Relatório Final

Auxílio Regular FAPESP - Processo 2018/23392-5

Representações unificadas considerando atributos visuais e de semântica textual em tarefas de reconhecimento em imagens

Juliana M. Crivelli / Moacir A. Ponti

Instituto de Ciências Matemáticas e de Computação — Universidade de São Paulo

São Carlos, SP

Março de 2019 - Janeiro de 2020

Resumo

Com a complexidade multidimensional que imagens apresentam e com seu crescente volume disponível como base para os mais diversos tipos de análises, surge a necessidade de criar representações para imagens que sejam mais compactas e que ainda assim contenham as informações relevantes para cada tarefa. Assim, este trabalho investiga o espaço latente de imagens quando apresentadas à CNNs e propõe uma forma de obter representações que aumentam a acurácia em tarefas de classificação porém são mais compactas que o espaço original. O método desenvolvido parte da hipótese de que a combinação de diferentes tipos de características extraídas é capaz de melhorar a performance em uma dada representação. Para isso, o trabalho explora características visuais, extraídas com CNNs, características semânticas e espaciais obtidas com o auxílio da rede YOLO v3 e um método de fusão via concatenação direta. Como contribuição secundária, identificamos um possível método para detecção de exemplos adversariais ao analisar a compatibilidade do classificador SVM como avaliador da representação proposta.

Abstract

With the multidimensional complexity that images present and their growing volume, there is a necessity of creating more compact representations that still retain the relevant information needed for each kind of analysis. Therefore, this work does an investigation on the latent space of images through CNNs and propose a method to obtain more accurate representations with more compact space. The developed method has a hypothesis that the combination of different types of extracted characteristics is capable of improving performance in a given representation. To test that, we used visual characteristics extracted with CNNs, semantic and spatial characteristics obtained with YOLO v3 and a method of direct concatenation. As a secondary contribution, we identified a plausible method for detecting adversarial examples when analyzing the SVM classifier as an evaluation metric for the proposed representation.

Sumário

1 Introdução

O sistema cognitivo-visual humano é capaz de analisar e interpretar com precisão imagens complexas, de modo quase instantâneo, abstraindo conceitos visuais a partir de múltiplos elementos em uma cena. Por exemplo, é possível categorizar uma fotografia com pessoas no tema trabalho ou férias com base em atributos visuais como as roupas com as quais as pessoas estão vestidas, e os objetos na cena. Humanos também são capazes de aprender a partir da visualização de poucos exemplos, e gerar uma variedade de descrições textuais em cima da mesma cena, abordando diferentes aspectos observados [Gonzalez and Woods, 2002].

Nesse contexto, o estudo de representações digitais de imagens é relevante para diversas aplicações, em particular porque o espaço original das imagens digitais (contendo pixels com suas codificações de intensidade e cor) possui alta dimensionalidade, em muitos casos inviabilizando seu uso direto em tarefas de reconhecimento como: classificação, agrupamento e recuperação. Assim, é comum utilizar um passo intermediário que extrai características das imagens, gerando um espaço mais compacto que possa ser utilizado nas tarefas mencionadas [Penatti et al., 2012].

Métodos como Bag of Visual Words (BoVW) [Fei-Fei and Perona, 2005], Fisher Vectors [Perronnin et al., 2010] foram propostos em meados da década de 2000 e até próximo a 2010, sendo o estado da arte nessa década por melhor codificar características visuais em uma cena, obtendo dicionários visuais com os quais relacionar características (de cor, textura, orientação, etc.) encontradas em imagens [Haralick et al., 1973; Ponti et al., 2016]. Na década de 2010, com o surgimento métodos de redes neurais profundas (*deep networks*) em suas versões convolucionais (CNNs, do inglês *Convolutional Neural Networks*), essas passaram a dominar as soluções de visão computacional quando se trata da extração de características. Entre as redes neurais, destacamos os métodos AlexNet, VGGNet, Inception e ResNet [Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2016]. Mais do que extrair características, essas redes permitem realizar o aprendizado de características ou representações [Bengio et al., 2013].

Diversos estudos mostraram que características obtidas a partir de redes neurais profundas podem ser aplicadas com sucesso a problemas de reconhecimento visual [de Nazare et al., 2018; Cavallari et al., 2018; Bui et al., 2018; Ponti et al., 2019]. Ainda que produzindo maior acurácia do que métodos de extração de características tradicionais [Sharif Razavian et al., 2014], as características baseadas em CNNs são pouco interpretáveis, em particular quando considerados contextos, planos de fundo e posições dos objetos nas cenas [Ponti et al., 2017], e prejudicando a performance de sistemas com poucos dados anotados [Fu et al., 2014]. Além disso, CNNs podem ser enganadas por ataques de ima-

gens adversariais (geradas com o propósito de causar predições errôneas) [Nguyen et al., 2015].

Assim, identificou-se espaço para estudos que analisem as representações geradas por CNNs, bem como no enriquecimento dessas representações de maneira a melhorá-las para uso em tarefas de reconhecimento. Esse projeto investigou as características visuais obtidas a partir de camadas de redes neurais convolucionais, e ainda a combinação dessas características com atributos semânticos e espaciais obtidos a partir de anotações ou por meio de redes neurais de detecção de objetos [Redmon et al., 2015].

Como ponto de partida, foram utilizadas representações de imagens produzida por CNNs pré-treinadas [Sharif Razavian et al., 2014] para a obtenção de características visuais abstratas e o espaço de características obtidas é avaliado através dos métodos de visualização t-SNE (t-distributed stochastic neighbor embedding) [Maaten and Hinton, 2008] e PCA (Principal Component Analysis). Em seguida, análises com diferentes espaços de características, e resultados da fusão de características visuais e semânticas, avaliados comparando o seu desempenho com um classificador SVM.

A seguir, apresentamos conceitos fundamentais utilizados no projeto na Seção 2. Posteriormente, apresentamos na Seção 3 o método considerado no projeto e os recursos utilizados para os experimentos realizados. Na Seção 4 detalhes do desenvolvimento e resultados obtidos. Finalmente, conclusões e trabalhos futuros são reportados na Seção 5.

2 Conceitos fundamentais

Nesta seção são apresentados, de forma breve, conceitos e técnicas que servem como base para entender os resultados obtidos. Em particular descrevemos características visuais, e características obtidas a partir de redes neurais profundas aplicadas a problemas de reconhecimento visual. Ainda, descrevemos brevemente métodos de aprendizado de máquina e aprendizado profundo que apoiam essas tarefas. Os trabalhos de Ponti et al. [2016] e Penatti et al. [2012] possuem maiores detalhes dos conceitos e aplicações de características visuais; e os trabalhos de Bengio et al. [2013], Ponti et al. [2017], Goodfellow et al. [2016] são referências relevantes nas áreas de aprendizado de características a partir de redes neurais.

2.1 Representações e características

Há diferentes tipos de características que podem ser obtidas a partir do conteúdo visual, gerando representações abstratas e de baixo nível ou ainda representações semânticas de alto nível. Representações abstratas dizem respeito à características visuais de baixo nível, ou que possam ser computadas a partir de pixels: cor, textura, orientação entre outros. Em contraste, as representações semânticas estão relacionadas ao conteúdo percebido pelo sistema visual humano, incluindo categorias de objetos, cenas ou ações. Finalmente, o texto permite flexibilidade semântica para interpretar uma cena, gerando descrições ainda mais complexas.

2.1.1 Características visuais

As características de imagens manuais (em inglês é comumente usado o termo *handcrafted*), computados diretamente a partir dos pixels, mais comumente exploradas são cor, textura e forma ou orientação dos objetos e regiões em uma imagem:
— **Cor**: modo de perceber e interpretar uma determinada frequência de luz. O espectro de luz visível aos humanos corresponde aproximadamente a faixa de 400 a 700nm e suas cores podem ser especificadas como a combinação em diferentes proporções de vermelho, verde e azul (Modelo RGB) [Ponti et al., 2016; Ponti and Escobar, 2013].
— **Textura**: descreve a imagem em termos de sua suavidade, rugosidade e regularidade. As principais abordagens entre os descritores de texturas partem de medidas estatísticas, técnicas estruturais e baseadas no espectro de Fourier [Haralick et al., 1973].
— **Orientação**: características que capturam a forma dos objetos por meio das orientações das principais regiões e bordas, como por exemplo o método de Histograma de Orientação do Gradiente (HOG) [Dalal and Triggs, 2005].

A literatura recente também utiliza, em substituição aos métodos acima, **características *CNN off-the-shelf***, utilizando uma rede neural convolucional pré-treinada como extrator de características. Modelos baseados nas redes VGGNet [Simonyan and Zisserman, 2014], Inception [Szegedy et al., 2015], ResNet [He et al., 2016] e MobileNet [Howard et al., 2017] são escolhas comuns. Nesse caso, se fornece como entrada as imagens e as características são as saídas de alguma das camadas da rede neural. Apesar da perda da interpretabilidade dessas representações (em que torna-se difícil identificar quais características dizem respeito aos aspectos visuais), essas se mostraram superiores do ponto de vista de sua capacidade discriminativa, em particular para tarefas de classificação e recuperação de imagens baseada em conteúdo [Sharif Razavian et al., 2014; Ponti et al., 2017; dos Santos and Ponti, 2019; Bui et al., 2018].

2.1.2 Atributos semânticos

Em contraste com os atributos visuais obtidos a partir dos pixels e suas relações na imagem, os atributos semânticos se referem ao significado simbólico dos elementos na imagem, ou ainda a interpretação de uma cena. Em aplicações de aprendizado de máquina que lidam com classificação, a classe obtida para cada entrada pode ser um atributo semântico, associando a imagem a uma categoria entre as pré-estabelecidas na aplicação.

2.1.3 Atributos espaciais

Geralmente vinculados a informação semântica, os atributos espaciais são relativos ao posicionamento de objetos ou de suas características no espaço da imagem. Podem ser dados em aproximação, por exemplo com *bounding box* (um retângulo que delimita o espaço ocupado), como os apresentados em conjunto com a categoria por objeto em [Redmon et al., 2015] ou por um conjunto de pixels segmentados, como em [Lin et al., 2014].

2.1.4 Atributos textuais

É possível estabelecer uma relação entre o conteúdo visual e sua semântica e uma descrição textual que um humano pode produzir ao anotar esse conteúdo. Um exemplo ocorre em recuperação de imagens baseada em conteúdo: nesse cenário os atributos textuais formam uma string de busca pelas imagens desejadas Wang and Zhang [2009]. Outro trabalho usa uma descrição textual das características visuais Tsai et al. [2012], com a vantagem de diminuir o tamanho da representação mantendo a performance do sistema.

2.2 Aprendizado de Máquina

Aprendizado de máquina é um campo da ciência da computação contido dentro da área de inteligência artificial, utilizando técnicas matemáticas e estatísticas, de teoria da informação e até mesmo inspiradas na biologia (como as redes neurais) para criar programas capazes de melhorar sua performance em uma tarefa através de experiência [Mitchell, 1997].

O aprendizado se caracteriza por meio da inferência de uma função ou mapeamento, aprendida por meio da inspeção de exemplos a partir dos quais se computa os parâmetros do modelo de forma que esse seja capaz de generalizar para dados futuros [Mello and Ponti, 2018]. Por exemplo, desejamos inferir $f : X \rightarrow Y$, em que X é o espaço de imagem (ou representação das imagens), e Y é o espaço dos rótulos (ou classes, categorias) a qual a imagem pertence. Ou $f : X \rightarrow \hat{X}$, em que $\hat{X}$ é uma reconstrução ou versão processada da imagem X.

Avaliar o quanto a função aprendida se aproxima da função desejada para a aplicação depende do tipo de tarefa. Para a classificação (designar uma classe para um exemplo dado), por exemplo, é comum utilizar medidas baseadas no erro entre a saída da função e o rótulo real.

Neste trabalho utilizamos redes neurais, um paradigma de aprendizado de máquina construído a partir de múltiplas unidades simples, comumente chamadas de neurônios. Esses compõem camadas que se conectam de modo a obter $f()$. Conforme citado anteriormente, apesar da dificuldade na interpretabilidade dos pesos e resultados intermediários obtidos em uma rede neural, elas tem se mostrado eficazes para problemas de classificação de imagem [Ponti et al., 2017]. Entre os conceitos essenciais para o entendimento de redes neurais e dos métodos utilizados neste trabalho podemos destacar Goodfellow et al. [2016]:

– **Neurônio**: unidade básica de uma camada densa ou convolucional em uma rede neural. Realiza uma combinação linear de seus valores de entrada. Cada neurônio possui um conjunto de pesos, que determinam a importância de uma determinada entrada para a saída daquele neurônio.

– **Camada densa**: todos os neurônios de uma camada densa (também conhecida como *fully connected* ou FC) estão conectados a todos os neurônios da camada anterior. Deste modo, as entradas de um neurônio são todas as saídas da camada prévia, o qual gera uma única saída.

– **Camada convolucional**: recebe uma imagem (ou tensor) como entrada e cada neurônio da camada aplica nela um filtro em uma operação de convolução. O filtro atua como uma matriz de pesos em cima de um pixel e seus vizinhos e é o mesmo para toda a imagem naquele neurônio, gerando $\hat{X}$. A saída é uma nova imagem, ou tensor.

Deep learning: permite aprender representações hierárquicas por meio da com-

posição de camadas [Goodfellow et al., 2016]. Assim, ao invés de aprender uma única função que mapeia o espaço de entrada no espaço objetivo, métodos profundos visam aprender uma série de funções aninhadas que comumente representam as L camadas de redes neurais:

$$
\begin{aligned}
f_1 &: X \rightarrow X_1 \\
f_2 &: X_1 \rightarrow X_2 \\
&\cdots \\
f_L &: X_{L-1} \rightarrow X_L
\end{aligned}
$$

3 Materiais e Métodos

3.1 Visão geral do método

A Figura 1 exibe a sequência de procedimentos do uso de diferentes atributos para o aprendizado de representações. Esses atributos são combinados ou selecionados de maneira a melhorar sistemas de reconhecimento, como por exemplo a classificação ou a busca visual. Assim, esse projeto investigou métodos que extraem atributos (1, 2 e 3), e posteriormente os combinando em uma representação unificada (4), a qual é finalmente avaliada em cenários de reconhecimento (5).

Primeiramente, foi feita uma análise dos espaços de caraterísticas (1) e sua robustez. A seguir, as etapas seguintes investigaram outros atributos (2,3), bem como sua combinação (4);

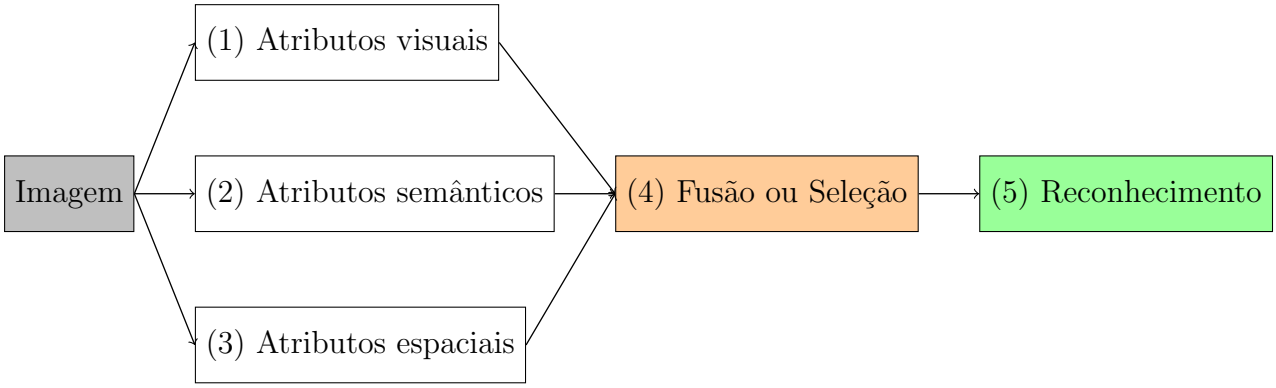

Figura 1: Sequência completa dos procedimentos para investigar diferentes atributos, combinar e realizar reconhecimento sobre o aprendizado realizado.

3.2 Investigação e implementação

O projeto foi estruturado em 3 etapas com experimentos distintos mas que avançam gradualmente na direção do método proposto através da aquisição de conhecimentos que se complementam e são aplicados na etapa final. No primeiro momento, foi explorado o conceito de espaço latente e visualização. Em seguida, o método de avaliação foi estudado. Por fim, o experimento final implementou o método proposto através de:

- **Extração de características visuais**: o principal método investigado consiste na extração de espaços de características por meio do uso de redes neurais do tipo CNN pré-treinadas [Sharif Razavian et al., 2014] ou com ajuste fino quando a base de

dados possui exemplos rotulados disponíveis. O espaço de características foi definido como o espaço latente (também chamado de *feature embedding*) das CNNs, obtidos das saídas das camadas da rede, na maior parte das vezes a penúltima camada, imediatamente anterior à predição.

- **Extração de características semânticas**: utilizando as anotações existentes na base de dados (na forma de oráculo), ou por meio de redes neurais que detectam e localizam objetos utilizando o método YOLO [Redmon et al., 2015].

- **Extração de características espaciais**: utilizando anotações existentes na base de dados (na forma de oráculo).

- **Combinação das representações**: como forma de comparar o espaço latente gerado, aplicamos ainda combinações das representações [Ponti Jr, 2011] por meio de fusão por concatenação (*early fusion*): em que os espaços são concatenados.

3.3 Bases de Dados

3.3.1 CIFAR-10

Para o propósito de estudos iniciais sobre o espaço latente foi utilizada a base de dados CIFAR-10, pois é simples o suficiente para prototipagem rápida dos experimentos mas ainda assim possui uma certa complexidade que exige maior atenção à arquitetura da rede e treinamento para obter maior acurácia. A base contém imagens pertencentes a dez classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck (avião, automóvel, pássaro, gato, cervo, cachorro, sapo, cavalo, navio e caminhão). A base consiste em 6000 imagens coloridas por classe, de 32 por 32 pixels. Os grupos de treinamento e teste possuem respectivamente 50000 e 10000 imagens. A Figura 2 possui exemplos de imagens presentes na base de dados.

3.3.2 ImageNet e exemplos com formas aleatórias

Com o objetivo de estudar o método de avaliação escolhido, a eficácia e a coerência do classificador SVM foram avaliadas em cima de duas bases diferentes porém com as mesmas classes possíveis como resultado do classificador.

Como base de referência para a avaliação foi utilizada a ImageNet [Deng et al., 2009], na qual foi pré-treinada a rede VGG-16 (modelo disponível para uso público através da ferramenta keras). Mais especificamente, foram utilizados 50 exemplos de cada classe correspondente àquelas que foram previstas com um grande valor de acurácia na base de comparação.

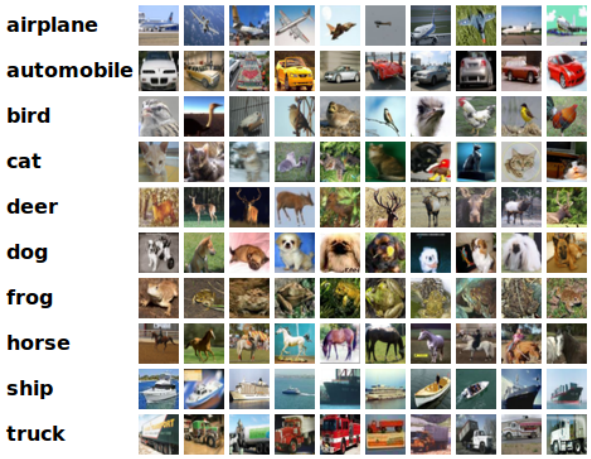

Figura 2: Exemplos de imagens da base de dados CIFAR-10, disponível em ¡https://www.cs.toronto.edu/ kriz/cifar.html¿. Acesso em 06/06/2020.

Como base de comparação foi criado um gerador de imagens aleatórias que utiliza formas geométricas poligonais, retas e curvas de bézier de acordo com esquemas de cores para criar exemplos abstratos, que a princípio não pertencem a classe alguma. Entre as 10 000 imagens geradas, foram selecionadas para o estudo aquelas que obtiveram uma previsão de classe com valor de acurácia acima do limite estabelecido. O experimento é detalhado na seção Desenvolvimento. A Figura 3 apresenta alguns dos exemplos gerados.

Figura 3: Exemplos de imagens aleatórias geradas

3.3.3 PASCAL VOC

Para a avaliação final foi utilizado o dataset PASCAL Visual Object Classes (VOC), que contém em um de seus subsets imagens de uma ou mais pessoas suas respectivas bounding boxes (informação espacial na forma de anotações textuais) e a ação que elas estão realizando, por exemplo tocar um instrumento musical, caminhar ou usar o computador. São ao total 10 classes de ações e uma classe de outros, que abrange casos onde foram detectadas pessoas porém as ações realizadas não se encaixam nas 10 categorias. Hipotisamos que informações relativas ao contexto auxiliariam a prever a ação, mesmo que o objetivo final seja apenas classificar a ação realizada. Por esta razão, além

do treinamento que obtém características visuais através de CNNs investigamos outras características semânticas das imagens utilizando a rede YOLO. Assim, esta base se apresenta ideal para o proposto neste projeto, pois contém características espaciais anotadas e as imagens nela contidas apresentam um campo visual amplo, não se focando em um único objeto possibilitando também a análise do contexto.

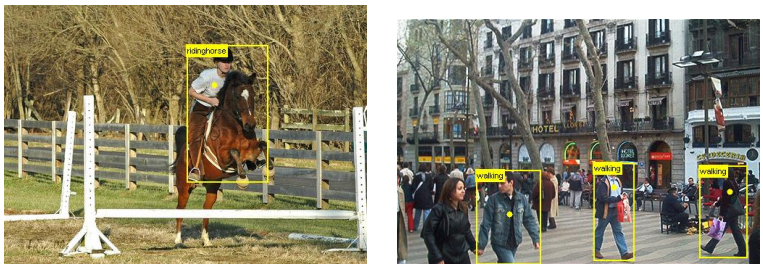

Figura 4: Exemplos do dataset PASCAL VOC para detecção de pessoas e ações

3.4 Métodos de Visualização

O espaço de características gerado é multidimensional. Deste modo é necessário utilizar um método de redução da dimensionalidade para possibilitar a visualização. É desejado que o número de variáveis do conjunto de dados seja reduzido preservado ao máximo a informação contida nele. Assim, buscou-se aplicar dois métodos bastante conhecidos na literatura, descritos a seguir.

3.4.1 PCA (*principal component analysis*)

O método PCA funciona identificando quais componentes são mais relevantes para a informação e reorganizando-a de modo a apresentar uma representação mais compacta, que apesar de possuir uma certa perda de precisão ainda é capaz de reconstituir a informação original com um alto grau de fidelidade. Geralmente é necessária a normalização das variáveis para que uma não tenha predominância sobre outra. Em seguida busca-se identificar variáveis altamente correlacionadas, que podem apresentar informações redundantes e portanto podem ser simplificadas. Para isso é preciso computar a matriz de covariância e seus autovalores e autovetores.

Novas variáveis (denominadas componentes principais) são geradas através de combinações lineares entre os autovetores e os dados originais (ambos transpostos). As componentes principais são ordenadas de tal forma que a primeira representa a maior quantidade de informação possível a se codificar com uma variável, maximizando a variância entre as variáveis originais projetadas em um determinado eixo. A quantidade de informação decai conforme nos aproximamos da componente principal n, para um conjunto original de n variáveis. Isso decorre da ordenação dos autovetores de modo que seus

autovalores (que representam o quanto aquele autovetor é relevante) fiquem em ordem decrescente. Podemos compor nosso novo espaço utilizando as duas primeiras componentes principais de modo a criar uma visualização 2D.

3.4.2 T-SNE (*t-Distributed Stochastic Neighbor Embedding*)

Enquanto o PCA busca manter pontos distantes na geometria original distantes na geometria gerada, preocupando-se com relações globais, o método t-SNE possui uma abordagem local e não linear, preservando agrupamentos mas não necessariamente a relação entre os demais agrupamentos no espaço. O método converte as distâncias euclidianas entre cada par para uma medida probabilística de similaridade baseada na projeção da distância em uma distribuição normal. Pontos atrelados aos dados originais são gerados em uma dimensão menor, com posicionamento aleatório. A cada passo o algoritmo busca aproximar as distâncias entre os pontos gerados para as distâncias medidas no espaço original ao minimizar uma função de custo através do gradiente descendente. Para prevenir alta densidade de pontos em uma única região na visualização em menor dimensão é utilizada a distribuição t de Student no cálculo das similaridades para os pontos no espaço desejado.

3.5 Rede YOLO v3

A rede YOLO v3 (You Only Look Once) pode ser usada para a extração de características semânticas e fornece bounding box de objetos detectados, prevendo por regressão logística a probabilidade de existir um objeto no bounding box, em oposição a algum outro elemento como grama ou água. Os labels atribuídos ao bounding box são multiclasses, o que melhora a classificação no caso de domínios com classes com interseções (por exemplo uma base que contenha as classes pessoa e mulher) [Redmon et al., 2015]. O vetor resultado é composto de uma série de bounding box, suas coordenadas e a previsão da probabilidade da caixa pertencer a cada uma das classes.

Utilizamos a YOLO treinada no dataset Common Objects in Context (COCO) que contém imagens de cenas complexas do dia-a-dia, com um ou mais objetos em seus contextos naturais, sem se fixar a uma vista e posicionamento ditos icônicos ([Lin et al., 2014]). A base possui 91 tipos de objetos, somando 2,5 milhões de instâncias anotadas dentro de 328 mil imagens. Cada imagem possui atreladas a ela as categorias de objetos presentes, suas instâncias segmentadas e um conjunto de 5 legendas textuais.

3.6 Avaliação

As representações geradas e suas combinações são avaliadas medindo a eficiência em tarefas de reconhecimento, em particular em problemas de classificação e busca visual. As métricas de avaliação de cada tarefa podem ser usadas como maneira de medir a qualidade das representações geradas.

Para classificação, a escolha foi o método Support Vector Machines (SVM) [Cortes and Vapnik, 1995], que possui garantias de aprendizado superiores quando comparado a outros algoritmos de classificação. O SVM aprende um hiperplano linear de separação das classes do problema, o que permite avaliar o espaço de características em termos da sua capacidade discriminativa [Mello and Ponti, 2018]. Assim, a acurácia computada a partir de um classificador SVM fornece uma medida de separabilidade linear do problema em questão, sendo assim um meio de avaliar e comparar os espaços de características.

4 Desenvolvimento e Resultados

4.1 Experimento 1: Estudo do espaço latente

Nesta etapa do projeto buscou-se trabalhar com características visuais e semânticas. O método investigado para a obtenção de características visuais foi a extração de espaços de características por meio do uso de redes neurais do tipo CNN pré-treinadas [Sharif Razavian et al., 2014] ou com ajuste fino quando a base de dados tiver exemplos rotulados disponíveis. Para a extração de características semânticas foram utilizadas as anotações existentes na base de dados (na forma de oráculo).

Foram testadas 2 arquiteturas de redes: a primeira possui arquitetura com menos camadas e baseada na arquitetura VGGNet (ver Figura 5). A segunda rede tem uma arquitetura mais profunda e com mais estratégias de treinamento e regularização como *batch normalization* e *dropout*, e foi treinada utilizando uma quantidade maior de dados através de *data augmentation* e por um número maior de épocas (ver Figura 6).

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 32, 32, 3)	0
conv2d_1 (Conv2D)	(None, 28, 28, 16)	1216
conv2d_2 (Conv2D)	(None, 24, 24, 16)	6416
conv2d_3 (Conv2D)	(None, 22, 22, 32)	4640
conv2d_4 (Conv2D)	(None, 20, 20, 32)	9248
conv2d_5 (Conv2D)	(None, 18, 18, 32)	9248
conv2d_6 (Conv2D)	(None, 18, 18, 64)	2112
flatten_1 (Flatten)	(None, 20736)	0
dense_1 (Dense)	(None, 128)	2654336
dense_2 (Dense)	(None, 128)	16512
dense_3 (Dense)	(None, 10)	1290

Total params: 2,705,018
Trainable params: 2,705,018
Non-trainable params: 0

Figura 5: Sumário da rede 1

Como ponto de partida, essas redes neurais foram utilizadas para estudar métodos de visualização e o comportamento dos exemplos apresentados à rede através de suas camadas intermediárias. Para facilitar a distinção visual entre as classes nos gráficos gerados, os casos de teste foram restringidos a quatro classes. As redes foram treinadas e testadas com a mesma base de dados (CIFAR-10).

Utilizando os métodos PCA e t-SNE foram geradas visualizações para cada camada da rede, possibilitando a observação da convergência dos dados para a classe prevista

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 32, 32)	896
activation_1 (Activation)	(None, 32, 32, 32)	0
batch_normalization_1 (Batch	(None, 32, 32, 32)	128
conv2d_2 (Conv2D)	(None, 32, 32, 32)	9248
activation_2 (Activation)	(None, 32, 32, 32)	0
batch_normalization_2 (Batch	(None, 32, 32, 32)	128
max_pooling2d_1 (MaxPooling2	(None, 16, 16, 32)	0
dropout_1 (Dropout)	(None, 16, 16, 32)	0
conv2d_3 (Conv2D)	(None, 16, 16, 64)	18496
activation_3 (Activation)	(None, 16, 16, 64)	0
batch_normalization_3 (Batch	(None, 16, 16, 64)	256
conv2d_4 (Conv2D)	(None, 16, 16, 64)	36928
activation_4 (Activation)	(None, 16, 16, 64)	0
batch_normalization_4 (Batch	(None, 16, 16, 64)	256
max_pooling2d_2 (MaxPooling2	(None, 8, 8, 64)	0
dropout_2 (Dropout)	(None, 8, 8, 64)	0
conv2d_5 (Conv2D)	(None, 8, 8, 128)	73856
activation_5 (Activation)	(None, 8, 8, 128)	0
batch_normalization_5 (Batch	(None, 8, 8, 128)	512
conv2d_6 (Conv2D)	(None, 8, 8, 128)	147584
activation_6 (Activation)	(None, 8, 8, 128)	0
batch_normalization_6 (Batch	(None, 8, 8, 128)	512
max_pooling2d_3 (MaxPooling2	(None, 4, 4, 128)	0
dropout_3 (Dropout)	(None, 4, 4, 128)	0
flatten_1 (Flatten)	(None, 2048)	0
dense_1 (Dense)	(None, 10)	20490

Total params: 309,290
Trainable params: 308,394
Non-trainable params: 896

Figura 6: Sumário da rede 2

na última camada. No caso específico deste projeto não foi necessário o passo de normalização do vetor de características extraído das camadas intermediárias da rede em preparação para a aplicação do PCA pois ele já apresentava valores dentro da mesma faixa.

O método t-SNE é custoso computacionalmente. Para a realização das análises cada método foi testado individualmente e em uma combinação. O PCA foi utilizado primeiramente para trazer os vetores de características para dimensões mais gerenciáveis pelo t-SNE, que foi responsável pela segunda fase de redução de dimensionalidade. O processo diminuiu o tempo necessário para gerar a visualização do espaço 2-D.

Os testes foram realizados utilizando ambas as redes, que apresentaram resultados similares. Optamos por visualizar as saídas da rede mais simples utilizando PCA na Figura 7 e t-SNE na Figura 8. O t-SNE apresentou uma maior capacidade de agrupar os exemplos teste de acordo com suas classes, com menos sobreposição. De fato, as representações das redes neurais permitem maior separabilidade das classes quanto mais próxima a camada está da saída da rede neural. O resultado da combinação dos métodos (Figura 9) foi muito semelhante ao t-SNE individualmente, porém foi executado com maior velocidade.

4.1.1 Comportamento de imagens artificiais

Em uma etapa de transição para o segundo experimento utilizamos uma rede VGG-16 pré-treinada na base de dados ImageNet para gerar previsões para as 1000 classes pertencentes à essa base de dados.

Simulamos um ataque à rede neural, oferecendo os exemplos aleatórios e verificando quais classes eram atribuídas às imagens geradas. As features obtidas em diferentes ca-

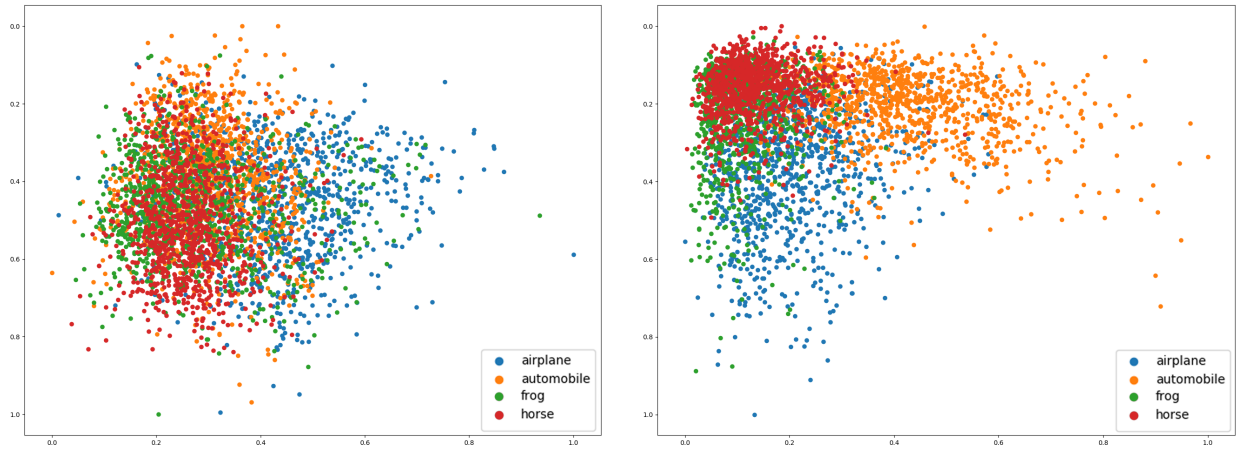

Figura 7: Utilizando a Rede 1 e o método PCA (a) 1ª camada convolucional (b) Última camada densa

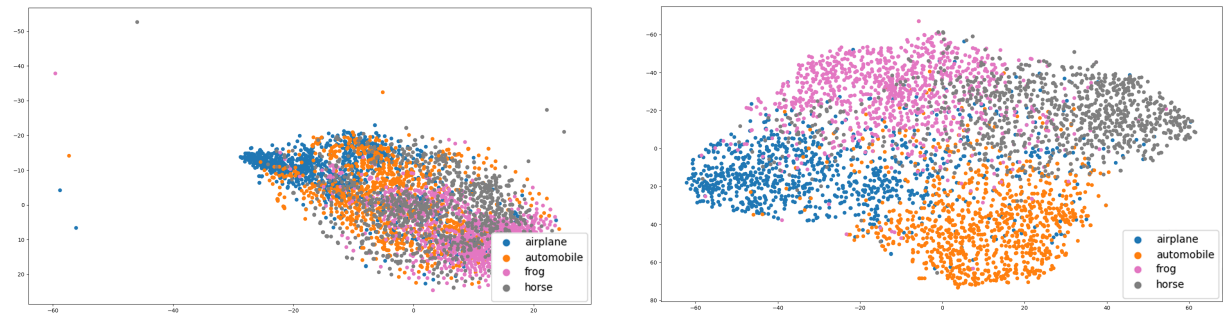

Figura 8: Utilizando a Rede 1 e o método t-SNE (a) 1ª camada convolucional (b) Última camada densa

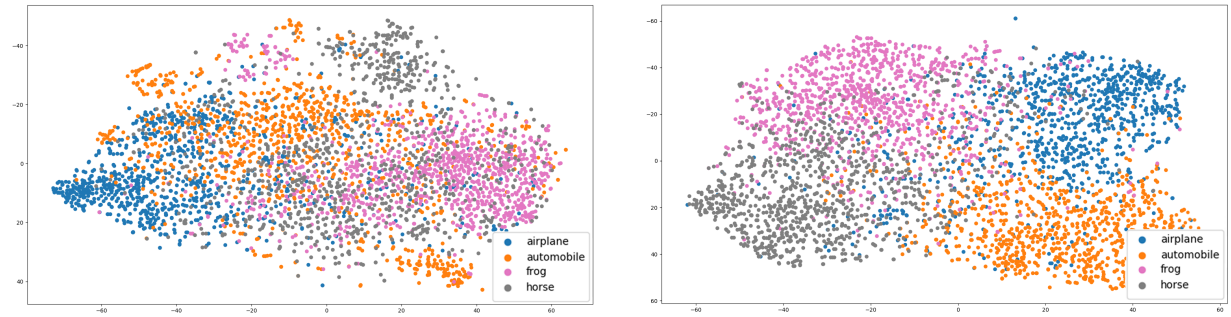

Figura 9: Utilizando a Rede 1 e o método PCA para reduzir a dimensionalidade e em seguida o t-SNE para gerar a visualização final (a) 1ª camada convolucional (b) Última camada densa

madas, intermediária na Figura 10, e final na Figura 11, foram visualizadas para entender o comportamento das representações.

A fim de comparação, apenas resultados de imagens pertencentes a classes obtidas por exemplos com mais de 45% de confiança em suas previsões foram considerados. Aplicamos

as mesmas etapas de redução de dimensionalidade e visualização acima nos exemplos de teste da base ImageNet e os exemplos gerados com formas aleatórias. A atribuição de cores na visualização gerada foi realizada de acordo com a classe ImageNet prevista. Os pontos representados por "•"são do tipo exemplo aleatório e os identificados pelo símbolo "X"são do tipo real.

É possível notar na representação do meio da rede, que há uma separação entre imagens que pertencem ao conjunto de exemplos reais (independente da classe) e imagens geradas aleatoriamente, conforme a Figura 10. Os exemplos aleatórios possuem dispersão maior no espaço e somente nos últimos estágios da rede se aproximam dos seus respectivos representantes entre os exemplos reais. Mesmo assim, ainda estão distribuídos em uma área de raio consideravelmente maior que os exemplos reais. Isso pode ser observado a partir do exemplos aleatórios classificados como "macaw", os quais estão espalhados por todo o espaço projetado, levantando questionamentos acerca do motivo de suas classificações com grau de confiança aceitável. Na etapa seguinte realizamos uma investigação para descobrir se trata-se da rede de fato reconhecer um objeto em específico ou se os resultados de exemplos retratados pelo gráfico são puramente aleatórios.

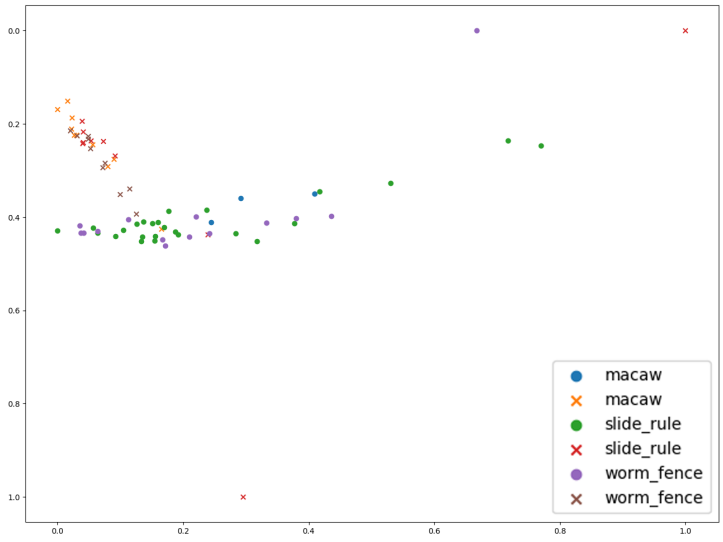

Figura 10: Exemplos aleatórios (o) e reais (x) em um bloco intermediário de camadas convolucionais da VGG-16

4.2 Experimento 2: Detecção de Exemplos Artificiais

Utilizamos um classificador do tipo SVM (Support Vector Machines) linear para investigar se o espaço de características obtido na classificação de imagens aleatórias é compatível com o da classificação de imagens reais.

Treinamos dois classificadores SVM, um utilizando a penúltima camada da VGG-16 (treinada na ImageNet) em previsões de imagens reais pertencentes ao conjunto de teste

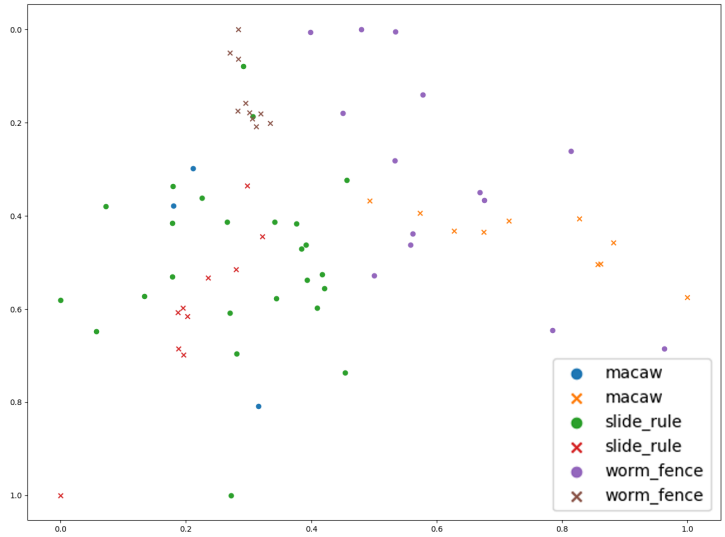

Figura 11: Exemplos aleatórios (o) e reais (x) no último bloco fully connected da VGG-16

da base ImageNet (denominado R) e outro, também utilizando penúltima camada da rede, porém executando previsões de imagens aleatórias (denominado A).

A seguir, o classificador treinado para classificar imagens do tipo oposto (o modelo R utilizado para classificar imagens aleatórias, e vice-versa). No caso da existência da proximidade entre os dois espaços de características, seria esperado que o classificador R corretamente atribuísse as classes de imagens aleatórias (utilizamos como parâmetro de comparação as classes previstas pela VGG-16). Analogamente para o classificador A.

O experimento foi realizado utilizando 50 exemplos (tanto reais quanto aleatórios) para as 10 classes mais frequentes entre aquelas previstas para imagens aleatórias e previstas com probabilidade de pertencer à classe superior a 0,1. Observamos que, levando em conta que para as 1000 classes da Imagenet o limiar de previsão aleatória é 0,001. Portanto o limiar utilizado indica probabilidade 100 vezes maior do que a aleatória.

Os resultados de acurácia obtidos foram: 0,294 para o classificador R, isto é, 29,4% das vezes as imagens aleatórias utilizadas no teste obtiveram a mesma classe mais provável prevista pela VGG-16. Para o classificador A, a acurácia foi de 0,01, indicando que as classes das imagens reais testadas coincidiram com a maior previsão de classe em apenas 1% das imagens apresentadas ao classificador.

Em ambos os casos as classes previstas apresentaram poucas variações, quase sempre prevendo a mesma classe X para o classificador R e a mesma classe Y para o classificador A. Isto evidencia como o espaço de características não tem proximidade, apesar da alta acurácia prevista pela rede. Assim, como a rede neural possui um classificador (codificado na última camada) com alta capacidade em termos do espaço de funções admissíveis, o resultado de probabilidade atribuído a um exemplo artificial não corresponde ao conceito da classe aprendida, e pode levar a interpretações incorretas. Nosso resultado oferece

uma nova forma de detectar exemplos artificiais (possivelmente adversariais) oferecidos para a rede, por meio da análise do espaço de características na camada anterior à da predição.

4.3 Experimento 3: Fusão de atributos

Para analisar se a fusão de características visuais, semânticas e espaciais melhora a performance em tarefas de classificação realizamos uma comparação entre os resultados obtidos pelas seguintes redes e classificadores na tarefa de prever ações anotadas na base VOC:

1. CNN pré-treinada com a Imagenet, avaliada com SVM linear treinado com o espaço latente (penúltima camada) de exemplos da base VOC através da CNN.

2. Rede YOLOv3 pré-treinada na MS COCO, avaliada com SVM linear treinado com características extraídas pela rede em cima de exemplos da base VOC.

3. CNN pré-treinada com a Imagenet, rede YOLOv3 pré-treinada na MS COCO, avaliadas com SVM linear treinado com concatenação direta do espaço latente da primeira rede (compactado para 100 dimensões com PCA) e das características extraídas pela segunda rede.

Como forma de acelerar a execução dos experimentos e suas variações, já que carregar em memória mais de 6 mil imagens coloridas de tamanhos variados é uma tarefa demorada e que exige uma quantidade considerável de memória, a geração de vetores de características em espaços latentes foi separada da análise de suas performances. Os vetores obtidos foram divididos em conjuntos de treinameto e teste na proporção 2:1.

No caso da CNN pré-treinada optou-se pela Resnet50 devido a sua penúltima camada mais compacta em relação às demais opções (tamanho 2048). Após a predição dos exemplos da VOC a classe de maior probabilidade foi salva, assim como os vetores latentes gerados. O resultado foi compactado utilizando PCA para 100 dimensões por cada exemplo.

Para a YOLO, as previsões de bounding box de objetos foram consideradas apenas se atingissem o valor mínimo de 50% de confiança e estivessem entre os 10 objetos detectados com maior confiança. O vetor gerado para cada imagem contêm as coordenadas x e y máximas e mínimas e o número da classe prevista para cada objeto detectado acima do limite. Caso a imagem não atinja 10 bounding boxes que atendem as condições, o restante do espaço é preenchido com cópias do vetor [0, 0, 0, 0, 80], representando coordenadas vazias e uma classe não existente. Deste modo, conseguimos garantir a padronização de

tamanho dos vetores semânticos. O tamanho total da representação concatenada é de 150 componentes, uma compactação significativa.

Existem diversos métodos para realizar a fusão de características, por exemplo a utilização de autoencoder [Cavallari et al., 2018]. Neste trabalho foi explorada a fusão por concatenação simples, ou seja, o vetor de características no espaço latente foi extendido e as informações semânticas de detecção de objetos e textuais fornecidas pelas anotações de classe do objeto (codificadas numericamente) foram colocadas ao lado dos dados já existentes. O conjunto das três informações passou a ser considerado um único vetor para propósitos de treinamento e teste. Um cuidado necessário foi de realizar operações de normalização para que os valores de um tipo de característica não predominasse em relação a outra.

Na Tabela 2 podemos observar os resultados obtidos ao executar o classificador SVM treinado nas condições descritas acima e utilizando os dados processados e particionados conforme descrito anteriormente. Os experimentos foram realizados 10 vezes e o resultado registrado na tabela é a média aritmética dos valores obtidos. É possível notar que há uma melhoria na classificação ao utilizar a combinação das representações: visuais, semânticas e textuais, quando comparado ao uso individual de cada representação. Salienta-se ainda que o conjunto final de vetores gerado é mais compacto do que o obtido através de uma utilização padrão de uma CNN pré-treinada.

Tabela 1: Métricas obtidas nos experimentos 1 a 3

	Train Score	Test Score	Accuracy	Precision	Recall
1 - ResNet50	0.8038	0.5216	0.5216	0.5460	0.5103
2 - YOLOv3	0.3954	0.3854	0.3854	0.0147	0.0287
3 - ResNet50 + YOLOv3	0.8534	0.5634	0.5634	0.5659	0.5340

Tabela 2: Desvio padrão para métricas obtidas

	Train Score σ	Test Score σ	Accuracy σ	Precision σ	Recall σ
1 - ResNet50	0.0034	0.0125	0.0125	0.0265	0.0159
2 - YOLOv3	0.0073	0.0152	0.0152	0.0006	0.0005
3 - ResNet50 + YOLOv3	0.0096	0.0168	0.0168	0.0170	0.0165

5 Conclusões

Este projeto apresentou duas investigações principais para a área de extração e aprendizado de características: a primeira no estudo das representações ao longo das redes CNNs e da sua análise com relação a exemplos reais e gerados artificialmente. A segunda, expande a primeira ao enriquecer a representação com características semânticas e espaciais adicionais às visuais, mostrando que há ganho de acurácia nessas características adicionais mesmo utilizando uma representação compacta.

O experimento utilizando imagens reais e imagens geradas artificialmente demostrou a importância da análise do espaço latente, observando camadas anteriores a preditiva como uma ferramenta extra para validar ou não os resultados apresentados pela rede. Mostrou-se ainda um promissor ponto de partida para um método de detecção de exemplos adversariais, onde uma rede neural deve ser capaz de detectar e separar exemplos falsos, gerados por uma outra rede pra se aproximar de integrantes plausíveis de um dataset.

No caso da fusão de características visuais, semânticas e espaciais um método simples como a concatenação foi capaz de apresentar melhoras nos resultados. Futuros experimentos podem caminhar na direção de explorar novos métodos de fusão, extração e seleção das características semânticas e textuais para aprimorar classificação e outras tarefas, tais como a busca visual. Dando continuidade a essa linha, o método pode ser expandido com características adicionais semânticas utilizando codificação do tipo Word2Vec em descritores e anotações textuais. Outros métodos de fusão ainda podem ser explorados, tais como *late fusion* com o auxílio de autoencoders.

Referências

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Bui, T., Ribeiro, L., Ponti, M., and Collomosse, J. Sketching out the details: Sketch-based image retrieval using convolutional neural networks with multi-stage regression. *Computers & Graphics*, 71:77–87, 2018.

Cavallari, G. B., Ribeiro, L. S., and Ponti, M. A. Unsupervised representation learning using convolutional and stacked auto-encoders: a domain and cross-domain feature space analysis. In *SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI 2018)*, 2018.

Cortes, C. and Vapnik, V. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

de Nazare, T. S., da Costa, G. d. B. P., de Mello, R. F., and Ponti, M. A. Color quantization in transfer learning and noisy scenarios: an empirical analysis using convolutional networks. In *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 377–383. IEEE, 2018.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

dos Santos, F. P. and Ponti, M. A. Alignment of local and global features from multiple layers of convolutional neural network for image classification. In *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 241–248. IEEE, 2019.

Fei-Fei, L. and Perona, P. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531. IEEE, 2005.

Fu, Y., Hospedales, T. M., Xiang, T., Fu, Z., and Gong, S. Transductive multi-view embedding for zero-shot recognition and annotation. In *European Conference on Computer Vision*, pages 584–599. Springer, 2014.

Gonzalez, R. C. and Woods, R. E. *Digital Image Processing*. Pearson/Prentice Hall, 2002.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Haralick, R. M., Shanmugam, K., et al. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL `http://arxiv.org/abs/1405.0312`.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Mello, R. F. and Ponti, M. A. *Machine Learning: A Practical Approach on the Statistical Learning Theory*. Springer, 2018.