Por exemplo, (Ciregan et al., 2012) propuseram a *Multi-Column Deep Convolutional Neural Networks* (*MCDNN*) que consiste em usar várias *CNN*

e depois efetuar uma média dos resultados apresentados por cada. Os resultados obtidos apresentaram uma melhoria relativa de 30%, no mínimo, quando comparada com outros algoritmos e testando com vários *datasets*.

A deteção de faces continua a ser alvo de pesquisa devido à sua crescente importância no quotidiano. Em particular, a análise e deteção de faces é um aspeto importante quer na classificação de imagem quer noutras áreas como a vídeo-vigilância e a biométrica (Ye et al., 2015; Dahmane et al., 2017).

O método proposto por (Sun et al., 2017) utiliza *Region-Based CNN* (*RCNN*) para deteção de faces conseguindo obter resultados que outros métodos publicados para o mesmo efeito. Por outro lado, (Ye et al., 2015) recorreram a *Multi-Layer Restricted Boltzmann Machines* (*MLRBM*) na deteção de faces, apresentando eficácia de aproximadamente 85% e 93% para pequeno (30^o a 60^o) e grandes (60^o a 90^o) ângulos de rotação da face, respetivamente. Para reduzir a área da localização da face recorreu-se à deteção da cor da pele para acelerar a convergência do algoritmo. A partir duma *CNN* previamente treinada para reconhecer pessoas (Dahmane et al., 2017) conseguiu, através de mecanismos de transferência de aprendizagem, detetar o sorriso e o género da pessoa com performances de 90.69% e 88.14% respetivamente.

2.6 Atenção visual

Atenção visual é um mecanismo do sistema visual humano que foca a atenção em certas partes duma imagem, antes de mudar a atenção para outras regiões (Hu et al., 2008). Isto é, o humano seleciona gradualmente determinadas zonas para focar a sua atenção e combina a informação obtida nas regiões escolhidas por forma a recriar a cena observada, (Mnih et al., 2014; Rensink, 2000/01/). Recorrendo a caraterísticas como intensidade, cor, orientação e tamanho o córtex visual primário é capaz de selecionar a região para qual será focada a atenção através da informação sensorial recebida.

Em pesquisas anteriormente efetuadas mostraram que atenção visual segue duas abordagens de cima para baixo (*top-down*) e de baixo para cima (*bottom-up*). O mecanismo de cima para baixo é responsável pela rápida mudança da seleção da atenção para as caraterísticas visuais salientes, de provável importância. O mecanismo de baixo para cima depende de caraterísticas de baixo nível, como por exemplo, cor, textura e brilho. Por outro lado, a atenção de cima para baixo afirma que a informação de objetos domina a seleção da atenção, isto é, na presença de objetos conhecidos e e caraterísticas de baixo nível salientes a atenção será redirecionada para o objetos (Connor et al., 2004; Lu et al., 2012).

Atualmente, o desenvolvimento de modelos de atenção visual que simulam o sistema visual humano têm atraído cada vez mais interesse para a visão computacional devido ao facto de as caraterísticas visuais humanas serem capazes de selecionar o alvo de forma rápida e precisa, (Wu, 2017; Vikram et al., 2011; Cretu et al., 2015; Sun et al., 2013).

O uso de redes neuronais para a classificação de imagens e deteção de objetos tem apresentado elevada precisão, mas devido a altos custos computacionais no treino destes algoritmos. Amplas

CNNs, normalmente usadas, têm um tempo de treino que pode demorar dias, mesmo com recurso a múltiplas unidades de processamento gráfico *GPUs*, apesar de em alguns casos ser efetuado o *downsample* da imagem para reduzir o processamento necessário, (Krizhevsky et al., 2012). Tendo em consideração o elevado custo computacional, (Mnih et al., 2014) apresentou uma rede neuronal, baseada na atenção visual, que seletivamente escolhe uma região e processa em alta resolução apenas apenas às regiões selecionadas, em vez de processar a imagem toda de uma só vez. Este novo modelo, conseguiu superar as *CNNs* em tarefas de classificação de imagem. Mais modelos de atenção visual têm sido usados na classificação de imagem e deteção de objetos, por exemplo (Guo et al., 2014) desenvolveram um método de deteção de objetos baseado na atenção visual seletiva, e conseguiram obter precisão ao mesmo que outros algoritmos para o efeito, mas diminuindo a área de pesquisa drasticamente, efetuando a deteção mais rapidamente.

Capítulo 3

Análise contextual de imagens

Neste capítulo, será explicitado como foi desenvolvido do sistema de análise contextual de imagem, o seu funcionamento e os resultados por ele obtidos. Este sistema, consiste na integração de várias ferramentas que permitem retirar informação sobre uma imagem. Com este sistema, pretende-se efetuar a classificação semântica da imagem, deteção de faces e objetos e caraterísticas de baixo nível, nomeadamente cores dominantes. É esperado que da classificação de imagem que consiga identificar certos cenários normalmente presentes em imagens de fotojornalismo, por exemplo, cenários de guerra.

3.1 Sistema de análise de imagem

Como anteriormente mencionado, o sistema de análise de imagem consiste na junção de várias ferramentas para retirar conteúdo duma imagem. Para tal, recorreu-se às *APIs* da Clarifai (Clarifai) e Microsoft (Microsoft), o detetor de objetos YOLO (Redmon and Farhadi, 2018) e as bibliotecas Dlib (King, 2009) e TensorFlow (TensorFlow, 2008), mencionados no capítulo 2. De seguida, será clarificada a arquitetura deste sistema, como foi efetuada a integração das várias ferramentas e os respetivos *outputs*.

3.1.1 Arquitetura do sistema de análise de imagens

Primeiramente, é necessário fornecer ao sistema a imagem que se pretende analisar. De seguida, a imagem é passada como argumento das várias ferramentas que trabalham em paralelo. Depois de obtidos os resultados das várias *APIs* e bibliotecas, estes são guardados num ficheiro de texto que contém o mesmo nome da imagem. Para ilustrar o que foi anteriormente mencionado pode-se visualizar a arquitetura do módulo de análise de imagem na Figura 3.1. Enquanto as o YOLO, Dlib e TensorFlow correm localmente, as *APIs* da Clarifai e Mircosoft efetuam o processamento na *cloud* e como tal é necessário fazer *upload* da imagem.

O YOLO é responsável pela deteção de objetos na imagem. Como o YOLO foi treinado com o COCO *dataset* (Lin et al., 2014) os objetos que por ele identificados correspondem às categorias deste *dataset*. Portanto, o YOLO deteta 80 tipos de objetos, como por exemplo, pessoas, bicicletas

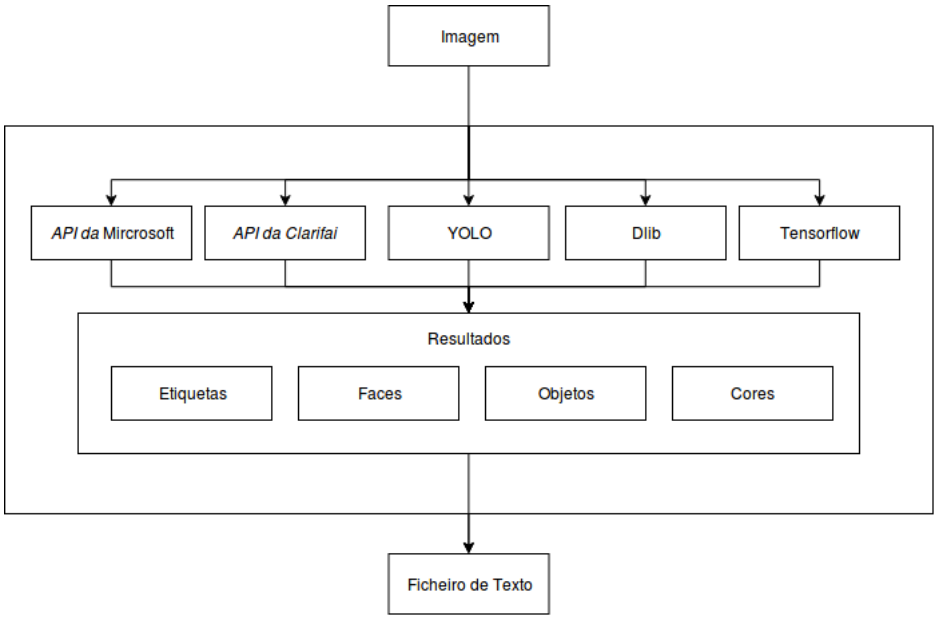


Figura 3.1: Arquitetura do módulo de análise de imagem

e carros. A deteção de faces é efetuada com recurso ao Dlib e etiquetagem da imagem é efetuada graças às*APIs* da Clarifai, Microsoft e TensorFlow.

3.1.2 Formato dos dados retornados pelas ferramentas


```
"data": {
  "concepts": [
    {
      "id": "ai_HLmqFqBf",
      "name": "train",
      "app_id": null,
      "value": 0.9989112
    },
    {
      "id": "ai_fvlBqXZR",
      "name": "railway",
      "app_id": null,
      "value": 0.9975532
    },
    {
      "id": "ai_Xxjc3MhT",
      "name": "transportation system",
      "app_id": null,
      "value": 0.9959158
    },
    {
      "id": "ai_6kTjGfF6",
      "name": "station",
      "app_id": null,
      "value": 0.992573
    },
```


Figura 3.2: Exemplo do formato dos dados fornecidos pela *API* da Clarifai, retirado de (Clarifai)

Anteriormente à integração das várias ferramentas foi necessário conhecer o formato em que os dados eram retornados. A *API* da Clarifai apresenta a informação resultante da análise da imagem em formato JSON como pode ser visualizado na Figura 3.2. Do mesmo modo, a *API* da Microsoft também apresenta as suas etiquetas no formato JSON, Figura 3.3. Por outro lado, o YOLO guarda as deteções num vetor em que cada elemento contém o objeto detetado, as coordenadas da caixa delimitadora do obejto e a respetiva confiança. O Dlib retorna dois objetos por cada face detetada, um dos objetos contêm as coordenadas caixas delimitadoras das faces detetadas e outro com as coordenadas de pontos de interesse na face. O TensorFlow retorna um objeto com as etiquetas e as respetivas confianças.

FEATURE NAME:	VALUE
Description	{ "tags": ["train", "platform", "station", "building", "indoor", "subway", "track", "walking", "waiting", "pulling", "board", "people", "man", "luggage", "standing", "holding", "large", "woman", "yellow", "suitcase"], "captions": [{ "text": "people waiting at a train station", "confidence": 0.833099365 }] }
Tags	[{ "name": "train", "confidence": 0.9975446 }, { "name": "platform", "confidence": 0.995543063 }, { "name": "station", "confidence": 0.9798007 }, { "name": "indoor", "confidence": 0.927719653 }, { "name": "subway", "confidence": 0.838939846 }, { "name": "pulling", "confidence": 0.431715637 }]

Figura 3.3: Exemplo do formato dos dados fornecidos pela *API* da Microsoft, retirado de (Microsoft)

3.1.3 Fusão de etiquetas

Uma caraterística importante num classificador de imagem é não apresentar etiquetas diferentes que possuam o mesmo significado. Como tal, este era um aspeto importante ao juntar as etiquetas das *APIs* da Microsoft e da Clarifai e do TensorFlow. Mas como estas plataformas possuem mais de mil etiquetas torna-se impraticável averiguar se são retornadas etiquetas com significados semelhantes. Devido a isto, não foi possível efetuar nenhuma verificação ao reunir as etiquetas.

3.1.4 Sincronização de processos

Uma vez que cada ferramenta corre em paralelo através de *threads* é necessário sincronizar os processos das várias ferramentas para que não escrevam os resultados em simultâneo no ficheiro de texto. Para evitar o acesso em simultâneo das diferentes ferramentas ao recurso partilhado recorreu-se a variáveis de bloqueio. Deste modo, quando alguma ferramenta está prestes a escrever no ficheiro, a variável que permite armazenar os dados no ficheiro é bloqueada (impedindo o uso desta variável pelas restantes ferramentas) e é desbloqueada depois de os dados estarem armazenados.

3.1.5 Tempos de processamento

O tempo de processamento em aplicações deste género torna-se um fator crucial sobretudo quando se pretende analisar *datasets* de imagens ou vídeos de elevadas dimensões. O tempo de processamento está dependente de vários fatores sendo os principais o processador, as dimensões da imagem.

Tabela 3.1: Tempos de processamento por imagem para imagens de diferentes dimensões

Resolução	Número de pixeis	Tempo(s)	Resolução	Número de pixeis	Tempo(s)
1024x773	791552	22,741	772x513	396036	16,820
1024x768	786432	20,502	960x598	574080	18,292
1024x680	696320	19,642	1000x666	666000	19,052
800x602	481600	17,613	750x1024	768000	20,208
1024x683	699392	19,959	1024x697	713728	23,711
1024x683	699392	19,828	950x650	617500	18,834
964x650	626600	18,777	1024x683	699392	19,838

Para saber qual o tempo de processamento médio de uma imagem mediu-se o tempo que levou a analisar várias imagens com diferentes resoluções cujos resultados podem ser observados na Tabela 3.1. Deste teste resultou que a média de tempo de processamento é de 19,7 segundos, e que o número médio de pixeis destas imagens é de 65827. Apesar de o tempo de processamento não parecer muito elevado para uma imagem, caso fosse para analisar um vídeo com 30 *frames* por segundo em que cada *frame* tivesse cerca de 65827 pixeis seriam precisas cerca de de 10 horas para analisar apenas um minuto de vídeo. Em suma, este sistema de analise de imagem não poderia ser utilizado para um extenso volume de imagens ou vídeo.

3.2 Resultados da análise de imagens

O sistema de análise de imagem foi testado em várias imagens para avaliar a sua performance, de seguida serão apresentados os resultados para as Figuras 3.4, 3.5, 3.6 e 3.7. Apesar das várias ferramentas terem sido integradas os resultados serão apresentados por ferramenta, por forma a facilitar a apresentação dos resultados.

Figura 3.4: Imagem de teste, retirada de (flickr)

Figura 3.5: Imagem de teste, retirada de (wikipedia)

Figura 3.6: Imagem de teste, retirada de (flickr)

Figura 3.7: Imagem de teste, retirada de (flickr)

3.2.1 Clarifai

3.2.1.1 Exemplo 1

Na Tabela 3.2, pode ser visualizado o retorno da *API* da Clarifai quando testada com a Figura 3.4. A *API* apresenta várias etiquetas em que a maioria remete para um cenário de guerra, como por exemplo, *soldier*; *military*; *war*. Ou seja, concluir-se que a maioria das etiquetas está de acordo com o conteúdo presente na Figura 3.4. Apesar de a Clarifai possuir um modelo para a deteção de faces não foi capaz de identificar alguma na Figura 3.4.

Na Tabela 3.3, pode-se observar as cores dominantes e as respetivas percentagens da Figura 3.4, apresentadas pela Clarifai.

Tabela 3.2: Etiquetas e respetivas confianças, apresentadas pela Clarifai para a Figura 3.4

Etiqueta	**Confiança**	**Etiqueta**	**Confiança**
people	0.9989209	*gun*	0.97074586
soldier	0.99752223	*army*	0.9618315
military	0.9969752	*uniform*	0.9599729
war	0.99595886	*rifle*	0.9570518
group together	0.99436593	*military uniform*	0.93984854
skirmish	0.99253607	*battle wound*	0.93361914
group	0.9905261	*many*	0.92765486
adult	0.9887457	*several*	0.92529845
man	0.9770794	*wear*	0.9086869
weapon	0.9730576	*four*	0.901989

Tabela 3.3: Cores dominantes da Figura 3.4 apresentadas pela Clarifai

65%	22%	13%

3.2.1.2 Exemplo 2

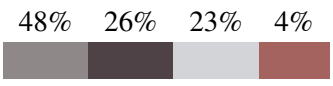
Testando com a Figura 3.5 e observando os resultados na Tabela 3.4, pode-se concluir que a *API* da Clarifai algumas etiquetas incorretas com confianças elevadas, nomeadamente *religion*, *event*, *election*, *leader*. Apenas uma das etiquetas (*calamity*) remete para o que realmente está representado na imagem. Tendo em conta as Tabelas 3.2 e 3.4, verifica-se que um dos problema da Clarifai é o facto de ter etiquetas com confianças elevadas incoerentes com as imagens. Na Figura 3.8 verifica-se que a Clarifai detetou três faces na Figura 3.5, mas não conseguiu detetar uma face que está apenas parcialmente visível. Na Tabela 3.5 podem ser observadas as cores dominantes e as respetivas percentagens apresentadas pela *API* para a Figura 3.5.

Figura 3.8: Deteção de faces na Figura 3.5 efetuada pela Clarifai

Tabela 3.4: Classificação da Figura 3.5 efetuada pela *API* da Clarifai

Etiqueta	**Confiança**	**Etiqueta**	**Confiança**
people	0.9930318	*calamity*	0.6832775
man	0.9708795	*person*	0.6666744
group	0.9529328	*entertainment*	0.6634735
adult	0.91966903	*wear*	0.6534209
group together	0.8744072	*music*	0.6515429
religion	0.8073762	*portrait*	0.6501198
event	0.7902778	*offense*	0.6499754
family	0.7779739	*administration*	0.6442683
election	0.7376717	*competition*	0.63811356
leader	0.7258925	*famous*	0.6254179

Tabela 3.5: Cores dominantes da Figura 3.5 apresentadas pela Clarifai



3.2.1.3 Exemplo 3

Depois de analisadas as etiquetas retornadas pela Clarifai (Tabela 3.7), verifica-se que a maioria das etiquetas remete para moda, como por exemplo *fashion*, *model*, *glamour*. Como tal a classificação efetuada está em concordância com o conteúdo da imagem, uma vez que nesta está presente uma modelo num desfile de moda. Apesar de na Figura 3.6 estarem presentes enumeras faces, a Clarifai só conseguiu detetar três delas, como se pode observar na Figura 3.9. Na Tabela 3.6 podem ser observadas as cores dominantes e as respetivas percentagens da Figura 3.6.

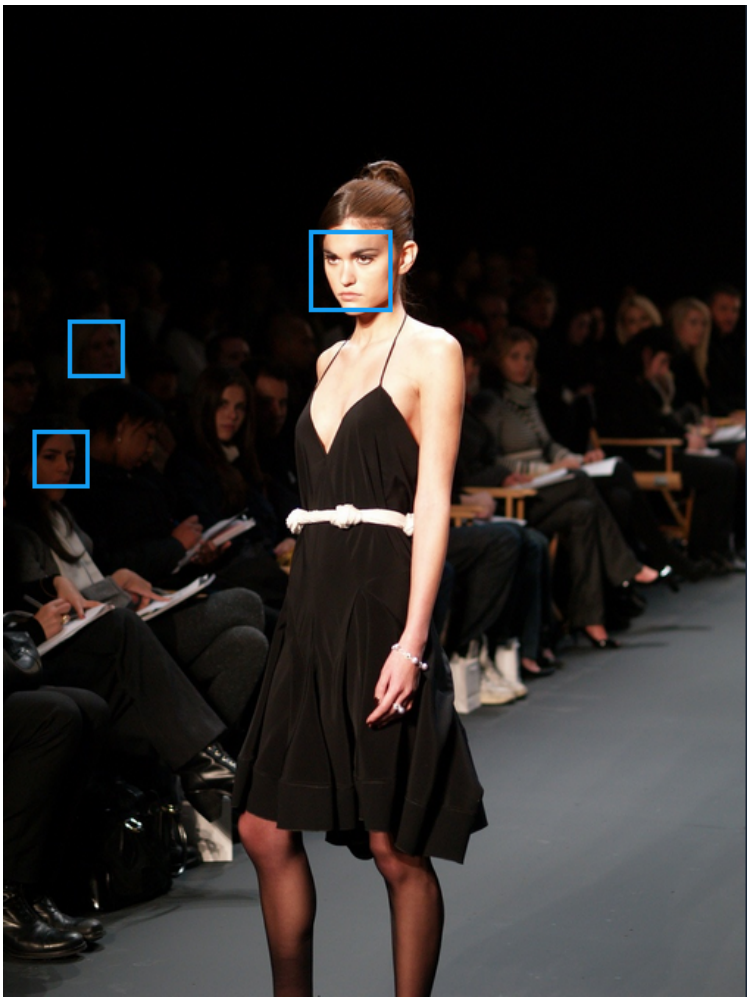


Figura 3.9: Deteção de faces na Figura 3.6 efetuada pela Clarifai

Tabela 3.6: Cores dominantes da Figura 3.6 apresentadas pela Clarifai

70%	15%	7%	4%	3%	1%

3.2.1.4 Exemplo 4

A Clarifai não foi capaz de detetar nenhuma das duas faces presentes na Figura 3.7. Quanto às etiquetas retornadas, verifica-se que algumas apontam para um desastre, nomeadamente *calamaty*, *flood* e *earthquake*, mas a imagem não representa nenhum desastre. As etiquetas também demonstram a presença de pessoas na imagem. Na Tabela 3.9 estão presentes as cores dominantes da Figura 3.7.

Tabela 3.7: Etiquetas e respetivas confianças, apresentadas pela Clarifai para a Figura 3.6

Etiqueta	**Confiança**	**Etiqueta**	**Confiança**
fashion	0.9981661	*fashionable*	0.92061955
woman	0.9908792	*punk*	0.90418106
model	0.9887384	*sexy*	0.88405323
glamour	0.9865721	*sneakers*	0.8809515
wear	0.9789034	*public show*	0.8672739
haute couture	0.9733077	*skirt*	0.851493
collection	0.9720407	*festival*	0.8367977
dress	0.96221495	*stage*	0.8305191
backstage	0.9557904	*fall*	0.830196
people	0.92552876	*menswear*	0.82677203

Tabela 3.8: Etiquetas e respetivas confianças, apresentadas pela Clarifai para a Figura 3.7

Etiqueta	**Confiança**	**Etiqueta**	**Confiança**
people	0.99152136	*one*	0.9119046
calamity	0.99095047	*flood*	0.91067994
child	0.9883883	*outdoors*	0.9051598
home	0.98577017	*house*	0.90120864
adult	0.9657085	*wear*	0.89621794
family	0.9581764	*earthquake*	0.89483285
two	0.93417	*man*	0.8927358
travel	0.9339062	*daylight*	0.889803
group	0.9210057	*interaction*	0.8859656
boy	0.91417575	*war*	0.86998

Tabela 3.9: Cores dominantes da Figura 3.7 apresentadas pela Clarifai

38%	33%	17%	9%	2%

3.2.2 Microsoft

3.2.2.1 Exemplo 1

A *API* da Microsoft apresenta várias etiquetas de modo a classificar a imagem, mas apenas apresenta a respetiva confiança das etiquetas que a têm mais elevada. Na Tabela 3.10 pode-se observar as etiquetas geradas pela *API* para a Figura 3.4 e na Tabela 3.11 a confiança apresentada por algumas dessas etiquetas. Como se pode visualizar pela Tabela 3.11 as etiquetas com maior confiança descravem um pouco a imagem, mas são muito genéricas, nenhuma das etiquetas remete para o cenário de guerra presenta na Figura 3.4. Por outro lado, algumas das etiquetas apresentadas na Tabela 3.4 são contraditórias, por exemplo: novo (*young*) e velho (*old*). Neste caso, a *API* da Clarifai obteve resultados bastante melhores que a da Microsoft.

A *API* da Microsoft consegue detetar e analisar faces e conseguiu detetar uma face na Figura 3.4 como se pode visualizar na Figura 3.10. A *API* retornou que a face pertence a um homem

Figura 3.10: Face detetada pela *API* da Microsoft na Figura 3.4

Tabela 3.10: Etiquetas apresentadas pela *API* da Microsoft

outdoor	*man*	*boy*	*black*	*baby*	*sheep*
grass	*group*	*white*	*young*	*grassy*	*eating*
person	*cow*	*herd*	*old*	*playing*	*woman*
field	*people*	*cattle*	*holding*	*grazing*	*dirt*
sitting	*photo*	*laying*	*standing*	*dog*	*riding*

Tabela 3.11: Etiquetas e respetiva confiança apresentadas pela *API* da Microsoft

Etiqueta	**Confiança**
outdoor	0.999871254
sky	0.995403647
grass	0.9912868
field	0.779475749

de 46 anos. Por outro lado, a *API* também apresenta as cores dominantes de fundo, primeiro plano e de realce de uma imagem. Na Tabela 3.12 podem ser observadas as cores dominantes correspondentes à Figura 3.4.

Tabela 3.12: Cores dominantes apresentadas pela *API* da Microsoft

Fundo	**Primeiro plano**	**Realce**

3.2.2.2 Exemplo 2

Figura 3.11: Deteção de faces na Figura 3.5 pela *API* da Microsoft

Na Tabela 3.13 podem-se observar as etiquetas retornadas pela *API* no teste efetuado com a Figura 3.5. De um modo geral, as etiquetas apresentadas encontram-se de acordo com o conteúdo da imagem. Por outro lado, nenhuma das etiquetas remete para o trágico acontecimento presente na Figura 3.5. Na Tabela 3.14 pode-se visualizar as etiquetas com maior confiança e a respetiva confiança. Apesar destas etiquetas fazerem todo o sentido, são um pouco genéricas.

Tabela 3.13: Etiquetas apresentadas pela *API* da Microsoft para a Figura 3.5

person	outdoor	people	group	woman	young
building	standing	posing	holding	old	white
man	snow	sign	wearing	red	street

Tabela 3.14: Etiquetas e respetiva confiança apresentadas pela *API* da Microsoft para a imagem 3.5

Etiqueta	**Confiança**
person	0.998486757
ground	0.993778
man	0.967517257
outdoor	0.9600123
standing	0.7731561

A análise de faces da *API* da Microsoft detetou três faces como se pode observar na Figura 3.11. Além da deteção a *API* retornou que as três faces pertencem a homens, quando na verdade

são dois homens e uma mulher. Por outro lado, a *API* retorna a idade para essas faces em que as idades apresentadas pela *API* foram 84, 75, 65 (da esquerda para a direita). Para a pessoa da esquerda 84 anos parece uma idade elevada. Na Tabela 3.15 são apresentadas as cores dominantes de fundo, primeiro plano e de realce fornecidas pela *API*.

Tabela 3.15: Cores dominantes apresentadas pela *API* da Microsoft

Fundo	**Primeiro plano**	**Realce**

3.2.2.3 Exemplo 3

Como se pode verificar pelas Tabelas 3.16 e 3.17 a Microsoft não apresenta nenhuma etiqueta que que esteja relacionada com moda, ao contrário da Clarifai. Mesmo assim, a *API* foi capaz de identificar pessoas presentes na imagem, como se pode verificar nas etiquetas *person*, *woman*, *man*. Por outro lado, a *API* retornou a etiqueta *road* (Tabela 3.17) com uma confiança elevada (0.97), mesmo não estando nenhuma estrada na imagem. A *API* conseguiu detetar duas faces na Figura 3.6, como se pode confirmar na Figura 3.12. Na Tabela 3.18 podem ser visualizadas as cores dominantes apresentadas pela *API*.

Tabela 3.16: Etiquetas apresentadas pela *API* da Microsoft para a Figura 3.6

person	*man*	*woman*	*holding*	*crowd*	*black*
road	*street*	*board*	*group*	*walking*	*trick*
people	*young*	*riding*	*standing*	*suit*	*doing*

Tabela 3.17: Etiquetas e respetiva confiança apresentadas pela *API* da Microsoft para a Figura 3.6

Etiqueta	**Confiança**
Person	0.9999393
Road	0.9659275
People	0.618116
Crowd	0.215744391

Tabela 3.18: Cores dominantes apresentadas pela *API* da Microsoft

Fundo	**Primeiro plano**	**Realce**

3.2.2.4 Exemplo 4

Na Figura 3.13 observam-se as duas faces detetadas pela *API* da Microsoft. Segundo a *API* estão presentes duas raparigas de idades 4 e 6. As classificação da Figura 3.7 vai de encontro a esta deteção uma vez que algumas etiquetas (Tabela 3.19)são *child*, *little* e *girl*. As etiquetas com mais

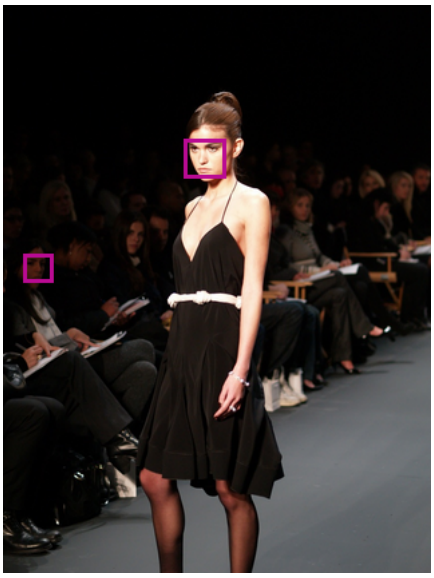


Figura 3.12: Deteção de faces na Figura 3.6 efetuada pelo *API* da Microsoft

confiança apresentadas são *groud*, *outdoor*, *little*, *young* e *dirt* e estão todas em concordância com a imagem. Na Tabela 3.21 podem ser visualizadas as cores dominantes apresentadas pela *API*.

Tabela 3.19: Etiquetas apresentadas pela *API* da Microsoft para a Figura 3.7

outdoor	*child*	*boy*	*playing*	*dirt*	*toddler*
little	*holding*	*toy*	*kite*	*snow*	*walking*
young	*small*	*girl*	*beach*	*standing*	*fliyng*

Tabela 3.20: Etiquetas e respetiva confiança apresentadas pela *API* da Microsoft para a Figura

Etiqueta	**Confiança**
ground	0.999755561
outdoor	0.9992723
little	0.972155333
young	0.924059331
dirt	0.251314729

Tabela 3.21: Cores dominantes apresentadas pela *API* da Microsoft

Fundo	**Primeiro plano**	**Realce**

3.2.3 TensorFlow

3.2.3.1 Exemplo 1

Quanto ao modelo pré treinado do TensorFlow, podem ser observados as etiquetas por ele apresentadas na Tabela 3.22 quando testado com a Figura 3.4. A etiqueta com mais confiança

Figura 3.13: Deteção de faces na Figura 3.7 efetuada pelo *API* da Microsoft

Tabela 3.22: Etiquetas e confiança apresentados pelo TensorFlow para a Figura 3.4

Etiqueta	**Confiança**
stretcher	0.82618
assault rifle, assault gun	0.07654
rifle	0.03631
half track	0.00504
tank, army tank, armored combat vehicle	0.00480

apresentada é maca (*stretcher*), no entanto nenhuma maca está presente na Figura 3.4. As restantes etiquetas apresentadas têm confiança muito baixa, e algumas não estão de acordo com a Figura 3.4 mesmo remetendo para um cenário de guerra. Em suma, o TensorFlow apresentou maus resultados para a imagem em questão.

3.2.3.2 Exemplo 2

Os resultados obtidos pelo TensorFlow quando testado na Figura 3.5, como se pode verificar pela Tabela 3.23. Em primeiro lugar, as etiquetas têm todas confiança inferior a 0.15. Em segundo lugar, apenas duas etiquetas podem ser consideradas corretas (*blue jean* e *suit*). Observando os resultados da Tabelas 3.22 e 3.23 verifica-se que as etiquetas retornadas pelo TensorFlow têm confianças demasiado baixas, mas mesmo assim ainda apresenta algumas etiquetas que estão de acordo com as imagens.

3.2.3.3 Exemplo 3

A classificação da imagem 3.6 pelo TensorFlow pode ser visualizada na Tabela 3.24. Apesar de as etiquetas apresentadas terem confianças baixas algumas estão corretas quando comparando com a imagem como *stage* e *miniskirt*.

Tabela 3.23: Classificação da Figura 3.5 pelo TensorFlow

Etiqueta	**Confianca**
coho, cohoe, coho salmon, blue jack, silver salmon, Oncorhynchus kisutch	0.13277
jean, blue jean, denim	0.12187
bow tie, bow-tie, bowtie	0.05650
suit, suit of clothes	0.04396
pajama, pyjama, pj's, jammies	0.04333

Tabela 3.24: Classificação da Figura 3.6 pelo TensorFlow

Etiqueta	**Confianca**
stage	0.27470
miniskirt, mini	0.15852
jean, blue jean, denim	0.04142
gown	0.03911
flute, transverse flute	0.02564

3.2.3.4 Exemplo 4

A classificação da Figura 3.7 efetuada apresentou as etiquetas presentes na Tabela 3.25. Como se pode verificar, as tendas e casas presentes na imagem foram identificadas (*yurt*, *manufactured home*). Por outro lado, os vestidos das raparigas foram confundidos com o *sarong*. Mais uma vez, as confianças apresentadas são demasiado baixas.

Tabela 3.25: Classificação da Figura 3.7 pelo TensorFlow

Etiqueta	**Confiança**
yurt	0.36166
sarong	0.06478
sandbar, sand bar	0.04359
mobile home, manufactured home	0.03214
cliff, drop, drop-off	0.02490

3.2.4 YOLO

3.2.4.1 Exemplo 1

Os resultados obtidos pelo YOLO para a Figura 3.4, podem ser observados na Figura 3.14. O YOLO retornou as coordenadas dos objetos detetados, assim como as respetivas percentagens de confiança. Pode-se verificar que o YOLO conseguir detetar várias pessoas corretamente, mas falhou ao detetar um cavalo na região em que está presente uma pessoa. Apesar desta falha, a confiança associada a esta deteção é de 0.58, isto é, apesar da deteção estar errada não é apresentada uma elevada confiança para a deteção. Além da imprecisão previamente mencionada, o YOLO também não conseguiu detetar uma das pessoas que aparece em primeiro plano da Figura 3.4.

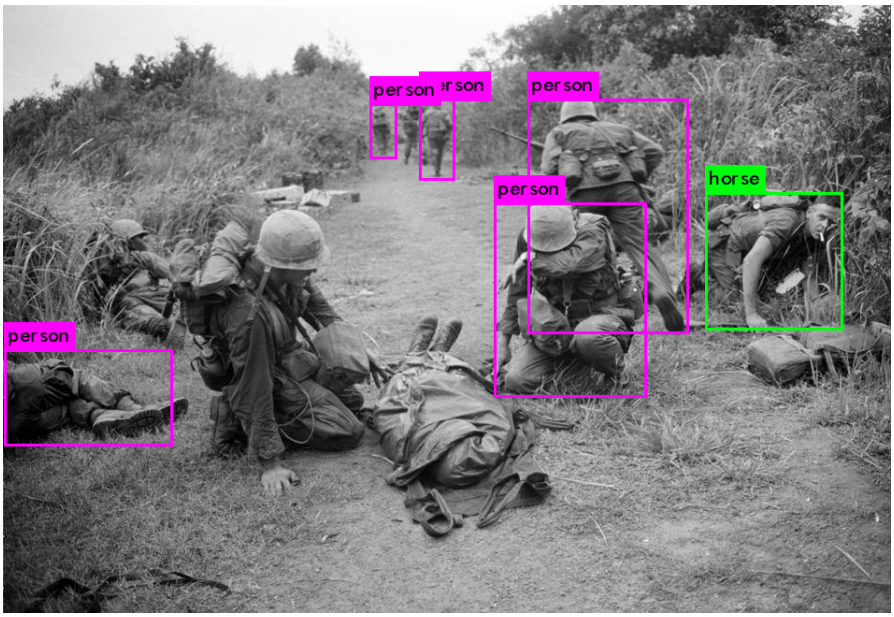


Figura 3.14: Deteções apresentadas pelo YOLO para a Figura 3.4

3.2.4.2 Exemplo 2

Como se pode verificar pela Figura 3.15, o YOLO conseguiu detetar corretamente todas as pessoas presentes na Figura 3.5 e duas gravatas. Todas as deteções apresentam uma confiança superior a 0.6 e a maioria das deteções apresenta uma confiança superior a 0.9. Apesar de o YOLO ter efetuado uma deteção errada na Figura 3.4, na Figura 3.5 só realizou deteções acertadas.

Figura 3.15: Deteção de objetos na Figura 3.5 pelo YOLO

3.2.4.3 Exemplo 3

O YOLO conseguiu detetar várias pessoas presentes na Figura 3.6, mas não foi capaz de detetor todas, como se pode verificar na Figura 3.16. As pessoas foram detetadas com confianças acima de 0.5.

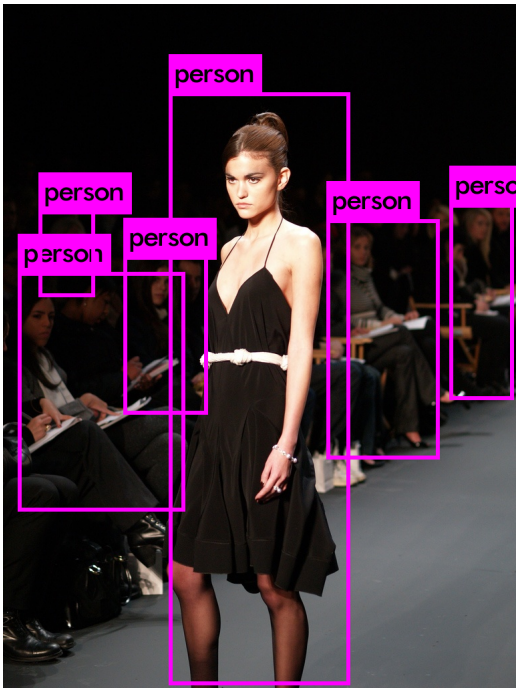


Figura 3.16: Deteção de objetos na Figura 3.6 efetuada pelo YOLO

3.2.4.4 Exemplo 4

Como se pode observar na Figura 3.17, o YOLO detetou as duas crianças Figura 3.7 presentes. As crianças foram detetadas com confianças de cerca de 100%.

3.2.5 Dlib

3.2.5.1 Exemplo 1

O Dlib identificou uma face na Figura 3.4 como se pode visualizar na Figura 3.18. Tendo em conta, que o Dlib conseguiu detetar a única face presente na imagem conclui-se que se comportou bem com a Figura 3.4. A confiança apresentada para a face detetada foi de 0.73. Apesar de o Dlib conseguir detetar alguns pontos de interesse em faces, neste caso não conseguiu, possivelmente por a face estar a ocupar uma região muito pequena na imagem.

3.2.5.2 Exemplo 2

O Dlib conseguiu detetar as três faces na Figura 3.5, como a Clarifai, mas além disso, conseguiu detetar vários pontos de interesse na face como se pode visualizar na Figura 3.19. Assim

Figura 3.17: Deteção de objetos na Figura 3.7 efetuada pelo YOLO

Figura 3.18: Face detetada pelo Dlib na Figura 3.4

conclui-se que os modelos do Dlib são bastante precisos e eficazes, apesar de na Figura 3.4 não ter sido capaz de localizar pontos de interesse na face detetada.

3.2.5.3 Exemplo 3

Na Figura 3.20 podem ser observadas as 5 faces detetadas na Figura 3.6. Na Figura 3.6 à esquerda podem ser observados a azul cinco pontos de interesse em cada face. À direita podem ser observados 68 pontos de interesse em cada face. O Dlib consegui detetar mais faces na Figura 3.6 do que as *APIs* da Microsoft e Clarifai.

Figura 3.19: Deteção de faces e de 5(esquerda) ou 68(direita) pontos de interesse nas respetivas faces

Figura 3.20: Deteção de faces e de 5(esquerda) ou 68(direita) pontos de interesse nas respetivas faces

3.2.5.4 Exemplo 4

O Dlib detetou as duas faces na Figura 3.7 como se pode observar na Figura 3.21, mas os modelos que permitem detetar pontos de interesse na face só conseguiram detetar pontos na face da esquerda como se pode visualizar na Figura 3.22

Figura 3.21: Deteção de faces e de 5(esquerda) ou 68(direita) pontos de interesse nas respetivas faces

Figura 3.22: Deteção de faces e de 5(esquerda) ou 68(direita) pontos de interesse nas respetivas faces

3.2.6 Conclusões

A partir dos resultados acima apresentados podemos concluir que o Clarifai foi o que obteve melhores resultados na classificação de imagem, uma vez que na maioria das vezes apresentava etiquetas que identificavam o "panorama geral" presente na imagem. Por outro lado, o YOLO consegue detetar objetos sem erros na maioria dos casos e quando apresenta um falso positivo, tem sempre confiança baixa. O Dlib conseguiu detetar faces e pontos de interesse nas faces com bastante precisão. A *API* da Microsoft obteve resultados relativamente aceitáveis na classificação de imagem. O TensorFlow apresentou na maioria das vezes etiquetas com baixa confiança e normalmente fora do contexto das imagens. Com exceção do TensorFlow, as restantes ferramentas conseguiram fornecer informação relevante sobre imagens de fotojornalismo.

Capítulo 4

Dataset

Para efetuar um detetor automático de zonas de interesse é necessário um *dataset* que permita treinar os algoritmos que efetuar a deteção. Como mencionado no capítulo 2 não foi encontrado uma *dataset* para um cenário de fotojornalismo e como tal foi necessário elaborar um. Neste capitulo será explicado como se elaborou o *dataset*, assim como a aquisição do respetivo *ground truth*.

4.1 Caraterização do *dataset*

O *dataset* é constituído por 256 imagens das quais 200 são de fotojornalismo, 31 de moda e 25 de eventos. As imagens foram obtidas através da pesquisa por imagens nas respetivas áreas, que podiam ser utilizadas não comercialmente. As imagens de fotojornalismo são na sua maioria imagens a preto e branco, retratando cenários de guerra, refugiados, desastres naturais, manifestações entre outros problemas contemporâneos. As imagens da moda consistem são provenientes de desfiles de moda. Quanto às imagens de eventos correspondem a jogos desportivos, conferências e eventos regionais. As imagens são de dimensões variáveis sendo que a imagem com mais pixeis tem uma resolução de 4644x3091 enquanto a imagem que possui menos pixeis uma resolução de 950x650. Na Figura 4.1 podem ser visualizadas algumas imagens que constituem o *dataset*.

Figura 4.1: Exemplos de imagens presentes no *dataset*

4.2 Aquisição de *ground truth*

De modo a desenvolver um detetor automático de zonas de interesse baseado na perceção humana é necessário não só um vasto conjunto de imagens, mas é também necessário obter anotações que consistem nas caixas delimitadoras das zonas de interesse. Para adquirir as anotações foi desenvolvida uma página *web* de modo a recolher retângulos em que as pessoas mostram interesse nas várias imagens por forma a posteriormente utilizar as anotações para treinar um algoritmo aprendizagem máquina.

4.2.1 Estrutura e funcionamento da página *web*

Em primeiro lugar, quando uma pessoa acede à página *web* são escolhidas cinco imagens para a respetiva pessoa anotar. A escolha das imagens para a anotação é baseada no número de anotações que cada imagem possui. Isto é, as cinco imagens escolhidas para um dado utilizador são sempre as cinco imagens menos anotadas até ao momento.

Do lado esquerdo do *website*, é apresentada a imagem ao anotador, como apresentado na Figura 4.2. Deste lado, podem ser selecionados cinco retângulos que contêm as zonas que mais captam a atenção do utilizador. Para selecionar um retângulo na imagem são necessários dois cliques, um para escolher o vértice do topo-esquerda e outro para o vértice fundo-direita. Do lado direito, são apresentados os recortes dos retângulos à medida que eles são selecionados, ou seja, pela ordem de relevância percetual, como pode ser observado na Figura 4.3. Depois de escolhidas as cinco áreas de interesse, é pressionado um botão que permite passar à anotação da imagem seguinte, até a anotação das cinco imagens estar completa.

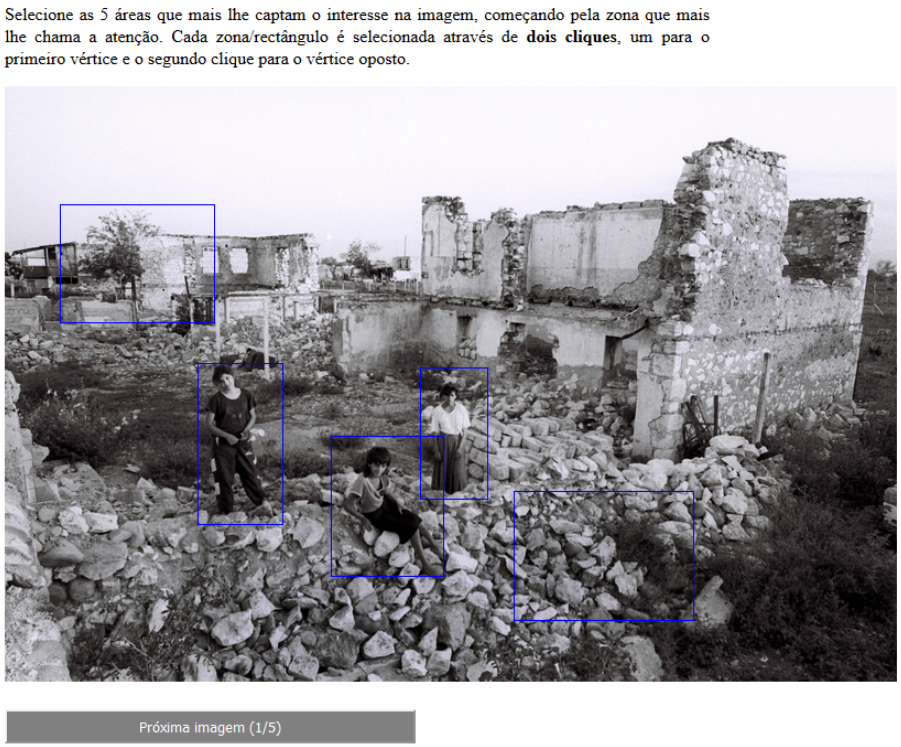


Figura 4.2: Aspeto da parte esquerda página *web* utilizada para a recolha de *ground truth*

Figura 4.3: Aspeto da parte direita da página *web* utilizada para a recolha de *ground truth*

4.2.2 Formato das anotações provenientes da página *web*

No momento em que é pressionado o botão, que pode ser visualizado na Figura 4.2, as coordenadas dos retângulos são guardadas num ficheiro XML. Para cada imagem existe um ficheiro XML. A estrutura ficheiro XML é constituída por um elemento raiz <data> que engloba os restantes elementos. Por cada pessoa que anota a imagem é criado o elemento <person>. Dentro do elemento <person> existem 5 elementos <rectangle>, que por sua vez incluem os elementos <top>, <left>, <width> e <height> que contêm as coordenadas dos cinco retângulos escolhidos. O elemento <rectangle> tem um atributo "score" varia de 1 a 5, sendo atribuído o 5 ao primeiro retângulo a ser escolhido, 4 ao segundo e assim sucessivamente. A hierarquia do XML pode ser

visualizada na Figura 4.4.

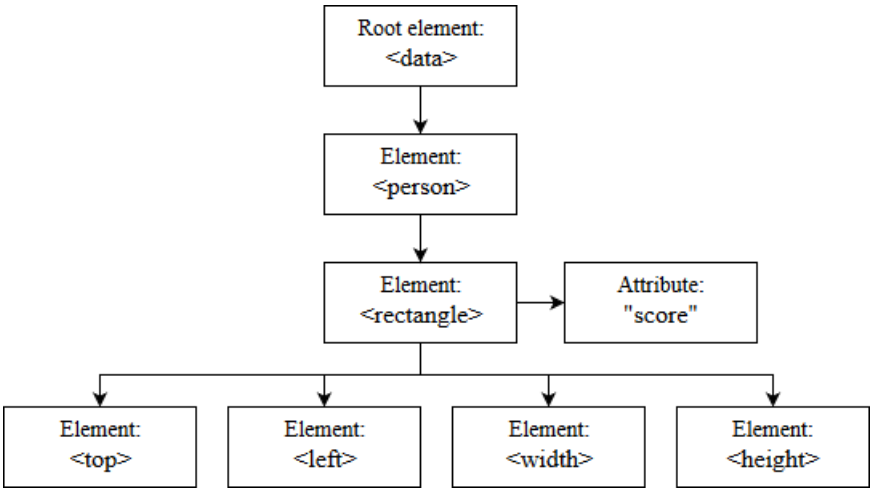


Figura 4.4: Estrutura do ficheiro de dados XML

4.3 Filtragem do *ground truth*

Depois de obtidos os retângulos selecionados pelas várias pessoas na página *web* é necessário filtrar os resultados por forma a eliminar anotações que poderão interferir negativamente na precisão do detetor. Por outro lado, é necessário agregar os retângulos que poderão conter o mesmo objeto, de modo a perceber quais os objetos ou zonas da imagem que foram selecionados mais vezes. Primeiramente, são eliminados os retângulos cuja área é inferior a 10 % ou superior a 80% da área da imagem, de modo a remover retângulos demasiado pequenos que não conseguem conter um objeto o que poderia interferir com os resultados. De seguida, são agregados os retângulos como será explicado de seguida.

4.3.1 Agregação dos retângulos

Como foi referido no capítulo 4.2, a cada retângulo está associado uma pontuação de um a cinco. Estas pontuações são somadas à medida que os retângulos são fundidos e é guardado num ficheiro XML os cinco retângulos com maior pontuação e o número de retângulos que contribuiu para a respetiva pontuação.

Para agregar os retângulos recorreu-se ao *non-maximum supression* (*NMS*) por ser um algoritmo simples e rápido e competitivo quando comparado com outros algoritmos com o mesmo propósito (Hosang et al., 2017).

O *NMS* é normalmente utilizado como pós-processamento de detetores de objetos com a finalidade de eliminar deteções redundantes. Como o objetivo da deteção de objetos consiste na obtenção de apenas uma deteção por objeto é habitual assumir que áreas com elevada sobreposição correspondem ao mesmo objeto (Hosang et al., 2017). Tendo em conta que o *NMS* só utiliza

apenas o *threshold* de sobreposição para efetuar decisões, pode haver situações em que objetos próximos sejam identificados como apenas um.

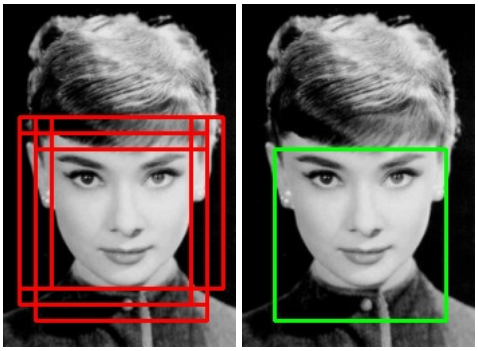


Figura 4.5: Resultados da aplicação do *NMS* (imagem retirada de Rosebrock)

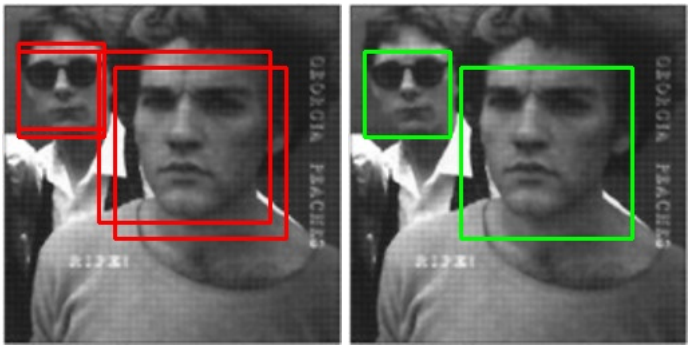


Figura 4.6: Resultados da aplicação do *NMS* (imagem retirada de Rosebrock)

Para aplicar o *non-maximum supression* é necessário fornecer ao algoritmo as coordenadas das caixas delimitadores bem como o *threshold* de sobreposição a aplicar. Em primeiro lugar, as caixas delimitadoras são ordenadas pela coordenada y do canto inferior direito. De seguida, ciclicamente, é escolhido um retângulo de referência em que o retângulo são escolhidos do último para o primeiro depois de ordenados como anteriormente mencionado. Depois de escolhido o retângulo de referência, este é comparado com os restantes retângulos para verificar se a percentagem de sobreposição é superior ao *threshold* de sobreposição, caso seja, o retângulo de comparação é eliminado, como se pode observar no Algoritmo 1. Isto é, como a sobreposição é superior ao *threshold* assume-se que estamos na presença do mesmo retângulo e como tal, mantém-se as coordenadas do retângulo de referência. Finalmente, quando percorridos todos os retângulos de referência é retornada a lista de retângulos. Para efetuar a implementação recorreu-se a uma função para o calculo do *NMS*, efetuada por Rosebrock.

Nas Figuras 4.5 e 4.6 podem ser visualizados os resultados da aplicação do *NMS* com um *threshold* de 30% de sobreposição. Do lado esquerdo das Figuras são apresentados os retângulos que serão enviados para o *NMS* e do lado direito os retângulos após ser aplicado o *NMS*. Como se pode verificar quando a sobreposição entre retângulos é superior ao *threshold* estes são aglomerados.

Algoritmo 1 Agregação de retângulos e pontuações

BoundaryBoxes ← lista com as coordenadas de cada retângulo e respetiva pontuação
threshold ← limiar sobreposição
se *BoundaryBoxes* está vazio **então**
    retorna lista vazia
fim se
area ← lista com as areas das *BoundaryBoxes*
enquanto não forem percorridos todos os retângulos **faz**
    RecRef ← retângulo de referência
    enquanto não forem percorridos todos os retângulos (exceto o de referência) **faz**
        RecCom ← retângulo de comparação
        Sobrep ← área de sobreposição entre o retângulo de referência e o de comparação
        se *Sobrep*/max(*area(RecRef), area(RecCom)*) > *threshold* **então**
            soma-se as pontuações dos retângulos *RecRef* e *RecCom*
            elimina-se *RecCom*
        fim se
    fim ciclo
fim de ciclo
retorna lista com os retângulos não eliminados

4.3.2 Formato das anotações após filtragem

Na Figura 4.7 pode ser observada a estrutura e hierarquia do ficheiro XML depois de aplicada a filtragem. A estrutura é relativamente parecida à da Figura 4.4 presente em 4.2.2. O ficheiro XML é constituído pelos retângulos e as suas respetivas coordenadas. Este ficheiro vai conter apenas os cinco retângulos (elemento <rectangle>) com maior pontuação, como foi acima mencionado. Como alguns retângulos foram aglomerados, o elemento <rectangle> deixou de ser um filho do elemento <person>, como acontecia no XML que era *input* do *NMS*, e passou a ter um atributo "persons" que consiste no número de pessoas que contribuiu para a pontuação associada ao retângulo, ou seja, será o número de retângulos aglomerados.

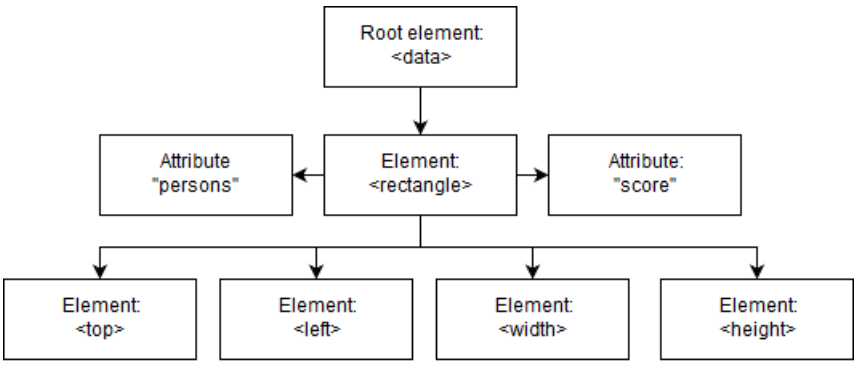


Figura 4.7: Estrutura do ficheiro de dados XML depois de aplicado o algoritmo de condensação

4.3.3 Resultados da filtragem do *ground truth*

Como foi mencionado anteriormente, depois da filtragem resultam apenas os cinco retângulos com maior pontuação. Nas Figuras 4.8, 4.9, 4.10 e 4.11 pode ser observado os retângulos antes da filtragem nas imagens à direita e os retângulos resultantes da filtragem nas imagens da direita. Os retângulos obtidos após a filtragem são apresentados a vermelho, amarelo, verde, azul, laranja consoante a pontuação de cada retângulo, em que vermelho corresponde à maior pontuação amarelo à segunda e assim sucessivamente. A partir destas figuras verifica-se que os retângulos provenientes da filtragem correspondem às zonas com maior concentração de retângulos nas figuras antes da filtragem.

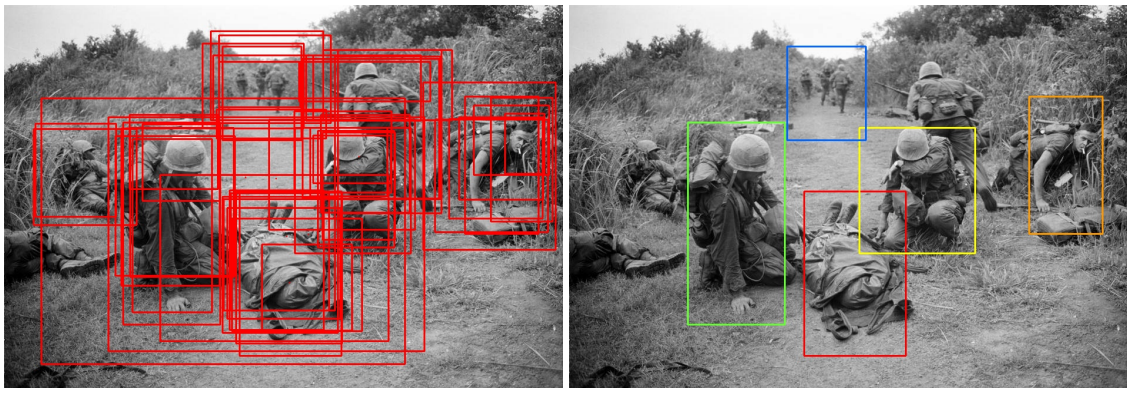


Figura 4.8: Retângulos resultantes das anotações de várias pessoas (direita) e cinco retângulos com mais pontuação depois de aplicada filtragem (esquerda)

Figura 4.9: Retângulos resultantes das anotações de várias pessoas (direita) e cinco retângulos com mais pontuação depois de aplicada filtragem (esquerda)

É importante validar o *ground truth* uma vez que os resultados do detetor de zonas de interesse vão estar diretamente dependentes das anotações do *dataset*. Uma vez que se tratam anotações percetuais não é possível validar de forma objetiva. Os retângulos resultantes da filtragem costumam conter objetos presentes em primeiro plano na imagem. Por esse motivo, as anotações podem ser aceites.

Figura 4.10: Retângulos resultantes das anotações de várias pessoas (direita) e cinco retângulos com mais pontuação depois de aplicada filtragem (esquerda)

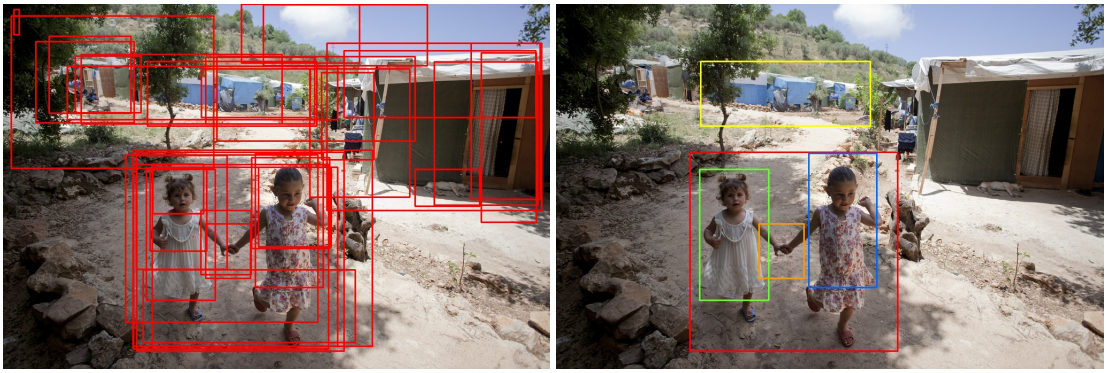


Figura 4.11: Retângulos resultantes das anotações de várias pessoas (direita) e cinco retângulos com mais pontuação depois de aplicada filtragem (esquerda)

4.4 Divulgação do *website*

Em primeiro lugar, o *website* presente no seguinte URL `https://paginas.fe.up.pt/~up201305177/tese/pages/`, foi divulgado apenas a 20 pessoas por forma a validar o funcionamento do *website* e do algoritmo de filtragem dos resultados. Com esta experiência, confirmou-se que o *website* conseguia guardar os retângulos selecionados pelas pessoas. Por outro lado, verificou-se o algoritmo de agregação dos retângulos conseguia fundir retângulos que continham o mesmo objeto, mas quando estava um objeto pequeno contido num objeto maior estes eram agregados apesar de serem objetos diferentes. Isto devia-se ao facto de na comparação entre dois retângulos a percentagem de sobreposição era calculada em relação ao retângulo de referência, ou seja, caso um objeto pequeno estiver contido num maior e o objeto menor for o retângulo de referência eles seriam agregados pois a sobreposição seria calculada através do quociente da área de sobreposição e a área do retângulo pequeno resultando numa sobreposição elevada. Alterando o calculo da sobreposição para o quociente da área de sobreposição e o máximo entre a área do retângulo de referência e o retângulo de comparação, resolveu o problema anteriormente mencionado. Na Figura 4.12 pode-se verificar o que foi anteriormente referido. Neste caso existe um retângulo nos olhos e outro na face. Uma vez que são objetos diferentes deviam permanecer os

dois retângulos depois de aplicado o *NMS*, mas tal não acontece como se verifica na Figura 4.12, mas depois de alterada a formula de calculo da percentagem de sobreposição o *NMS* retorna os dois retângulos como suposto, Figura 4.13.

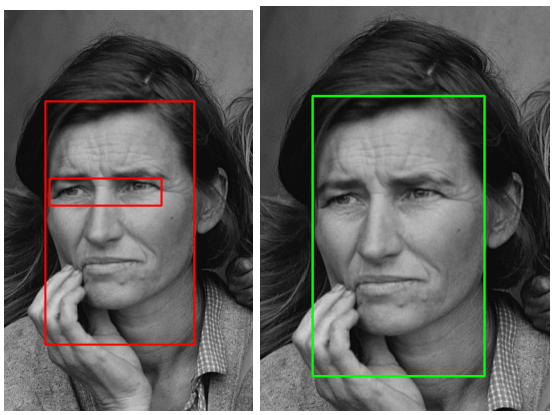


Figura 4.12: Funcionamento do *NMS* quando um objeto menor está contido dentro de outro

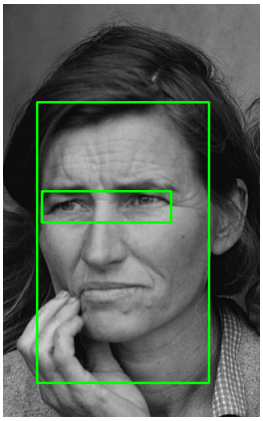


Figura 4.13: Funcionamento do *NMS* quando um objeto menor está contido dentro de outro, depois da alteração do calculo da sobreposição

Assegurado o bom funcionamento do sistema de obtenção de anotações, o *website* foi divulgado por *email* para todos alunos da Faculdade de Engenharia da Universidade do Porto (FEUP) por forma a obter o máximo de anotações para a mesma imagem. Quando as anotações são efetuadas por pessoas, estas estão dependentes do estado de espírito do anotador bem como da sua personalidade, daí a importância de obter o máximo número de anotações para uma dada imagem de modo que as anotações vão de encontro às preferências da maioria das pessoas.

Capítulo 5

Análise de caraterísticas percetualmente relevantes

De modo a entender quais as caraterísticas ou objetos mais relevantes nas imagens foi efetuada uma análise detalhada das regiões selecionadas pelos anotadores. Em primeiro lugar, foi efetuada a contagem de todos os objetos contidos nos retângulos resultantes da aglomeração, para todas as imagens. Em segundo lugar, verificou-se qual a soma das pontuações atribuídas a estes retângulos para cada categoria, de modo a perceber , por exemplo, se certos objetos eram selecionados menos vezes, mas com maior prioridade. Por fim, foi verificado com que prioridade atribuída a cada categoria quando esta era selecionada. Os resultados desta análise serão expostos de seguida.

5.1 Análise geral de caraterísticas percetualmente relevantes

Depois de aplicado o algoritmo de aglomeração, resultam os cinco regiões das imagens que mais interesse despertaram nos anotadores. De seguida, efetuou-se a contagem dos objetos presentes nessas regiões para a totalidade das imagens por forma a ser percetível quais os objetos mais relevantes.

Nas Figuras 5.1 e 5.2 podem ser observado o número de retângulos selecionados por cada categoria e os seu valor percentual, respetivamente. Destes gráficos pode-se concluir que a preferência é na maioria pessoas, crianças e bebés. Este resultado era expectável uma vez que as pessoas, na maioria dos cenários de aplicação do *dataset*, costumam ser o elemento central da imagem. Isto é, no fotojornalismo, moda e eventos as pessoas costumam ser parte essencial das fotografias. Do mesmo modo, partes do corpo como face, cabeça e cabelo também se encontram entre os mais selecionados.

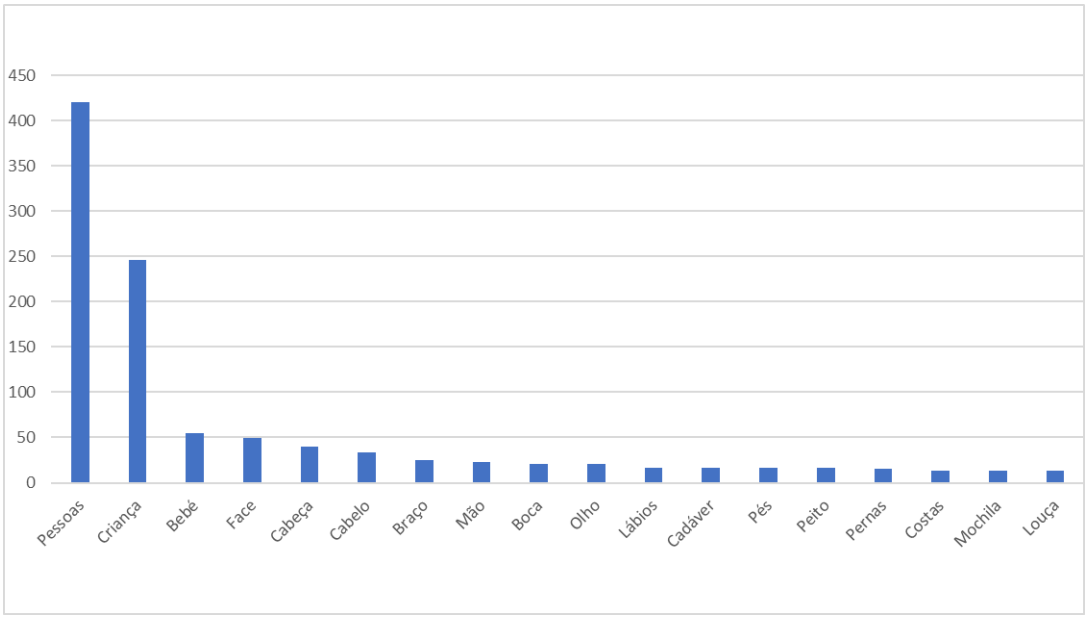


Figura 5.1: Histograma as caraterísticas escolhidas nas imagens do *dataset*

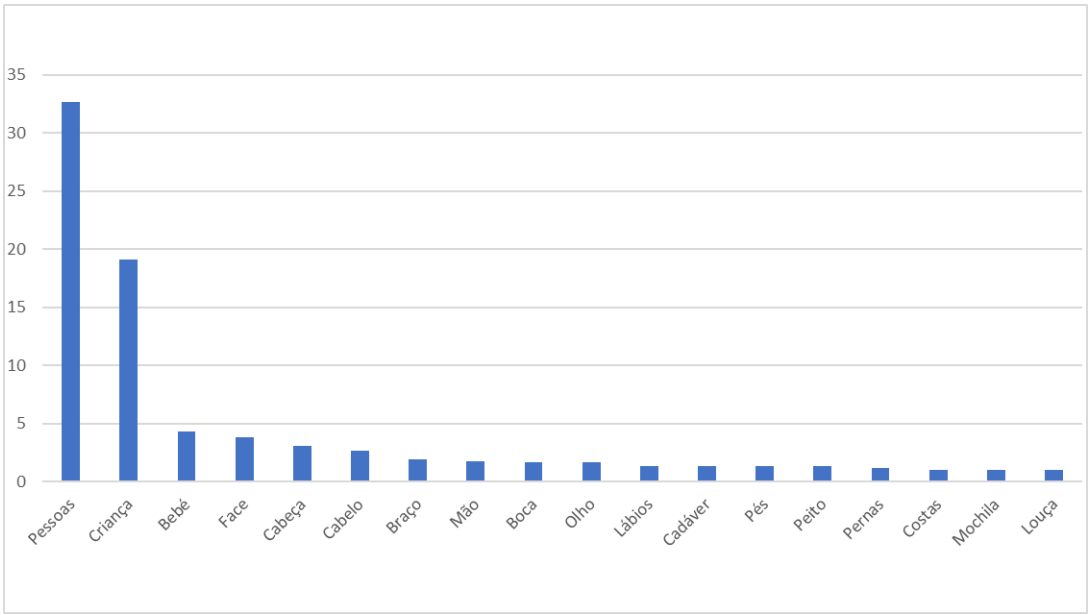


Figura 5.2: Histograma das caraterísticas escolhidas nas imagens do *dataset* (percentagem)

Nas Figuras 5.3 e 5.4 está presente a soma da pontuação que cada retângulo obteve para cada categoria e os seus valores em percentuais. Como se pode verificar a ordem verificada nas Figuras 5.1 e 5.2 não se manteve, o que significa que determinados objetos, apesar de selecionados menos vezes, quando escolhidos têm uma maior preferência. Pessoas continua a representar a maioria das escolhas certas categorias passaram a ter maior percentagem nesta análise, como por exemplo, faces, palavras, fogo, destroços, ecrã, lixo, fumo, vestido.

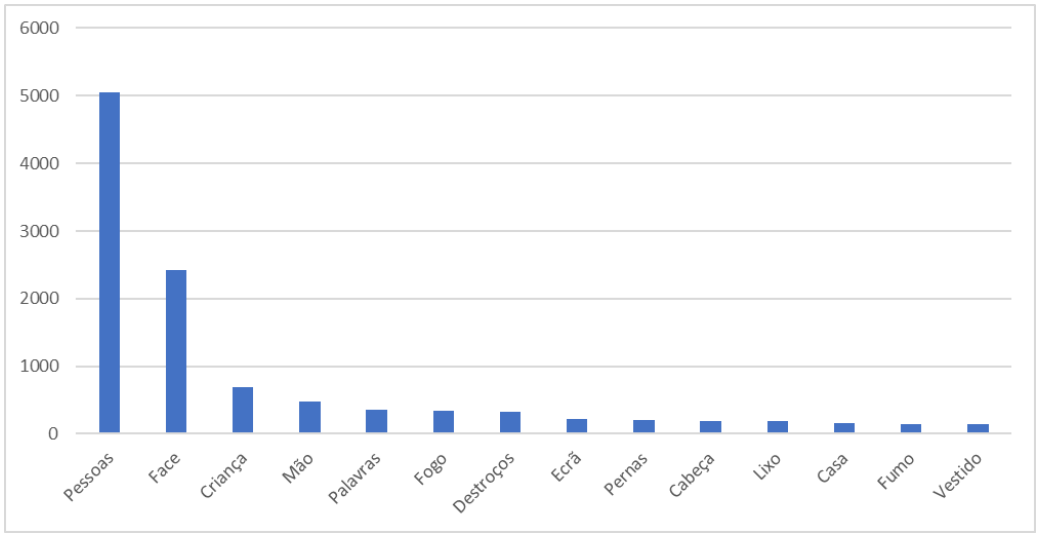


Figura 5.3: Histograma das pontuações das caraterísticas escolhidas nas imagens do *dataset*

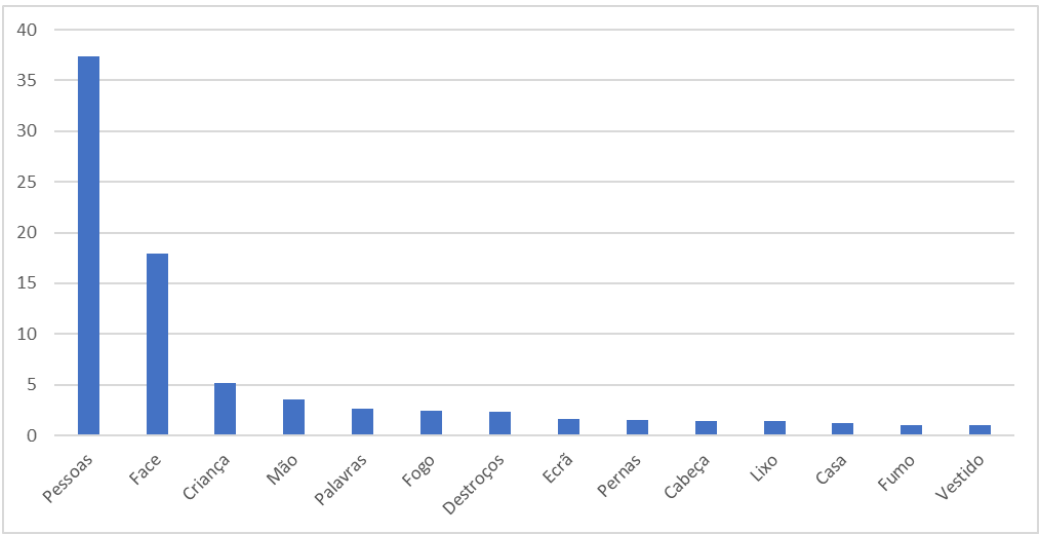


Figura 5.4: Histograma das pontuações das caraterísticas escolhidas nas imagens do *dataset*

5.2 Análise de caraterísticas percetualmente relevantes por caso de uso

Apesar da área de foco ser o fotojornalismo, é importante comparar com outras áreas para compreender se existem diferenças em cenários de aplicação distintos. Na moda, as fotografias focam-se, normalmente, na modelo e na sua roupa portanto espera-se que estas sejam os objetos mais relevantes nestas imagens. Por sua vez, as fotografias de eventos contém pessoas e multidões, e como tal, e de esperar que pessoas seja o mais relevante. Por outro lado, as imagens de fotojornalismo são mais abrangentes pelo que é mais difícil prever quais o tipo de objetos mais percetualmente importantes nestas imagens.

5.2.1 Fotojornalismo

Efetuando a mesma análise que em 5.1, mas considerando apenas as imagens de fotojornalismo, obteve-se os resultados das Figuras 5.5 e 5.6. Como se pode verificar, no fotojornalismo, pessoas continua a ser dos elementos mais selecionados, seguido de face e criança. Verifica-se que algumas das categorias presentes nestes gráficos são mais orientadas para o fotojornalismo, nomeadamente destroços, fogo, lixo, fumo, arma devendo-se ao facto de estas categorias estarem normalmente presentes em cenários de guerra, manifestações e poluição.

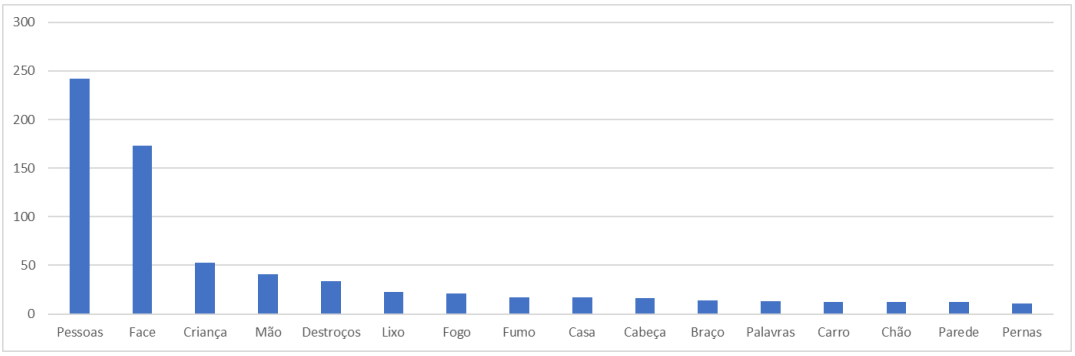


Figura 5.5: Histograma as caraterísticas escolhidas nas imagens de fotojornalismo

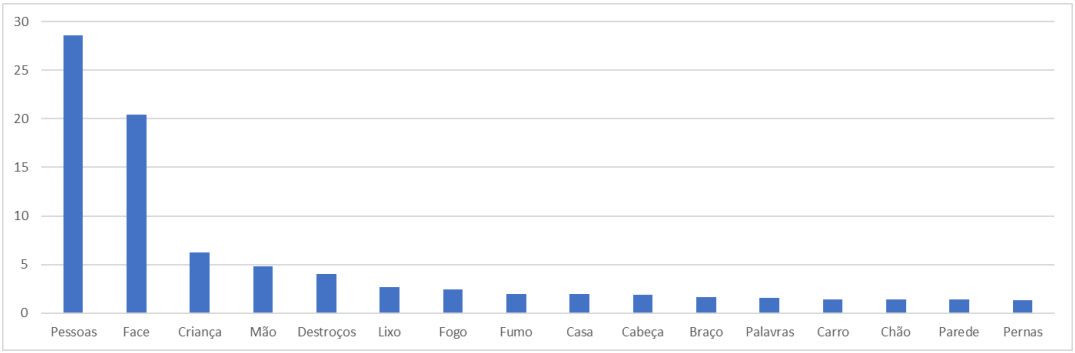


Figura 5.6: Histograma das caraterísticas escolhidas nas imagens de fotojornalismo (percentagem)

Tendo em conta a pontuação, observa-se que o fogo, apesar de não ter sido escolhido tantas vezes como destroços e lixo acaba por ter maior pontuação que estas, como se pode verificar comparando as Figuras 5.5 e 5.6 com 5.7 e 5.8. O facto do fogo se destacar deve-se ao facto da sua cor viva que faz contraste com o resto da imagem acabando por se destacar.

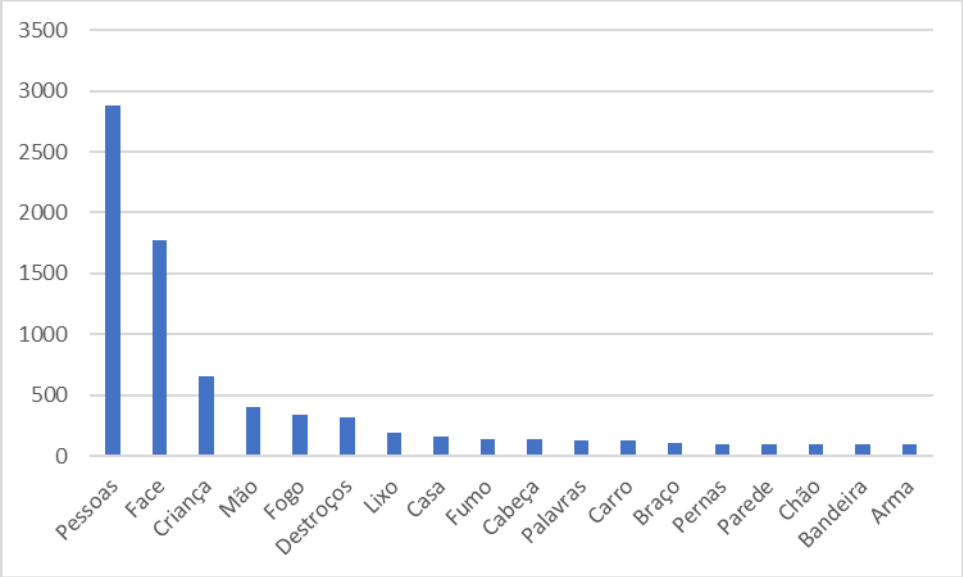


Figura 5.7: Histograma das pontuações das caraterísticas escolhidas nas imagens de fotojornalismo

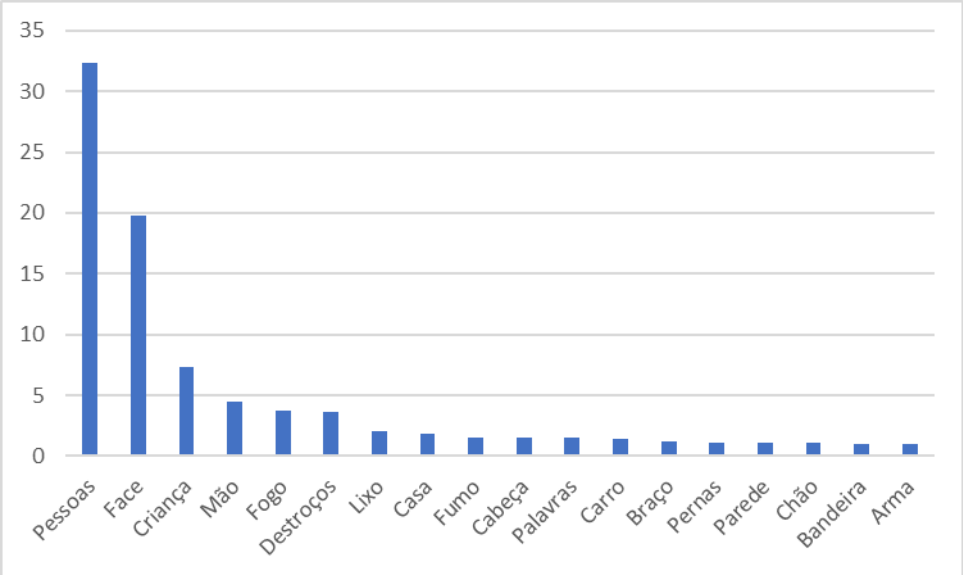


Figura 5.8: Histograma das pontuações das caraterísticas escolhidas nas imagens de fotojornalismo (percentagem)

5.2.2 Moda

Como seria de esperar, nas imagens de moda as categorias mais selecionadas são pessoas, peças de roupa e partes do corpo como se pode verificar pelas Figuras 5.9 e 5.10. Ao analisar as regiões de interesse tendo em conta a prioridade, Figuras 5.11 e 5.12, verificam-se algumas diferenças de preferência Por exemplo, vestido e palavras têm maior percentagem do que pernas o que não aconteceu quando a prioridade não era incluída.

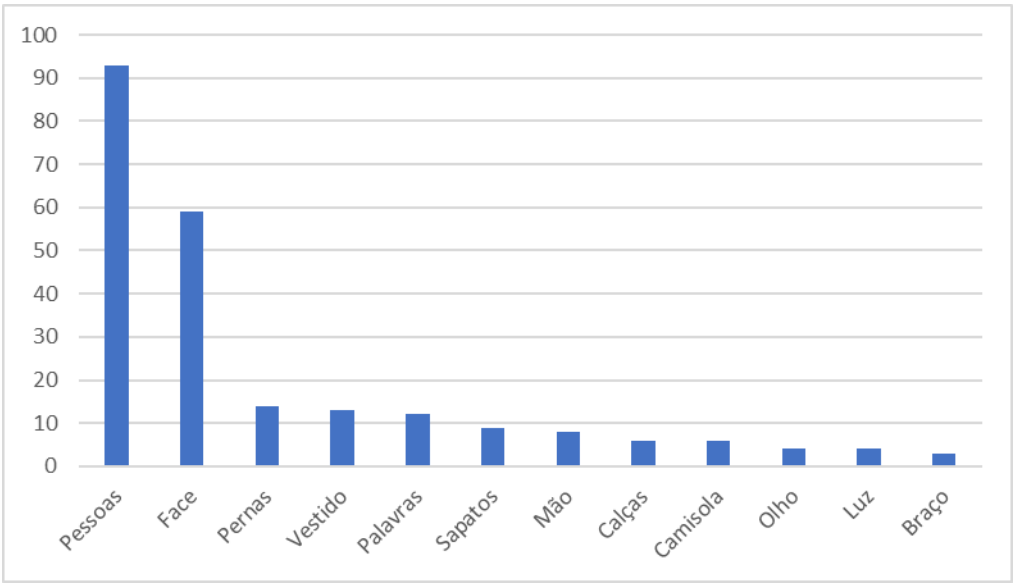


Figura 5.9: Histograma das caraterísticas escolhidas nas imagens de moda

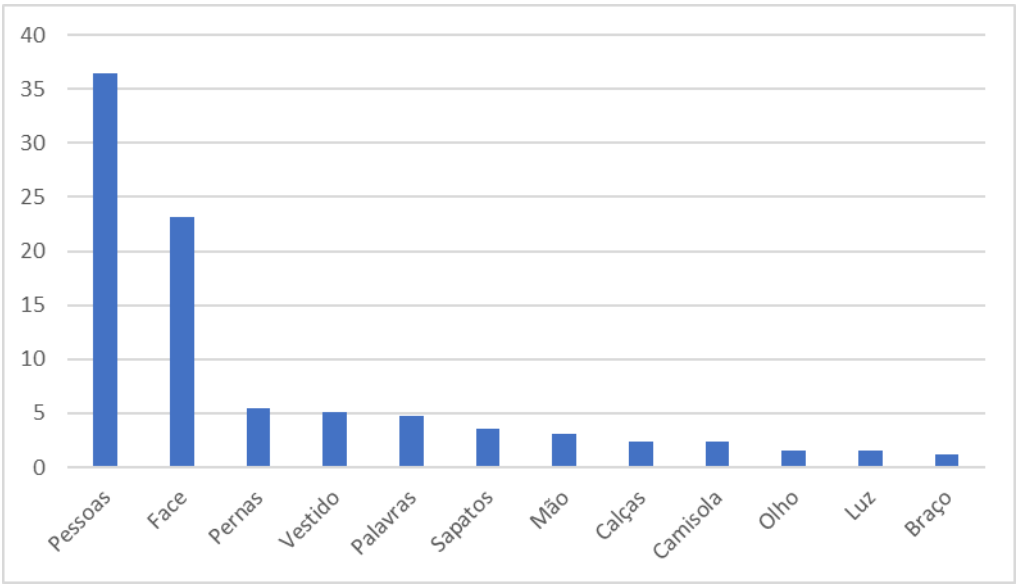


Figura 5.10: Histograma das caraterísticas escolhidas nas imagens de moda (percentagem)

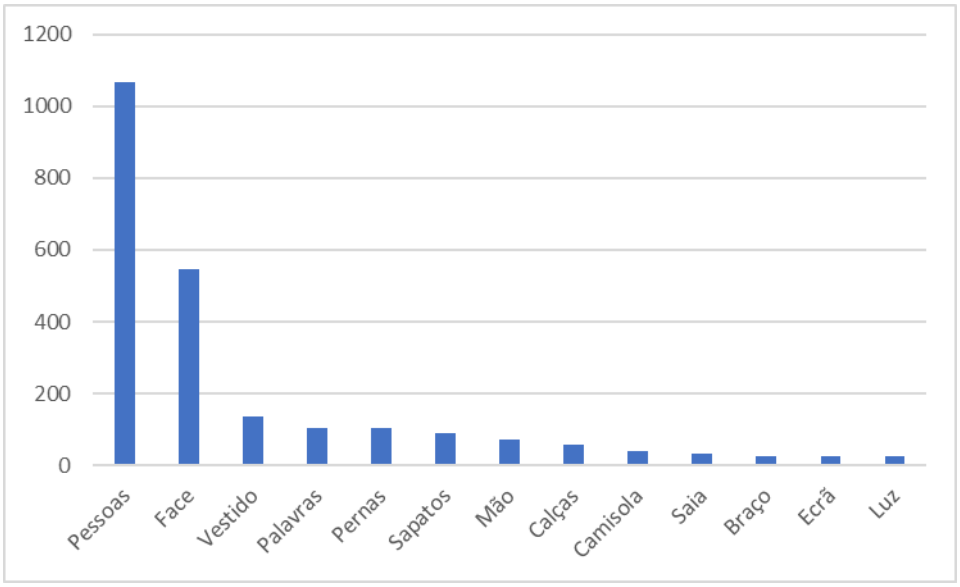


Figura 5.11: Histograma das pontuações das caraterísticas escolhidas nas imagens de moda

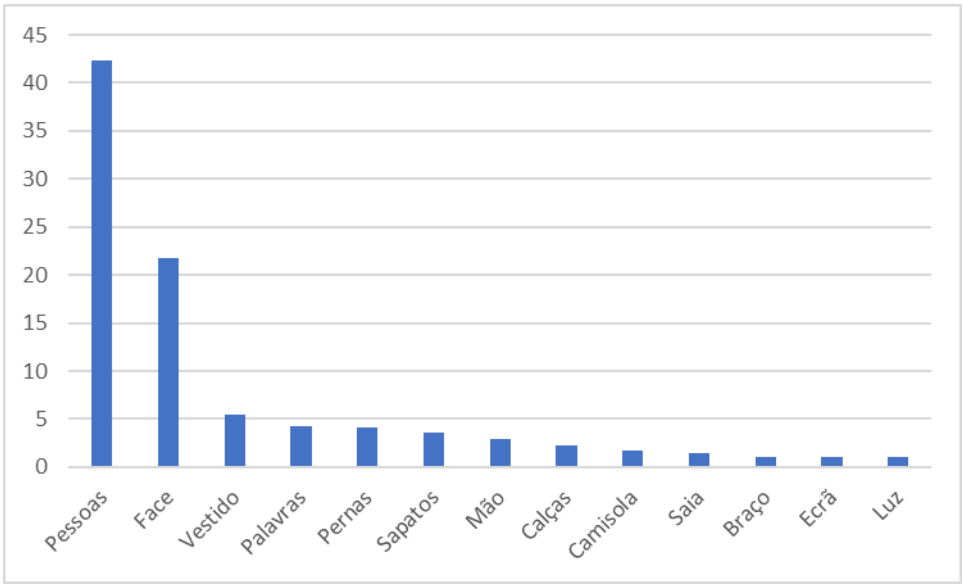


Figura 5.12: Histograma das pontuações das caraterísticas escolhidas nas imagens de moda (percentagem)

5.2.3 Eventos

Nas imagens de eventos, a categoria pessoas representa cerca de 46% das regiões selecionadas, confirmando a previsão anteriormente efetuada, como se pode observar nas Figuras 5.13 e 5.14. Palavras, ecrã, palco e plateia são objetos bastante selecionados devido ao facto de se destacarem das multidões normalmente presentes em eventos. Quando se considera a soma das pontuações ecrã e palco passam a ser mais relevantes que palavras e faces.

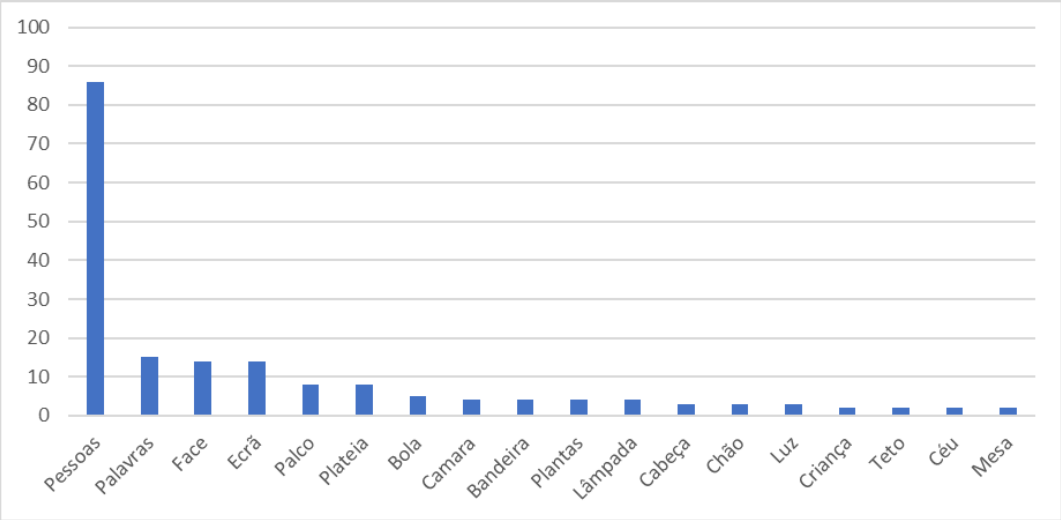


Figura 5.13: Histograma das caraterísticas escolhidas nas imagens de eventos

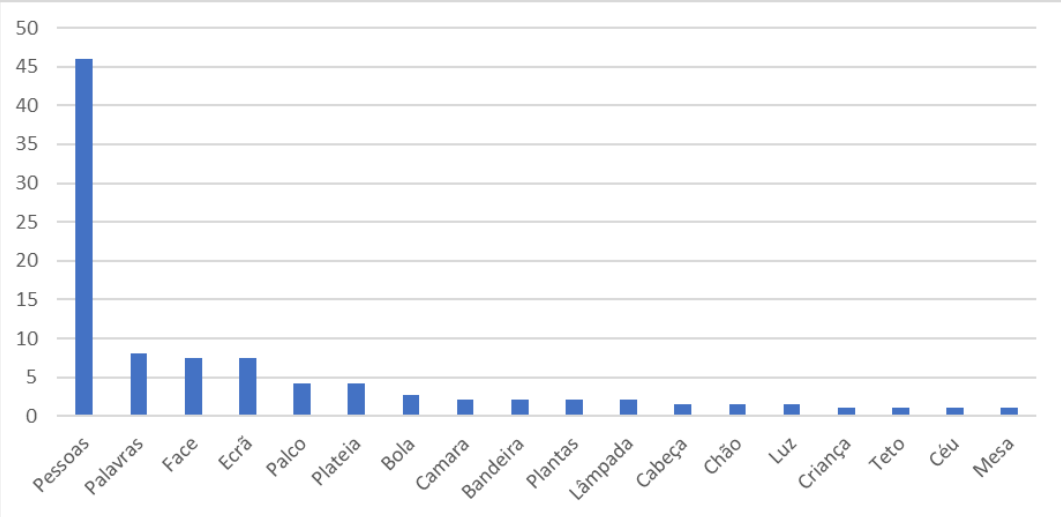


Figura 5.14: Histograma das caraterísticas escolhidas nas imagens de eventos (percentagem)

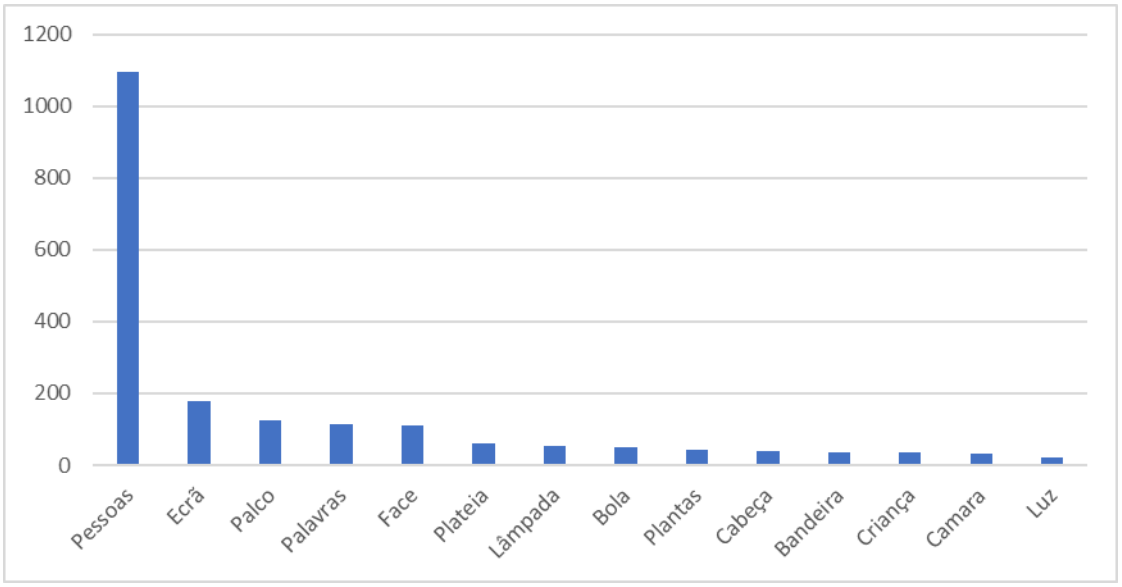


Figura 5.15: Histograma das pontuações das caraterísticas escolhidas nas imagens de eventos

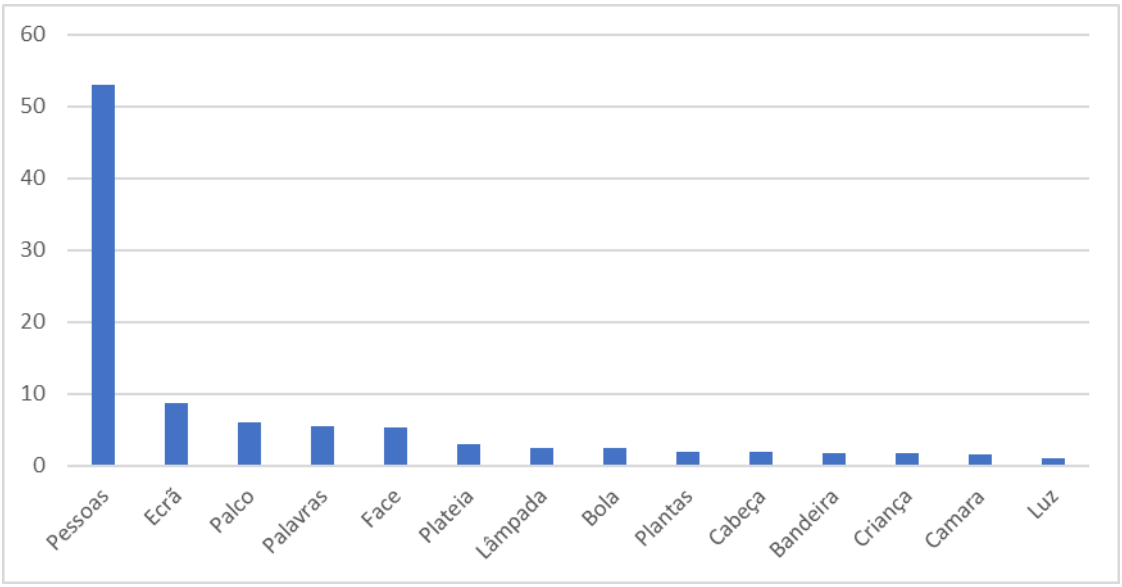


Figura 5.16: Histograma das pontuações das caraterísticas escolhidas nas imagens de eventos (percentagem)

5.3 Análise estatística tendo em conta a prioridade

Além da análise previamente efetuada, também é necessário averiguar a prioridade com que cada objeto é normalmente selecionado. Para tal, verificou-se a percentagem de cada prioridade, de 1 a 5, com que cada categoria era selecionada, nas imagens em que eram selecionadas, sendo 1 a região com maior preferência. O resultado desta análise pode ser observado na Figura 5.17. Como se pode observar, apenas algumas categorias foram selecionadas como primeira preferência em mais de 50% das vezes, sendo elas: pessoas, crianças, ecrã, fogo e palco. Fogo foi a categoria que mais se destacou uma vez que em 64% das imagens em que aparece demonstrou ser a maior região de interesse, provavelmente devido à sua cor intensa. Por outro lado, categorias como braço, olhos, pés, câmara e chão nunca foram a primeira preferência dos anotadores.

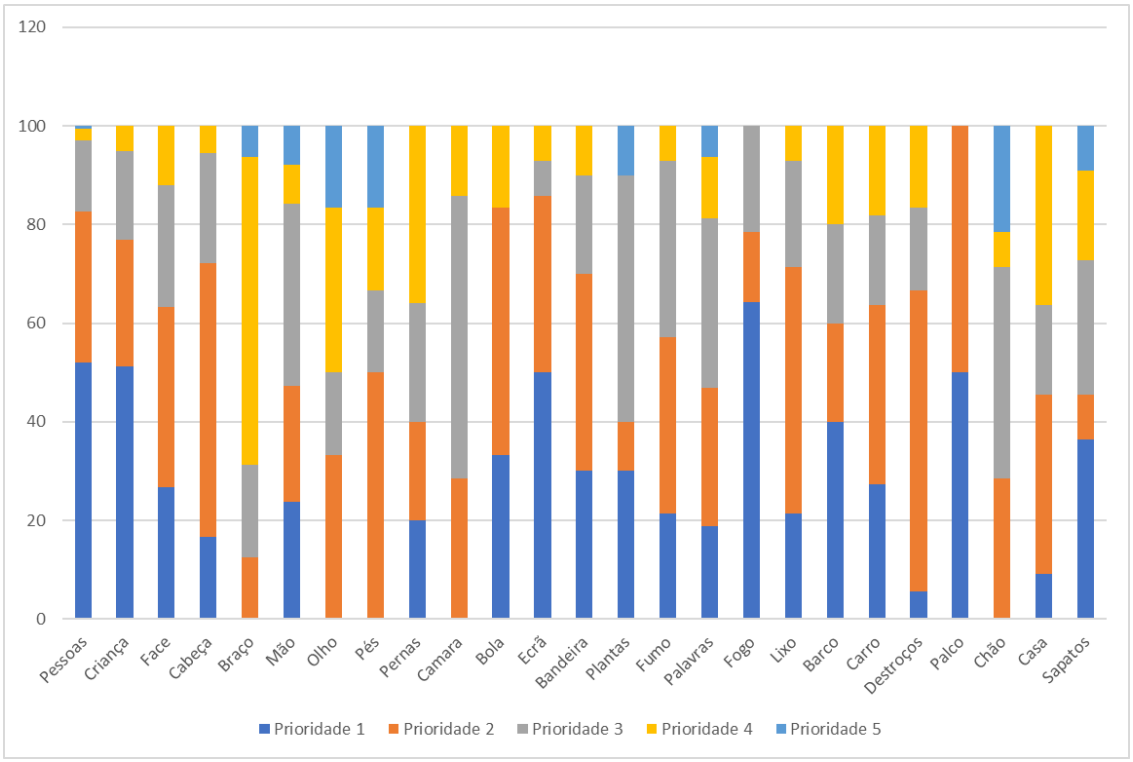


Figura 5.17: Percentagem de cada prioridade para cada categoria

5.4 Discussão sobre o tipo de caraterísticas/objetos percetualmente mais relevantes

Como se pode concluir, pessoas, crianças e partes do corpo são regiões da imagem que de um modo geral captam mais a atenção. Além destas, destroços, lixo e fogo e fumo são objetos de interesse em imagens de fotojornalismo. Por outro lado, nas imagens de moda as peças de roupa têm mais impacto do que nos restantes cenários de aplicação. Em imagens de eventos, certas categorias receberam mais atenção nomeadamente ecrã, palco, palavras e plateia. Mas o destaque vai para as categorias pessoas, crianças, ecrã, fogo e palco pois nas imagens em que estas estão presentes são selecionadas com a prioridade máxima mais de 50% das vezes.

Capítulo 6

Identificação automática de regiões de interesse

Depois da análise dos objetos percetualmente mais relevantes, o objetivo seria integrar detetores dos objetos mais relevantes provenientes desta análise em que o resultado destes detetores seriam as zonas que mais captam a atenção visual da pessoa, na imagem. Cada classe de objeto teria uma prioridade de acordo com a preferência que essa classe obteve na análise previamente efetuada. Deste modo, seria possível comparar o resultado dos detetores com o *ground truth* proveniente das anotações, avaliando a performance do detetor de regiões percetualmente relevantes. Mas como não foi possível obter detetores de todas as classes/categorias relevantes apenas foram usados detetores de pessoas, faces e olhos. Por outro lado, foi efetuada uma comparação entre a cor dominante, luminosidade e o *ground truth*.

6.1 Métrica de avaliação da performance do detetor

Antes da explicação da métrica utilizada foi necessário introduzir alguns conceitos chave para a avaliação da performance de detetores como falsos positivos, falsos negativos, verdadeiros negativos, verdadeiros positivos, precisão e sensibilidade. Falsos positivos consistem em previsões efetuadas pelo detetor que não têm correspondência com o *ground truth* e os falsos negativos consistem nem regiões em que não existe predição por parte do detetor nem *ground truth* nessa mesma região. Por outro lado, verdadeiros negativos consistem nas regiões do *ground truth* que não têm nenhuma correspondência nas previsões do detetor enquanto os verdadeiros positivos consistem nas previsões do detetor com correspondência no *ground truth*. A precisão é a percentagem de elementos selecionados que são relevantes, ou seja, será quantidade de previsões do detetor com correspondência no *ground truth* que será o quociente entre verdadeiros positivos e a soma de verdadeiros positivos e falsos positivos. A sensibilidade será o quociente entre verdadeiros positivos e a soma de verdadeiros positivos e falsos negativos, ou seja, será a percentagem de elementos do *ground truth* com correspondência nas previsões.

A performance do detetor de regiões de interesse foi avaliada através da métrica *mean Average Precision* (*mAP*), definida na competição PASCAL VOC 2012 (Everingham et al.). Segundo esta métrica, as previsões são comparadas com o *ground-truth* e é considerado um verdadeiro positivo caso as regiões a comparar pertençam à mesma classe e o quociente entre a área interseção e a reunião dessas regiões é superior a 50%. Na Figura 6.1 encontra-se ilustrado o calculo do quociente entre a área de interseção e reunião. Mas só é considerado um verdadeiro positivo caso a região do *ground truth* ainda não tenha sido utilizada por forma a evitar várias deteções para a mesma região. De seguida, a partir das correspondências efetuadas, é calculada a curva precisão/sensibilidade para cada classe. A área debaixo desta curva consiste *Average Precision* (*AP*) e por fim, o *mAP* consiste na média do *AP* para todas as classes.

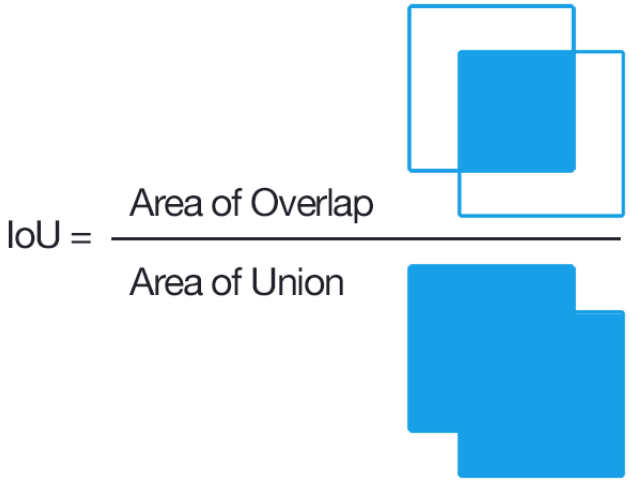


Figura 6.1: Quociente entre a área de interseção e reunião na verificação de verdadeiros positivos

6.2 Identificação de regiões de interesse através de detetores de objetos

Para detetar pessoas recorreu-se ao detetor de objetos YOLO enquanto para detetar faces e olhos recorram-se aos modelos pré treinados do Dlib. A partir da análise do capítulo anterior verificou-se que face era percetualmente mais relevante que olhos e que pessoas era mais relevante que faces e olhos. Por esta razão, foi atribuída a prioridade 1, 2 e 3 a pessoas, faces e olhos respetivamente, em que prioridade 1 significa maior prioridade.

Como a maior prioridade foi atribuída à classe pessoas as predições de pessoas por parte do YOLO serão comparadas com os retângulos com maior prioridade de todas as imagens. Uma vez que a categoria pessoas só obteve maior prioridade em 202 das 260 imagens a sensibilidade nunca poderia ser superior a 40,4%. Por outro lado, uma vez que o YOLO deteta todas as pessoas presentes na imagem e não apenas as que foram consideradas relevantes e presentes no *ground truth* é de esperar um valor baixo para a precisão uma vez que todas as pessoas detetadas pelo YOLO que não estejam no *ground truth* serão encarados como falsos positivos. Como se pode

verificar pela Figura 6.2 o *AP* para a primeira prioridade foi apenas de 3,03% como se pode observar na Figura 6.2. Apesar desta baixa precisão o número de verdadeiros positivos foi de 118, Figura 6.6, em que o máximo que poderia ser alcançado seria de 202 resultando num total de 142 falsos negativos. Este número de falsos negativos pode-se dever ou a imprecisões nas anotação por parte dos anotadores o que pode fazer com que não se atinja a sobreposição necessária para que haja correspondência entre a deteção e o *ground truth* ou a pessoa do *ground truth* pode não ter sido detetada pelo YOLO.

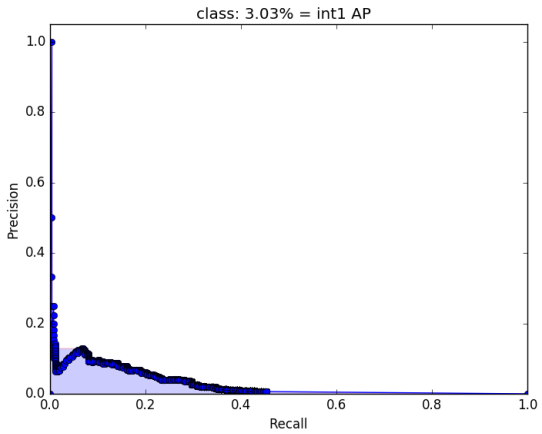


Figura 6.2: Curva precisão/sensibilidade da prioridade 1

Como se viu pela análise do capítulo anterior, quando as regiões selecionadas eram faces, eram na sua maioria escolhidas como segunda prioridade (36% das vezes), portanto, recorreu-se ao Dlib para detetar faces nas imagens e estas regiões que de posteriormente foram comparadas com o *ground truth* da segunda prioridade. Deste modo, obteve-se um *AP* de 0,01%, como se pode visualizar pela Figura 6.3. Este valor é baixo uma vez que foi detetado um número elevado faces sem interesse percetual resultando em 58927 falsos positivos, Figura 6.6. Por outro lado, o número de verdadeiros positivos foi de 20 (Figura 6.6), em que o máximo possível seria 54, uma vez que face foi a zona de segunda prioridade em 54 imagens e o número de falsos negativos é de 240.

Quanto à categoria olhos foi-lhe atribuída prioridade 3, obtendo um *AP* inferior a 0,01% apenas com 1 verdadeiro positivo, como se pode confirmar nas Figuras 6.4 e 6.6. Uma vez que olhos foi escolhido com terceira prioridade apenas uma vez (valor obtido pela análise do capítulo anterior) e existe um verdadeiro positivo conclui-se que não existem falsos negativos. Mais uma vez, a baixa precisão deve-se ao elevado número de falsos positivos e não a um baixo número de verdadeiros positivos.

Uma vez que não foram efetuadas deteções com quarta e quinta prioridade o *AP* para estas classe foi de 0% resultando num *mAP* de 0,61%, como se pode confirmar na Figura 6.5.

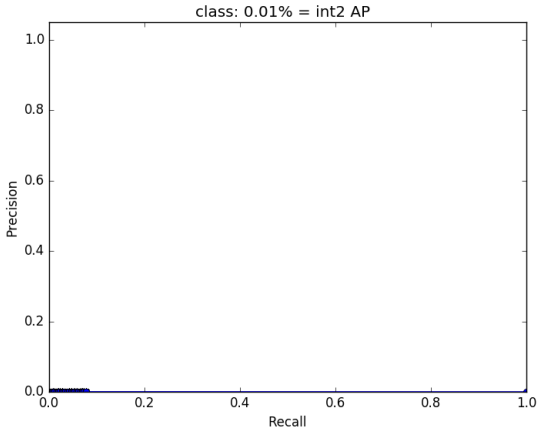


Figura 6.3: Curva precisão/sensibilidade da prioridade 2

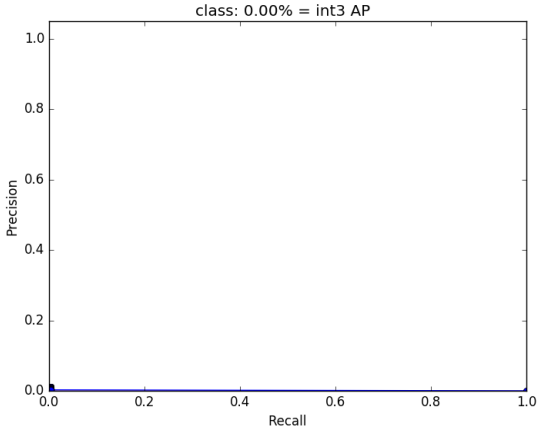


Figura 6.4: Curva precisão/sensibilidade da prioridade 3

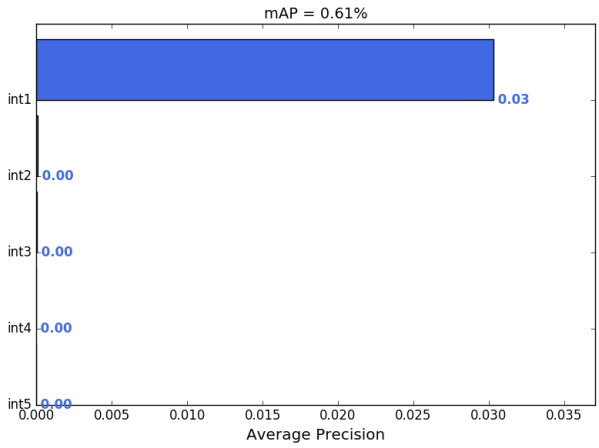


Figura 6.5: *mAP* e *AP* de cada classe

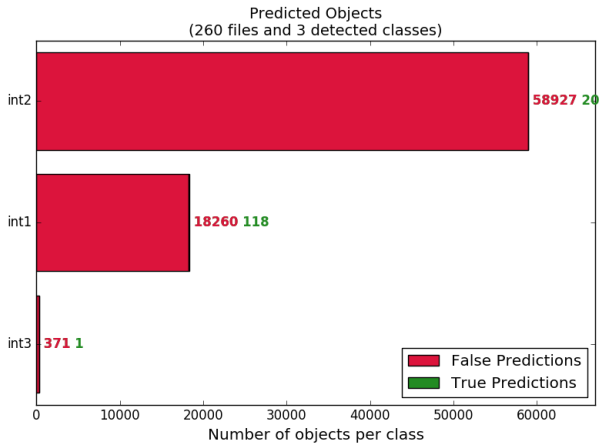


Figura 6.6: Contabilização de falsos positivos e verdadeiros positivos por classe

6.3 Identificação de regiões de interesse através de cor dominante e luminosidade

Para verificar a relação entre as cores dominantes e as regiões percetualmente mais relevantes começou por se efetuar a conversão do espaço de cores RGB para HSV, de seguida efetuou-se o histograma da componente *hue* (H) do HSV de modo a obter a cor dominante. Depois de obtida a a cor dominante, obtiveram-se as zonas da imagem cuja diferença de intensidade do *hue* é de 10, de modo a localizar a cor dominante na imagem. Depois de encontradas as zonas com *hue* próximo da cor dominante a imagem é etiquetada com conetividade 8 destas e de modo a permitir obter o retângulo delimitador da região com maior área.

Para obter a luminosidade, em primeiro lugar, converteu-se as imagens para o espaço de cores Lab e verificaram-se quais as regiões com a componente L superior a 70, ou seja, as zonas com elevada luminosidade. De seguida efetuou-se a etiquetagem com conetividade 8 para obter a região maior área.

Foram efetuadas duas experiências, primeiro foi atribuída às regiões da cor dominante prioridade 1 e à luminosidade prioridade 2 e comparou-se com o *ground truth* e em segundo lugar trocaram-se as prioridades, ou seja, cor prioridade 2 e luminosidade prioridade 1. Com a cor dominante em prioridade 1 e luminosidade com prioridade 2 obteve-se 7 verdadeiros positivos para ambas as prioridades, ou seja, das zonas com maior prioridade sete delas são regiões onde predomina a cor dominante e nas zonas com segunda maior prioridade sete delas contêm alta luminosidade.

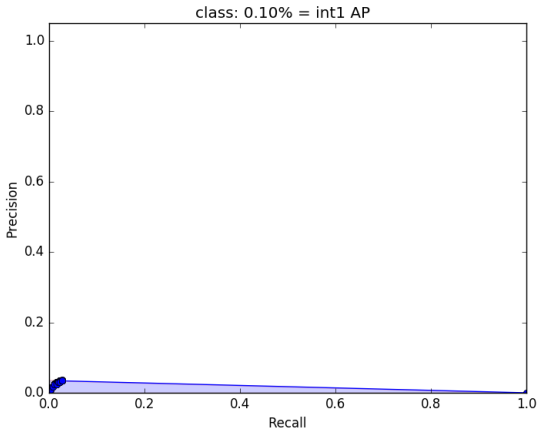


Figura 6.7: Curva precisão/sensibilidade da prioridade 1

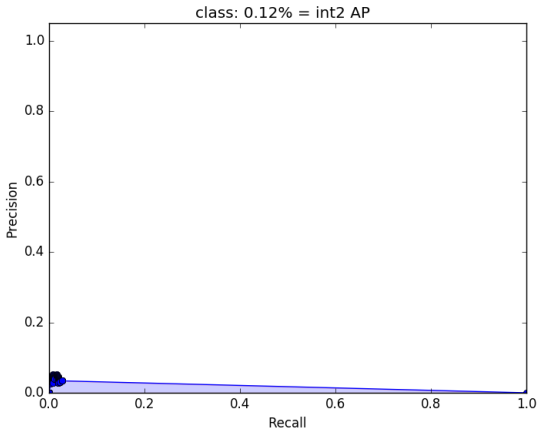


Figura 6.8: Curva precisão/sensibilidade da prioridade 2

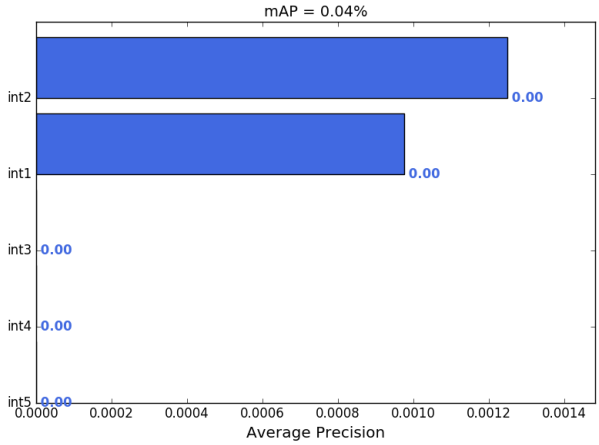


Figura 6.9: *mAP* e *AP* de cada classe

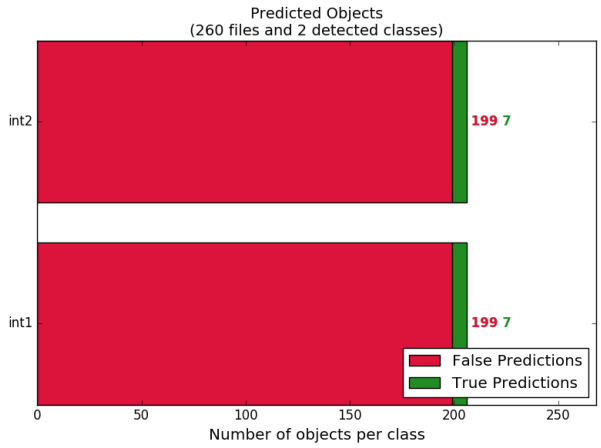


Figura 6.10: Contabilização de falsos positivos e verdadeiros positivos por classe

Por outro lado, quando atribuída prioridade 1 a luminosidade e prioridade 2 a cor o número de verdadeiros positivos é de 5 e 6 para a a luminosidade e cor, respetivamente. Como o número de verdadeiros positivos é muito inferior ao número de imagens a cor dominante e a luminosidade não tem muita influência na atenção visual das pessoas na imagem.

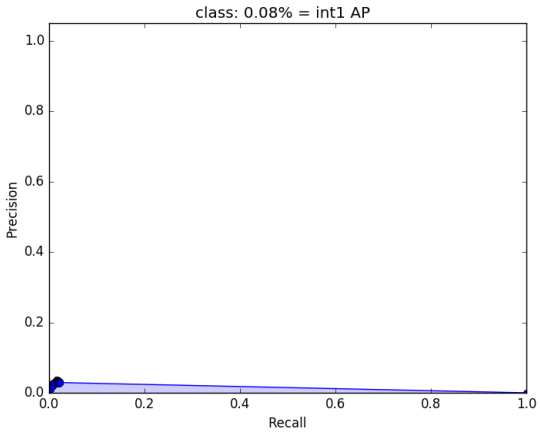


Figura 6.11: Curva precisão/sensibilidade da prioridade 1

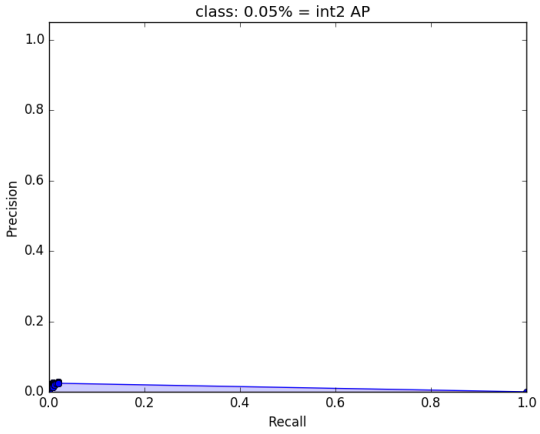


Figura 6.12: Curva precisão/sensibilidade da prioridade 2

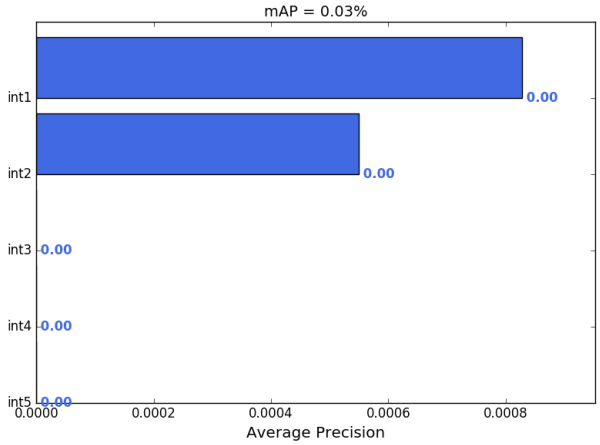


Figura 6.13: *mAP* e *AP* de cada classe

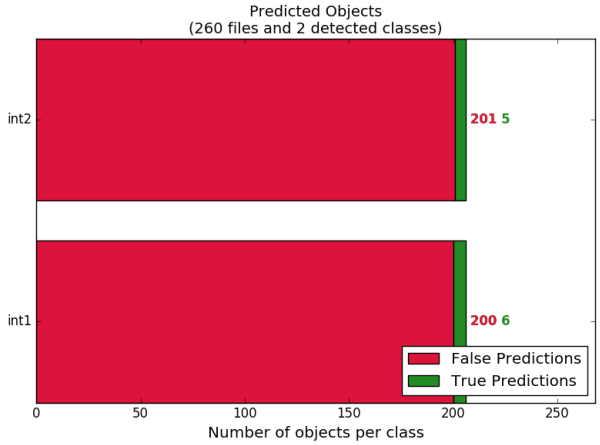


Figura 6.14: Contabilização de falsos positivos e verdadeiros positivos por classe

Capítulo 7

Conclusões e trabalho futuro

7.1 Conclusões

Em primeiro lugar, foi desenvolvido o sistema que efetua a análise contextual da imagem que realiza a classificação da imagem, deteta objeto e cores dominantes. De seguida, foram obtidas as imagens do *dataset* e desenvolveu-se a ferramenta de anotação que permitiu obter as regiões percetualmente das imagens que constituem o *dataset*. Posteriormente, analisou-se quais os objetos percetualmente mais relevantes e por fim desenvolveu-se um detetor de regiões de interesse que consiste na integração de detetores de pessoas, faces e olhos.

O sistema de análise contextual de imagem revelou ser capaz de identificar objetos, faces e pontos de interesse em faces com bastante precisão. Este sistema também consegue efetuar a classificação semântica de imagem, conseguindo identificar objetos na imagem, bem como identificar o contexto da imagem. A partir deste sistema, é também possível conhecer as cores dominantes da imagem. De um modo geral, este sistema tem uma boa performance, mas um tempo de processamento por imagem elevado tornando impraticável o seu uso para analisar grande volume de dados.

A análise dos objetos e caraterísticas percetualmente mais relevantes permitiu conhecer quais os objetos mais relevantes e o seu nível de importância. Desta análise, resultou que pessoas e partes do corpo foram selecionadas com maior frequência no processo de anotação, mas por outro lado alguns objetos foram selecionados menos vezes, mas quando selecionados era-lhe atribuída uma prioridade elevada, nomeadamente "fogo" 64% das vezes que foi selecionado obteve a prioridade máxima.

A identificação automática de regiões de interesse obteve maus resultados, com um *mAP* de apenas 0,61% quando utilizados detetores de objetos para detetar as regiões de interesse. Este resultado deve-se maioritariamente ao facto de não serem detetadas as pessoas, faces e olhos com maior interesse, mas sim todas presentes nas imagens levando a um elevado número de falsos positivos e por conseguinte um *mAP* muito baixo.

7.2 Trabalho futuro

Uma vez que os resultados obtidos na identificação automática de imagens não foram satisfatórios seria melhor seguir outra abordagem para a deteção e zonas de interesse. Essa abordagem seria usar o *ground truth* do *dataset* para treinar um algoritmo aprendizagem máquina do modo a se efetuar a deteção automática de regiões de interesse. Para tal, seria necessário aumentar o tamanho do *dataset* uma vez que para os algoritmos aprendizagem máquina obterem boa precisão necessitam de grande volume de dados para efetuar o treino.

Referências

Amazon. Amazon rekognition. Disponível em https://aws.amazon.com/rekognition/, acedido a última vez em 27 de janeiro de 2018.

A. Borji, M. M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722, Dec 2015. ISSN 1057-7149. doi: 10.1109/TIP.2015.2487833.

Sergi Caelles, Alberto Montes, Kevis-Kokitsi Maninis, Yuhua Chen, Luc Van Gool, Federico Perazzi, and Jordi Pont-Tuset. The 2018 davis challenge on video object segmentation. *arXiv:1803.00557*, 2018.

CAVIAR. Caviar project, Junho 2006. URL http://homepages.inf.ed.ac.uk/rbf/CAVIAR/. Último acesso em 2018/04/08.

D. Ciregan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, June 2012. doi: 10.1109/CVPR.2012.6248110.

Clarifai. Clarifai. Disponível em https://www.clarifai.com/, acedido a última vez em 27 de janeiro de 2018.

Charles E. Connor, Howard E. Egeth, and Steven Yantis. Visual attention: Bottom-up versus top-down. *Current Biology*, 14(19):R850 – R852, 2004. ISSN 0960-9822. doi: https://doi.org/10.1016/j.cub.2004.09.041. URL http://www.sciencedirect.com/science/article/pii/S0960982204007250.

Ana-Maria Cretu, Pierre Payeur, and Robert Laganière. An application of a bio-inspired visual attention model for the localization of vehicle parts. *Applied Soft Computing*, 31:369 – 380, 2015. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2015.02.032. URL http://www.sciencedirect.com/science/article/pii/S1568494615001337.

M. Dahmane, S. Foucher, and D. Byrns. Are you smiling as a celebrity? latent smile and gender recognition. In Fakhri Karray, Aurélio Campilho, and Farida Cheriet, editors, *Image Analysis and Recognition*, pages 304–311, Cham, 2017. Springer International Publishing. ISBN 978-3-319-59876-5.

A. Dixit and N. P. Hegde. Image texture analysis - survey. In *2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT)*, pages 69–76, April 2013. doi: 10.1109/ACCT.2013.49.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, January 2015.

flickr. flickr *website*. Disponível em https://www.flickr.com/, acedido a última vez em 18 de junho de 2018.

Google. Google vision api. Disponível em https://cloud.google.com/vision/, acedido a última vez em 19 de janeiro de 2018.

Mingwei Guo, Yuzhou Zhao, Chenbin Zhang, and Zonghai Chen. Fast object detection based on selective visual attention. *Neurocomputing*, 144:184 – 197, 2014. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2014.04.054. URL http://www.sciencedirect.com/science/article/pii/S0925231214006857.

T. Guo, J. Dong, H. Li, and Y. Gao. Simple convolutional neural network on image classification. In *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*, pages 721–724, March 2017. doi: 10.1109/ICBDA.2017.8078730.

Jan Hendrik Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. *CoRR*, abs/1705.02950, 2017. URL http://arxiv.org/abs/1705.02950.

Hossein Hosseini, Baicen Xiao, and Radha Poovendran. Google's cloud vision API is not robust to noise. *CoRR*, abs/1704.05051, 2017. URL http://arxiv.org/abs/1704.05051.

Yiqun Hu, Deepu Rajan, and Liang-Tien Chia. Detection of visual attention regions in images using robust subspace analysis. *Journal of Visual Communication and Image Representation*, 19(3):199 – 216, 2008. ISSN 1047-3203. doi: https://doi.org/10.1016/j.jvcir.2007.11.001. URL http://www.sciencedirect.com/science/article/pii/S104732030700096X.

Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10: 1755–1758, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc. URL http://dl.acm.org/citation.cfm?id=2999134.2999257.

L. Li, L. Feng, S. Liu, and Y. Liu. Local co-occurrence pattern for color and texture image retrieval. In *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, pages 1212–1217, June 2016. doi: 10.1109/WCICA.2016.7578339.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

Y. Lu, W. Zhang, C. Jin, and X. Xue. Learning attention map from images. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1067–1074, June 2012. doi: 10.1109/CVPR.2012.6247785.

Microsoft. Microsoft vision api. Disponível em https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/, acedido a última vez em 21 de janeiro de 2018.

G.A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39 – 41, 1995. ISSN 0001-0782. URL http://dx.doi.org/10.1145/219717.219748. lexical database;English nouns;English verbs;English adjectives;English adverbs;semantic relations;word definitions;WordNet;.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. *CoRR*, abs/1406.6247, 2014. URL http://arxiv.org/abs/1406.6247.

F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016.

Pexels. Pexels. Disponível em https://static.pexels.com/photos/61100/pexels-photo-61100.jpeg, acedido a última vez em 22 de janeiro de 2018.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, June 2016. doi: 10.1109/CVPR.2016.91.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

R.A. Rensink. The dynamic representation of scenes. *Visual Cognition*, 7(1-3):17 – 42, 2000/01/. ISSN 1350-6285. URL http://dx.doi.org/10.1080/135062800394667. dynamic representation;scenes;change blindness;focused attention;spatiotemporal coherence;virtual representation;stable object representation;.

Adrian Rosebrock. Pyimagesearch. Disponível em https://www.pyimagesearch.com/, acedido a última vez em 4 de junho de 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1):157–173, May 2008. ISSN 1573-1405. doi: 10.1007/s11263-007-0090-8. URL https://doi.org/10.1007/s11263-007-0090-8.

G. A. S. Saroja and C. H. Sulochana. Texture analysis of non-uniform images using glcm. In *2013 IEEE Conference on Information Communication Technologies*, pages 1319–1322, April 2013. doi: 10.1109/CICT.2013.6558306.

H. Shao, Y. Wu, W. Cui, and J. Zhang. Image retrieval based on mpeg-7 dominant color descriptor. In *2008 The 9th International Conference for Young Computer Scientists*, pages 753–757, Nov 2008. doi: 10.1109/ICYCS.2008.89.

Xiaoshuai Sun, Hongxun Yao, and Rongrong Ji. Visual attention modeling based on short-term environmental adaption. *J. Vis. Comun. Image Represent.*, 24(2):171–180, February 2013. ISSN 1047-3203. doi: 10.1016/j.jvcir.2012.01.014. URL http://dx.doi.org/10.1016/j.jvcir.2012.01.014.

Xudong Sun, Pengcheng Wu, and Steven C. H. Hoi. Face detection using deep learning: An improved faster RCNN approach. *CoRR*, abs/1701.08289, 2017. URL http://arxiv.org/abs/1701.08289.

TensorFlow. Tensorflow, 2008. Disponível em https://www.tensorflow.org/, acedido a última vez em 9 de junho de 2018.

Tadmeri Narayan Vikram, Marko Tscherepanow, and Britta Wrede. Impact of real-time visual attention on computer vision products and cognitive robotics. *Procedia Computer Science*, 7:332 – 333, 2011. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2011.09.059. URL http://www.sciencedirect.com/science/article/pii/S1877050911006193. Proceedings of the 2nd European Future Technologies Conference and Exhibition 2011 (FET 11).

wikipedia. wikipedia *website*. Disponível em https://pt.m.wikipedia.org, acedido a última vez em 18 de junho de 2018.

L. Wu. Detection of salient region of in-field rapeseed plant images based-on visual attention model. In *2017 2nd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pages 33–36, June 2017. doi: 10.1109/ACIRS.2017.7986060.

Xueyi Ye, X. Chen, H. Chen, Yafeng Gu, and Qiuyun Lv. Deep learning network for face detection. In *2015 IEEE 16th International Conference on Communication Technology (ICCT)*, pages 504–509, Oct 2015. doi: 10.1109/ICCT.2015.7399887.

S. S. Yu, S. Y. Huang, Y. H. Pan, and H. C. Wu. An easy dominant color extraction and edge valley histogram for image retrieval. In *2010 International Computer Symposium (ICS2010)*, pages 159–164, Dec 2010. doi: 10.1109/COMPSYM.2010.5685526.

Zalando. Fashion-mnist. URL https://www.kaggle.com/zalando-research/fashionmnist. Último acesso em 2018/04/08.

R. Zhang, X. Qian, and D. Ye. A modified gray-level difference algorithm for analysing gaussian blurred texture images. In *2011 4th International Congress on Image and Signal Processing*, volume 2, pages 833–837, Oct 2011. doi: 10.1109/CISP.2011.6100298.