Bruno Tinen

Classificação estética de fotografias por aprendizagem de máquina

São Paulo
2020

Bruno Tinen

Classificação estética de fotografias por aprendizagem de máquina

Versão Corrigida

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Ciências.

Área de concentração:
Engenharia de Controle e Automação Mecânica

Orientador:
Prof. Dr. Jun Okamoto Jr.

São Paulo
2020

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catalogação-na-publicação

Tinen, Bruno
Classificação estética de fotografias por aprendizagem de máquina / B. Tinen -- versão corr. -- São Paulo, 2020.
83 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.aprendizado computacional 2.redes neurais 3.reconhecimento de padrões 4.processamento de imagens 5.estética(arte) I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II.t.

Nome: TINEN, Bruno

Título: Classificação estética de fotografias por aprendizagem de máquina

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Ciências.

Aprovado em:

Banca Examinadora

Prof. Dr. _____

Instituição: _____

Julgamento: _____

Prof. Dr. _____

Instituição: _____

Julgamento: _____

Prof. Dr. _____

Instituição: _____

Julgamento: _____

À minha companheira de vida Mariana Moschione Castro e à minha querida mãe.
Foram vocês que tornaram este trabalho possível.

AGRADECIMENTOS

Durante toda a trajetória neste trabalho de pesquisa tive ajuda de diversas pessoas que me ajudaram não só tecnicamente mas também me deram todo o apoio moral que precisei - em especial minha mãe e minha namorada.

Agradeço ao professor Jun Okamoto Jr. por todo o tempo dedicado à este trabalho. Por me guiar na escolha das soluções, aprender comigo sobre todo o problema e me ajudar nos momentos de dificuldade.

Agradeço também ao professor Marcos Pereira-Barretto por me ajudar em todas as etapas iniciais da pós-graduação e, mesmo que tenha sido abandonada, a estudar um primeiro tema ao qual poderia ter dado continuidade neste mestrado.

Ao meu amigo João Ruivo por me ajudar do início ao fim deste projeto sempre que precisei de uma pessoa para discutir sobre todos os tipos de problema.

E à todos aqueles que me apoiaram direta ou indiretamente durante estes anos.

RESUMO

TINEN, Bruno. Classificação estética de fotografias por aprendizagem de máquina. 2020. Dissertação (Mestrado em Engenharia de Controle e Automação Mecânica) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2020.

A classificação estética de fotografias é um problema de separação de imagens como boas ou ruins esteticamente. Na fotografia, a qualidade e beleza de uma foto podem ser descritas segundo uma série de fatores, não só de cunho técnico, mas também de cunho emocional. Por se tratar de um processo trabalhoso mas que é baseado em regras estruturadas, pode-se considerar a utilização de algoritmos de aprendizado de máquina para automatizar a resolução do problema. Esses algoritmos possibilitam a avaliação da qualidade estética de uma fotografia através da extração e classificação de atributos. Para resolver este problema um sistema proposto consistindo numa rede neural profunda convolucional (DCNN) e uma máquina de vetores de suporte (SVM), numa abordagem denominada aprendizagem profunda usando L2-SVM (DLSVM) foi utilizado. A avaliação de diferentes arquiteturas de redes neurais e de camadas de saída foi feita com o intuito de encontrar a arquitetura de melhor desempenho e de avaliar se o uso de uma DLSVM leva a resultados superiores. As redes foram pré-treinadas usando o ImageNet Large Scale Visual Recognition Challenge no desafio de classificação de objetos em imagens para melhorar ainda mais o desempenho final do modelo, usando duas abordagens de transferência de conhecimento diferentes. Os resultados finais obtidos se comparam com os da literatura recente e as análises feitas formam uma base sobre a qual estudos futuros poderão ser feitos.

Palavras-chave: aprendizado computacional; redes neurais; reconhecimento de padrões; processamento de imagens; estética(arte).

ABSTRACT

TINEN, Bruno. Aesthetic classification of photographies using machine learning. 2020. Dissertação (Mestrado em Engenharia de Controle e Automação Mecânica) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2020.

The aesthetic classification of photographies is a problem of separating images as aesthetically good or bad. In photography the quality and beauty of a photograph can be described under a series of factors, not only technical ones but also emotional ones. As this is a time consuming process that is based on structured rules, it is possible to consider machine learning algorithms to automatize the problem resolution. These algorithms enable the aesthetic quality evaluation of a photograph through the extraction and classification of attributes. To solve this problem a system is proposed consisting of a deep convolutional neural network (DCNN) and a support vector machine (SVM), in an approach named deep learning using a L2-SVM (DLSVM) was used. The evaluation of different DCNN architectures and final layers was done with the objective of finding the architecture with the best performance and evaluate if the use of a DLSVM leads to supperior results. The networks were pre-trained using the ImageNet Large Scale Visual Recognition Challenge in the object classification challenge to improve even more the final performance of the model, using two different approaches to transfer learning. The final results are comparable to state-of-art ones and form a basis over which more studies can be done.

Keywords: machine learning; neural networks; pattern recognition; image processing; aesthetics(art)

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AVA	Aesthetic Visual Analysis
AWS	Amazon Web Services
DCNN	Deep Convolutional Neural Network
GCP	Google Cloud Platform
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
MCC	Matthews correlation coefficient
NAS	Neural Architechture Search
SVM	Support Vector Machine
VGG	Visual Geometry Group
DLSVM	Aprendizagem profunda usando L2-SVM
RNN	Rede Neural Recorrente

Sumário

1. Introdução

Na percepção visual da forma o equilíbrio, a harmonia e a clareza compõem o que o ser humano entende como estética e são considerados indispensáveis na formação de imagens, seja numa fotografia, pintura ou escultura [1]. Na fotografia, a qualidade e beleza de uma foto podem ser descritas segundo uma série de fatores, não só de cunho técnico, como a iluminação e composição da imagem, mas também de cunho emocional, tal como o momento histórico de uma foto e a expressão capturada nos rostos da pessoas. Assim, a classificação estética de uma fotografia não se resume somente avaliação de métricas, existe um componente importante que não é mensurável ou quantificável. Através de técnicas de aprendizado de máquina é possível classificar fotografias de maneira a levar em consideração tais fatores subjetivos ao treinar uma rede neural com exemplos de fotos classificadas em boas ou ruins por especialistas cuja análise foi além de simples métricas.

A Figura 1.1 apresenta duas fotografias que ficaram conhecidas pelo momento histórico e pela carga emocional que carregam consigo. A Figura 1.1a, de 1972 do fotógrafo Nick Ut [2], foi tirada logo após um bombardeio de Napalm durante a guerra do Vietnã e a expressão de desespero das crianças correndo com soldados logo atrás é muito impressionante. A Figura 1.1b, pelo fotógrafo Charlie Cole de 1989 [3], mostra um jovem parando uma fila de tanques na praça Tianmen, em Beijing. Ambas, apesar de terem sido tiradas há décadas, continuam sendo um retrato da situação vivida e fazem parte da coleção de fotos historicamente relevantes da história recente.

A solução manual do problema de classificação estética é através de uma avaliação de diversos aspectos da fotografia usando conhecimento especialista. Tomando como exemplo a Figura 1.2b, pode-se dizer que a fotografia vencedora do concurso da National Geographic de 2018, Mermaid, do fotógrafo Reiko Takahashi [4], tem uma paleta de cores bem definidas em tons de azul, a iluminação dá um foco maior ao objetivo da fotografia, o qual é a representação do animal marinho, a nitidez é boa e, analisando do ponto de vista da regra dos terços, o conteúdo é distribuído nos quadrantes centrais e inferiores,

Figura 1.1: Exemplos de fotografias conhecidas pelo seu apelo histórico e emocional.

(a) Fotografia das crianças após um ataque de bomba napalm.

Fonte: Nick Ut (1972)[2]

(b) Fotografia de jovem parando os tanques na praça Tianmen.

Fonte: Charlie Cole (1969) [3]

Figura 1.2: Exemplo de descritores concretos em fotografias.

(a) Fotografia sem métrica ou balanço de cores.

Fonte: Próprio autor.

(b) Fotografia com métricas e cores adequadas, vencedora do concurso da National Geographic de 2018.

Fonte: Reiko Takahashi (2018) [4]

com uma linha vertical bem definida, levando à um equilíbrio da imagem, enquanto que na fotografia de acervo próprio (Figura 1.2a) a nitidez não é boa, a iluminação não tem foco e apenas deixa a foto desequilibrada, não existe nenhum objetivo para a imagem e todo o conteúdo é mal distribuído. Usando essa análise, a fotografia vencedora de concurso é esteticamente boa enquanto que a de acervo próprio pode ser classificada como esteticamente ruim.

A classificação estética de fotografias pode ser vista como um problema de classificação binária intrínseca em imagens esteticamente boas e ruins [5]. Os fatores técnicos de uma imagem podem ser definidos e extraídos através de um conjunto de regras, similares àquelas utilizados por profissionais de fotografias. Não existe, entretanto, a mesma possibilidade de abordagem com relação aos fatores emocionais de uma imagem.

Apesar de ser possível definir o problema como uma classificação binária, existe o desafio de definir o que cada uma das classes do problema abrange. Estética pode ser definida de diferentes formas, por diferentes pessoas, não existindo um consenso acerca do significado de uma fotografia esteticamente atraente. Apesar desta ambiguidade, existem algumas propriedades visuais que fazem, de forma geral, uma imagem ser esteticamente mais atraente [6].

Por se tratar de um processo trabalhoso mas que é baseado em regras estruturadas pode-se considerar a utilização algoritmos de aprendizado de máquina para automatizar a resolução do problema. Esses algoritmos possibilitam a avaliação da qualidade estética de uma fotografia através da extração e classificação de atributos que englobam tanto os fatores emocionais quanto técnicos, num problema denominado classificação estética de fotografias por aprendizado de máquina.

Com o maior acesso das pessoas à formas de se tirar fotografias através de smartphones houve um grande aumento do número de fotografias disponíveis em acervos públicos e privados e uma característica importante de programas de gerenciamento de fotografia é conseguir recomendar as melhores fotos dentro de uma quantidade cada vez maior de imagens [7].

Para a classificação automática estética de fotografias, como em outros problemas de classificação por aprendizado de máquina, é necessária a construção de modelos computacionais que generalizem elementos que caracterizem fotos esteticamente boas e ruins. Estes modelos podem ser construídos a partir de descritores estéticos, os quais são distribuições de padrões de pixels que representam um fator estético – concreto ou genérico – em uma fotografia. Métodos de aprendizado de máquina são alimentados com estes descritores estéticos e modelos genéricos de classificação são gerados. Pode-se então utilizar

este modelo para a classificação de qualquer fotografia.

Os descritores estéticos concretos envolvem, em geral, regras de fotografia e descritores visuais [5]. Entretanto ainda é possível melhorar a qualidade destes classificadores para que possam replicar de maneira mais fiel o comportamento humano, incluindo fatores além dos puramente técnicos, quando descritores genéricos, gerados sem pressupostos de regras formais, são utilizados.

Aplicações que envolvem a classificação estética automática englobam a preservação de multimídia, a sumarização de coleções e a navegação por grandes arquivos de fotografias [8], ajudando os usuários tanto em tarefas de seleção quanto de edição de fotografias [5].

Este trabalho tem como objetivo fazer a classificação estética de fotografias por aprendizado de máquina. Para isto foram buscados descritores estéticos adequados e algoritmos de classificação que separem adequadamente as fotos em esteticamente agradáveis e ruins baseados nestes descritores que estão sendo testados. Por se tratar de uma classificação subjetiva, o problema está sendo abordado utilizando-se descritores genéricos com o objetivo de simular o processo cognitivo humano a respeito da avaliação de fotos. O sistema proposto consiste numa rede neural profunda convolucional (DCNN) e uma máquina de vetores de suporte (SVM), numa abordagem inspirada no artigo do Tang denominada aprendizagem profunda usando L2-SVM (DLSVM) [9]. Diversas arquiteturas diferentes de DCNN foram testadas para determinar aquela que leva aos melhores resultados. Além disso, tomando como base o problema de classificação de objetos, serão avaliadas duas abordagens distintas de transferência de conhecimento baseado em redes previamente treinadas com o banco de dados do ImageNet Large Scale Visual Recognition Challenge.

O melhor resultado obtido neste trabalho é com uma rede que possui uma taxa de acertos de 86,06% e é baseada na arquitetura Inception-ResNet-V2. Este resultado é superior aquele observado em trabalhos na área de classificação estética nos anos recentes. Os estudos feitos neste trabalho servem de base para que trabalhos futuros possam melhorar ainda mais o desempenho deste modelo.

2. Classificação Estética de Fotografias

O julgamento estético computacional de fotografias é um problema de classificação intrinsecamente binário no qual se busca encontrar as fotografias que são esteticamente atraentes. A classificação pode ser estendida para um problema multiclasse caso se queira obter uma avaliação do quão atraente uma fotografia é. Diferentemente da determinação da qualidade da imagem, que lida com características que representam ruídos de diferentes naturezas - ruídos gaussianos e erros de compressão por exemplo [10] – a classificação estética lida com características relacionadas à compreensão estética da imagem – como iluminação, cores, composição, etc.

Apesar de ser possível definir o problema como uma classificação binária, existe o desafio de definir o que cada uma das classes do problema abrange. Estética pode ser definida de diferentes formas, por diferentes pessoas, não existindo um consenso acerca da do significado de uma fotografia esteticamente atraente. Apesar desta ambiguidade, existem algumas propriedades visuais que fazem, de forma geral, uma imagem ser esteticamente mais atraente [6]. Encontrá-las e utilizá-las de forma a separar de forma clara quais são as fotografias esteticamente bonitas dentro de um determinado grupo é o objetivo do trabalho.

A classificação estética automática é em geral resolvida como um problema de aprendizado [11, 6, 5] seguindo a metodologia: define-se um conjunto de características que melhor descrevem a estética da fotografia, escolhe-se um banco de dados com imagens de treinamento com anotações referentes à avaliação e estética, extraem-se os descritores estéticos deste conjunto de treinamento e utiliza-se um método de aprendizado para a construção de um modelo a partir dos vetores de descritores que foram obtidos. Testes são então realizados com o modelo obtido e comparam-se os resultados com outros obtidos por outros autores.

Os trabalhos revisados se concentram em duas frentes de pesquisa bem definidas: a determinação e extração de descritores estéticos que melhor representem a estética de uma imagem e a utilização de modelos de aprendizado de máquina que melhor distinguam as fotografias esteticamente boas das ruins [12]. Além da melhoria da taxa de acertos, alguns trabalhos focam no desenvolvimento de frameworks para a melhoria estética das imagens, melhorando a composição [13] ou auxiliando na edição de retratos [14], por exemplo.

Com relação aos descritores estéticos, podem-se ter descritores concretos, que representam modelos matemáticos baseados em definições de conhecimento especialista sobre fotografia, e descritores genéricos, os quais são aprendidos por algoritmos de aprendizado de máquina a partir de um conjunto de fotografias previamente classificadas em esteticamente boas e ruins. Os modelos de aprendizado de máquina utilizados para a classificação estética podem ser divididos nos utilizados para a classificação binária e nos utilizados para a classificação multiclasse.

As seções deste capítulo descrevem as diferentes abordagens para a solução do problema, características de fotografias que podem ser utilizadas nos modelos e o métodos de aprendizado de máquina para a construção dos modelos a partir das características obtidas.

2.1 Síntese de Metodologias Adotadas na Classificação Estética de Fotografia

Para a resolução do problema de classificação estética foram vistas diferentes abordagens adotadas por autores, com respeito aos tipos de descritores utilizados e modelos de aprendizado de máquina gerados. Nos trabalhos mais recentes dois tipos de separação das imagens em esteticamente boas e ruins foi usada para o banco de dados AVA para a classificação binária. Na primeira versão, denominada AVA1, é usada a nota de corte 5, sendo usadas como esteticamente boas as com nota média acima de 5 e esteticamente ruins as com nota média inferior. A segunda versão, AVA2, usa como esteticamente boas a parcela de 10% das fotografias com melhor média de notas e as como esteticamente ruins as com 10% pior média de notas.

Gao et. al [15] propôs o uso de redes bayesianas com atributos estéticos concretos para a solução do problema. No total 17 atributos estéticos foram considerados para a construção do modelo com um resultado final de 72,7% de acurácia no banco de dados Memorability. Comparando-se com outros resultados da literatura o desempenho de redes bayesianas se mostrou inferior.

Marchesotti et. al propôs um modelo que combina atributos visuais e textuais para a resolução da classificação estética [16]. Para a parte textual foram usados os comentários de usuários das fotografias no banco de dados AVA, que contêm informações detalhadas sobre a opinião das pessoas sobre a estética das fotografias analisadas e para esta etapa do modelo foram usados histogramas de bag-of-words para representar o vocabulário uma rede elástica de vetores de suporte regressores para se encontrar os atributos visuais das imagens. Esses atributos foram então usados para treinar um modelo baseado em vetores de Fischer treinados com o gradiente estocástico descendente com uma perda logística. Neste trabalho usou-se a área sob a curva (AUC) para comunicar os resultados obtidos pelo modelo, com um resultado de 0,718 no melhor caso.

Zhang et. al [17] propôs um modelo baseado em graphlets para modelar os elementos dentro de uma fotografia e formar os atributos estéticos que alimentam uma SVM multiclasse, obtendo uma acurácia de 83,24% no banco de dados AVA. Como a classificação avaliada para este trabalho foi multiclasse, para as imagens do AVA dataset foram usadas as notas médias discretizadas num intervalo de $0,1$ a 1 com passo de $0,1$.

Mavridaki e Mezaris [8] usaram de atributos estéticos concretos, sem o uso de aprendizado profundo para geração dos atributos, com um classificador SVM binário. A acurácia do modelo no melhor caso foi de 77,08% usando o AVA2. Comparando com os resultados obtidos com descritores genéricos a abordagem que usa os descritores concretos teve um desempenho inferior, principalmente por terem uma capacidade de representação limitada.

Os demais trabalhos recentes de classificação estética estudados usam aprendizado profundo como forma de resolução do problema.

Dong et. al [18] usou a arquitetura AlexNet treinada inicialmente no ILSVRC para extração de parâmetros e então usou os atributos da camada de saída para treinar uma SVM binária numa segunda etapa. Note que nesta abordagem não houve treinamento dos parâmetros da DCNN com o banco de dados estético, apenas da SVM. Uma técnica usada neste trabalho que resultou numa melhoria de desempenho foi o uso da pirâmide espacial, uma técnica na qual além da imagem redimensionada também é usada a imagem segmentada em 5 regiões, com o objetivo de reduzir a perda de informação ao diminuir a imagem para que possa ser usada como entrada da DCNN. No melhor caso uma acurácia de 83,52% foi observada no AVA2.

Em [19] um modelo usando aprendizado profundo dependente do tipo de fotografia analisada foi proposto. Dada uma fotografia um grupo composto por imagens similares é construído e os modelos são

Tabela 2.1: Síntese de soluções para o problema de classificação estética de fotografias.

Descritor Estético	Método de Aprendizado	Trabalhos Relacionados
Concreto	Binário	Mavridaki e Mezaris (2015) [8], Datta, Joshi, Li et al. (2006) [6], Ke, Tang e Jing (2006) [11], Khan e Vogel (2012) [22], Luo e Tang (2008) [23], Dhar, Ordonez e Berg (2011) [24], Lo, Liu e Chen (2012) [25], Nishiyama, Okabe, Sato et al. (2011) [26], Wong e Low (2009) [27], Tang, Luo e Wang (2013) [28]
	multiclasse	Bhattacharya, Sukthankar e Shah (2010) [13], Zhang, Gao, Zimmermann et al. (2014) [17], Li, Loui e Chen (2010) [14], Li, Gallagher, Loui et al. (2010) [29], Jiang, Cerosaletti and Loui (2009) [30], Wu, Hu e Gao (2011) [12], Gao, Wang e Ji (2015) [15], Aydin, Smolic e Gross (2015) [31]
Genérico	Binário	Murray, Marchesotti e Perronnin (2012) [32], Marchesotti, Perronnin, Larlus et al. (2011) [5], Marchesotti e Perronnin (2013) [33], Marchesotti, Murray e Perronnin (2015) [16], Lu Lin, Sehn et al. (2015) [34], Ma, Liu e Chen (2017) [35], Talebi e Milanfar (2018) [36]
	Multiclasse	Lu, Lin Jin et al. (2014) [37], Kao, He e Huang (2016) [38], Wang, Chang, Dolcos et al. (2016) [39]

Fonte: Elaborada pelo autor.

treinados para estes grupos específicos. Os parâmetros foram extraídos com uma DCNN de 5 camadas e então foram utilizados para treinar uma SVM. No melhor caso a rede teve uma acurácia de 80,38% no AVA2.

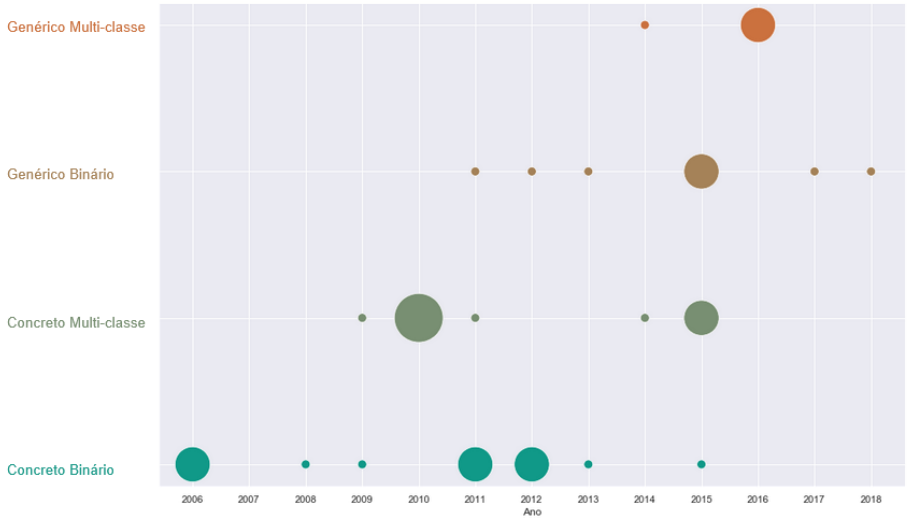
Um modelo de aprendizado de máquina profundo com múltiplas cenas (MSDLM) foi proposto em [20], que assim como em [18], usou a arquitetura AlexNet treinada no ILSVRC como base. No MSDLM as primeiras quatro camadas do AlexNet são usadas seguidas por camadas convolucionais para extração de detalhes de cena, usando os descritores "animal", "arquitetura", "humano", "paisagem", "noite", "planta" e "estático". Finalmente uma camada softmax completamente conectada foi usada. A rede foi pré-treinada usando o banco de dados estético CUHKPQ nos sete tipo diferentes de cenas. No melhor caso para o AVA dataset foi obtida uma acurácia de 84,88%, usando o AVA2.

Em 2017 a ILGNet [21], uma rede baseada no modulo Inception e no GoogLeNet cujo objetivo é ter parâmetros locais e globais da imagem para uso no classificador foi proposta, obtendo como melhor resultado acurácia de 85,53% para o treinamento com o AVA2. Como a abordagem era baseada na rede GoogLeNet, foi utilizada a rede previamente treinada no ILSVRC Large Scale Object Recognition Challenge para que então a rede fosse treinada usando o AVA, num processo de adaptação de domínio usando transferência de conhecimento.

Com o intuito de sintetizar as possíveis soluções a Tabela 2.1 mostra possíveis resoluções do problema e quais autores dos revisados as utilizaram.

A análise da Tabela 2.1 e da Figura 2.1 nos permite fazer algumas observações. A primeira é a que existe um maior volume de trabalhos relacionados com descritores estéticos concretos do que genéricos nos trabalhos mais antigos. Grande parte dos descritores genéricos foram estudados em trabalhos mais recente - com um destaque para Marchesotti, Perronnin e Murray - utilizando-se de classificadores binários e com uma concentração em classificação multiclasse genérica com a utilização de redes neurais e aprendizado

Figura 2.1: Distribuição das publicações relacionadas a classificação estética e suas abordagens ao longo dos anos.



Fonte: Elaborada pelo autor.

multitarefa.

A maior concentração de trabalhos com descritores concretos em trabalhos mais antigos se dá pelo fato dos trabalhos pioneiros na área adotarem esta metodologia e muitos autores tomarem como base para as suas linhas de pesquisas metodologias semelhantes. Foi apenas nos últimos 8 anos que trabalhos relacionados com características genéricas começaram a ser publicados. Com a demonstração de um desempenho superior dos descritores genéricos é possível que cada vez mais as linhas de pesquisa se direcionem para o seu uso em detrimento dos descritores concretos.

Além disso, com relação ao uso de classificadores multiclasse ou binários há uma igual separação do trabalho. No geral, a adoção de um ou outro classificador está relacionado apenas com o escopo que se deseja adotar ao problema.

Este trabalho está no escopo dos descritores estéticos genéricos, uma vez que serão gerados a partir de uma rede neural convolucional profunda, e por terem demonstrado cada vez mais um desempenho superior do que os descritores estéticos concretos. Já o método de aprendizado é binário, uma vez que o objetivo principal será o de classificar as fotos entre boas e ruins apenas, sem uma diferenciação mais minuciosa entre estas classes.

A Tabela 2.2 contém os resultados para trabalhos nos últimos anos referentes à classificação estética de imagens, para diferentes metodologias. Como pode-se observar, a base de dados AVA [40] é bastante utilizada; em apenas um dos trabalhos é utilizado o banco Memorability [41], pois possui dezessete rótulos estéticos, contra o único do AVA.

2.2 Descritores Estéticos

Os descritores estéticos são definidos como as propriedades que podem ser extraídas a partir de fotografias que conseguem modelar de forma aproximada um critério utilizado pela percepção humana para a classificação de imagens com relação à sua estética [5]. Estes descritores podem estar relacionados com aproximações dos princípios de fotografias [31], cor e composição [42], por exemplo. A seleção de descritores estéticos que são significativos para a determinação da qualidade de fotografias é uma etapa importante para que a classificação possa ser feita.

Tabela 2.2: Comparação da acurácia obtida por diversos autores e métodos em trabalhos recentes.

Banco de dados	Acurácia	Ano	Referência
AVA	77,08%	2015	Descritores concretos + SVM [8]
Memorability	72,7%	2015	Rede Bayesiana [15]
AVA	Área ROC=0,718	2015	Representações textuais + Vetores de Fischer [16]
AVA	83.24%	2014	Graphlets + SVM [17]
AVA	83.52%	2015	AlexNet + SVM [18]
AVA	80.38%	2015	Query Dependent Aesthetic Model [19]
AVA	84.88%	2016	MSDLM [20]
AVA	85.53%	2017	ILGNet [21]

Fonte: Elaborada pelo autor.

Pode-se dividir as características utilizadas em trabalhos anteriores em dois conjuntos: descritores concretos e descritores genéricos. Estes dois conjuntos diferem na forma como são obtidos e no escopo que representam. Os descritores concretos representam modelos matemáticos baseados em definições de conhecimento especialista sobre fotografia, e os descritores genéricos representam fatores genéricos que não possuem necessariamente uma correlação com regras de fotografia e são aprendidos por algoritmos de aprendizado de máquina a partir de um conjunto de fotografias previamente classificadas em esteticamente boas e ruins.

É importante ressaltar que é importante escolher descritores de forma a evitar a utilização de informações redundantes ou que tenham pouca relação com o problema. A presença destes descritores podem adicionar incerteza na predição da classe pelos algoritmos de aprendizado de máquina, reduzindo a eficiência do modelo [43].

2.2.1 Descritores Estéticos Concretos

Descritores estéticos concretos são modelos estatísticos de critérios que seres humanos usam para julgar fotografias. Estes critérios são definidos a partir de como os seres humanos compreendem o julgamento estético. Os descritores concretos, portanto, usam regras definidas por especialistas e não tentam aprender quais características são relevantes no problema de classificação.

Existem três fatores que tanto fotógrafos profissionais, amadores e não-fotógrafos consideram importantes para se distinguir fotografias boas de ruins: simplicidade, realismo e uso de técnicas fotográficas [11]. O maior consenso a respeito de fatores de distinção está na simplicidade; a foto deve possuir um objetivo claro e estruturado. O fator do realismo se refere a fotos que saem do comum, ou sejam, são surreais e atípicas, as quais são mais atraentes que fotos que capturam o cotidiano. Uso de técnicas fotográficas abordam o uso correto de contraste e nitidez, fatores que independem do conteúdo da foto, mas que são considerados degradantes caso sejam falhos.

A partir do conhecimento especialista acerca de fotografia é possível fazer modelagens estatísticas que aproximem as regras estabelecidas por relações entre os pixels de uma determinada imagem. A exposição à luz pode ser calculada, por exemplo, calculando-se a intensidade média dos pixels da fotografia [6].

Os descritores estéticos concretos podem ser divididos em descritores de baixo nível [6, 22], de alto nível [11, 13, 23, 24, 22, 25, 8] e semânticos. Os dois primeiros grupos são tipicamente utilizados de maneira conjunta nas analises [5].

Características de baixo nível são meios de descrever a fotografia por métricas relacionadas à percepção visual através de estatísticas simples [5]. São descritores isolados, que consideram as características de

Figura 2.2: Os pontos de intersecção dividindo a imagem em três partes horizontalmente e verticalmente são os focos para se colocar o elemento principal da fotografia. Esta fotografia, por exemplo, não tem nenhum elemento que segue esta regra.

Fonte: Elaborada pelo autor.

forma independente e sem abstrações que envolvam a composição explicita de informações.

Os descritores de baixo nível podem ser usados de forma local ou global [22]. Quando analisados de forma local, a fotografia é considerada de forma segmentada e cada um dos segmentos pode ser analisado de forma independente e as relações entre os segmentos podem ser considerados. Já descritores globais são calculados considerando a imagem como um todo, ignorando detalhes. O descritor global pode ser considerado como uma média dos descritores locais. Estes descritores podem estar relacionados com tons de cor e saturação, razões de tamanho, formatos de objetos e exposição a luz [6].

O uso de luz pode ser usado como um bom descritor estético por exemplo. Uma imagem com muita exposição ou muito escuras no geral são relacionadas a uma menor qualidade. Dada uma imagem de dimensões $X \times Y$ pode-se calcular a exposição média de luz como a intensidade média do valor do pixel no espaço HSV usando a Equação 2.1[6].

$$\frac{1}{XY} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} I_V(x, y) \tag{2.1}$$

Descritores de alto nível descrevem a distribuição de elementos e estrutura em uma fotografia [5]. Essas características são abstrações que modelam regras de simplicidade, composição e conteúdo [11, 23, 8]. De modo geral estes descritores são globais pois lidam com a percepção da combinação de elementos individuais locais. Estes fatores podem ser descritos então por características como a distribuição de bordas [11, 25], distribuição de cores [11, 23, 22, 14, 25, 26], contagem de tonalidade [11, 22, 25], uso de técnicas de borrar [11, 25] e composição (associada em geral à regra dos terços) [23, 24, 22, 14, 8].

A regra dos terços é uma aproximação da regra de ouro da matemática e diz que o elemento principal da imagem deve estar em uma das intersecções dada uma divisão em terços, horizontalmente e verticalmente - da fotografia, como pode ser observado na Figura 2.2. No geral pode ser observado que fotografias que seguem essa regra tem o objetivo da imagem se estendendo pelo centro da imagem. Assim, uma forma de modelar essa regra é calculado a média da intensidade HSV no retângulo central da imagem. A Equação 2.2 mostra o cálculo de média de tonalidade para o retângulo central de uma imagem[6].

$$\frac{9}{XY} \sum_{x=X/3}^{2X/3} \sum_{y=Y/3}^{2Y/3} I_H(x, y) \tag{2.2}$$

Pode-se utilizar também uma abordagem estrutural para a extração de características, com o uso de graphlets [17], ou o uso de regiões salientes [27]. O modelo de graphlets foi adotado com o intuito de

preservar as informações de estrutura de layout locais e globais e para descrever uma foto com informações de diferentes canais de forma colaborativa [17]. Já o método de regiões salientes parte do fato de que a principal distinção entre fotos profissionais e amadoras é a presença de um elemento principal, focando em extrair o elemento principal da fotografia, possibilitando a análise de descritores nesta região especifica [27].

Os descritores semânticos descrevem o conteúdo de uma fotografia de forma categorizada [13, 24, 32]. Estes descritores podem ser tanto entre fotografias ou dentro de uma mesma fotografia e a partir deles pode-se utilizar os descritores de alto e baixo nível para se encontrar relações entre as categorias semânticas.

Dentro de fotografias pode-se separar regiões de cenário tais como o céu e o suporte, ao fundo, e o componente principal da imagem, e aplicar-se os demais descritores estéticos específicas para cada região [13, 24]. Usando esta categorização pode-se utilizar um descritor que avalia o destaque do componente principal da foto com respeito ao fundo e a proporção entre céu e suporte por exemplo.

Entre fotografias os descritores semânticos se referem à categorias temáticas, de conceitos ou técnicas fotográficas usadas [32]. Um exemplo é o de grupos fotográficos sobre cenários e arquiteturas [28]. Estes grupos podem ser obtidos de forma automática com base não só nas fotografias mas também nos comentários de pessoas sobre as mesmas [33, 16].

Um exemplo do uso de descritores semânticos é na categoria de retratos, caracterizados pela presença de pessoas. Nesta categoria é possível utilizar descritores específicos para a busca de detalhes nas pessoas como o algoritmo Viola-Jones [24, 22] ou o Active Shape Model [14, 29], que podem buscar faces que estão presentes na fotografia para uma posterior análise. No caso do Active Shape Model [14, 29], ângulos e distâncias entre pontos da face – tais como a abertura de um sorriso – podem ser usados como características para a determinação de uma boa fotografia.

2.2.2 Descritores Estéticos Genéricos

O conjunto de descritores estéticos genéricos modelam as características de forma implícita nas fotografias. Estes descritores não possuem necessariamente uma relação direta com regras de fotografia ou com fatores visuais, tal como nos descritores concretos.

Os descritores genéricos são obtidos através de técnicas de aprendizado de máquina. A partir de um conjunto de fotografias classificadas previamente em esteticamente boas e ruins e com o uso de técnicas como a análise de componente principal ou redes neurais convolucionais, é possível encontrar os principais elementos que foram utilizados para a separação das fotos nestas categorias. Diferentemente dos descritores concretos, não há a limitação da descrição das regras por conhecimento especialista e, há a possibilidade, portanto, de modelar regras implícitas. Descritores genéricos são capazes de aprender de forma eficiente e explicita o que os descritores concretos tentam caracterizar de forma explicita [5].

Algumas técnicas que permitem a extração deste tipo de características são os Bag-Of-Visual-Words (BOVs) [5, 16], os vetores de Fischer [5, 32, 33, 16] e as redes neurais convolucionais profundas (DCNNs) [38, 35, 34, 7]. Aplicando-se os descritores genéricos localmente pode-se identificar diversas propriedades locais, tais como saturação e presença de bordas bruscas. A partir destas descrições locais pode-se obter a informação global da fotografia [5].

Na representação por Bag-Of-Visual-Words (BOVs) uma imagem é descrita por histogramas locais de características quantizadas [5]. A partir destas descrições por histogramas aprende-se um dicionário de palavras visuais a partir de um conjunto de imagens de treinamento. Cada palavra deste dicionário representa um agrupamento de histogramas. Os vetores de Fischer (VF) estendem os BOVs incluindo nas palavras informações de estatística de até segunda ordem a respeito dos histogramas (variância da distribuição por exemplo). Enquanto os BOVs tratam as imagens como distribuições discretas, os VFs as

tratam como distribuições contínuas [5].

Redes neurais convolucionais podem ser usadas para se categorizar fotos em mais de um quesito simultaneamente, utilizando como parâmetros de entrada estéticos e semânticos para a otimização da rede por exemplo. Assim, uma única rede é capaz de categorizar uma fotografia em grupos semânticos e em relação à estética. O uso de aprendizado múltiplo possibilita o compartilhamento de conhecimento, o que melhora os resultados das tarefas para as quais a rede está sendo criada [38].

Características genéricas se mostraram melhores na tarefa de descrição das fotografias, resultando em maiores taxas de acerto na classificação [5] e podem ser usadas em conjunto com características concretas [33, 16].

Uma das limitações do aprendizado de descritores estéticos por DCNNs é o fato da entrada ter tamanho fixo, sendo necessário transformar inicialmente a entrada - usando bordas e recortes por exemplo - o que pode levar a uma perda de detalhes e distorção na composição. A arquitetura Adaptative Layout-Aware Multi-Patch (A-Lamp) CNN oferece uma alternativa de solução para este problema, combinando diversas redes que se focam em pedaços menores da imagem selecionados de forma a representar as partes mais discriminantes e informativas, solução de rede também adotada na Deep Multi-Patch Aggregation Network (DMA-Net) [34], e uma solução holística responsável por representar a composição da imagem através de um grafo que contém os objetos contidos na imagem e as suas relações [35].

Outra forma de se obter descritores genéricos é através da utilização de transferência de conhecimento de DCNNs pré-treinadas para tarefas de classificação de objeto tais como a Inception v1 e v2, VGG16 e a MobileNet [36]. Uma abordagem similar é a utilizar diversas camadas de modelos pré-treinados, como o Inception, em conjunto com camadas convolucionais [7].

2.3 Tipos de classes de saída de classificação estética

A discriminação estética de fotografias é um problema de classificação binária, se forem consideradas as classes de imagens esteticamente boas e ruins, ou de classificação multiclasse, se for buscada uma quantificação da qualidade estética da fotografia.

Existem diversas razões para se optar pela classificação binária e não pela multiclasse. A classificação multiclasse exige uma maior quantidade de exemplos de treinamento, por ser mais refinada, e nem sempre estes dados estão disponíveis; nem sempre a distinção exata do valor estético é importante para a análise; e para diferentes categorias semânticas pessoas tendem a usar critérios diferentes, não havendo assim comparabilidade entre o valor dos resultados entre grupos semânticos na classificação multiclasse [30].

Um problema de se obter apenas a classificação sim/não ou a nota de uma imagem com relação à sua estética é não se obter os motivos e regras que levaram o algoritmo àquela conclusão. Utilizar métodos e bancos de dados que levem em conta outras características além de uma simples nota, tais como a coloração e os tons usados, compondo uma assinatura estética, pode ser uma solução para isto [31].

Os modelos sobre os quais os métodos de aprendizado são aplicados podem abranger características semânticas e estéticas de forma conjunta ou separada [32]. Utilizar as características semânticas de forma conjunta significa utilizá-las como uma outra propriedade concreta qualquer no momento do treinamento do algoritmo de aprendizado de máquina; utilizá-las de forma separada implica em utilizar as categorias semânticas como forma de pré-processamento dos demais descritores, selecionando os descritores mais adequados para determinadas situações semânticas de forma manual. Uma forma de se obter um modelo conjunto estético-semântico é o modelo conjunto de ranqueamento (JRM) [32]. Formas de classificação separada contam com o modelo de ranqueamento dependente e independente (IRM e DRM respectivamente) [32].

2.3.1 Classificação Binária

Na classificação binária de fotografias de maneira estética, diferentemente da classificação binária em categorias semânticas (como entre cidade e paisagem, fotografias e pinturas), não existe uma definição do que caracteriza as fotografias nestas classes, por ser uma classificação subjetiva na qual existe a possibilidade de divergência de opiniões, mesmo entre especialistas [11].

Os trabalhos de classificações binária estética de fotografias revisados utilizaram-se de técnicas probabilísticas de classificação com o uso de classificadores Naïve Bayes [11, 23], Support Vector Machines (SVMs) [6, 23, 24, 22, 25, 8, 26, 28, 27], Adaboost [23, 22] e DCNNs [35, 34, 36, 7].

Classificadores Naïve Bayes são utilizados para a combinação de descritores devido ao caráter não linear de alguns destes (como o contraste). O Naïve Bayes é derivado da aplicação do teorema de Bayes com a hipótese simplificadora de independência entre cada uma das evidências (no caso dos descritores). No caso da classificação de fotografias, entretanto, alguns dos descritores podem possuir correlações e assumir esta hipótese implica em modelar inadequadamente o relacionamento de algumas das hipóteses na realidade, o que pode impactar na taxa de acerto do classificador [11].

Classificadores por SVMs são bastante populares nestes problemas e foram utilizados por diversos autores. SVMs são um conjunto de métodos de aprendizado supervisado em que gerados classificadores binários nos quais as informações de um conjunto de treinamento são observadas a partir de espaços vetoriais cuja ordem possibilite a separação destes dados em duas classes distintas [6]. A distância entre o classificador e a informação mais próxima de uma determinada classe é chamada de margem e o conjunto de margens define os chamados vetores de suporte.

Além de classificadores lineares pode-se obter classificadores polinomiais, de tangentes hiperbólicas e de funções radiais [43]. Nos problemas de classificação estética é usado em geral um kernel linear para a SVM [8, 28, 27], pois existe a crença que a separação entre fotos esteticamente boas e ruins é linearmente separável se os descritores utilizados são suficientemente discriminativos [27]. O uso de kernels polinomiais em SVMs possibilita a avaliação da contribuição de cada descritor para a classificação, que podem ser obtidos a partir dos termos da função do kernel [24].

O uso de SVMs é justificado pelo seu melhor desempenho, tendo um desempenho marginalmente superior quando comparado aos métodos de K vizinhos próximos, florestas aleatórias, classificação por regressão e o Adaboost [22].

Caso se deseje utilizar propriedades semânticas com SVMs é possível treinar diversos classificadores, um para cada um dos grupos semânticos. Pode-se, por exemplo, criar classificadores para grupos que representam animais, arquiteturas, seres humanos, cenas noturnas, plantas e imagens estáticas [25]. Este mesma utilização da SVM, separação para análise de grupos específicos, pode ser utilizada para analisar a influência de descritores concretos para a resolução do problema [28].

É possível também utilizar-se Gradientes Estocásticos Descendentes (SGD) em conjunto com SVMs [5, 32, 33, 16]. SGD é um algoritmo de otimização de aprendizado assintótico, que diminui o número de passos para a convergência do treinamento da SVM e que melhora a escalabilidade da etapa de aprendizado, importante quando é utilizado um número muito grande de descritores [44].

A classificação binária também pode ser feita utilizando uma camada de rede neural completamente conectada combinada com uma função de ativação, como por exemplo a softmax, como última camada do algoritmos de classificação [35, 34, 36, 7].

2.3.2 Classificação e Regressão Multiclasse

O uso da classificação multiclasse no problema de classificação estética de fotos tem como objetivo obter-se uma quantificação da estética da fotografia, um valor similar a uma nota. Isto possibilita uma definição

mais clara do quão esteticamente atraente uma fotografia é e explicita os casos de medianos, em que a fotografia não é considerada nem muito boa e nem muito ruim.

Para a classificação multiclasse pode-se utilizar métodos de regressão, como a regressão linear [29] e o Support Vector Regression (SVR) [13, 30, 29], técnicas baseadas no teorema de Bayes, seja como parte da modelagem de dependência de características [17], seja com o uso de uma rede Bayesiana [15] e redes neurais [38, 37, 39].

Na classificação multivariável um dos métodos mais utilizados é o Support Vector Regression (SVR) [13, 30, 29]. O SVR toma como base os hiperplanos de separação de classe do SVM. O objetivo do SVR é encontrar uma função da margem entre os pontos de treinamento e os hiperplanos de classificação, mantendo-se a condição da margem grande de separação. Usar os descritores de SVR implica em encontrar o vetor de diferenças de notas entre a fotografia analisada e aquelas utilizadas na etapa de treinamento do classificador [30].

Uma outra abordagem do SVR é o Support Vector Distribution Regression (SVDR), que leva em conta a distribuição de probabilidades da avaliação de qualidade estética da imagem [12]. O SVDR demonstrou resultados marginalmente iguais ou inferiores ao SVR e com relação a redes neurais.

Em sua formulação original, SVMs foram projetadas para a tarefa de classificação binária. Existem meios de estender esta formulação para classificação multiclasse, com a combinação de diversos classificadores binários ou por um método direto no qual todos os dados são considerados e uma otimização é realizada [45]. É possível utilizar um SVM multiclasse para o problema de categorização de imagens [17].

Redes Bayesianas são representações de grafos das relações modeladas pelo teorema de Bayes entre os descritores estéticos e são utilizadas quando se deseja avaliar fotografias com relação a vários fatores simultaneamente, classificando-se, além da estética, fatores como a atratividade e a memorabilidade [15].

Pode-se também utilizar métodos baseados em multitarefas, tais como redes neurais profundas [38, 37, 39]. Estes métodos tem como objetivo aumentar a generalização dos modelos, aprendendo diversos aspectos do problema simultaneamente. Em redes neurais é necessário normalizar os dados de aprendizado para que a categorização possa ser feita adequadamente [37].

3. Aprendizado de Máquina para Classificação Estética

Aprendizado estatístico é o conjunto de ferramentas para análise de dados, podendo ser feita de modo supervisionado ou não-supervisionado e é uma área que abrange regressões, analises de discriminantes, modelos generalizados, classificação, modelos generativos e aprendizado de máquina [46]. Através do aprendizado estatístico é possível criar um modelo para a predição ou estimação de uma variável de saída a partir de uma ou mais entradas [46]; no caso da classificação estética de imagens estas classes se referem a imagens esteticamente boas ou não.

A utilização de modelos de aprendizado de máquina para a classificação estética de fotografias de fotografias é o que permite a classificação automática das fotografias dado um conjunto de treinamento pré confeccionado, tanto na classificação quanto na geração de descritores estéticos genéricos.

Dentre os modelos utilizados se destacam as redes neurais convolucionais profundas (DCNNs) para a tarefa de geração de descritores estéticos genéricos e os métodos derivados de máquinas de vetores de suporte (SVMs) na tarefa de classificação – tanto na classificação binária [6, 23, 24, 22, 25, 8, 26, 28, 27] quanto na classificação multiclasse [13, 30, 29]. Uma solução que mista que usa DCNNs com uma última camada de L2-SVM inspirada em [9] será utilizada neste trabalho.

As seções deste capítulo fazem a revisão teórica dos dois modelos de aprendizado de máquina utilizados nesse trabalho com o objetivo de construir a base para que o sistema de classificação em si possa ser desenvolvido.

3.1 Redes Neurais Profundas

A terminologia aprendizagem profunda ou "deep learning" é derivada da profundidade de camadas em um modelo de redes neurais [47]. Cada uma das camadas de uma rede neural é uma representação de uma função e a sequência das camadas representa o encadeamento dessas funções. Quanto mais profunda uma rede maior é o tamanho deste encadeamento.

Uma forma de utilizar modelos de aprendizagem profunda para classificação de imagens é através de uma rede neural convolucional profunda (DCNN) treinada de forma supervisionada para a extração de atributos. Pode-se então utilizar estes atributos para a classificação das imagens com algum outro modelo, por exemplo uma SVM ou uma regressão logística. Este é o modelo que foi utilizado em [48], com uma abordagem que também utiliza o conceito de transferência de conhecimento, com os atributos treinados com um conjunto de imagens sendo usados para uma tarefa de classificação arbitrária.

Modelos de redes convolucionais profundas tem sido aplicados à tarefas de reconhecimento visual em larga escala com sucesso, tendo superado o desempenho de outros métodos em competições de reconhecimento de objetos em larga escala [48]. A profundidade da rede, em geral, é diretamente proporcional à capacidade de generalização da mesma.

Cada camada da rede implementa funções distintas, cada qual com um objetivo diferente. Nas seções

deste capítulo são descritas, baseado em [47], as camadas e técnicas que irão compor a rede neural deste projeto.

3.1.1 Camada convolucional

A camada convolucional de uma rede profunda implementa uma rede neural convolucional (CNN), que são redes especializadas no processamento de dados que possuem uma estrutura matricial, como no caso de imagens digitais. A diferença entre uma uma CNN e uma rede neural convencional está na operação utilizada: enquanto a CNN utiliza a convolução de um kernel com a entrada a rede convencional utiliza apenas uma multiplicação matricial (entre a camada de entrada e os pesos dos neurônios).

A convolução no caso contínuo, entre uma função de entrada x e uma função de pesos, ou kernel, w pode ser escrita como 3.1.

$$S(t) = \int x(a) \cdot w(t-a)da \tag{3.1}$$

Entretanto, no caso de imagens estamos lidando com uma função discreta que se estende em duas dimensões. Assim, pode-se definir a convolução, tendo I como os dados da imagem, K o kernel, m e n as dimensões do kernel e S a saída como 3.2.

$$S(i,j) = \sum_m \sum_n I(i-m, j-n) \cdot K(m,n) \tag{3.2}$$

A motivação por trás do uso de convolução é devido a três fatores: interações esparsas, compartilhamento de parâmetros e representações equivariantes. As interações esparsas se devem ao fato de que quando se utiliza kernels, pode-se detectar atributos pequenos que estão espalhados numa imagem, por exemplo, utilizando uma quantidade menor de parâmetros armazenados, reduzindo os requisitos operacional e de memória do modelo e melhorando a eficiência estatística. O compartilhamento de parâmetros se refere ao fato de que o kernel é aplicado à toda entrada, diferentemente do peso de um neurônio de uma rede neural convencional, que é utilizado apenas para uma entrada específica. Além disso, por conta da convolução, uma forma particular de compartilhamento de parâmetros leva a camada convolucional a possuir uma equivariância a translação, pois um atributo pode ser identificado independentemente de onde se encontra na entrada.

Sendo D_F a largura e altura de uma entrada quadrada numa camada convolucional, M a quantidade de canais na entrada, D_K a largura e altura da saída quadrada numa camada convolucional e N a quantidade de canais na saída, o custo computacional de uma camada de convolução convencional pode ser dada por 3.3 [49].

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \tag{3.3}$$

3.1.1.1 Convolução separável espacialmente

A convolução separável espacialmente, como o nome diz, separa a convolução de acordo com as dimensões espaciais da entrada. Num exemplo bidimensional, como no caso de imagens, a separação é feita na largura e altura da imagem. Assim, um kernel 3×3 que numa convolução tradicional seria aplicado de uma vez à uma imagem é separado em dois kernels equivalentes ao primeiro, o primeiro 3×1 e segundo 1×3, como pode ser observado na Equação 3.4.

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \tag{3.4}$$

O principal objetivo de se separar os kernels em dois é reduzir a quantidade de operações necessárias para se computar a convolução. No caso de uma janela 3×3, um passo do kernel no pixel equivale a 9 multiplicações enquanto que para a convolução espacial separável apenas 6 multiplicações se fazem necessárias, ou seja, uma redução de 33%. O ponto negativo do uso de convoluções deste tipo é a redução dos kernels que podem ser representados numa camada, uma vez que apenas os kernels separáveis em dois serão usados, uma fração do domínio de kernels normais.

Desconsiderando os canais de uma imagem, o custo da convolução separável espacialmente é dada pela Equação 3.5.

$$2 \cdot D_K \cdot D_K \cdot D_F \tag{3.5}$$

3.1.1.2 Convolução separável em profundidade

A convolução separável em profundidade é similar à separável espacialmente mas, além das dimensões espaciais também considera a dimensão de profundidade. No caso de imagens RGB a profundidade é a dimensão dos canais de cores. Na convolução separável em profundidade o kernel também é dividido em dois, um para a convolução em profundidade e um para um convolução pontual.

Na convolução em profundidade é feita a convolução convencional por nível de profundidade, utilizando um número de kernels correspondente à dimensão da profundidade e resulta numa saída de mesma profundidade que a entrada. Todos os kernels aplicados possuem a mesma janela. No caso de uma imagem RGB poderiam ser utilizados três kernels de janela 3×3 por exemplo. O custo da convolução em profundidade é dado pela Equação 3.6.

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F \tag{3.6}$$

A convolução pontual é sempre feita numa janela 1×1 mas que se extende ao longo da dimensão de profundidade e tem como objetivo fazer a convolução nesta dimensão. Se apenas 1 janela de convolução for utilizada a saída terá a sua dimensão de profundidade reduzida à unidade, uma vez que a convolução irá aglutinar a dimensão de profundidade. Caso mais de uma camada profundidade seja desejada na saída pode-se utilizar de mais de uma janela nesta convolução, empilhando as saídas de cada uma das convoluções pontuais no final. Se três dimensões sejam desejadas na saída da convolução de profundidade 3 janelas de kernel pontual podem ser utilizadas. Uma outra forma de entender a convolução pontual é interpretá-la como um alongador de dimensão de profundidade que aplica uma função de não linearidade (ReLU, softmax por exemplo). O custo da convolução pontual é dada pela Equação 3.7.

$$M \cdot N \cdot D_F \cdot D_F \tag{3.7}$$

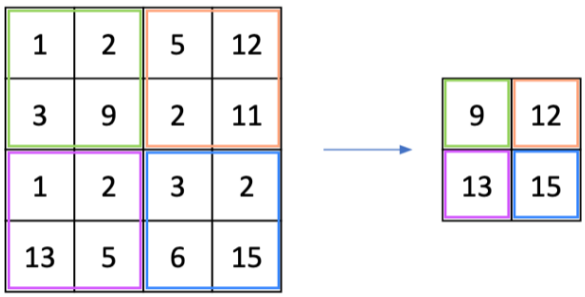
Assim, o custo total da convolução é dada pela Equação 3.8. Comparando com o custo da convolução convencional 3.3, pode-se concluir que há uma redução com relação ao tamanho da entrada e aos canais de saída, como pode ser observado na Equação 3.9. Reduções entre 8 e 9 vezes foram observadas quando convoluções separáveis em profundidade foram usadas ao invés da convolução convencional com apenas uma pequena perda de acurácia [49].

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F \tag{3.8}$$

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2} \tag{3.9}$$

Da mesma forma que a convolução espacialmente separável, a convolução em profundidade é aplicada

Figura 3.1: Exemplo de aplicação do max pooling considerando uma janela quadrada de lado 2.



Fonte: Elaborada pelo autor.

com o objetivo de reduzir o número de operações necessárias para se realizar a convolução numa camada de rede neural.

3.1.2 Camada de subamostragem

A subamostragem, ou pooling, é uma função tipicamente utilizada com o intuito de substituir a saída de uma camada em cada local pelo resumo estatístico das saídas adjacentes àquele local. O max pooling, por exemplo, substitui a saída pela saída máxima dentro de uma vizinhança retangular próxima, como pode ser observado na Figura 3.1.

A camada de subamostragem tem duas funções. A primeira se refere a auxiliar a representação do modelo a se tornar invariante a pequenas translações da entrada. A segunda é a de reduzir o tamanho da rede se utilizada com uma redução de resolução, pois é possível realizar o pooling não apenas a cada 1 pixel, mas a cada k pixels. Essa redução melhora a eficiência computacional da rede, diminuindo também os requisitos de memória. O uso de pooling pode ser entendido como uma hipótese forte de que a camada deve ser invariante a translação. Se esta for verdadeira, pode-se melhorar a eficiência estatística da rede como um todo.

3.1.3 Camada completamente conectada

Uma camada completamente conectada implementa uma rede neural convencional completamente conectada, ou seja, todos os neurônios da camada estão conectados com todas as entradas, com cada conexão com seu respectivo peso. Esta camada pode ser representada como a multiplicação entre a camada de entrada e uma matriz de pesos, existindo um peso para cada neurônio para cada entrada correspondente.

A camada completamente conectada é a responsável por fazer a classificação propriamente dita, baseada nos atributos que foram extraídos pelas camadas de convolução e pooling.

3.1.4 Aprendizagem residual

O aumento da profundidade de redes neurais pode levar à degradação do desempenho dos modelos devido à um aumento da complexidade de otimização do modelo. Entretanto, existe uma solução por construção que leva a modelos mais profundos que devem ter pelo menos a acurácia dos modelos mais rasos. Considerando uma rede rasa e a sua contraparte mais profunda, se todas as camadas adicionadas na rede mais profunda forem camadas que fazem a identidade da entrada indica que existe uma construção que tem um erro de treinamento que não pode ser superior ao erro de treinamento da sua contraparte mais rasa [50].

Com o fato de que as camadas adicionadas em redes mais profundas são mapeamentos de funções identidade, pode-se considerar $\mathcal{H}(x)$ como o mapeamento de algumas camadas de uma DCNN. Assumindo

que as entradas e saídas deste conjunto de camadas tenha o mesmo tamanho e que este conjunto de camadas é capaz de aproximar uma função residual, pode-se definir a função $\mathcal{F}(x) \coloneqq \mathcal{H}(x) - x$. Assim, o mapeamento residual pode ser reescrito como a Equação 3.10.

$$\mathcal{H}(x) = \mathcal{F}(x) + x \tag{3.10}$$

Em casos reais este mapeamento para uma função identidade em geral não é ótima mas já foi mostrado que as funções residuais formam uma precondição razoável que aumenta o desempenho da rede [50].

3.1.5 Camada de Dropout

O uso de uma camada de dropout é uma forma, com baixo custo computacional, de se regularizar um modelo, ou seja, adiciona uma modificação com a intenção de reduzir o erro de generalização do modelo, mas não o de treinamento.

Especificamente, a camada de dropout treina um conjunto consistindo de todas as subredes que podem ser formadas removendo-se unidades de uma determinada camada. A remoção pode ser feita simplesmente pela multiplicação da saída daquela unidade por zero. Este processo gera uma quantidade exponencialmente grande de redes treinadas.

Em um treinamento de um modelo com uma camada de dropout, a cada iteração do treinamento uma diferente máscara que será aplicada àquela camada é selecionada. A máscara define quais são os neurônios ativos naquele passo de treinamento e é controlada por um parâmetro que define a probabilidade de um neurônio estar desativado.

A principal vantagem da camada de dropout se deve ao fato de possuir custo computacional baixo, tendo um custo de $O(n)$, onde n se refere ao número de neurônios da camada da rede em que está sendo aplicado.

Finalmente a camada de dropout pode ser combinada com outras formas de regularização (como bagging e ensemble) com o intuito de levar a melhores resultados na redução do erro de generalização do modelo.

3.1.6 Camada de normalização de lote

A camada de normalização de lote (do inglês *batch normalization*) é um método de reparametrização adaptativa, que tem por objetivo ajudar no processo de otimização durante o treinamento de uma rede neural profunda.

Uma rede neural profunda representa a composição de muitas funções em diversas camadas. Durante o processo de atualização de parâmetros na etapa de treinamento de uma rede existe a hipótese de que quando os parâmetros de uma camada são atualizados os demais se mantêm constantes, mas, na prática, todas as camadas são atualizadas de forma simultânea.

A normalização em lote reduz de forma significante o problema de coordenar as atualização ao longo de diversas camadas e pode ser aplicada em qualquer camada de entrada ou escondida.

Basicamente, a camada faz a normalização das ativações de uma camada. Sendo H um mini-lote de ativações de uma camada, μ a média e σ o desvio padrão, podemos normalizar usando segundo a Equação 3.11.

$$H' = \frac{H - \mu}{\sigma} \tag{3.11}$$

Sem essa normalização, cada atualização na rede tem um efeito extremo nos pesos das unidades. A normalização em lote, tem, portanto, a função de estabilizar o processo de aprendizagem. No caso de redes

neurais convolucionais, é importante normalizar o mapa de atributos em todas as localizações espaciais, para garantir que as estatísticas deste mapa sejam as mesmas independentemente da localização espacial.

3.1.7 Otimizador Adam

Adam [51] ("Adaptative moment estimation") é um otimizador estocástico baseado em gradiente que requer apenas gradientes de primeira ordem e, portanto, é bastante eficiente em termos de uso de memória.O método combina dois outros métodos de otimização – RMSProp [52] e AdaGrad [53] – combinando a utilização de momento com redimensionamento de gradientes e correlação de viés nas estimativas. O Algoritmo 1 mostra a aplicação do método. Parte-se do cálculo do gradiente e faz-se a etapa de atualização de parâmetros baseado em dois fatores de momento corretivos.

Algoritmo 1 O algoritmo do otimizador Adam.

ϵ: Taxa de aprendizagem
ρ_1, ρ_2: Taxa de decaimento de momento
δ: Fator de estabilização numérica
θ: Parâmetros iniciais
Inicializa-se o primeiro e segundo momentos: s=0, r=0
Inicializa-se o passo t=0
while critério de parada do
  Escolhe-se um lote de exemplos $\{x^1, ..., x^m\}$ com classes y^i respectivamente
  $g \leftarrow \frac{1}{m}\nabla L(f(x^i;\theta), y^i)$
  $t \leftarrow t + 1$
  $s \leftarrow \rho_1 s + (1 - \rho_1)g$
  $r \leftarrow \rho_2 r + (1 - \rho_2)r$
  $\hat{s} \leftarrow \frac{s}{1-\rho_1^t}$
  $\hat{r} \leftarrow \frac{r}{1-\rho_2^t}$
  $\Delta\theta = -\epsilon\frac{\hat{s}}{\sqrt{\hat{r}}+\delta}$
  $\theta \leftarrow \theta + \Delta\theta$
end while

3.2 Arquiteturas de Redes Neurais Profundas

O ImageNet Large Scale Visual Recognition Challenge (ILSVRC) é uma competição de referência na área de detecção de objetos e classificação de imagens e tem sido usada como base de comparação para a avaliação do progresso de novos algoritmos de aprendizagem de máquina. No ILSVRC 2012 a rede AlexNet, com arquitetura baseada em DCNN, obteve um desempenho superior aos demais algoritmos que estavam também na competição [54] iniciando um período de grande aplicação de DCNNs em problemas relacionados a imagens. Diversas arquiteturas de DCNNs foram desenvolvidas ao longo dos anos, tendo como exemplo o VGG [55] e o Inception [56], e, a cada ano, melhorias significativas com relação ao ano anterior foram observadas no desafio de classificação de objetos do ILSVRC.

Esta sessão tem como objetivo apresentar diferentes arquiteturas de DCNNs que foram desenvolvidas nos últimos anos, comparando as suas diferenças e desempenho no ILSVRC. As arquiteturas avaliadas são: VGG16 [55], MobileNet [49], ResNet50 [50], NASNetMobile [57], MobileNet V2 [58], Inception V3 [59] e Inception-ResNet V2 [60].

3.2.1 VGG

O VGG [55] foi publicado em 2014 e tem uma arquitetura composta por uma imagem de entrada de $224 \times 224 \times 3$ que passa por uma série de camadas convolucionais, cada uma com um campo receptivo

pequeno de 3×3 pixels, o qual é o menor tamanho para se ter a noção de direções e centro. O passo da janela de convolução é de 1 pixel, de forma a manter a resolução do vetor em cada uma das camadas. Intercalando as camadas de convolução existem também camadas de max pooling, cada uma numa janela de 2×2 pixels com um passo de 2. Em todas as combinações propostas em [55], 5 camadas de max pooling são utilizadas. Um conjunto de 3 camadas completamente conectadas dá seguimento à rede, as duas primeiras com 4096 neurônios e a última com 1000 neurônios correspondentes às classes do ILSVRC. A última camada da rede utiliza softmax.

As primeiras camadas do VGG possuem 64 neurônios, as após o primeiro max pooling 128, após o segundo 256 e a partir do terceiro max pooling todas possuem 512 neurônios. Dentre as diferentes composições destas camadas, as composições mais conhecidas são VGG16, que possui uma combinação de 16 camadas – 13 de convolução, 5 de max pooling e 3 completamente conectadas – e o VGG19 com 19 camadas – 16 de convolução, 5 de max pooling e 3 completamente conectadas.

Uma das principais características do VGG é o uso de convoluções com um pequeno campo receptivo (3×3) ao invés de usar campos maiores como 7×7. É importante notar, entretanto, que a duas camadas de convolução de janela 3×3 sem pooling entre elas correspondem à uma camada de janela 5×5; três camadas de convolução 3×3 correspondem à uma janela de 7×7. Usar este tipo de composição incorpora mais classificadores ReLU ao modelo, fazendo a função de classificação ser mais discriminativa e diminui o número de parâmetros: sendo C o número de nós numa camada 3 camadas de convolução com janela 3×3 requerem $3(3^2C) = 27C^2$ enquanto uma camada de janela 7×7 requer $7^2C = 49C^2$.

3.2.2 ResNet50

Publicada em 2015, a ResNet50 faz parte de uma família de arquiteturas de redes baseadas no conceito de aprendizado profundo residual apresentado na sessão 3.1.4, sendo essa família também composta pela ResNet-34, ResNet-101 e ResNet-152, nas quais o número representa a quantidade de camadas da rede [50].

A função de aprendizagem residual 3.10 é modelada através de ligações de atalho e somas elemento a elemento, como pode ser observado na Figura 3.2. Esse bloco pode ser definido formalmente como 3.12, onde $\mathcal{F}(x, \{W_i\})$ é a função residual que deve ser aprendida.

$$y = \mathcal{F}(x, \{W_i\}) + x \tag{3.12}$$

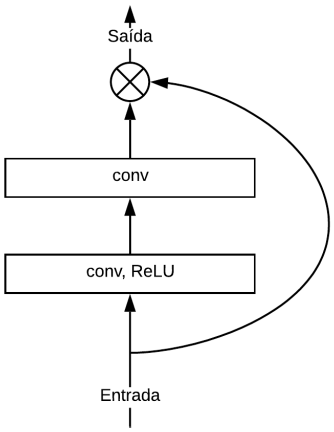
A arquitetura ResNet é baseada na VGG, com uma imagem de entrada com $224 \times 224 \times 3$ pixels e usando convoluções de janela 3×3 em camadas com um número crescente de neurônios. começando com camadas com 64 neurônios e terminando com camadas com de até 2048 neurônios. A diferença com relação ao VGG é que a cada certo número de camadas – 2 para arquiteturas de até 34 camadas e 3 para arquiteturas a partir de 50 camadas – uma ligação de atalho de identidade é utilizada.

Como bloco base para a construção dos ResNet, nos casos de 18 e 34 camadas, 2 camadas convolucionais com janela 3×3 são utilizadas, com uma ligação de atalho ao final. Para os casos a partir de 50 camadas uma estrutura de gargalo é utilizada, com três camadas convolucionais, com janelas 1×1, 3×3 e 1×1 e então uma ligação de atalho. As camadas 1×1 são responsáveis por alterar as dimensões de entrada e saída num bloco base do ResNet. Este bloco modificado para arquiteturas mais profundas é utilizado para reduzir os parâmetros e o tempo de treinamento da rede.

3.2.3 MobileNet

A arquitetura da MobileNet foi publicada em 2015 e tem como o principal objetivo ser uma arquitetura leve para uso em aplicações móveis. Ela é composta por convoluções separáveis em profundidade, com

Figura 3.2: Bloco residual. A conexão de atalho entre a entrada e a saída garante que o bloco também é capaz de aprender a função identidade.



Fonte: Elaborada pelo autor.

exceção da primeira camada, na qual é feita uma convolução convencional [49]. O bloco básico da MobileNet pode ser observado na Figura 3.3(a). Todas as camadas (inclusive entre cada uma das etapas da convolução separável em profundidade) é seguida por uma camada de normalização de lote e uma ativação ReLU. Ao invés de utilizar max pooling, a redução de dimensionalidade ao longo da rede é feita através de convoluções com passo maior do que 1. A última camada é composta por uma camada completamente conectada e um classificador softmax. No total a MobileNet conta com 28 camadas – 1 camada inicial com convolução convencional, 13 camadas com convoluções separáveis em profundidade e a última camada totalmente conectada com o classificador softmax.

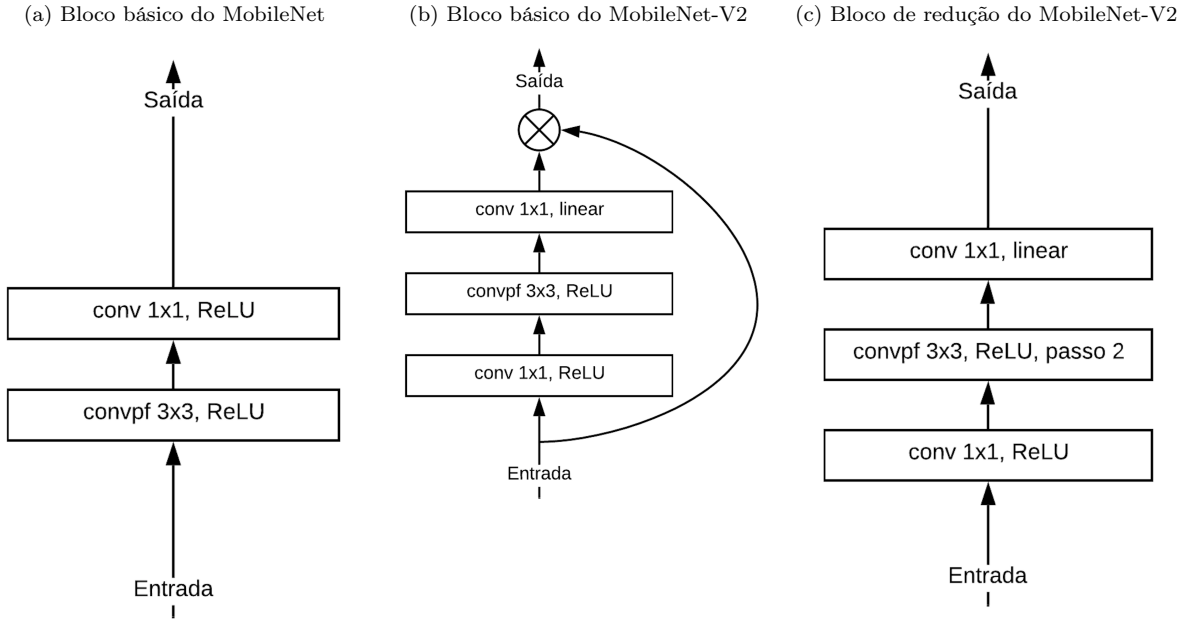
As camadas da MobileNet foram construídas de forma a se otimizar o tempo computacional, colocando todo o custo computacional nas convoluções pontuais que podem ser computadas usando funções altamente otimizadas para multiplicação de matrizes gerais (GEMM), usando 95% do tempo de computação nestas convoluções, que correspondem à 75% dos parâmetros da rede.

Na MobileNet também são apresentados parâmetros para diminuir o tamanho da rede. O primeiro é o parâmetro $\alpha\epsilon(0,1]$ de diminuição de largura da rede de forma uniforme. O segundo é o parâmetro $\rho\epsilon(0,1]$ que multiplica a imagem de entrada, reduzindo a representação de cada camada pelo mesmo multiplicador. O parâmetro ρ pode ser determinado implicitamente através da resolução da entrada. Ao final, o custo computacional da MobileNet é dado pela Equação 3.13, sendo que $\rho = 1$ e $\alpha = 1$ é o custo baseline da MobileNet.

$$D_K \cdot D_K \cdot \alpha M \cdot \rho D_F \cdot \rho D_F + \alpha M \cdot \alpha N \cdot \rho D_F \cdot \rho D_F \tag{3.13}$$

A MobileNet-V2 [58] introduz na MobileNet a ideia de camadas de gargalo lineares, cujo objetivo é reduzir a dimensionalidade das camadas convolucionais impedindo que as não-linearidades introduzidas pelas camadas de ativação, como a ReLU, destruam informação. Isso é feito através de um bloco de convolução de gargalo, que faz a expansão através de ativações lineares. Este bloco usa residuais invertidos: ao invés de aplicar o atalho da ligação residual após uma operação de expansão aplica-se após todas as operações de redução de dimensão. A MobileNet-V2 também é composta por blocos básicos, com o da Figura 3.3(b) sendo bloco fundamental e Figura 3.3(c) o bloco de redução de dimensão.

Figura 3.3: Blocos básicos da arquitetura MobileNet.



Fonte: Elaborada pelo autor.

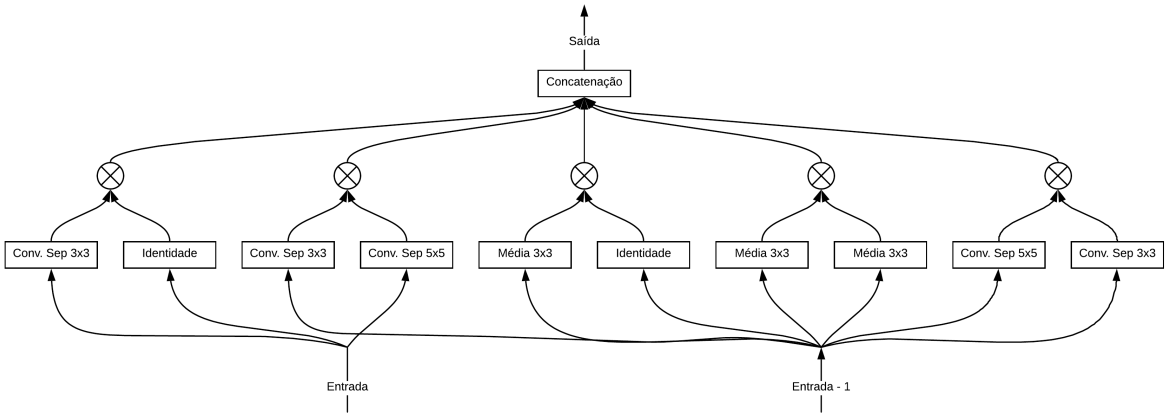
3.2.4 NASNet

Publicada em 2017, a NASNet é resultado de um método de busca por DCNNs usando o framework de Busca de Arquitetura Neural – Neural Architecture Search (NAS) – na qual um controlador composto por uma rede neural recorrente (RNN) é usado para construir redes sobre um dataset especifico, testá-las e selecionar as melhores, num processo contínuo de melhoria. A arquitetura encontrada para a NASNet é baseada no dataset CIFAR-10. A sua motivação é o fato de que a maioria das redes no estado da arte são compostas por blocos de filtros e por não linearidades interconectadas, sendo então possível encontrar um bloco genérico composto por estes sub-blocos que compõem essas redes [57].

Na arquitetura são propostos dois blocos básicos: um bloco convolucional normal, que retorna uma saída de mesmas dimensões que a entrada (célula normal) e um bloco convolucional de redução, em que a altura e largura da saída são reduzidas pela metade (célula de redução). Na célula de redução é aplicada uma operação inicial com passo 2 para que o tamanho seja reduzido. A RNN então faz o aprendizado das células convolucionais ótimas através de aprendizado por reforço, combinando as seguintes operações através de somas elemento a elemento ou concatenações das seguintes operações:

- Identidade
- Convolução separável espacialmente 1×3 e 3×1
- Convolução separável espacialmente 1×7 e 7×1
- Convolução dilatada 3×3
- Pooling de média 3×3
- Max pooling 3×3, 5×5 e 7×7
- Convolução convencional 1×1 e 3×3
- Convolução separável em profundidade 3×3, 5×5 e 7×7

Figura 3.4: Bloco base do NASNet-A.



Fonte: Elaborada pelo autor.

Dentre as arquiteturas obtidas, a chamada NASNet-A foi a que obteve o melhor desempenho e as célula convolucional normal é apresentada na Figura 3.4. Um bloco de redução também foi criado e é usado para etapas de reduções de dimensionalidade. Originalmente as arquiteturas NASNet foram treinadas com o banco de dados CIFAR-10 mas para o ImageNet as mesmas células convolucionais que foram aprendidas para o CIFAR-10 são utilizadas, com a diferença de mais blocos serem utilizados. Isto se deve ao fato de que as imagens do CIFAR-10 possuírem resolução de 32×32 enquanto as do ImageNet possuem resolução de 224×224. A arquitetura NASNet-A, com $N = 4$ é a arquitetura chamada de NASNetMobile enquanto para $N = 6$ é chamada de NASNetLarge.

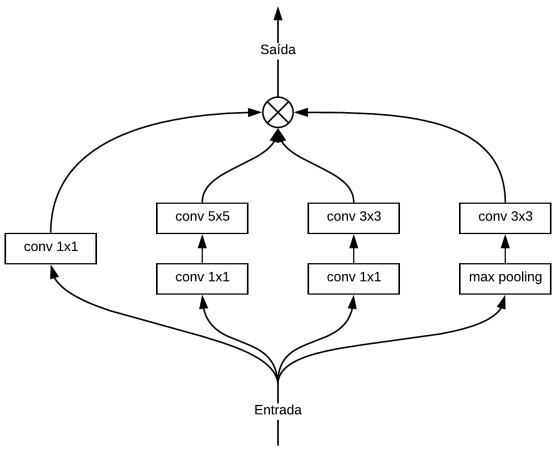
3.2.5 Inception

O Inception tem como motivação principal a construção de uma rede com uma maior profundidade e largura sem aumentar drasticamente o custo computacional e número de parâmetros da rede. Uma forma fundamental de resolver os problemas relacionados ao aumento da rede seria trocar as camadas completamente conectadas por camadas esparsas, entretanto a computação nestas estruturas esparsas é ineficiente. A principal ideia da arquitetura Inception é tentar aproximar estes filtros esparsos através de componentes densos [56]. A segunda ideia da arquitetura Inception é reduzir a dimensão sempre que os requisitos computacionais forem aumentar demais.

Toda a arquitetura é baseada no módulo Inception, que pode ser observado na Figura 3.5, o qual é empilhado ao longo da rede. O módulo é composto por filtros convolucionais de kernels 1×1, 3×3, 5×5 e max poolings 3×3 aplicados paralelamente à entrada da camada, com uma concatenação dos resultados ao final para a produção da saída da camada. Antes dos filtros 3×3 e 5×5 de convolução são usadas convoluções 1×1 para reduzir a dimensionalidade e funcionar como uma camada de não linearidade devido à sua ativação ReLU. Uma convolução 1×1 é também necessária após o max pooling para ajustar os canais da saída. Algumas camadas de max pooling com passo 2 são utilizadas entre alguns módulos Inception para reduzir a resolução da camada.

O Inception-V3 é uma arquitetura baseada no Inception [56], aprimorando algumas das estruturas propostas inicialmente com o objetivo de melhorar a escalabilidade da arquitetura original. O Inception-V3 melhora o Inception-V1 introduzindo 3 tipos novos de módulos. O primeiro destes módulos, Inception-A da Figura 3.6(a), usa o conceito de menores campos receptivos do VGG. Assim ao invés de utilizar uma convolução 5×5, duas convoluções 3×3 em sequência são utilizadas. O segundo destes módulos, Inception-B da Figura 3.6(b), substitui as convoluções com dimensões maiores por convoluções separáveis

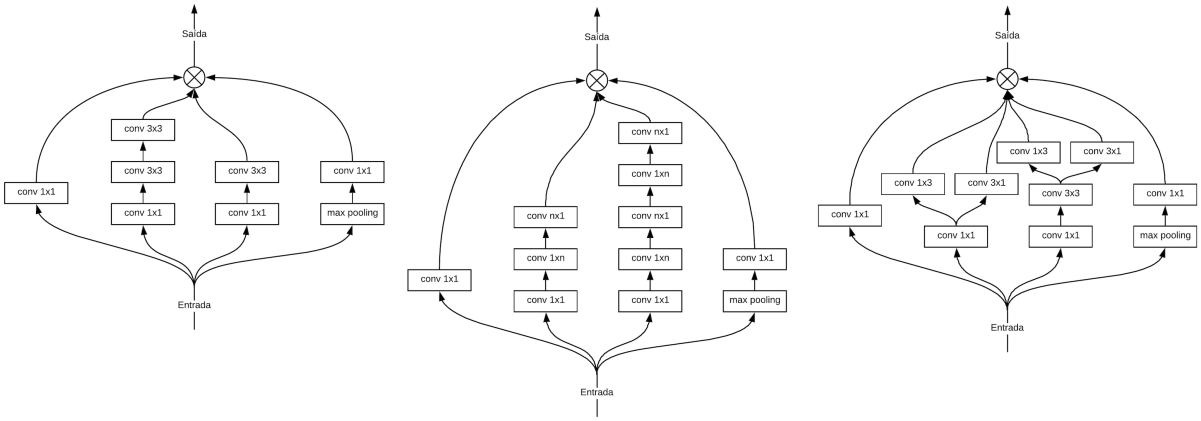
Figura 3.5: O bloco base da arquitetura Inception.



Fonte: Elaborada pelo autor.

Figura 3.6: Blocos da arquitetura Inception-V3.

(a) Bloco Inception-A (b) Bloco Inception-B (c) Bloco Inception-C



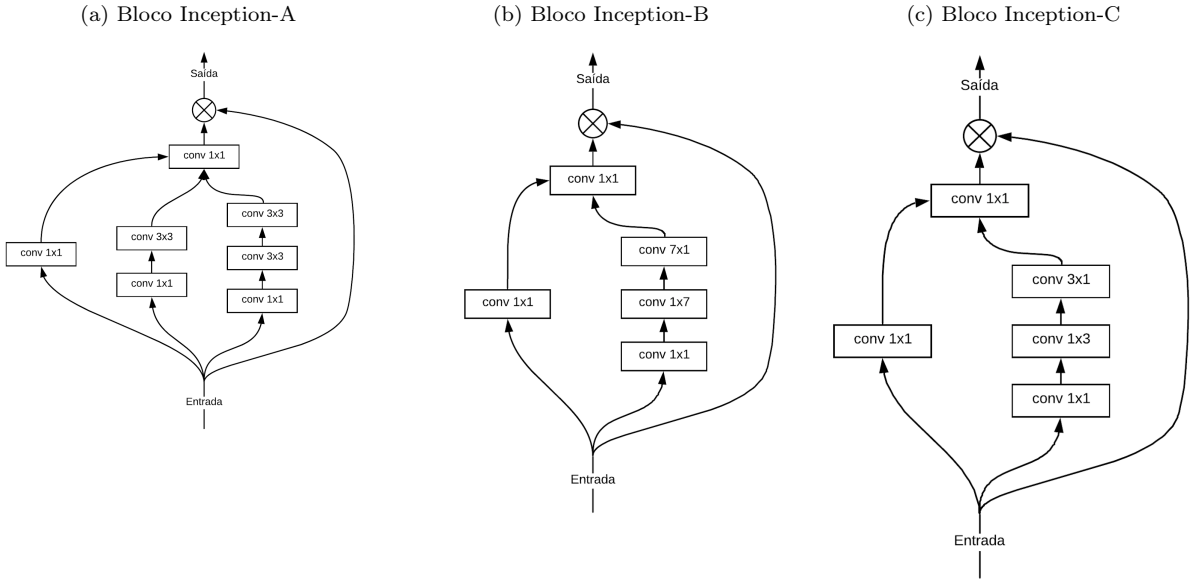
Fonte: Elaborada pelo autor.

linearmente: $1 \times n$ e $n \times 1$. O último dos módulos, Inception-C da Figura 3.6(c), tem como objetivo expandir a quantidade final de filtros, separando algumas das convoluções em duas saídas 1×3 e 3×1 em paralelo. Além disso, as convoluções iniciais de kernel maior também foram substituídas por um conjunto de convoluções 3×3.

Para evitar gargalos de representação antes das camadas de max pooling são utilizados módulos Inception com passo 2, com o objetivo de aumentar o número de filtros antes de realizar a redução de dimensionalidade espacial.

O Inception-ResNet-V2 [60] é uma arquitetura que tem como base o Inception-V3 mas incorpora a ideia de conexões residuais introduzidas pela ResNet. Assim, foram criados três blocos baseados nos blocos Inception-A e Inception-B do Inception-V3, com a diferença principal de usarem uma conexão de atalho entre a entrada e a saída do módulo. Os blocos, de forma análoga ao Inception-V3, também são nomeados como Inception-A (Figura 3.7(a)), Inception-B (Figura 3.7(b)) e Inception-C (Figura 3.7(c)). Os blocos de Inception que usam as conexões residuais tem um tempo de treinamento melhor o que possibilita a criação de redes de maior tamanho.

Figura 3.7: Blocos da Arquitetura Inception-ResNet-V2.



Fonte: Elaborada pelo autor.

Tabela 3.1: DCNN acurácia Top-1 and Top-5 no dataset de validação do ILSVRC Object Classification Challenge.

Arquitetura	Ano de Publicação	Acurácia Top-1	Acurácia Top-5	Parâmetros treináveis
VGG16	2014	71.3%	90.1%	138,357,544
MobileNet	2015	70.4%	89.5%	4,253,864
ResNet50	2015	74.9%	92.1%	25,636,712
NASNetMobile	2017	74.4%	91.9%	5,326,716
MobileNet-V2	2018	71.3%	90.1%	3,538,984
Inception-V3	2016	77.9%	93.7%	23,851,784
Inception-ResNet-V2	2017	80.3%	95.3%	55,873,736

Fonte: Elaborada pelo autor.

3.2.6 Discussão sobre as arquiteturas de DCNN

Todas as DCNNs apresentadas possuem diferenças estruturais e estas diferenças se refletem no seu desempenho perante a tarefa de classificação de objetos. Observando as arquiteturas de acordo com a Tabela 3.1 e comparando a arquitetura menos performática (VGG16) com a mais performática (Inception-ResNet-V2), pode-se observar um aumento da acurácia Top-1 em aproximadamente 9% e a Top-5 em aproximadamente 5%, com uma diminuição da quantidade de parâmetros treináveis. Esta melhoria foi feita em um espaço entre os trabalhos de apenas 3 anos, mostrando que melhorias estão sendo obtidas continuamente e que possivelmente novas arquiteturas com melhor desempenho serão propostas nos próximos anos.

Outra análise importante dos dados da Tabela 3.1 é com relação aos parâmetros treináveis em si. Não necessariamente uma rede com mais parâmetros, ou seja, com mais neurônios ou mais profunda, resulta em uma rede com melhor desempenho. É necessário usar estes parâmetros da melhor forma possível de forma a evitar que dados relevantes sejam descartados ao longo do treinamento. Além disso, redes com mais parâmetros tem um maior custo computacional que levam à redução da sua aplicabilidade em problemas reais.

Finalmente, uma análise final dos dados desta tabela é com relação às arquiteturas com um número reduzido de parâmetros. O principal objetivo destas redes é ampliar o leque de aplicação de DCNNs,

Tabela 3.2: Comparação de técnicas utilizadas nas arquiteturas de DCNN.

Arquitetura	Composição de Kernel 3×3	Convolução separável em profundidade	Convolução separável espacialmente	Atalho Residual	Representação de Esparcialidade
VGG16	X				
MobileNet		X			
ResNet50				X	
NASNetMobile	X	X	X		
MobileNet-V2		X		X	
Inception-V3			X		X
Inception-ResNet-V2	X		X	X	X

Fonte: Elaborada pelo autor.

em especial em ambientes móveis, que têm requisitos de memória e processamento mais restritos. Um menor número de parâmetro implica em uma menor representatividade e generalização do problema quando comparadas com arquiteturas mais robustas, entretanto os resultados mostram que é possível obter resultados comparáveis a arquiteturas com um número de parâmetros ordens de grandeza maiores.

As técnicas apresentadas nos trabalhos mais antigos foram reaproveitadas e combinadas em trabalhos mais recentes. A Tabela 3.2 mostra as principais técnicas utilizadas na construção das arquiteturas, desde a composição de kernels 3×3 ao invés de usar kernels com janela maiores até a representação de esparcialidade através de convoluções em paralelo com diversos tipos de kernel. A introdução de convoluções separáveis e de atalhos residuais diminuíram significativamente o número de parâmetros treináveis necessários para que resultados equivalentes fossem obtidos. A introdução da representação de esparcialidade pelo bloco Inception aumentou significativamente o desempenho das redes na tarefa de classificação. Finalmente a combinação das diversas técnicas apresentadas ao longo dos anos possibilitou a construção de redes com a melhor eficácia, em particular a Inception-ResNet-V2 para redes robustas e a NASNetMobile para redes móveis.

É importante ressaltar que todas as análises feitas até então se referem apenas à tarefa de classificação de objetos usando a base de dados da ILSVRC Object Classification Challenge. É também objetivo deste trabalho analisar se as mesmas observações poderão ser feitas sobre estas diferentes arquiteturas para o problema de classificação estética após a transferência de conhecimento.

Alguns princípios gerais já foram propostos para a criação de arquiteturas de redes convolucionais em [59]. São eles:

1. Evitar gargalos de representação; no geral deve-se diminuir o tamanho da representação de forma gradual até a camada de saída;

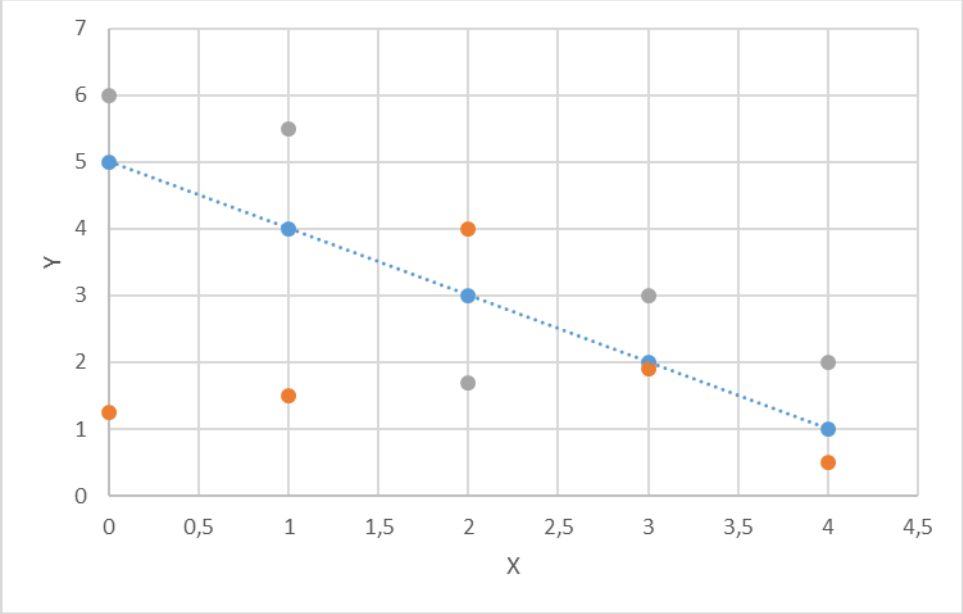
2. Representações de mais alto nível são mais facilmente processáveis localmente. Ter mais filtros por entrada possibilita o aprendizado de atributos menos correlacionados;

3. Agregações espaciais podem ser feitas em dimensões menores sem perder representatividade, ou seja, aplicar pooling antes de se fazer a convolução é uma otimização válida;

4. Deve-se balancear amplitude e profundidade da rede;

Todos estes princípios foram observados em maior ou em menor grau quando as arquiteturas foram analisadas por exemplo, as reduções graduais ao longo das arquiteturas e as agregações após as camadas de pooling são características encontradas em todas as redes. As redes Inception em especial aplicam o princípio 2 em maior grau, com mais filtros por entrada. Redes mais antigas, como por exemplo a VGG16 possuem uma relação profundidade contra amplitude pior, o que leva a redes que tem um custo computacional mais elevado.

Figura 3.8: Exemplo de um classificador em uma SVM no caso bidimensional.



Fonte: Elaborada pelo autor.

3.3 Máquinas de Vetores de Suporte

Máquinas de vetores de suporte (SVMs) formam uma classe de classificadores que usam um espaço de hipóteses linear que separam atributos que estão em espaços de atributos com muitas dimensões. São treinadas com um algoritmo baseado na teoria de otimização e o viés de treinamento é baseado na teoria de aprendizado estatístico desenvolvido por Vladmir Vapnik e a sua equipe [61].

Nas SVMs buscam-se hiperplanos que melhor dividem as diferentes classes de hipóteses a partir de atributos previamente definidos. Um hiperplano ótimo é aquele que melhor otimiza o problema de generalização das SVMs, de forma a previnir o overfitting e controlar as margens deste hiperplano [61]. Os denominados vetores de suporte são os elementos do subconjunto de pontos de treinamento que definem os classificadores, os pontos que suportam as equações de predição [43].

A margem é uma métrica definida de forma como a distância entre o classificador e o ponto de treinamento de uma determinada classe mais próximas [43]. Matematicamente, dado um exemplo definido por um vetor $\overrightarrow{x_i}$ e uma saída y_i com respeito a um hiperplano definido pelo vetor de pesos $\overrightarrow{w}$ e um viés b a margem γ_i é dada pela Equação 3.14 [61]. Caso a margem $\gamma_i > 0$ o exemplo $(\overrightarrow{x_i}, y_i)$ é classificado corretamente pelo hiperplano.

$$\gamma_i = y_i(< \overrightarrow{w}, \overrightarrow{x_i} > +b) \tag{3.14}$$

No exemplo bidimensional da Figura 3.8, os pontos vermelhos e azuis representam classes distintas e os eixos se referem a dois atributos distintos. Neste caso o classificador da SVM buscado é linear, o qual melhor separa as duas classes de hipóteses a partir das evidências. Note que nem todos os exemplos são classificados corretamente.

3.3.1 Modelos de Support Vector Machine

O modelo mais simples e que foi o primeiro a ser introduzido foi o classificador de margem máxima. Este classificador só funciona com problemas linearmente separáveis, mas é o de mais fácil entendimento e forma a base para modelos mais complexos. Neste modelo o hiperplano é definido pelas margens máximas dos

atributos. O problema deste modelo, e o que o restringe à problemas linearmente separáveis, é o fato de que ele sempre gera um erro de treinamento zero, ou seja, é totalmente consistente com os exemplos de treinamento [61].

Para o caso linear separável, dado um conjunto de dados $\{\overrightarrow{x_i}, y_i\}, i = 1, ..., l, y_i \epsilon \{-1, 1\}, \overrightarrow{x_i} \epsilon R^d$ pode-se supor que existe um hiperplano separador entre as classes positivas ($y_i = 1$) e negativas ($y_i = -1$). Os pontos que $\overrightarrow{x_i}$ no hiperplano satisfazem $< \overrightarrow{w}, \overrightarrow{x_i} > +b = 0$, com $\overrightarrow{w}$ normal ao hiperplano e $|b|/||\overrightarrow{w}||$ a distância do hiperplano à origem. Sendo d_+ e d_- as menores distâncias entre o hiperplano e o exemplo positivo e negativo respectivamente mais próximo a margem também pode ser definida como $d_+ + d_-$. Encontrar o melhor hiperplano separador implica em maximizar a margem. Supondo um caso em que todos os dados são separáveis pode-se formular o problema como a Inequação 3.15.

$$y_i(< \overrightarrow{w}, \overrightarrow{x_i} > +b) - 1 \geq 0, \forall i \tag{3.15}$$

Tendo em mente que as classes podem ser apenas negativas ou positivas, ou seja, $y_i \epsilon \{-1, 1\}$, pode-se encontrar dois hiperplanos em que há a equidade da Equação 3.15, os quais são os hiperplanos referentes aos dados $\{\overrightarrow{x_i}, y_i\}$ mais próximos do hiperplano separador. São eles $< \overrightarrow{w}, \overrightarrow{x_i} > +b = 1$ e $< \overrightarrow{w}, \overrightarrow{x_i} > +b = -1$ e a distância desses hiperplanos à origem é respectivamente $|1 - b|/||\overrightarrow{w}||$ e $| - 1 - b|/||\overrightarrow{w}||$. Assim, nesse caso, a margem é simplesmente $2/||\overrightarrow{w}||$. Para encontrar o hiperplano separador basta então resolver o problema de otimização de maximização de margem, ou seja:

- maximizar w^2
- sujeito a $y_i(< \overrightarrow{w}, \overrightarrow{x_i} > +b) - 1 \geq 0, \forall i$

Dados os multiplicadores de Langrange $\alpha_i, i = 1, ..., l$ pode-se definir o Lagrangiano do problema como 3.16.

$$L_P \equiv 1/2||\overrightarrow{w}||^2 - \sum_{i=1}^{l} \alpha_i y_i(< \overrightarrow{w}, \overrightarrow{x_i} > +b) + \sum_{i=1}^{l} \alpha_i \tag{3.16}$$

Esse problema de otimização pode ser resolvido maximizando-se L_P com a condição de que o gradiente de L_P com relação a $\overrightarrow{w}$ e b seja nulo e que $\alpha_i \geq 0$. As condições de gradiente resultam em 3.17 e 3.18 que, quando substituídas em 3.16 resultam em 3.19.

$$\overrightarrow{w} = \sum_{i=1}^{l} \alpha_i y_i \overrightarrow{x_i} \tag{3.17}$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{3.18}$$

$$L_D \equiv \sum_{i=1}^{l} \alpha_i - 1/2 \sum_{i=1,j=1}^{l,l} \alpha_i \alpha_j y_i y_j < \overrightarrow{x_j}, \overrightarrow{x_i} > \tag{3.19}$$

Assim, encontrar o hiperplano separador é maximizar L_D com relação aos multiplicadores de Lagrande com as condições 3.17 e 3.18.

Outra solução do problema pode ser feita através da descrição do problema primal através das condições de Karush-Kuhn-Tucker (KKT) 3.20, 3.21, 3.22, 3.23 e 3.24. Como o problema da SVM é convexo, as condições KKT são necessárias e suficientes para encontrar a solução. Assim, resolver as condições equivale a achar a solução da SVM.

$$\frac{\partial}{\partial w_v} L_P = w_v - \sum \alpha_i y_i x_w = 0, v = 1, ..., d \tag{3.20}$$

$$\frac{\partial}{\partial b} L_P = -\sum \alpha_i y_i = 0 \tag{3.21}$$

$$y_i(\overrightarrow{x_i} \cdot \overrightarrow{w} + b) - 1 \geq 0, i = 1, ..., l \tag{3.22}$$

$$\alpha_i \geq 0, \forall i \tag{3.23}$$

$$\alpha_i(y_i(\overrightarrow{w} \cdot \overrightarrow{x_i} + b) - 1) = 0, \forall i \tag{3.24}$$

Para o caso em que a SVM é treinada em dados não separáveis é possível encontrar uma solução utilizando a mesma abordagem que a usada no caso linear, com o uso do problema dual Lagrangiano e as condições de KKT. A diferença neste caso é a introdução de um parâmetro ξ_i que atenua a condição 3.15 apenas quando necessário, penalizando os pontos que violam as condições de margem. Assim, deve-se otimizar as condições 3.25, 3.26 e 3.27 para o caso de dados não separáveis.

$$\overrightarrow{x_i} \cdot \overrightarrow{w} + b \geq 1 - \xi_i, y_i = 1 \tag{3.25}$$

$$\overrightarrow{x_i} \cdot \overrightarrow{w} + b \leq -1 + \xi_i, y_i = -1 \tag{3.26}$$

$$\xi_i \geq 0, \forall i \tag{3.27}$$

O classificador de margem branda utiliza medidas mais robustas que a margem, como a distribuição de margens. Estas medidas toleram ruído e pontos fora da curva, considerando mais do que os pontos mais próximos à fronteira para a definição do hiperplano separador. Existem os modelos de norma-1 e norma-2, definidos pela ordem do Lagrangiano que será otimizado na busca pelo hiperplano separador [61].

3.3.2 Modelos de SVM norma-L1 e norma-L2

O modelo de norma-L1 (L1-SVM) consiste no problema de otimização das condições 3.28, 3.29 e 3.30.

$$\min 1/2 \overrightarrow{w}^T \cdot \overrightarrow{w} + C \sum \xi_i \tag{3.28}$$

$$tal\ que\ \overrightarrow{w}^T \cdot \overrightarrow{x_i} \cdot t_i \geq 1 - \xi_i, \forall i \tag{3.29}$$

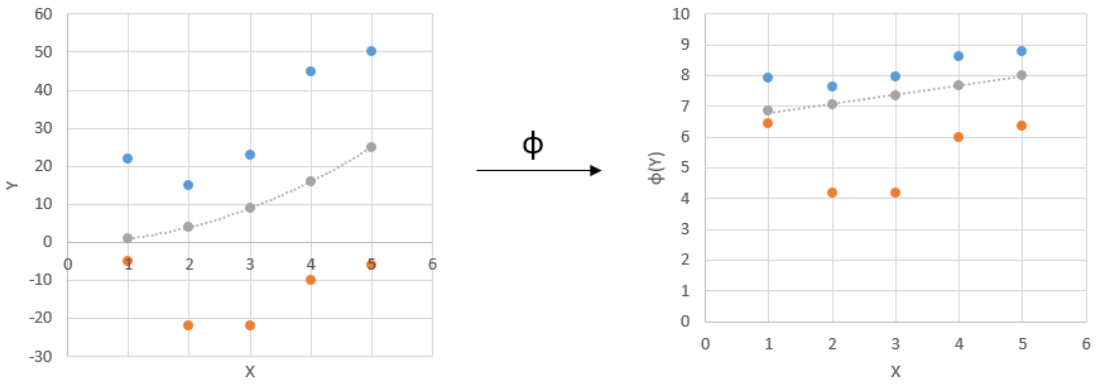
$$\xi_i \geq 0, \forall i \tag{3.30}$$

Pode-se incluir um viés unitário em todos os pontos de treinamento de forma a se obter o problema de otimização sem restrições da Equação 3.31, que também é conhecida como a forma primal da L1-SVM, com a perda de Hinge padrão.

$$\min_w \quad \frac{1}{2} \cdot \overrightarrow{w}^T \cdot \overrightarrow{w} + C \sum \max(1 - \overrightarrow{w}^T \cdot \overrightarrow{x}_n \cdot t_n, 0) \tag{3.31}$$

Como a L1-SVM não é diferenciável, a variação de norma-L2 (L2-SVM) é comumente utilizada. O problema de otimização de uma L2-SVM, a qual busca minimizar a perda de Hinge, pode ser representada

Figura 3.9: Exemplo de transformação de espaços de atributos através de um kernel.



Fonte: Elaborada pelo autor.

pela Equação 3.32, que minimiza o erro quadrático de Hinge [9]. Diferentemente da L1-SVM, a L2-SVM é diferenciável e impõe uma penalidade maior para pontos que violam a condição de margem.

$$\min_{w} \quad \frac{1}{2}\overrightarrow{w}^T \cdot \overrightarrow{w} + C\sum \max(1 - \overrightarrow{w}^T \cdot \overrightarrow{x}_n \cdot t_n, 0)^2 \tag{3.32}$$

3.3.3 Espaço de atributos introduzidos por kernels

Além de classificadores lineares pode-se obter classificadores polinomiais, de tangentes hiperbólicas e de funções radiais [43]. Isto é realizado através de funções de kernel, as quais mapeiam os dados de entrada para um espaço vetorial distinto do inicial. Isto equivale a, dado um vetor de entrada $\overrightarrow{x_i}$, utilizar uma função de kernel Φ que faz a projeção num espaço vetorial $F = \{\Phi(x)|x\epsilon X\}$.

Considerando como um exemplo em que a função $f(x) = x^2$ divide os exemplos de entrada em dois grupos distintos. Pode-se então utilizar uma função como $\Phi(x) = ln(x + 30)$ como sendo o kernel e, ao se transformar os pontos de entrada, obtêm-se um espaço vetorial de exemplos que são linearmente separáveis. A Figura 3.9 demonstra graficamente o exemplo.

No caso de SVMs, note que a formulação dual do problema depende apenas do produto escalar $\overrightarrow{x_i} \cdot \overrightarrow{x_j}$ e, aplicar uma função de kernel equivale a ter $K(\overrightarrow{x_i}, \overrightarrow{x_j}) = \phi(\overrightarrow{x_i}) \cdot \phi(\overrightarrow{x_j})$. O kernel em SVMs então é a função $K(\overrightarrow{x_i}, \overrightarrow{x_j})$, não sendo necessário conhecer ϕ explicitamente. Para que uma função de kernel possa ser usada ela deve atender às condições de Mercer, que diz que existe um mapeamento ϕ e uma expansão (Equação 3.33) se e apenas se para todo $g(x)$, a Equação 3.34 é finita então vale a Equação 3.35 [62].

$$K(x, y) = \sum \phi(x)_i \cdot \phi(y)_i \tag{3.33}$$

$$\int g(x)^2 dx \tag{3.34}$$

$$\int K(x, y) \cdot g(x) \cdot g(y) dx dy \geq 0 \tag{3.35}$$

Algumas das funções de kernel comumente utilizadas e que satisfazem as condições de Mercer são a polinomial de grau d (Equação 3.36), a de base radial Gaussiana (Equação 3.37) e exponencial (Equação 3.38) e o perceptron multicamada (Equação 3.39) [63].

$$K(x,y) = (< x,y > +1)^d \tag{3.36}$$

$$K(x,y) = exp(-\frac{||x-y||^2}{2\sigma^2}) \tag{3.37}$$

$$K(x,y) = exp(-\frac{||x-y||}{2\sigma^2}) \tag{3.38}$$

$$K(x,y) = tanh(\rho < x,y > +\varrho) \tag{3.39}$$

3.4 Função objetivo e função perda

A função objetivo, também conhecida como função de custo, é a função que se deseja minimizar ou maximizar em qualquer processo de otimização. O processo de treinamento em aprendizagem de máquina é um processo de otimização de um modelo estatístico que melhor generaliza um determinado problema. Diferentes funções de perda podem ser usadas no processo de treinamento de uma DCNN, dentre elas as baseadas em entropia cruzada e na perda de Hinge, que serão apresentadas nessa seção. Para SVMs a função de perda normalmente utilizada é a perda de Hinge.

Sendo $\mathcal{L}_i$ o erro, ou perda, para um exemplo i de treinamento e m o número de exemplos de treinamento a função objetivo, ou de custo, pode ser definida como 3.40.

$$C = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}_i \tag{3.40}$$

3.4.1 Entropia Cruzada

Uma das funções de erro, ou perda, mais conhecidas é a de erro quadrático médio, entretanto este tipo de função tem um desempenho ruim quando utilizadas em unidades ReLU ou softmax, podendo falhar em treinar um modelo de DCNN. Nesses casos pode-se utilizar a entropia cruzada, ou log-likelihood negativa. A entropia cruzada entre o valor real y e a saída do algoritmo de aprendizagem y' pode ser definida como 3.41.

$$\mathcal{L} = y \cdot log(y') + (1-y) \cdot log(1-y') \tag{3.41}$$

Pode-se também introduzir um peso w, cujo principal objetivo é balancear o treinamento em casos em que o base de dados seja desbalanceada, ou seja, a quantidade de exemplos em cada uma das classes é desigual. Nesse caso, a função perda é representada em 3.42 .

$$\mathcal{L} = -\left(y \cdot log(y') \cdot w + (1-y) \cdot log(1-y')\right) \tag{3.42}$$

Caso a classificação seja binária, w é uma constante. Caso uma classificação multiclasse w é um vetor. O parâmetro w é um multiplicador sobre o erro dos valores verdadeiros.

No caso binário, $w > 1$ força a o aumento do peso dos falsos negativos na função de custo e $w < 1$ aumento o peso dos falsos positivos na função de perda. De forma simplificada, caso existam mais exemplos negativos do que positivos, deve-se utilizar $w > 1$ e para o caso de mais exemplos positivos do que negativos utiliza-se $w < 1$. Uma forma de se obter um valor de w adequado para utilização na rede é usar a proporção entre os exemplos positivos e negativos da base.

3.4.2 Perda de Hinge

A perda de Hinge é uma função de perda usada em classificadores de margem máxima, como as SVMs. No caso binário a perda de Hinge para um único exemplo pode ser definida como 3.43 [64].

$$\mathcal{L} = \max(1 - y' \cdot y, 0) \tag{3.43}$$

Assim, dado um conjunto de valores possíveis de $y = \{0, 1\}$, quando o valor predito tem o mesmo valor que a classe real $\mathcal{L} = 0$ e quando tem valores diferentes $\mathcal{L} = 1$.

É possível estender este conceito para o caso multiclasse. Sendo$f$ o conjunto de atributos de um determinado exemplo de treinamento, $w_{(f,l)}$ como o conjunto de pesos de atributos de para a saída correta l pode-se definir a perda de Hinge multiclasse para um exemplo de treinamento como 3.44 [65].

$$\mathcal{L} = \max(1 + \max_{l' \neq l} \sum w_{(f,l')} - \sum w_{(f,l)}, 0) \tag{3.44}$$

3.4.3 SVM como camada de classificação de redes neurais

O uso de SVMs em conjunto com redes convolucionais pode ser feito como um processo de múltiplas camadas. Uma DCNN pode ser treinada com o intuito de aprender uma representação interna do problema e esta representação pode ser usada posteriormente como entrada no treinamento de uma SVM linear. Este tipo de abordagem no geral melhora o desempenho do modelo mas os atributos de camadas anteriores da DCNN não são otimizadas com relação à função objetivo da SVM.

Tang em seu artigo [9] demonstrou que o uso de uma camada de classificação SVM ao invés da softmax é benéfica, mostrando resultados melhores na MNIST, CIFAR-10 e reconhecimento de faces. Essa abordagem otimiza o problema primal da SVM e os gradientes podem ser usados no algoritmo de back propagation para aprender os atributos de mais baixo nível, corrigindo o problema de abordagens que usam a DCNN e a SVM em separado. Trabalhos anteriores já haviam estudado esta abordagem mas com o erro de Hinge ao invés de usar do erro da L2-SVM. O erro da L2-SVM, diferentemente do erro de Hinge, é diferenciável e dá uma maior penalidade à erros. Nomeada como aprendizagem profunda usando uma L2-SVM (DLSVM), ela obteve um erro de 0,87% contra 0,99% da camada softmax no MNIST e de 11,9% contra 14,0% no CIFAR-10. Além disso, nos experimentos do artigo de referência a L2-SVM teve um desempenho ligeiramente superior ao da L1-SVM.

Não há um entendimento total sobre qual o motivo do melhor desempenho, mas acredita-se que se deva mais à regularização superior do que ao resultado de uma melhor otimização de parâmetros.

Tomando como ponto de partida a função perda do problema de otimização de uma L2-SVM da Equação 3.32 e substituindo o vetor de entrada x pelo penúltimo vetor de ativações h e diferenciando em relação a este vetor de ativações obtêm-se a Equação 3.45.

$$\frac{\partial l(w)}{\partial h_n} = -2 \cdot C \cdot t_n \cdot w(\max(1 - w^T \cdot h_n \cdot t_n, 0)) \tag{3.45}$$

Utilizando-se dessa equação o algoritmo de back propagation pode ser utilizado da mesma forma que com camadas de ativação baseadas em softmax [9]. Numa camada de saída com k neurônios a classe classificada é dada pela Equação 3.46, onde$a_k(x)$ é o valor da ativação do k-ésimo neurônio da camadas.

$$\arg\max_k a_k(x) \tag{3.46}$$

3.5 Transferência de Conhecimento

A transferência de conhecimento é o nome dado ao conjunto de métodos de aprendizagem de máquina que visam aprimorar o aprendizado em uma *tarefa objetivo* a partir do conhecimento que foi aprendido em uma ou mais *tarefas origem* relacionadas. O fluxo de informação nestes algoritmos é unidirecional, as tarefas origem não possuem nenhum conhecimento a respeito da tarefa objetivo que receberá as informações. Métodos de transferência tendem a ser fortemente acoplados com os algoritmos de aprendizagem de máquina usados para aprender estas tarefas, podendo ser considerados como extensões destes algoritmos. Os principais algoritmos de transferência de conhecimento se concentram nos de aprendizagem indutiva – redes neurais, redes Bayesianas e redes de Markov – e em aprendizagem por reforço [66].

O principal caso de uso de transferência de conhecimento ocorre quando há uma quantidade limitada de dados para a tarefa objetivo. Com a criação de bancos de dados cada vez maiores, como no caso de classificação de objetos em imagens e o banco do ILSVRC Object Classification Challenge, o uso de um banco de dados relacionado ao da tarefa objetivo se torna uma solução atraente.

O impacto da transferência de conhecimento num dado problema pode ser medido de três formas ao longo da fase de treinamento: o desempenho inicial do modelo, o tempo de treinamento até a estabilização do treinamento e o desempenho final do modelo. Se a transferência de conhecimento leva a uma piora do desempenho do modelo diz-se que uma *transferência negativa* ocorreu.

Seguindo a notação adotada em [67, 68] pode-se definir um domínio $\mathcal{D}$, que é composto por um espaço de atributos $\mathcal{X}$ e uma distribuição de probabilidades marginal $P(X), X = \{x_1, ..., x_n\}$, onde X é um exemplo particular que pertence a $\mathcal{X}$. Dado o domínio, uma tarefa $\mathcal{T}$ como uma função de predição f e um espaço de classes $\mathcal{Y}$. A função de predição é aprendida através de exemplos de treinamento $\{X_i, y_i\}, X_i \epsilon \mathcal{X}, y_i \epsilon \mathcal{Y}$. Na classificação binária de fotografias, X_i é uma fotografia do banco de dados e y_i a respectiva classe, esteticamente boa ou não.

Sendo a tarefa origem T_S e a tarefa objetivo T_T, pode-se definir f_S a função de predição de tarefa origem,$\mathcal{D}_S = \{(X_{S1}, Y_{S1}), ..., (X_{Sn}, Y_{Sn})\}$ como o domínio da tarefa origem, f_T a função objetivo e $\mathcal{D}_T = \{(X_{T1}, Y_{T1}), ..., (X_{Tn}, Y_{Tn})\}$ o domínio da tarefa objetivo. A função objetivo que representa o classificador final ao término da transferência de conhecimento.

Assim, formalmente define-se a transferência de conhecimento de um domínio $\mathcal{D}_S$ de uma determinada tarefa T_S para uma tarefa T_T em um domínio $\mathcal{D}_T$ como a melhoria da função f_T usando informações da tarefa e domínio origem, com $\mathcal{D}_S \neq \mathcal{D}_T$ ou $T_S \neq T_T$.

Para o caso de transferência de conhecimento com DCNNs, inicialmente é feito o treinamento da rede em uma tarefa origem e então um segundo treinamento é feito na tarefa objetivo, mantendo-se os parâmetros da rede pré-treinada como o estado inicial do segundo treinamento, com uma taxa de aprendizado menor do que a taxa padrão. É possível fazer o treinamento apenas da última camada, parcialmente de algumas camadas, ou de todas as camadas da rede: no caso em que se treina os pesos da última camada os parâmetros da tarefa origem são usados diretamente no classificador modelado pela última camada; no caso em que se treina todos os parâmetros da rede a idéia é usar a tarefa origem como um ponto de partida para que os atributos ótimos para a tarefa objetivo possam ser encontrados.

3.5.1 Classificações de transferência de conhecimento

Existem diferentes espectros pelos quais é possível fazer a classificação de um problema de transferência de conhecimento tais como a existência de dados anotados, a relação entre os domínios de atributos origem/destino e o modo como o problema de transferência será resolvido. Tais definições serão apresentadas nesta sessão.

Primeiramente, pode-se classificar como a transferência de conhecimento homogênea problemas em

que $\mathcal{X}_S = \mathcal{X}_T$ enquanto que na transferência de conhecimento heterogênea os atributos da origem é diferente do espaço de atributos da tarefa objetivo [67].

Outra forma de classificar problemas de transferência de conhecimento é baseada em diferentes situações que podem existir entre as tarefas e domínios destino e origem e a existência de dados anotados [68]. Neste caso a transferência de conhecimento pode ser classificada como indutiva, não-supervisionada e transdutiva. Na transferência de conhecimento indutiva $T_S \neq T_T$. Neste caso os dados da tarefa destino induzem o modelo à obter a função objetivo f_T. A transferência de conhecimento não-supervisionada é similar à transferência de conhecimento indutiva mas com foco em tarefas de aprendizagem não supervisionada tal como a clusterização. Neste caso não existem dados anotados nem no domínio origem e nem no domínio objetivo. Finalmente, na transferência de conhecimento transdutiva $\mathcal{D}_S \neq \mathcal{D}_T$. Neste caso muitos dados anotados existem no domínio origem mas nenhum dado anotado existe no domínio objetivo. Pode-se ainda observar neste caso situações em que $\mathcal{X}_S \neq \mathcal{X}_T$ ou $\mathcal{X}_S = \mathcal{X}_T$ mas $P(X_S) \neq P(X_T)$.

Com relação à forma como será resolvida a forma de transferência de conhecimento pode-se ter a transferência de dados, de representação de atributos, de parâmetros e de conhecimento relacionado [68]. O objetivo na transferência de dados é a reutilização de alguns dos dados do domínio origem no domínio destino. Na transferência de representação a busca por uma representação que diminua a diferença entre os domínios destino e origem é o que é buscado. No caso da transferência de parâmetros deve-se descobrir parâmetros e hipóteses relacionadas entre as tarefas. Na transferência de conhecimento relacionado o objetivo é a construção de um mapa entre o conhecimento do domínio origem e o domínio destino.

A transferência de conhecimento heterogênea os espaços de atributos origem e objetivo são distintos [67]. Existem duas formas principais para a resolução de problemas relacionados à diferença de espaço de atributos: a transformação simétrica e a transformação assimétrica. Na transformação simétrica ambos os espaços de origem e objetivo são transformados em um espaço de atributos comum, unificando os domínios, enquanto que na assimétrica o espaço de atributos origem é transformado no espaço de atributos objetivo.

Um ponto importante é o fato de que os exemplos dos espaços origem e objetivo tem o mesmo domínio, que é o de imagens e fotografias. Desta forma pode-se assumir que não existem diferenças significativas entre os domínios e assim que as diferenças entre os espaços de atributos é resolvida não é necessária mas nenhuma adaptação de domínio.

3.5.2 Transferência negativa

O principal objetivo da aplicação da transferência de conhecimento é melhorar o desempenho do modelo na tarefa objetivo, entretanto quando o desempenho do modelo final é impactada negativamente pela transferência de conhecimento diz-se que uma transferência negativa ocorreu. O principal motivo deste resultado é a falta de relacionamento entre os domínios origem e objetivo [67].

Esta falta de relacionamento pode ser de forma global, ou seja, todo o conjunto de dados da tarefa origem é pouco relacionada à tarefa objetivo. Ela também pode se apresentar de forma pontual, como em casos de big data ou de múltiplas origens de dados, em que apenas alguns dos conjuntos de dados são relevantes ao problema de transferência enquanto que outros impactam negativamente o aprendizado.

Uma das causas apontadas como raiz deste problema é a diferença entre as distribuições condicionais de domínios origens e a diferença de distribuição de classes entre as tarefas origem e destino [67].

Uma das formas de se evitar a transferência negativa é dando prioridade aos conjuntos de dados que são mais relacionados à tarefa objetivo, evitando usar conjuntos de dados com baixa relevância para a tarefa objetivo. Entretanto, a maioria das abordagens de transferência de conhecimento não leva em consideração este tipo de impacto uma vez que a forma de mensurar a transferência negativa é difícil de definir além de poder levar a um overfitting na tarefa objetivo uma vez que se limita a quantidade de

informação cujo conhecimento é transferido. Ainda é uma área que precisa de mais pesquisa para que sua aplicação se torne relevante [67].

4. Sistema Proposto para Classificação Estética de Fotografias

Propõe-se, para a resolução do problema de classificação estética de fotografias, a utilização de um algoritmo de aprendizado de máquina composto por uma DCNN e uma SVM, numa abordagem inspirada no modelo proposto por Tang em [9] denominada DLSVM. O escolha do uso da DLSVM ao invés de uma camada de saída softmax é por conta de seu melhor resultado observado em outros problemas de classificação de imagens.

Esta proposta tem como base propostas já apresentadas na literatura, DCNN e SVM, entretanto, dentre os trabalhos revisados, ainda não existe uma abordagem que faça uso de uma DLSVM para a classificação estética de fotografias. Assim, a abordagem é inédita e a avaliação do desempenho desta abordagem na classificação estética de imagens é a principal contribuição deste trabalho.

Com o objetivo de se utilizar ao máximo do conhecimento já existente sobre DCNNs para melhorar o desempenho do classificador estético este trabalho também propõe a utilização de arquiteturas de DCNN que possuem um bom desempenho na tarefa de classificação de objetos através da transferência de conhecimento. Assim, foram analisadas diferentes arquiteturas treinadas no ILSVRC Object Classification Challenge e a transferabilidade do conhecimento entre o problema de classificação de objetos e classificação estética.

Para a resolução de qualquer problema de classificação por aprendizado de máquina é necessário definir o banco de dados para testes e treinamento, além do tipo e da estrutura do modelo de aprendizado em si. No caso de classificação estética foi necessário escolher primeiramente um banco de dados de fotografias previamente classificadas quanto à sua estética. Além disso, como o modelo de aprendizagem envolve DCNNs, foi necessário definir a estrutura e os hiperparâmetros de forma a se obter os melhores descritores estéticos genéricos possíveis. Inicialmente algumas DCNNs foram criadas do zero com o intuito de validar a possibilidade de sua utilização para a classificação estética e para que se pudesse aprofundar o conhecimento sobre as camadas que compõem a rede. Na proposta final DCNNs com arquiteturas já propostas em outros trabalhos na literatura foram utilizadas.

É importante destacar que como a classificação é de imagens, requisitos maiores de computação são necessários para que o treinamento seja feito em um curto período de tempo devido ao grande volume de dados que deve ser analisado. Para isso, provisionar uma infraestrutura na nuvem é essencial para a solução do problema.

As seções deste capítulo descrevem o banco de dados e os modelos de aprendizagem de máquina que foram utilizadas para a solução do problema de classificação estética.

4.1 Banco de dados

A base de dados AVA (Aesthetic Visual Analysis) [40] é uma base de dados voltada para projetos de classificação estética de imagens que contém aproximadamente 255.000 imagens, cada qual com três tipos

Figura 4.1: Três primeiras linhas do arquivo de anotações AVA.txt.

```
1 953619 0 1 5 17 38 36 15 6 5 1 1 22 1396
2 953958 10 7 15 26 26 21 10 8 1 2 1 21 1396
3 954184 0 0 4 8 41 56 10 3 4 0 0 0 1396
```

Fonte: Naila Murray, Luca Marchesotti e Florent Perronnin [40].

Figura 4.2: Três primeiras linhas do arquivo de anotações tags.txt.

```
1 Abstract
24 Action
31 Advertisement
```

Fonte: Naila Murray, Luca Marchesotti e Florent Perronnin [40].

de anotação: notas dadas por usuários numa escala de 1 a 10, categoria semântica (natureza por exemplo) e anotação de estilo (que corresponde à um desafio, como por exemplo silhuetas). A coleção de imagens foi coletada a partir dos dados, tanto com relação às fotos propriamente ditas, quanto com relações às notas e categorias/tipos de competição, de competições de fotografias obtidos no site `www.dpchallenge.com`.

O AVA dataset é composto por três arquivos de metadados. O primeiro, "AVA.txt", possui as informações das notas, o identificador da imagem no AVA, o identificador do arquivo no `www.dpchallenge.com`, o identificador das categorias semânticas e identificador do desafio no qual a fotografia participou. O segundo arquivo, "tags.txt", é o relacionamento entre o identificador da categoria semântica e o seu nome. O terceiro arquivo, "challenges.txt", é a relação entre o identificador do desafio e o seu nome. As três primeiras linhas de cada um dos arquivos de metadados podem ser observados nas Figuras 4.1, 4.2 e 4.3 respectivamente. É possível visualizar estes arquivos como um diagrama entidade relacionamento, como pode ser observado na Figura 4.4.

No banco de dados original as fotografias não são fornecidas, apenas o identificador das mesmas. Para acessar a fotografia é necessário obtê-la diretamente do site, usando `https://www.dpchallenge.com/image.php?IMAGE_ID=<Identificadordaimage>` para fazer o download da imagem. Existem alguns sites de bancos de dados acadêmicos que fornecem o banco de dados na íntegra, com as imagens já compiladas para uso.

Comparando-se o banco de dados AVA com outros bancos como o Photo.net [69] e o CUHK [11], nota-se que o primeiro possui a maior quantidade de imagens e tais imagens já estão devidamente anotadas com relação à sua classificação estética baseada no feedback dos usuários do site `www.dpchallenge.com`. Os outros bancos citados possuem uma quantidade inferior de informações e nem todas as meta-descrições que são fornecidas no AVA. Em [40], é feita uma comparação mais detalhada da base de dados AVA com outros bancos, mostrando a superioridade em diversos aspectos com relação a eles.

4.1.1 Análise do conteúdo da base

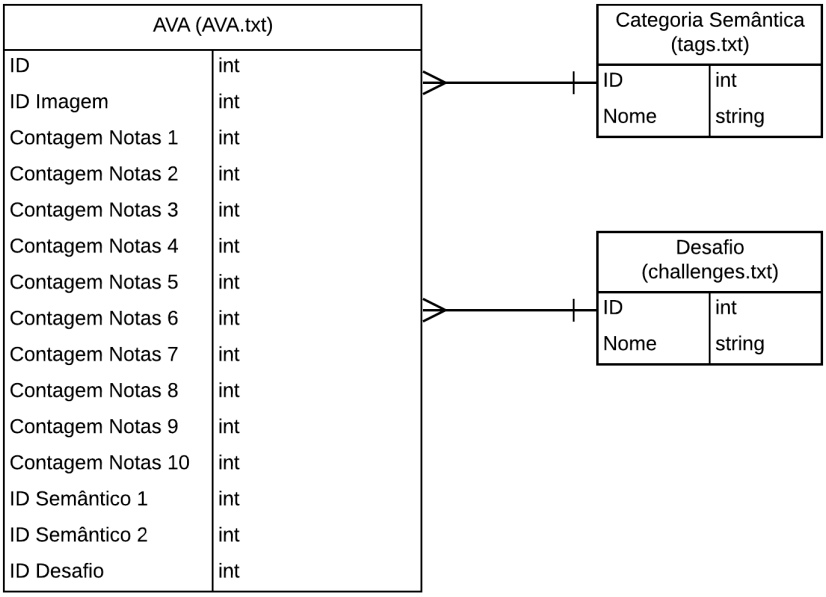
Com o intuito de extrair um conjunto de dados para análises preliminares da base de dados AVA, diminuindo o tempo de processamento e agilizando as etapas de análise de parâmetros iniciais, alguns estudos foram realizados sobre os dados fornecidos pelo banco. Utilizar o banco de dados inteiro implicaria não só em um tempo de processamento maior, mas também em apenas um ligeiro ganho de desempenho de

Figura 4.3: Três primeiras linhas do arquivo de anotações challenges.txt.

```
1396 100_Meters
1004 100_Years_Old
1329 100_Years_Old_II
```

Fonte: Naila Murray, Luca Marchesotti e Florent Perronnin [40].

Figura 4.4: Diagrama de entidade-relacionamento para o AVA Dataset.



Fonte: Elaborada pelo autor.

Tabela 4.1: Contagem dos cinco desafios com maior número de fotografias.

Identificador do desafio	Contagem
1005	1108
430	828
616	704
536	678
707	646

Fonte: Elaborada pelo autor.

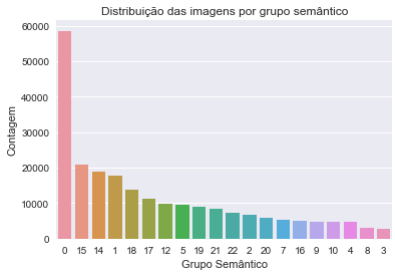
classificação, sendo, portanto, altamente não recomendado.

Primeiramente analisou-se a distribuição das imagens com relação ao identificador do seu desafio relacionado. No total, existem 1398 desafios distintos contabilizados no banco. A Tabela 4.1 mostra os cinco desafios com a maior contagem de fotografias. Note que o desafio com maior fotografias possui 1108 fotografias distintas. Como este número não é muito elevado, optou-se por analisar as fotografias do banco com relação ao seu grupo semântico.

Agrupando-se então as fotografias por grupo semântico obteve-se a Figura 4.5. Neste atributo, 0 é o grupo semântico que indica a inexistência de uma categoria pré-definida, grupo este com o maior número de fotografias. Os próximos grupos semânticos com maior número de fotografias são 15, natureza, 14 paisagens e 1 abstrato.

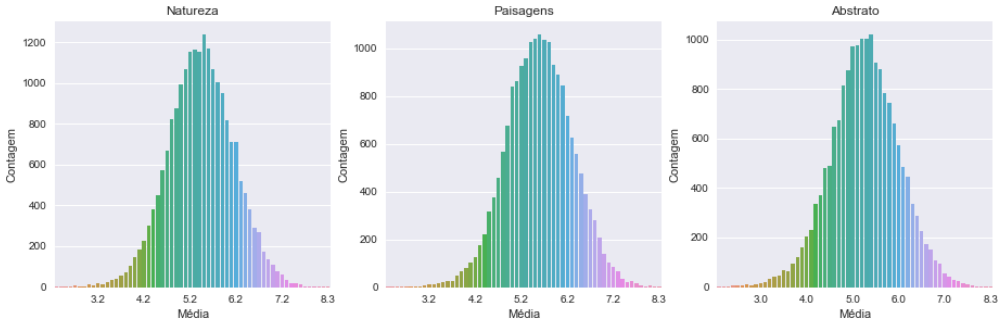
Utilizando-se estas três categorias, uma segunda análise foi realizada: avaliar a distribuição da média das notas por toda categoria. A Figura 4.6 mostra um histograma das notas médias de cada fotografia com uma casa decimal de precisão. Um primeiro ponto a se notar é a distribuição normal das notas, que se repete independentemente das categorias. Isso faz possível que a escolha dos extremos das distribuições garantirá a obtenção de dois grupos distintos. Utilizar as categorias de forma completa, apesar de elevar o número de exemplos, insere no treinamento exemplos de categorização duvidosa que podem diminuir a capacidade de discernimento final do classificador. Assim, uma possível alternativa para se utilizar as imagens dessas categorias se resume a utilizar os extremos das distribuições como representantes dos

Figura 4.5: Distribuição com os 20 grupos semânticos com mais fotografias.



Fonte: Elaborada pelo autor.

Figura 4.6: Distribuição de médias nas três categorias estudadas.



Fonte: Elaborada pelo autor.

grupos de imagens esteticamente boas e ruins, usando, por exemplo, as imagens cuja média está a um desvio padrão da média.

Outro ponto importante é a diferença das médias entre categorias. Assim, caso as categorias sejam utilizadas de forma independente, pode-se garantir a obtenção de grupos de imagens de teste e treinamento mais consistentes.

4.1.2 Serialização do conteúdo

Por se tratar de um problema relacionado a imagens, que demandam uma quantidade grande de memória durante o processamento, foi necessário desenvolver uma forma de utilizar o banco que não sobrecarregasse a memória RAM do computador utilizado e tivesse o menor impacto no tempo de treinamento.

A solução encontrada foi a utilização do formato TFRecord, o qual é o formato recomendado para utilização com o Tensorflow. O TFRecord é um formato de arquivo que guarda uma sequência de strings binárias de tal forma que o acesso não pode ser feito de forma aleatória, sendo assim adequado em situações em que é necessário realizar o streaming de uma quantidade grande de dados.

Como o TFRecord é apenas uma sequência de strings binários, é possível armazenar em conjunto com os pixels da imagem metadados relacionados. No caso deste projeto foram armazenados a classe à qual pertence a imagem e a imagem em si.

Os dois últimos atributos são necessários para fazer a reconstrução da imagem posteriormente. Arquivos do tipo TFRecord podem ser fragmentados, o que auxilia no armazenamento caso a quantidade de informações armazenadas nestes arquivos seja muito grande.

4.2 Modelos de aprendizagem de máquina utilizados neste projeto

Alguns modelos diferentes foram usados para solucionar o problema de classificação estética com aprendizado de máquina neste projeto. Os primeiros modelos foram compostos por uma rede neural apenas, para validar se redes neurais poderiam ser utilizadas para generalizar o problema antes que otimizações fossem aplicadas. Uma segunda abordagem foi feita usando a rede Inception, transferindo o conhecimento de atributos da ILSVRC Object Classification Challenge e retreinando toda a rede. Finalmente, a abordagem final é baseada numa DLSVM também transferindo o conhecimento de atributos do ILSVRC Object Classification Challenge, usando diferentes arquiteturas de DCNNs como base. O mesmo modelo da abordagem final mas com uma camada de saída softmax também foi utilizado para que as melhorias do uso de uma DLSVM ao invés de uma camada softmax pudessem ser mensuradas.

4.2.1 Modelos de redes neurais usados

Os modelos iniciais utilizados foram compostos apenas de redes neurais. O intuito deste modelo foi a validação da utilização de DCNNs e para estudos sobre as camadas que compõem a rede.

O bloco básico que forma uma unidade convolucional no modelo é composta por duas camadas: uma camada de normalização de lote, uma camada convolucional. Em algumas das camadas utiliza-se também o dropout e a subamostragem após a aplicação da camada convolucional.

Com relação as kernels escolhidos, inicia-se a rede com convoluções com atributos de kernel de janela menor (3×3) e, para camadas posteriores, aumenta-se o tamanho deste kernel para 5×5, 7×7 e 9×9. A ideia é utilizar as camadas anteriores para atributos de mais baixo nível, tais como bordas e cantos. As camadas posteriores compõem os atributos de mais alto nível. Quanto mais posterior a camada, mais alto o nível representado pelo atributo.

A subamostragem reduz a largura da imagem com o passar das camadas. Todas as subamostragens utilizadas são de janela 2×2, ou seja, reduzem as dimensões horizontal e vertical da imagem por dois e, consequentemente, o tamanho da imagem em um quarto.

A camada de dropout é utilizada no final da rede, com o intuito de ajudar na regularização dos parâmetros aprendidos pela camada que contém os atributos finais.

Duas redes iniciais foram criadas, uma com quatro e outra com seis camadas convolucionais. Ambas possuem duas camadas completamente conectadas. Um esquema da rede com quatro camadas pode ser observado na Figura 4.7; a rede com seis camadas tem uma estrutura muito semelhante, adicionando apenas mais dois conjuntos de camadas de convolução e normalização de lote.

Para a rede de 4 camadas a estrutura usada foi a seguinte:

- Convolução 3×3 com 32 neurônios
- Max pooling 2×2 e passo 2
- Convolução 5×5 com 64 neurônios
- Max pooling 2×2 e passo 2
- Duas convoluções 7×7 com 64 neurônios
- Max pooling 2×2 e passo 2
- Camada completamente conectada com 1024 neurônios
- Dropout de 0,4

- Camada completamente conectada com 2 neurônios

Para a rede de 6 camadas a estrutura usada foi a seguinte:

- Convolução 3×3 com 32 neurônios
- Max pooling 2×2 e passo 2
- Convolução 5×5 com 64 neurônios
- Convolução 5×5 com 32 neurônios
- Max pooling 2×2 e passo 2
- Duas convoluções 7×7 com 64 neurônios
- Max pooling 2×2 e passo 2
- Convolução 9×9 com 64 neurônios
- Max pooling 2×2 e passo 2
- Camada completamente conectada com 1024 neurônios
- Dropout de 0,4
- Camada completamente conectada com 2 neurônios

4.2.2 Modelo usando DLSVM

O modelo com DLSVM proposto é baseado na abordagem do Tang [9], usando uma DCNN com a perda da L2-SVM como função obtido como apresentado na subseção 3.4.2.

Levando-se em conta que o problema de classificação é binário a última camada desta arquitetura é uma camada com dois neurônios completamente conectada, substituindo uma camada completamente conectada de 1000 neurônios que seria comumente utilizada na classificação de objetos. Cada um dos dois neurônios representa uma das classes, utilizando-se sempre uma saída hot-encoded. Enquanto a DCNN extrai os atributos da imagem, a camada de dois neurônios é responsável pela classificação, usando estes atributos gerados pela DCNN e tendo como saída a classe da imagem.

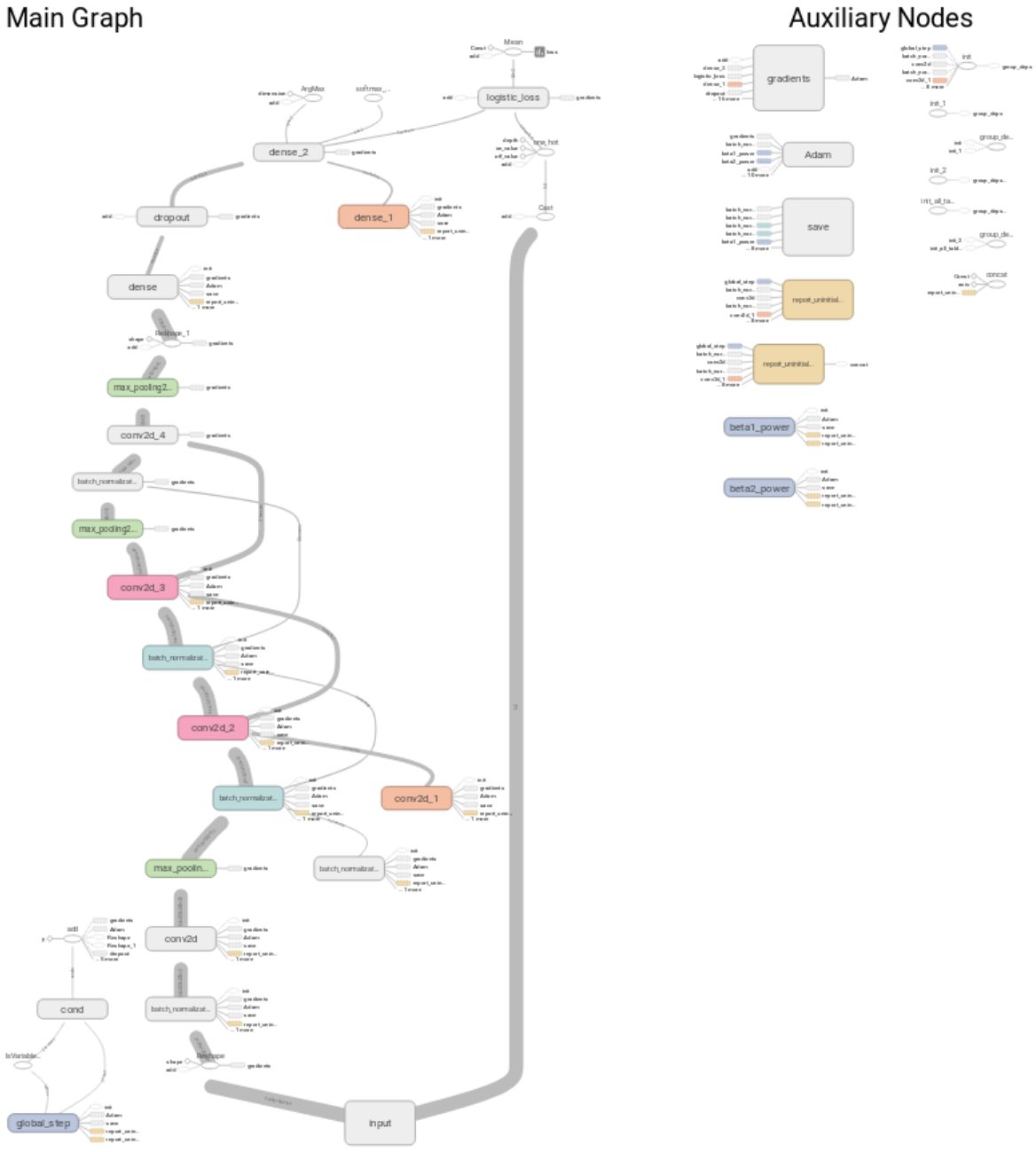
Diferentes arquiteturas de DCNNs foram testadas, todas elas com os atributos previamente treinados com no ILSVRC Object Classification Challenge. Para que fosse possível comprar os resultados dos experimentos com as diferentes arquiteturas de DCNN, entre os experimentos apenas a arquitetura foi alterada, mantendo os hiperparâmetros, a função de custo e a camada de saída inalterados.

Um modelo similar usando softmax e gradiente estocástico descendente ao invés da função de custo da DLSVM foi também utilizada para que seus resultados pudessem servir como base de comparação.

Como será utilizada transferência de conhecimento para adaptar um rede previamente treinada para classificação de objetos para a nova tarefa de classificação estética é necessário definir o problema formalmente. Para o caso da transferência de conhecimento para a classificação estética de fotografias a partir dos atributos do ILSVRC Object Classification Challenge pode-se dizer que:

- $T_S \neq T_T$: a classificação binária estética é diferente de uma classificação de objetos, que é multi-classe. Mesmo que as tarefas sejam distintas elas são relacionadas uma vez que ambas se referem a classificações de imagens;
- $\mathcal{D}_S = \mathcal{D}_T$: como ambos os datasets que serão usados são datasets de imagens pode-se considerar os domínios iguais;

Figura 4.7: Estrutura de rede neural profunda criada para o modelo inicial.



Fonte: Elaborada pelo autor.

- $\mathcal{X}_S \neq \mathcal{X}_T$: o problema estético é binário enquanto o do ILSVRC é um problema multiclasse. Logo o espaço de atributos é distinto.

Portanto, pode-se definir a transferência de conhecimento para a classificação estética de fotografias a partir dos atributos do ILSVRC Object Classification Challenge como uma transferência de conhecimento heterogênea, indutiva e de parâmetros. Para que essa definição fosse possível foi necessário assumir a hipótese de que os domínios dos dois problemas são suficientemente semelhantes.

5. Implementação e Experimentos

Para a construção dos modelos iniciais de testes de DCNNs optou-se pela utilização do Tensorflow, biblioteca de aprendizagem de máquina baseada em tensores e focada em redes neurais profundas. Ela já conta com diversas ferramentas que auxiliam na construção da estrutura da rede neural e ajudam nas etapas de treinamento e testes. A linguagem de programação escolhida foi o Python, uma linguagem já estabelecida para problemas de aprendizagem de máquina e que possui suporte ao Tensorflow.

O sistema de final foi implementado usando o framework Keras, que é um framework de alto nível escrito em Python e que usa o Tensorflow como base de suas computações. O principal diferencial com relação ao Tensorflow é a sua facilidade de uso, uma vez que o Keras possui diversas funções de alto nível que operam sobre funções de mais baixo nível disponíveis no Tensorflow. Todas as funcionalidades existentes no Tensorflow podem acessadas através do Keras.

Também foi utilizado o serviço CometML com o intuito de armazenar o histórico de todos os testes que foram feitos com as diferentes configurações de rede. Para cada um dos experimentos com DCNN foi criado um experimento no CometML, que armazena as informações do código usado, dos hiperparâmetros utilizados, da saída do programa ao longo do tempo e dos dados de controle do experimento, tais como a acurácia e perda de treinamento e validação.

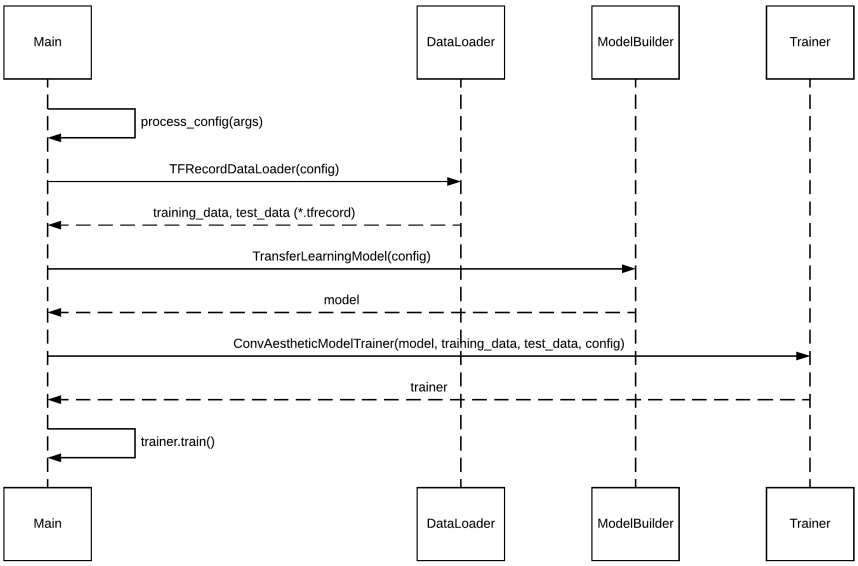
Para a implementação foram considerados dois fluxos principais: treinamento, representado na Figura 5.1, e testes, representado na Figura 5.2. Ambos os fluxos se iniciam a partir do processamento de um arquivo JSON de configuração, que contém as informações de arquivos para carregar o banco de dados, hiperparâmetros para o modelo, tags e metadados. Um exemplo deste arquivo de configuração se encontra no Algoritmo 2. Dadas estas configurações são carregados os dados de treinamento e teste, no caso do fluxo de treinamento, e validação, no caso de validação, e o modelo é criado com base nos hiperparâmetros deste mesmo arquivo.

Finalmente, no caso de treinamento, uma entidade de treinamento é criada, também com parâmetros baseados no arquivo de configuração. Esta será a entidade responsável por treinar o modelo e salvar os resultados obtidos e parâmetros da rede ao longo do tempo. Para o caso de validação, os pesos da DCNN são carregados no modelo e então o modelo é validado, tendo como saída informações como a taxa de acurácia e a matriz de confusão do teste.

5.1 Infraestrutura de aprendizagem

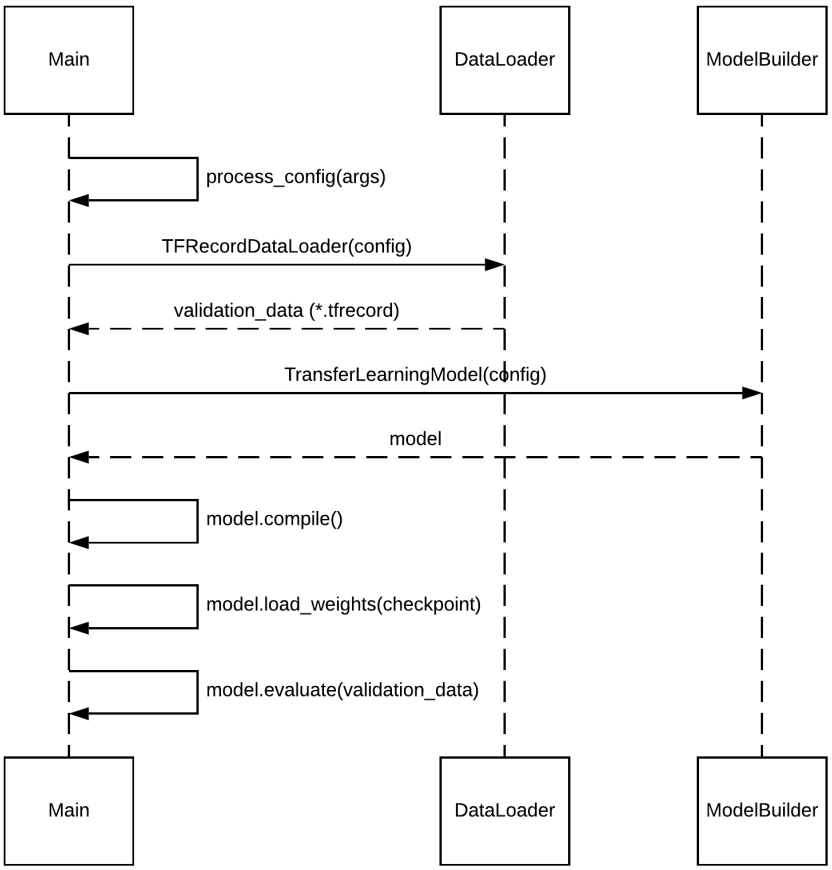
Processos de aprendizagem de máquina requerem tanto poder de processamento quanto memória volátil para armazenamento rápido de variáveis. Em ambientes locais de desenvolvimento nem sempre é possível ter computadores com um poder adequado para o treinamento destes tipos de modelo, ficando-se assim, restrito a modelos mais simplificados. Estes problemas ficam ainda mais aparentes quando se está lidando-se com imagens, pois a quantidade de informações contidas em um único exemplo é ainda maior. Considerando-se, por exemplo, uma imagem de 400×400 tem-se 480.000 valores por imagem, considerando uma imagem com 3 canais (RGB).

Figura 5.1: Diagrama sequencial de treinamento da DCNN.



Fonte: Elaborada pelo autor.

Figura 5.2: Diagrama sequencial de testes da DCNN.

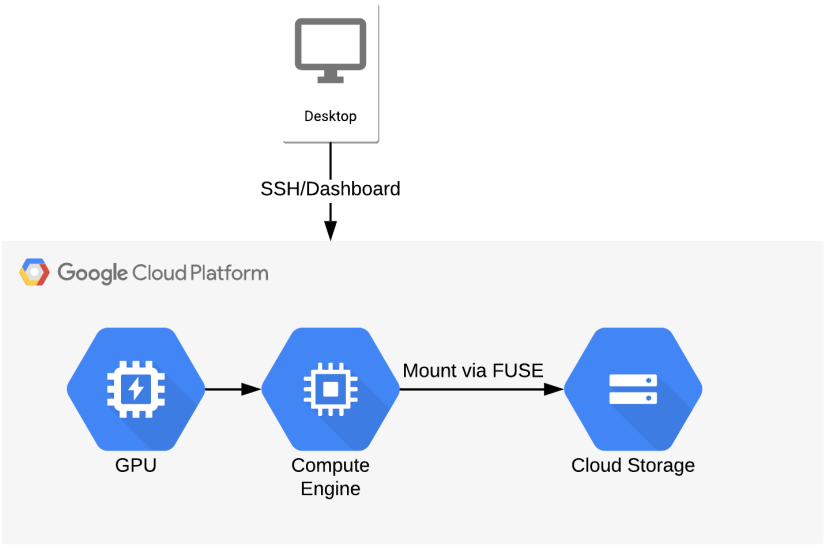


Fonte: Elaborada pelo autor.

Algoritmo 2 Exemplo de arquivo de configuração para o programa de treinamento e testes de DCNNs.

```
{
    "exp":{
        "name":"conv_aesthetic_inceptionV2resnet"
    },
    "model": {
        "learning_rate": 0.0001,
        "optimizer": "adam"
    },
    "loader": {
        "training_records": [
                    "ava_training_01of04.tfrecord",
                    "ava_training_02of04.tfrecord",
                    "ava_training_03of04.tfrecord",
                    "ava_training_04of04.tfrecord"
                ],
        "val_records": [
                    "ava_dev_01of04.tfrecord",
                    "ava_dev_02of04.tfrecord",
                    "ava_dev_03of04.tfrecord",
                    "ava_dev_04of04.tfrecord"
                ],
        "test_records": [
                    "ava_test_01of04.tfrecord",
                    "ava_test_02of04.tfrecord",
                    "ava_test_03of04.tfrecord",
                    "ava_test_04of04.tfrecord"
                ],
        "shuffle_buffer": 100,
        "batch_size": 50,
        "num_classes": 2,
        "image_size": [-1, 224, 224, 3]
    },
    "trainer":{
        "num_epochs": 20,
        "verbose_training": true
    },
    "callbacks": {
        "checkpoint_monitor": "val_loss",
        "checkpoint_mode": "min",
        "checkpoint_save_best_only": true,
        "checkpoint_save_weights_only": true,
        "checkpoint_verbose": true,
        "tensorboard_write_graph": true,
        "checkpoint_dir": "./model"
    },
    "comet_api_key": "abc",
    "exp_name": "aesthetic",
    "tags": ["inceptionv2resnet", "hinge", "transfer"],
    "checkpoint_dir": "./conv_aesthetic_nasnetmobile/checkpoints/"
}
```

Figura 5.3: Estruturação dos serviços do GCP utilizados.



Fonte: Elaborada pelo autor.

Assim existe a necessidade de provisionar um ambiente em que um poder de processamento mais poderoso possa ser utilizado. A solução é a utilização de computação na nuvem. Existem diversos serviços que possibilitam o provisionamento de máquinas virtuais que têm acesso a memória e placas gráficas apropriadas para processos de treinamento de modelos de aprendizado de máquina. Alguns destes serviços são o Amazon Web Services (AWS) e o Google Cloud Plataform (GCP). Por uma questão de existência de fundos gratuitos para testes e a existência de hardware especializado para o treinamento via Tensorflow, foi escolhida a plataforma do Google.

5.1.1 Estruturação dos serviços utilizados

O modelo com a arquitetura de nuvem utilizada pode ser vista na Figura 5.3. Existem dois serviços que devem ser provisionados para que um treinamento de aprendizagem de máquina possa ser feito: um de armazenamento e um de computação. Para o armazenamento foi criado um bucket do Google Storage. A função deste bucket é armazenar tanto o banco de dados quanto os modelos treinados. Os bancos de dados foram criados em um ambiente local no formato TFRecord e foram então enviados para este bucket. Estes arquivos então podem ser acessados por qualquer máquina com o link para o bucket, inclusive máquinas na nuvem.

Já com relação à computação, foi criada uma máquina na nuvem através do serviço Google Compute Engine. A máquina provisionada possui 4 CPUs Intel Broadwell, 16GB de memória, uma GPU Nvidia Tesla K80 com 11GB de memória de GPU. Estes atributos podem ser alterados com facilidade caso necessário, tornando-a muito flexível. Com relação ao sistema operacional foi instalado um Ubuntu 16.04 e todas os requisitos foram instalados na máquina manualmente. Foi necessário também montar o bucket do Google Storage via FUSE, para que os arquivos de banco de dados e de modelo pudessem ser manipulados de forma mais eficiente.

5.2 Experimentos

Uma série de experimentos foi realizada com o intuito de se estudar o melhor modelo para classificação estética. Os primeiros modelos foram feitos usando redes neurais convolucionais com o intuito de estudar a aplicabilidade deste tipo de modelo para o problema em questão. Em seguida foi feito um experimento para avaliar a aplicação de DCNNs e transferência de conhecimento, usando como base dados do ILRSVC Object Classification Challenge. Finalmente, os modelos finais foram feitos usando diferentes arquiteturas de DCNNs e camadas de saída, também usando como base o ILRSVC Object Classification Challenge.

A maior parte dos resultados será apresentada na forma de acurácia e o Matthews correlation coefficient (MCC). A acurácia pode ser calculada pela Equação 5.1 e o MCC pela Equação 5.2. O MCC é uma métrica equilibrada do desempenho de um classificador binário, levando em conta as taxas de erros para as classes positivas e negativas.

$$VP = Verdadeiros\ Positivos$$

$$FP = Falsos\ Positivos$$

$$VN = Verdadeiros\ Negativos$$

$$FN = Falsos\ Negativos$$

$$Acurácia = \frac{(VP + VN)}{VP + FP + VN + FN} \tag{5.1}$$

$$MCC = \frac{VP + VN - FP + FN}{\sqrt{(VP + FP)(VP + FN)(VN + FP)(VN + FN)}} \tag{5.2}$$

5.2.1 Treinamento da rede neural convolucional

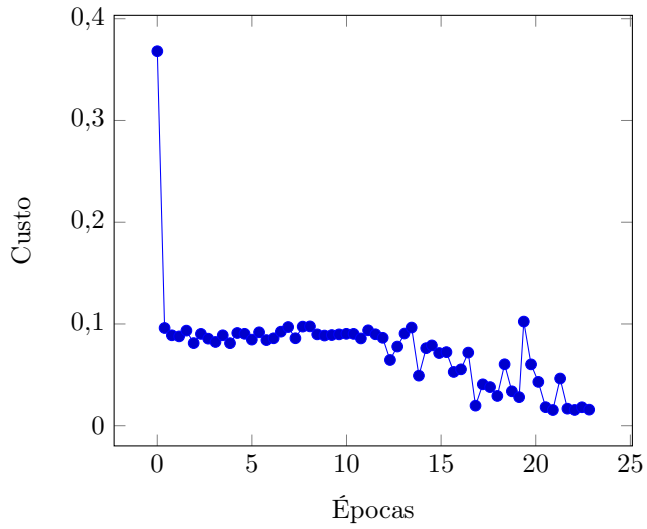
Foram realizados testes com as redes de quatro e seis camadas de convolução para analisar a convergência dos algoritmos de otimização durante o treinamento das redes neurais. Para os testes foi utilizado o banco de dados AVA levando em consideração apenas a categoria natureza, para conseguir validar a estrutura do modelo escolhido. Nessa categoria existe um total de aproximadamente vinte mil imagens.

Como separação das fotos esteticamente boas das ruins foi adotado o critério da nota média 5. Se a foto possuísse uma nota inferior a 5 ela é considerada da classe negativa e se for igual ou superior a 5 da categoria positiva. Dessa forma obteve-se um total de 5841 exemplos negativos e 14983 exemplos positivos, ou seja, aproximadamente 28% dos exemplos são negativos e 72% positivos. Deste total, foi reservado aproximadamente 20% dos exemplos para que os testes para validação do modelo pudessem ser realizados, num total de 4160 imagens, deixando 16664 imagens para a etapa de treinamento. É importante notar que a base de dados é desbalanceada e, para corrigir isto, foi utilizada uma função de custo que penaliza o erro sobre falsos positivos uma vez que uma proporção maior de exemplos positivos faz o modelo tender a ficar enviesado para classificar a saída como positiva.

Os testes foram realizados com lotes de 64 imagens por mini-batch de treinamento. Uma época de treinamento é composta por 261 mini-batches.

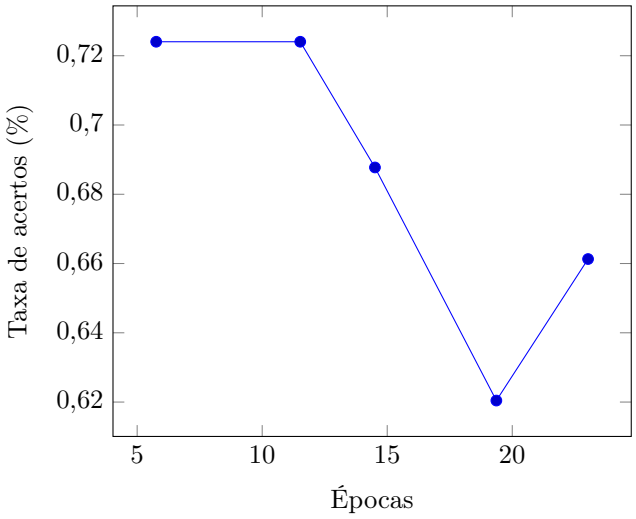
Um fator importante de ser observado num primeiro momento é a queda inicial do valor da função de custo e em sequência uma gradativa diminuição da mesma na Figura 5.4. O valor inicial elevado representa apenas o estado inicial do modelo, o qual teve os pesos gerados de forma aleatória. Em seguida houve um ajuste do modelo ao conjunto de treinamento, com a constante especialização do mesmo para representação deste conjunto. O platô inicial representa um intervalo em que não houve

Figura 5.4: Evolução do custo no conjunto de testes para uma rede de 4 camadas convolucionais.



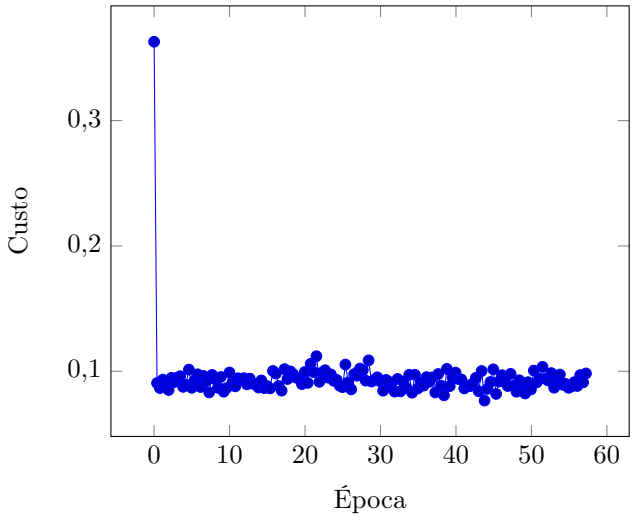
Fonte: Elaborada pelo autor.

Figura 5.5: Evolução da acurácia no conjunto de testes para uma rede de 4 camadas convolucionais.



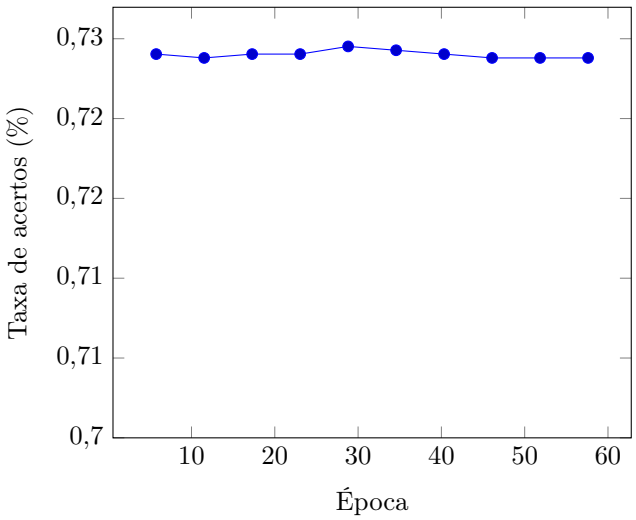
Fonte: Elaborada pelo autor.

Figura 5.6: Evolução do custo no conjunto de testes para uma rede de 6 camadas convolucionais.



Fonte: Elaborada pelo autor.

Figura 5.7: Evolução da acurácia no conjunto de testes para uma rede de 6 camadas convolucionais.



Fonte: Elaborada pelo autor.

mudanças significativas em como o modelo representava a estética da imagem. A partir do mini-batch 3000, entretanto, o modelo começou a melhor representar o problema, o que é refletido como a diminuição do custo.

Ao final do treinamento estabelecido para esta rede neural, pode-se observar que houve um decremento da taxa de acertos sobre o conjunto de validação na Figura 5.5. Apesar de parecer um resultado ruim, na verdade essa súbita mudança mostra uma alteração bastante importante da rede quando comparada ao seu estado inicial. Vale lembrar que a rede foi treinada com um banco de dados desbalanceado e que, portanto, a taxa inicial de acertos apresentada é basicamente o reflexo de uma rede que classifica todos os exemplos como da classe positiva.

Apesar de não estar visível nos gráficos, gradativamente a rede começou a classificar parte dos exemplos como negativos e isso refletiu em como a rede classifica todos os exemplos.

O cenário para o caso de uma rede com 6 camadas convolucionais foi bastante diferente daquele observado no caso com 4 camadas convolucionais. Primeiramente deve-se observar a função de custo na Figura 5.6. O início é bastante similar ao da Figura 5.4, mas logo se percebe que, mesmo após 15000

Tabela 5.1: Resultados com redes neurais convolucionais. Fonte: Elaborada pelo autor.

Rede	Acurácia	MCC
4 camadas convolucionais	66,18%	0,0725
6 camadas convolucionais	73%	0

Fonte: Elaborada pelo autor.

mini-batches, não houve nenhum movimento significativo para uma redução no valor desta função de custo.

Esse comportamento reflete na Figura 5.7, em que a taxa de acertos apenas oscilou na casa dos 73%, valor referente à proporção de exemplos positivos no banco de dados. Basicamente este modelo não foi capaz de generalizar o problema, mantendo apenas a solução trivial de classificar todas as suas entradas como sendo da classe positiva.

A Tabela 5.1 mostra a acurácia e o coeficiente de correlação de Mathews para os testes com a rede convolucional de múltiplas camadas. Analisando a Tabela 5.1 nota-se que apesar da acurácia elevada, em especial no caso de 6 camadas, o MCC é próximo de zero. Este resultado indica que o modelo obtido tem pouca correlação com o resultado real em si. No caso de 6 camadas principalmente o coeficiente zerado indica que o modelo obtido classifica as fotos em apenas uma das classes.

5.2.2 Inception com a categoria natureza do AVA dataset

Baseado em [48], foi criada uma arquitetura de rede neural convolucional que classifica, com taxas de erro de 21,2% no Top-1 e de 5,6% no Top-5, objetos em imagens [56].

Dois testes foram conduzidos a fim de avaliar a capacidade de transferência de conhecimento e classificação de um modelo que inicialmente foi criado para a classificação de objetos em imagens para a avaliação estética de imagens. Estes testes foram feitos utilizando-se apenas a categoria de natureza do banco de dados AVA. No primeiro teste todas as imagens da categoria foram utilizadas, considerando-se como esteticamente positivas as imagens com nota acima da média e as negativas as imagens abaixo da média. Neste teste foi obtida uma acurácia na validação cruzada de 68,8%

O segundo teste consistiu na elaboração de grupos esteticamente positivos e negativos mais consistentes. Para tal, foram consideradas como esteticamente positivas as imagens com nota com um desvio padrão acima da média e as esteticamente negativas as com nota com um desvio padrão abaixo da média. Neste teste obteve-se uma acurácia de 80,8%.

Os resultados positivos dos testes com o uso de transferência de conhecimento foram os motivadores para a utilização desta abordagem para melhorar o desempenho final do modelo.

5.2.3 DCNNs usando como base atributos do ILSVRC

Três experimentos foram realizados usando diferentes 7 arquiteturas distintas de DCNN foram avaliadas: VGG16 [55], MobileNet [49], ResNet50 [50], NASNetMobile [57], MobileNet V2 [58], Inception V3 [59] e Inception-ResNet V2 [60]. Em todos eles as redes foram pré-treinadas com o banco de dados de reconhecimento de objetos do ILSVRC.

O primeiro experimento teve como objetivo principal avaliar uma aplicação de transferência de conhecimento dos atributos do ILSVRC diretamente no problema de classificação de imagens. Neste experimento usou-se uma DLSVM e, mantendo todos os atributos da rede com exceção da última camada imutáveis, realizou-se o treinamento da rede.

Nos dois outros experimentos todos os atributos da rede foram treinados. No segundo experimento usou-se uma DCNN com uma camada de saída tradicional softmax com a função objetivo de entropia cruzada. Este experimento teve como objetivo obter um resultado de base de comparação para a avaliação

da melhoria de desempenho ao usar uma DLSVM. O último experimento teve como objetivo avaliar uma rede DLSVM no cenário da transferência de conhecimento com possibilidade de ajuste de todos os parâmetros da rede durante o treinamento.

5.2.3.1 Parâmetros experimentais

O tamanho da imagem de entrada, independentemente da arquitetura de DCNN, foi adotada como $224 \times 224 \times 3$. Este tamanho foi fixado para garantir que os resultados obtidos pelas diferentes redes partissem do mesmo conjunto de dados pois imagens maiores implicam em mais informação que poderia levar a um ganho de desempenho de uma ou outra rede. Além disso o tamanho escolhido foi baseado no tamanho que é adotado como padrão pela maior parte das arquiteturas que estão sendo estudadas. Em arquiteturas que esperam imagens de maior tamanho um padding exterior foi adotado de forma a deixar a imagem no tamanho adequado.

Para a confecção do banco foi adotada uma estratégia diferente dos testes de redes neurais convolucionais profundas. Como a distribuição de medias no AVA dataset é aproximadamente gaussiana, selecionou-se como imagens de boa qualidade aquelas cuja nota média é igual a média mais um desvio padrão e como imagens de má qualidade estética aquelas cuja nota média é igual à média de todas as imagens menos um desvio padrão. Todas as imagens do banco foram consideradas na construção deste subconjunto, obtendo-se ao final 38306 exemplos negativos e 39577 positivos. Destas imagens foram separadas 5000 para o conjunto de validação e 5000 para o conjunto de testes. Estes conjuntos de 5000 imagens tinham, cada um, 2500 exemplos negativos e 2500 positivos. As imagens foram redimensionadas para as dimensões $224 \times 224 \times 3$, para todas as arquiteturas testadas, inclusive as que esperavam tamanhos de entrada maiores.

Uma camada de dropout com valor de 0,5 é utilizada na saída da primeira etapa, ou seja, na saída da DCNN. Este dropout visa evitar o overfitting. O valor de 0,5 é o que leva ao valor máximo de regularização no caso linear [70]. Um ajuste mais fino deste valor poderia levar a um melhor desempenho dos modelos obtidos.

A função de perda utilizada, tanto durante o treinamento quanto durante os testes, é a perda quadrática de Hinge, de forma a fazer com que a segunda etapa do classificador se comporte como uma SVM linear. Para aproximar ainda mais o comportamento da última camada a uma SVM de margem suave, uma regularização L2 foi adicionada a esta camada com um fator de custo $C = 0,001$, da mesma forma que apresentado na subseção 3.4.3. O fator de custo utilizado é o mesmo adotado por Tang [9] e foi utilizado por apresentar bons resultados de referência.

O otimizador utilizado foi o Adam. Este otimizador foi inicializado com uma taxa de treinamento, lr, de 0,0005. O valor inicial da taxa de treinamento é 2 vezes menor do que a padrão recomendada normalmente para o Adam (0,001) [51] e foi adotada com a intenção de diminuir a velocidade inicial de treinamento. A taxa de treinamento é atualizada ao final de cada lote de iterações utilizando-se a Equação 5.3. O valor de decaimento de 0,003 foi utilizado em todas as atualizações de taxa de treinamento.

$$lr_{new} = \frac{lr_{old}}{1 + decay * iteration} \tag{5.3}$$

O decaimento utilizado foi um decaimento simples, que já é implementado no otimizador Adam do Keras. Outras opções de atualização de taxa de aprendizado que poderiam ser utilizadas são a apresentada no artigo original do Adam, na Equação 5.4. Outra opção que tem apresentado bons resultados é o uso de taxas de aprendizagem cíclicas [71]. A principal motivação por trás de taxas cíclicas é encontrar os valores ótimos para este hiperparâmetro, sem que diversos experimentos precisem ser realizados. Usar outras formas de atualização de taxa de aprendizagem poderia levar a resultados mais próximos ao ótimo, mas não foram exploradas neste trabalho.

Tabela 5.2: Matrizes de confusão para experimentos com transferência de conhecimento no caso da DLSVM e treinamento apenas dos parâmetros da última camada.

(a) ResNet50

		Previsto	
		Positivo	Negativo
Real	Positivo	0	2506
	Negativo	0	2494

(b) VGG16

		Previsto	
		Positivo	Negativo
Real	Positivo	1840	664
	Negativo	640	1856

(c) MobileNet

		Previsto	
		Positivo	Negativo
Real	Positivo	2196	308
	Negativo	1166	1330

(d) MobileNet-V2

		Previsto	
		Positivo	Negativo
Real	Positivo	1989	519
	Negativo	926	1566

(e) Inception-ResNet-V2

		Previsto	
		Positivo	Negativo
Real	Positivo	2289	216
	Negativo	1587	908

(f) Inception-V3

		Previsto	
		Positivo	Negativo
Real	Positivo	2349	156
	Negativo	1818	677

(g) NASNet-Mobile

		Previsto	
		Positivo	Negativo
Real	Positivo	2349	154
	Negativo	1698	799

Fonte: Elaborada pelo autor.

Tabela 5.3: Resultados obtidos com transferência de conhecimento no caso da DLSVM e treinamento apenas da última camada.

	Acurácia	MCC
VGG16	73,92%	0,478
MobileNet	70,52%	0,437
ResNet50	49,88%	0
NASNetMobile	63,94%	0,329
MobileNet V2	71,10%	0,427
Inception V3	60,52%	0,055
Inception-ResNet V2	62,96%	0,085

Fonte: Elaborada pelo autor.

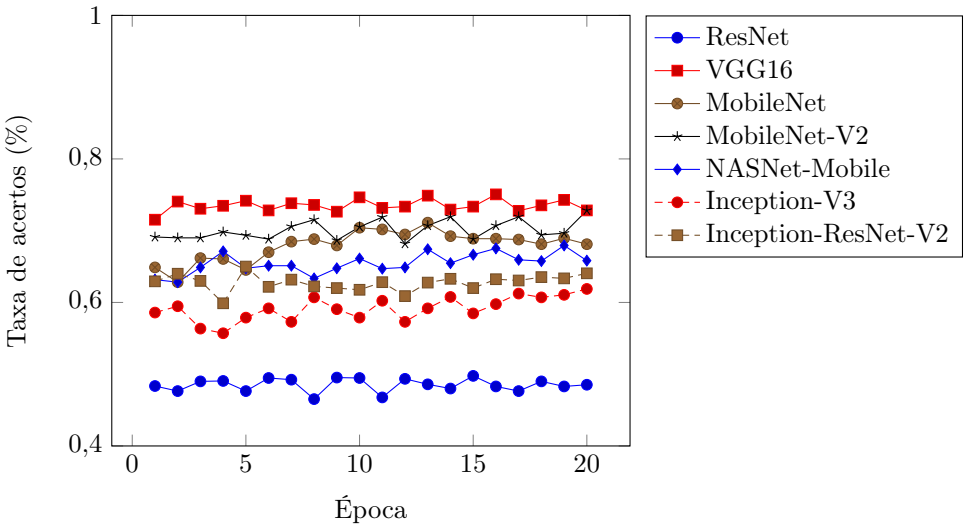
$$lr_{new} = \frac{lr_{old}}{\sqrt{iteration}} \tag{5.4}$$

5.2.3.2 Resultados

As matrizes de confusão, os resultados obtidos ao final do treinamento e as curvas de acurácia e de custo ao longo do treinamento para o caso da DLSVM com o treinamento apenas da última camada, podem ser observados respectivamente na Tabela 5.2, na Tabela 5.5, na Figura 5.7 e na Figura 5.4 para cada uma das arquiteturas de DCNNs utilizadas.

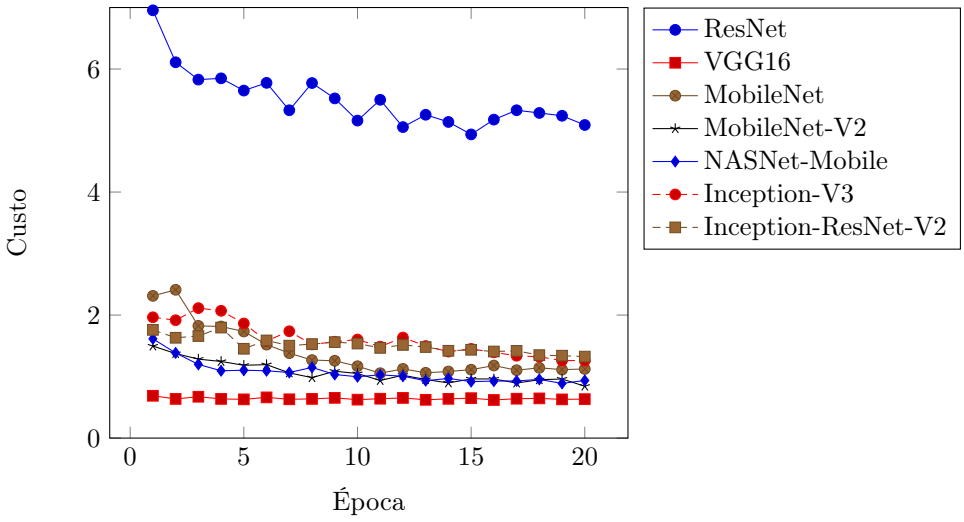
As matrizes de confusão, os resultados obtidos ao final do treinamento e as curvas de acurácia e de custo ao longo de treinamento para o caso de DCNN com saída softmax com diferentes arquiteturas e treinamento de todas as camadas da rede podem ser observados respectivamente na Tabela 5.4, na Tabela 5.5, na Figura 5.9 e na Figura 5.10. Esses resultados servirão como base de comparação para a avaliação

Figura 5.8: Evolução da acurácia no conjunto de testes para redes DLSVM com diferentes arquiteturas de DCNN com treinamento apenas dos parâmetros da última camada.



Fonte: Elaborada pelo autor.

Figura 5.9: Evolução do custo no conjunto de testes para redes DLSVM para diferentes arquiteturas de DCNN com treinamento apenas dos parâmetros da última camada.



Fonte: Elaborada pelo autor.

Tabela 5.4: Matrizes de confusão para experimentos com transferência de conhecimento e treinando todas as camadas com a última camada softmax.

(a) ResNet50

		Previsto: Positivo	Previsto: Negativo
Real	Positivo	2166	410
	Negativo	461	1963

(b) VGG16

		Previsto: Positivo	Previsto: Negativo
Real	Positivo	2573	0
	Negativo	2427	0

(c) MobileNet

		Previsto: Positivo	Previsto: Negativo
Real	Positivo	2096	479
	Negativo	377	2048

(d) MobileNet-V2

		Previsto: Positivo	Previsto: Negativo
Real	Positivo	2036	541
	Negativo	310	2113

(e) Inception-ResNet-V2

		Previsto: Positivo	Previsto: Negativo
Real	Positivo	2308	273
	Negativo	539	1880

(f) Inception-V3

		Previsto: Positivo	Previsto: Negativo
Real	Positivo	2011	562
	Negativo	373	2054

(g) NASNet-Mobile

		Previsto: Positivo	Previsto: Negativo
Real	Positivo	2487	91
	Negativo	1155	1267

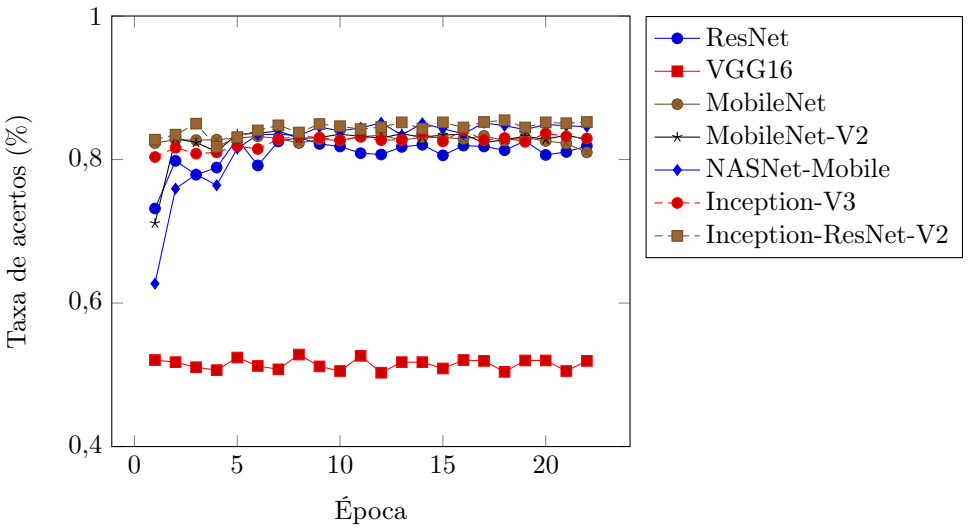
Fonte: Elaborada pelo autor.

Tabela 5.5: Resultados obtidos com transferência de conhecimento e treinando todas as camadas com a última camada softmax.

	Acurácia	MCC
VGG16	51,46%	0
MobileNet	82,88%	0,658
ResNet50	82,58%	0,652
NASNetMobile	75,08%	0,548
MobileNet V2	82,98%	0,663
Inception V3	81,30%	0,628
Inception-ResNet V2	83,76%	0,678

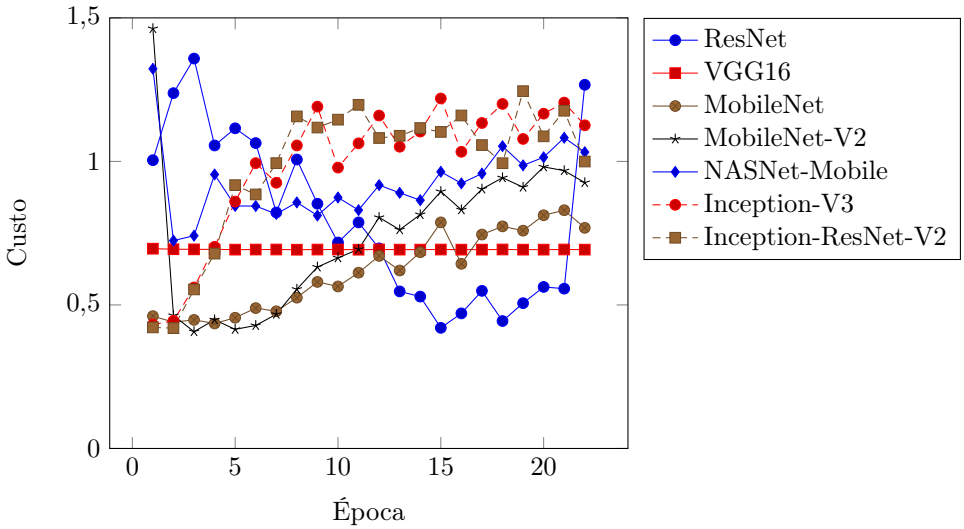
Fonte: Elaborada pelo autor.

Figura 5.10: Evolução da acurácia no conjunto de testes com diferentes arquiteturas de DCNN, saída softmax e treinamento de todas as camadas da rede.



Fonte: Elaborada pelo autor.

Figura 5.11: Evolução do custo no conjunto de testes com diferentes arquiteturas de DCNN, saída softmax e treinamento de todas as camadas da rede.



Fonte: Elaborada pelo autor.

Tabela 5.6: Matrizes de confusão para experimentos de transferência de conhecimento da DLSVM para diferentes arquiteturas de DCNN e treinamento de todas as camadas da rede.

(a) ResNet50

		Previsto	
		Positivo	Negativo
Real	Positivo	2150	467
	Negativo	358	2025

(b) VGG16

		Previsto	
		Positivo	Negativo
Real	Positivo	2125	379
	Negativo	525	1971

(c) MobileNet

		Previsto	
		Positivo	Negativo
Real	Positivo	2104	402
	Negativo	453	2041

(d) MobileNet-V2

		Previsto	
		Positivo	Negativo
Real	Positivo	2094	411
	Negativo	375	2120

(e) Inception-ResNet-V2

		Previsto	
		Positivo	Negativo
Real	Positivo	2191	316
	Negativo	381	2112

(f) Inception-V3

		Previsto	
		Positivo	Negativo
Real	Positivo	2084	426
	Negativo	333	2157

(g) NASNet-Mobile

		Previsto	
		Positivo	Negativo
Real	Positivo	2247	267
	Negativo	555	1931

Fonte: Elaborada pelo autor.

das melhorias de desempenho decorrentes do uso da DLSVM.

As matrizes de confusão, os resultados obtidos ao final do treinamento e as curvas de acurácia e de custo ao longo do treinamento para o caso de DLSVM com diferentes arquiteturas de DCNNs com treinamento de todas as camadas da rede, podem ser observados respectivamente na Tabela 5.6, na Tabela 5.7, na Figura 5.12 e na Figura 5.13.

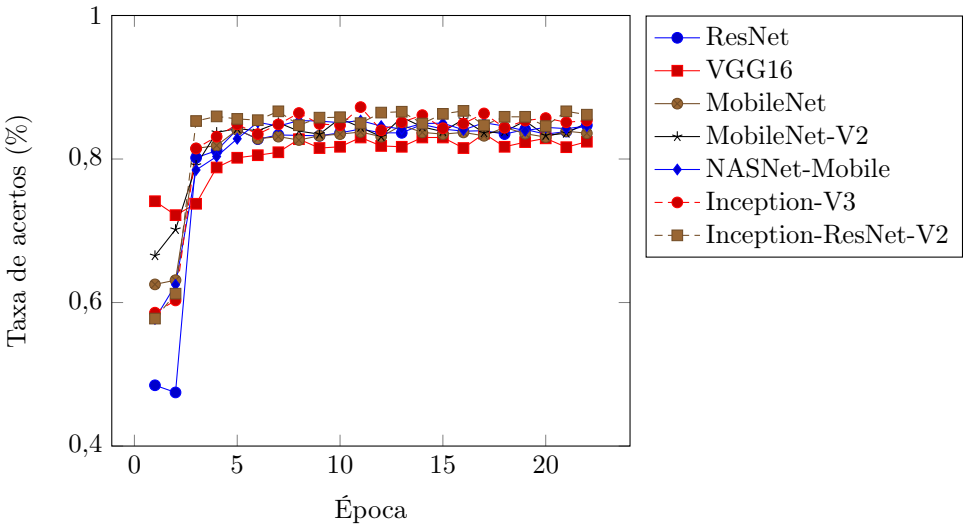
Finalmente a Tabela 5.8 mostra os tempos médios de treinamento em segundos para uma época para cada um dos testes realizados para as DCNNs.

Tabela 5.7: Resultados obtidos com o modelo DLSVM para diferentes arquiteturas de DCNN e treinamento de todas as camadas da rede. Fonte: Elaborada pelo autor.

Rede	Acurácia	Melhoria com Relação ao Treinamento Apenas da Última Camada	Melhoria com Relação ao uso de softmax	MCC
VGG16	81,92%	+10,82%	+59,19%	0,639
MobileNet	82,90%	+17,56%	+0,02%	0,658
ResNet50	83,50%	+67,04%	+1,11%	0,671
NASNetMobile	84,18%	+32.72%	+11,29%	0,684
MobileNet V2	84,28%	+18.53%	+1,57%	0,686
Inception V3	84,82%	+40,15%	+4,33%	0,697
Inception-ResNet V2	86,06%	+32,71%	+2,75%	0,721

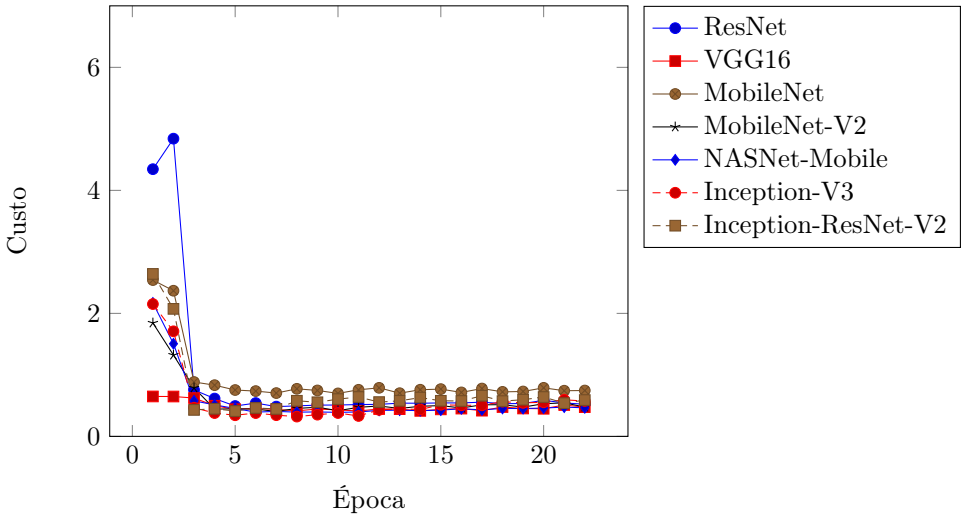
Fonte: Elaborada pelo autor.

Figura 5.12: Evolução da acurácia no conjunto de testes para DLSVM com diferentes arquiteturas de DCNN e treinamento de todas as camadas da rede.



Fonte: Elaborada pelo autor.

Figura 5.13: Evolução do custo no conjunto de testes para redes DLSVM com diferentes arquiteturas de DCNN e treinamento de todas as camadas da rede.



Fonte: Elaborada pelo autor.

Tabela 5.8: Duração média em minutos de uma época de treinamento para as redes DLSVM com diferentes arquiteturas de DCNNs.

	DLSVM		DCNN Softmax
Rede	Treinamento da última camada apenas (s)	Treinamento de todas as camadas (s)	Treinamento de todas as camadas (s)
VGG16	3102	1840	2335
MobileNet	2342	780	991
ResNet50	3710	1677	2100
NASNetMobile	1828	1618	2137
MobileNet V2	2091	849	1141
Inception V3	1432	1267	1604
Inception-ResNet V2	994	2958	2138

Fonte: Elaborada pelo autor.

6. Análise dos Resultados

Os resultados obtidos nos experimentos serão analisados sob diferentes espectros a fim de evidenciar características importantes que uma análise isolada dos resultados como puramente numéricos não é capaz de revelar. Além de comparar os resultados dos experimentos entre si também será realizada uma comparação com os resultados obtidos pela literatura.

Desta forma, a rede foi treinada para os casos mais extremos de classificação, nas quais as diferenças entre aquilo que é esteticamente bom ou ruim são mais evidentes. Esta composição de rede levou a resultados melhores do que aqueles obtidos na Tabela 2.2.

As hipóteses iniciais estabelecidas para o uso de transferência de conhecimento serão avaliadas, uma comparação entre as diferentes arquiteturas de DCNN com relação ao seu desempenho (tanto de acurácia quanto de tempo) será realizada. Ao final pretende-se levantar hipóteses e estabelecer conclusões que poderão ser utilizadas como referência para que trabalhos futuros possam investigar o problema de classificação estética de imagens usando transferência de conhecimento de forma mais profunda.

6.1 Aplicabilidade da transferência de conhecimento

É preciso validar primeiramente se os resultados obtidos corroboram com a hipótese que foi necessário se estabelecer para a definição do problema de transferência de conhecimento. Tomando como base os resultados obtidos nos experimentos de rede neural convolucional apenas nota-se que, com exceção do caso da arquitetura ResNet com o treinamento apenas da última camada, todos os experimentos usando transferência de conhecimento levaram a resultados melhores.

Desconsiderando a arquitetura ResNet para o caso de transferência de conhecimento com ajuste dos parâmetros apenas da última camada, a análise das matrizes de confusão, nas quais as taxas de verdadeiros positivos e verdadeiros negativos são ambas diferentes de zero, permite afirmar que métodos de transferência de conhecimento baseados no ILSVRC Object Classification Challenge é aplicável ao problema de classificação estética. Outros trabalhos também tiveram resultados baseadas nesta observação [17, 18, 19, 72, 21].

A aplicabilidade direta das arquiteturas de DCNN com o treinamento apenas da última camada também demonstrou resultados positivos. Pode-se concluir que os bancos de dados AVA e ILSVRC Object Classification Challenge são uma representação do mesmo domínio e a hipótese inicialmente assumida baseada no fato de que ambos os bancos de dados são de imagens é verdadeira.

Comparando os resultados da Tabela 5.5 com os da Tabela 2.2 fica claro que usar somente os atributos baseados em classificação de objetos não é suficiente para superar outros resultados obtidos na literatura; mesmo os trabalhos mais antigos que não utilizam DCNN tem resultados melhores. Assim, a segunda conclusão baseada nesta análise é: mesmo que aplicáveis, os atributos especializados em classificação de objetos no ILSVRC não são ótimos para a classificação estética.

Por outro lado, comparando os resultados da Tabela 5.7 com os da Tabela 2.2, independentemente da arquitetura de DCNN, os resultados obtidos são superiores à qualquer outro método utilizado. Assim, a

terceira conclusão é: o baseline para métodos de estado da arte para classificação estética são DCNNs.

Finalmente comparando os resultados da Tabela 5.5 com os da Tabela 5.7 e os da Tabela 5.5 pode-se afirmar que independentemente da DCNN e da camada de saída o treinamento de todas as camadas da rede levou a resultados superiores. Este fato complementa as segundas e terceiras conclusões desta sessão, uma vez que ao se fazer ao se treinar a rede como um todo há a especialização dos atributos descritores do modelo, que passam a ser otimizados para a tarefa de classificação estética.

6.2 Comparação do desempenho da DLSVM contra softmax

Um dos objetivos deste trabalho é avaliar se os ganhos observados em [9] também podem ser obtidos no problema de classificação estética. Comparando os resultados dos experimentos de treinamento com todas as camadas usando a softmax como última camada e a DLSVM na Tabela 5.7 pode-se dizer que houveram sempre ganhos no uso da DLSVM.

Desconsiderando o caso extremo da VGG16 na qual a rede treinada com o softmax não conseguiu generalizar o problema o desempenho da DLSVM sempre foi superior, mesmo que de forma marginal. O maior ganho foi na arquitetura NASNetMobile, entretanto os ganhos nas redes baseadas em Inception também foram significativos.

Assim, dado os resultados deste experimento é justificável o uso de DLSVM para o problema de classificação estética uma vez que ela levará sempre a resultados similares ou melhores ao final do treinamento do modelo.

6.3 Resultados com as arquiteturas de DCNN analisadas

Através da análise dos resultados obtidos para as diferentes arquiteturas de DCNN deseja-se avaliar as diferenças entre as arquiteturas, entre os resultados obtidos para a tarefa de classificação de objetos e a de classificação estética de forma individual e as diferenças entre o treinamento apenas da última camada e de todas as camadas.

Primeiramente analisando o extremo superior do experimento treinando apenas a última camada da rede, três arquiteturas obtiveram os melhores resultados: VGG16, MobileNet-V2 e MobileNet. A VGG16 é a rede mais antiga das analisadas, mas foi a que obteve o melhor desempenho neste primeiro teste com relação à acurácia e ao MCC. Analisando as taxas de acerto por classe na Tabela 5.4, diferentemente das outras arquiteturas a VGG16 teve uma distribuição de acertos nas classes positivas e negativas que se assimila à distribuição real da base de testes. Uma hipótese para o melhor desempenho da VGG16 neste caso é com relação ao maior número de parâmetros que a rede possui. Uma vez que a última camada possui mais atributos a especialização dos mesmos frente à tarefa de classificação de objetos é menor do que a das demais redes e assim tem uma maior compatibilidade com outros problemas similares que podem tentar ser resolvidos através da transferência de conhecimento. Outra hipótese é com relação à simplicidade dos atributos representados na camada de saída da VGG16. Esta hipótese é reforçada pelos seguintes fatos:

1. Comparando os resultados com DLSVM, a arquitetura VGG16 é a que obteve o pior resultado, indo da mais performática para a menos performática;

2. É a arquitetura mais antiga que usa apenas convoluções convencionais de kernel 3×3. Assim os atributos finais, apesar de numerosos, representam um espaço menor do que as arquiteturas que usam convoluções maiores e/ou atalhos residuais;

Estes dois fatos mostram que os atributos representados pela VGG16 são mais simples, logo mais genéricos e, portanto, menos especializados para a classificação de objetos. Assim possuem um melhor desempenho inicial na transferência de conhecimento, mas um menor com o treinamento da rede com um todo uma vez que o espaço dimensional que os atributos podem representar é menor.

Um fato curioso com relação aos resultados da Tabela 5.4 é com relação ao bias inicial dos classificadores. Na maior parte das arquiteturas houve um bias com relação à classificação na classe positiva. Uma hipótese é a de que os atributos usados na classificação de objetos tenham uma correlação maior com atributos que caracterizam uma boa foto. Fazendo uma relação com os descritores estéticos concretos, atributos que descrevem fotos sem foco, com composições ruins ou desalinhadas não possuem uma grande relevância para o problema de classificação de objetos.

Finalmente, olhando para o extremo inferior do espectro de desempenho das arquiteturas de DCNN no caso do treinamento apenas da última camada está a ResNet-50. Esta arquitetura não conseguiu classificar fotos como esteticamente boas ou ruins, classificando todas as fotos do conjunto de testes como ruins. Uma hipótese que pode ser levantada para este caso é o fato de atalhos residuais serem blocos mais especializados para um determinado tipo de problema entretanto não foram levantadas evidências que suportem este fato.

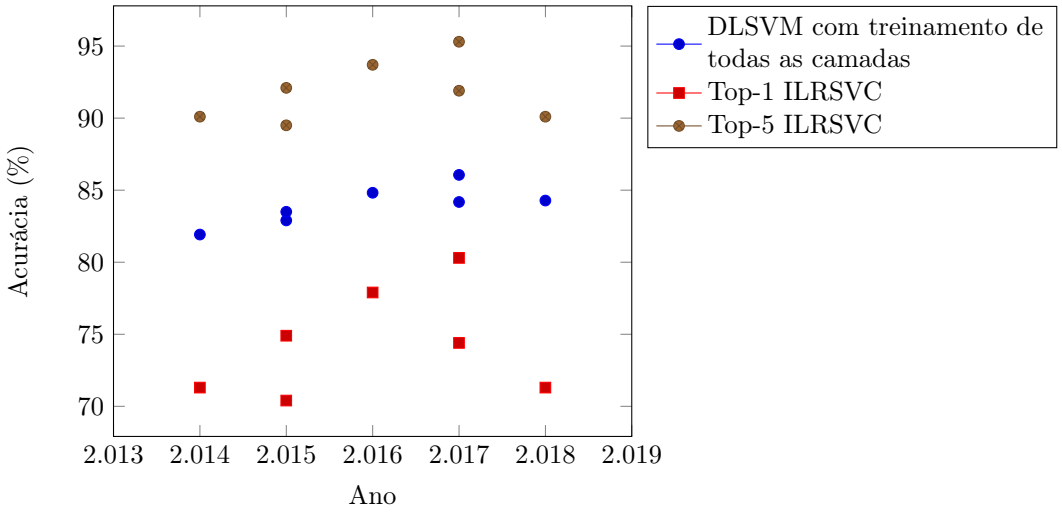
Agora analisando os resultados do experimento de DCNNs com saída softmax, a análise da Tabela 5.5 mostra que a VGG16 não conseguiu generalizar o problema. Ao longo de todo o treinamento não houve nenhuma mudança significativa no erro de treinamento e, uma hipótese que pode ser levantada é que a função de custo utilizada neste caso não tem um bom comportamento para a arquitetura treinada. Por outro lado todas as outras arquiteturas de redes conseguiram resolver de forma comparável ao estado da arte, com a Inception-ResNet-V2, MobileNet V2 e MobileNet com os melhores resultados. Os resultados para todas as arquiteturas, com exceção da VGG16 e da NASNetMobile, foram bastante próximos.

Excluindo o caso extremo da VGG16, um fato interessante de ser observado sobre o treinamento das DCNNs com saída softmax é sobre o comportamento da função de custo ao longo do treinamento na Figura 5.11. Diferentemente do erro de Hinge quadrático, o gradiente estocástico descendente tem uma curva ao longo do treinamento que passa por um mínimo e então aumenta. Entretanto, mesmo com o aumento deste erro há uma melhoria da acurácia de treinamento sobre o conjunto de validação. Uma explicação deste fato se deve ao fato de que o treinamento visa maximizar a acurácia de validação e ao longo do treinamento os exemplos de validação passam a ser classificados de forma mais extrema. Assim, os exemplos classificados erroneamente passam a ter um valor de taxa de erro superior. Da mesma forma, os exemplos classificados corretamente passam a ser classificados com uma certeza ainda maior.

Analisando os resultados do experimento com DLSVM treinando todas as camadas de forma isolada, uma primeira análise da Tabela 5.7 permite concluir que todas as redes DCNN analisadas conseguem resolver de forma satisfatória o problema de classificação estética ou seja, não houve em nenhum caso uma transferência negativa de conhecimento. O desempenho das redes tem uma correlação direta com o ano de publicação da rede como pode ser observado na Figura 6.1. Este fato está diretamente ligado com o fato dos blocos utilizados em cada uma das arquiteturas, com arquiteturas mais complexas possuindo um desempenho maior do que aquelas menos complexas. Além disso, comparando os resultados arquitetura a arquitetura pode-se observar também uma correlação direta do desempenho da rede no problema de classificação de objetos com o desempenho na classificação estética, servindo também como mais um resultado que suporta a hipótese inicial de que os espaços representados pelo banco de dados AVA e o ILSVRC Object Classification Challenge são equivalentes.

Comparando as arquiteturas de maior desempenho da Tabela 5.7 nota-se que as redes de maior desempenho são aquelas baseadas no bloco Inception. Essa informação permite formular a hipótese que redes que representam esparcialidade representam um espaço maior e, portanto, são capazes de gerar modelos mais genéricos. Outro ponto interessante pode ser observado com relação às redes Inception-ResNet-V2,

Figura 6.1: Comparação da acurácia das DCNNs ao longo dos anos.



Fonte: Elaborada pelo autor.

ResNet50 e MobileNet-V2. Todas estas arquiteturas usam um atalho residual em sua implementação e todas elas desempenharam melhor que arquiteturas equivalentes que não usam esta técnica. A rede que obteve o melhor desempenho, assim como no caso da classificação de objetos, foi a Inception-ResNet-V2. A Inception-ResNet-V2 é a rede que possui a maior quantidade de técnicas distintas aplicadas e o uso delas em conjunto promoveu resultados superiores dos que aqueles observados pela aplicação das técnicas em separado.

Fazendo a análise das curvas de treinamento das redes das Figuras 5.12 e 5.13, nota-se uma consistência das curvas, independentemente da arquitetura utilizada. A primeira época tem um resultado bastante próximo daquele observado no experimento treinando apenas a última camada da DLSVM, o que é esperado dado que inicialmente os dois experimentos têm características bem próximas. Há então um súbito crescimento da acurácia e decrescimento do custo nas próximas duas épocas seguidas por uma série de épocas em que a variação dos parâmetros é menor, mas em que há a convergência para o estado final para todas as redes depois de aproximadamente 5 épocas. Uma conclusão clara disto é o fato de que o número de épocas utilizadas foi muito maior do que o necessário para a estabilização. Para os parâmetros utilizados de decaimento da taxa de treinamento e configuração do banco de dados um total de 7 épocas é suficiente para que mais épocas de treinamento da rede se tornem desnecessárias.

Comparando os resultados dos experimentos com DLSVM, seguindo a VGG16, a MobileNet e a MobileNet-V2 foram as redes que tiveram a menor melhoria com relação ao primeiro experimento. Este é um resultado esperado, dado que estas três arquiteturas são formadas por blocos mais simples. No caso da MobileNet e MobileNet-V2, ambas as redes foram projetadas com o objetivo de desempenho em ambientes de requisitos computacionais mais restritos, sacrificando um pouco do desempenho de acurácia para obter um maior desempenho de computação. Por outro lado, as arquiteturas que tiveram a maior taxa de melhoria foram a ResNet-50, Inception-V3 e Inception-ResNet-V2. Primeiramente a ResNet-50 teve a melhoria com a maior taxa devido ao seu desempenho ruim no experimento com treinamento apenas da última camada. Como no experimento com o treinamento de todas as camadas a arquitetura não só passou a generalizar o modelo, mas também conseguiu atingir um patamar equivalente às outras arquiteturas testadas o aumento é justificado. As duas outras redes têm em comum o bloco Inception. Uma possível hipótese é que o bloco Inception, por tratar os atributos de forma paralela formando uma rede com camadas com camadas horizontais mais complexas do que utilizando apenas nós do mesmo tipo por camada, a rede gera atributos mais especializados e que, assim, precisam de uma etapa de treinamento para terem um desempenho mais próximo do ótimo.

Tabela 6.1: Correlação entre os resultados da classificação estética e classificação de objetos considerando todas as arquiteturas de DCNNs estudadas.

Parâmetros usados para a correlação	ρ
x=ILRSVC Top-1, y=DLSVM e treinamento da última camada	-0,540
x=ILRSVC Top-5, y=DLSVM e treinamento da última camada	-0,583
x=ILRSVC Top-1, y=Softmax e treinamento de todas as camadas	0,364
x=ILRSVC Top-5, y=Softmax e treinamento de todas as camadas	0,330
x=ILRSVC Top-1, y=DLSVM e treinamento de todas as camadas	0,826
x=ILRSVC Top-5, y=DLSVM e treinamento de todas as camadas	0,712

Fonte: Elaborada pelo autor.

Um resultado importante de toda esta análise é com relação à escolha da arquitetura adequada para problemas de classificação estética. Um dos parâmetros mais relevantes é com relação à disponibilidade de tempo e recursos computacionais para treinar a rede. Caso esses requisitos sejam restritos a escolha deve ser feita dando prioridade para redes com blocos mais simples e que os atributos de classificação de objetos podem ser usados de forma mais direta tais como VGG16, a MobileNet e a MobileNet-V2, treinando apenas a última camada. Se por outro lado for possível usar o treinar a rede como um todo a escolha deve ter como prioridade redes que usem elementos que podem levar a uma melhor especialização, tais como Inception-V3 e Inception-ResNet-V2. Além disso, optar por usar a DLSVM ao invés de uma camada de saída softmax leva a resultados iguais ou superiores.

O segundo resultado importante desta análise é a correlação entre o desempenho das redes nos problemas de classificação estética usando DLSVM treinando todas as camadas e classificação de objetos, calculada usando a Equação 6.1 e com os resultados apresentados na Tabela 6.1. Existe uma alta chance de que arquiteturas de que DCNNs com melhor desempenho desenvolvidas para o ILSVRC Object Classification Challenge tenham também um melhor desempenho na classificação estética quando se parte de uma rede pré-treinada. Assim, a evolução do estado da arte no problema de classificação de objetos automaticamente leva a evolução do estado da arte na classificação estética e pesquisadores podem usar esta informação como ponto de partida para o aperfeiçoamento de seus modelos. Este resultado só se sustenta para casos em que o treinamento de todas as camadas e com DLSVM é feito uma vez que a correlação foi negativa nos casos com o treinamento apenas da última camada e não muito expressivo no caso usando como camada de saída a softmax.

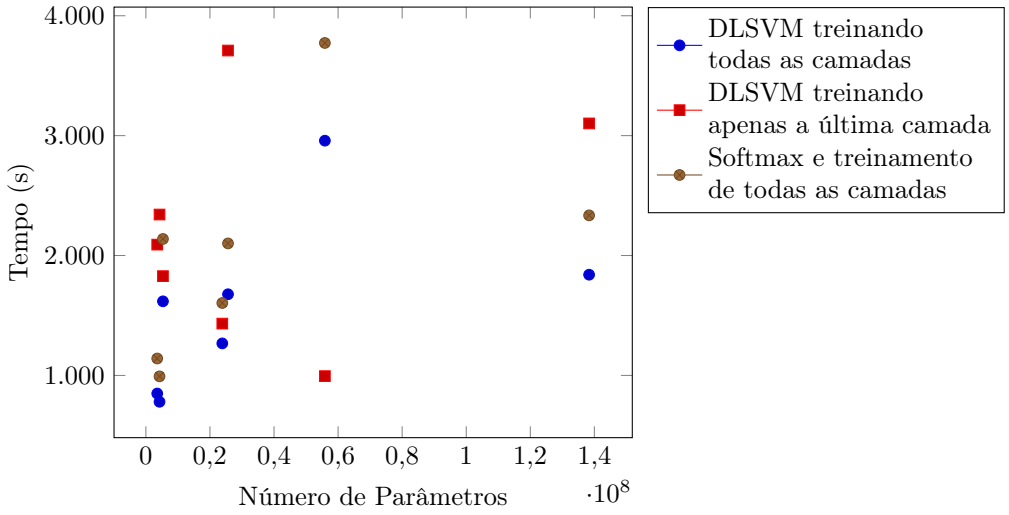
$$\rho = \frac{cov(X, Y)}{\sqrt{var(X) * var(Y)}} \tag{6.1}$$

O terceiro resultado desta análise é que é possível atingir resultados satisfatórios usando a estratégia de montagem de DCNNs proposta neste projeto, com uma DLSVM usando uma rede pré-treinada no ILSVRC Object Classification Challenge. O modelo com o melhor desempenho supera aquele obtido no estado da arte de outros projetos estudados na literatura.

6.4 Tempo de treinamento

Os resultados obtidos para os tempos de treinamento por época foram totalmente diferentes daquilo que era esperado. Em todos os experimentos a mesma configuração de hardware foi utilizada. Uma expectativa válida é que as arquiteturas mais complexas teriam um tempo de treinamento maior e que os experimentos com treinamento de todas as camadas fossem mais demorados do que aquele com o treinamento apenas da última camada da rede. Observando a Tabela 5.8 na maioria das arquiteturas estudadas, o tempo para o treinamento do caso do treinamento apenas da última camada foi superior ao tempo do caso treinamento de todas as camadas com exceção da Inception-ResNet-V2. Além disso,

Figura 6.2: Comparação dos tempos de treinamento entre os experimentos.



Fonte: Elaborada pelo autor.

Tabela 6.2: Correlação entre os tempos de treinamento e o número de parâmetros treináveis da rede. r.

Parâmetros usados para a correlação	ρ
x=DLSVM com treinamento de todas as camadas, y=DLSVM com treinamento apenas da última camada	-0,262
x=DLSVM com treinamento de todas as camadas, y=Softmax com treinamento de todas as camadas	0,999
x=parâmetros, y=DLSVM com treinamento apenas da última camada	0,264
x=parâmetros, y=DLSVM com treinamento de todas as camadas	0,489
x=parâmetros, y=Softmax com treinamento de todas as camadas	0,476

Fonte: Elaborada pelo autor.

calculando a correlação entre os tempos dos dois experimentos obtêm-se o valor de $-0,262$ o que mostra uma baixa correlação entre os dados, mostrando que os tempos do primeiro experimento com DLSVM não fornecem nenhuma informação sobre os tempos do segundo experimento com DLSVM.

Um resultado que era esperado e ocorreu é um menor tempo de treinamento por época para as redes MobileNet e MobileNet-V2, que foram otimizadas para desempenho computacional. Apesar deste resultado não ter ocorrido para o caso com treinamento apenas da última camada, para o caso com treinamento de todas as camadas elas foram as redes com o treinamento mais veloz independentemente da camada de saída. Outro resultado que apenas foi verdadeiro para o caso do treinamento com todas as camadas, DLSVM e softmax, foi o tempo de treinamento maior para as redes mais complexas, no caso a Inception-ResNet-V2 que possui os blocos mais complexos e a VGG16 que possui o mais elevado número de parâmetros. Finalmente, o tempo de treinamento independe da camada de saída utilizada, como pode ser concluído analisando a correlação de quase 1 entre os resultados de tempo de treinamento para o treinamento de todas as camadas da rede com DLSVM e softmax.

Comparando os tempos de treinamento com o número de parâmetros da rede ambas as redes tiveram uma correlação positiva, como pode ser observado na Tabela 6.2, com o experimento treinamento de todas as camadas com uma correlação maior do que o experimento com treinamento apenas da última camada. Observando também a Figura 5.5, podemos notar que ambas as dispersões poderiam ser aproximadas por retas, mas o desvio no caso com treinamento da última camada seria maior.

Uma hipótese que pode se estabelecer seria o fato do framework de aprendizagem profunda em si não ter uma boa otimização para casos em que apenas a última camada da rede é treinada.

7. Conclusões

Este trabalho apresentou um modelo de DLSVM usando como base arquiteturas de DCNNs pré-treinadas no ILSVRC Object Classification Challenge comparando os resultados com diferentes configurações de treinamento e camada de saída.

As principais conclusões obtidas foram:

- A constatação de que o uso de uma DLSVM produz resultados iguais ou melhores que o uso de uma camada softmax para o problema de classificação estética. Este resultado é compatível com o trabalho que apresentou a DLSVM [9];
- A constatação da alta correlação entre os problemas de classificação de objetos e estética. Melhorias no estado da arte no primeiro problema implicam em melhorias no estado da arte do segundo problema usando transferência de conhecimento partindo de uma rede pré-treinada no ILSVRC Object Classification Challenge e fazendo o treinamento de todas as camadas da rede;
- Um modelo baseado na Inception-ResNet-V2 usando DLSVM para classificação estética que possui um resultado melhor do que o obtido no estado da arte;

Usando o modelo com DLSVM e Inception-ResNet-V2 para classificar as fotografias da Figura 1.2, a fotografia de acervo próprio foi classificada como esteticamente ruim e a fotografia vencedora de concurso como esteticamente boa, gerando resultados iguais àqueles decorrentes da análise manual das fotos.

Todos estes resultados estabelecem uma base sobre a qual mais estudos podem ser desenvolvidos. Os métodos de transferência de conhecimento aplicados neste projeto não analisaram o impacto dos diferentes hiperparâmetros das redes no desempenho final. Como trabalhos futuros algumas linhas de estudo podem ser exploradas.

As principais linhas de trabalhos futuros para a expansão deste estudo são:

- Avaliar como os diferentes hiperparâmetros (taxa de aprendizagem, decaimento, dropout) e como eles afetam o desempenho do modelo;
- Comparar como uma diferente escolha de subgrupo do AVA dataset impacta o resultado no desempenho obtido;
- Avaliar o desempenho do modelo fora do AVA dataset, comparando com outros bancos de dados;
- Avaliar arquiteturas de DCNNs mais modernas;
- Desenvolver um modelo baseado no ensemble das saídas produzidas por redes baseadas em diferentes arquiteturas;
- Aplicar métodos para ampliar o conjunto de dados de treinamento e testes e avaliar o impacto destas técnicas no desempenho do modelo;

A conclusão final é que o trabalho atingiu aos objetivos inicialmente estabelecidos: obter um modelo base com desempenho similar ao da literatura, construir um conhecimento sobre classificação estética que poderá ser utilizado e expandido em trabalhos futuros e avaliar o uso de uma DLSVM para no problema de classificação estética..

Referências Bibliográficas

[1] FILHO, J. G. *Gestalt do objeto*. Sao Paulo, SP, Brasil: Escrivuras Editora, 2009.

[2] UT, N. *Phan Thi Kim Phuc (center) flees with other children after South Vietnamese planes mistakenly dropped napalm on South Vietnamese troops and civilians.* [s.n.], 1972. Disponível em: <https://www.worldpressphoto.org/collection/photo/1973/37161/1/1973-Nick-Ut-WY>.

[3] COLE, C. *A demonstrator confronts a line of People's Liberation Army tanks on Chang'an Avenue, Beijing, during protests for democratic reform on Tiananmen Square.* [s.n.], 1989. Disponível em: <https://www.worldpressphoto.org/collection/photo/1990/33708/1/1990-charlie-cole-wy>.

[4] TAKAHASHI, R. *Mermaid*. [s.n.], 2018. Disponível em: <https://www.nationalgeographic.com/contests/travel-photo-contest/2018/winners/#/prod-yourshot-1527368-11798474.jpg>.

[5] MARCHESOTTI, L. et al. Assessing the aesthetic quality of photographs using generic image descriptors. *Proceedings of the IEEE International Conference on Computer Vision*, p. 1784–1791, 2011.

[6] DATTA, R. et al. Studying aesthetics in photographic images using a computational approach. *Computer Vision ECCV 2006*, p. 288–301, 2006.

[7] JIN, X. et al. Ilgnet: Inception modules with connected local and global features for efficient image aesthetic quality classification using domain adaptation. *8th International Conference on Wireless Communications & Signal Processing, WCSP 2016*, 2016.

[8] MAVRIDAKI, E.; MEZARIS, V. A comprehensive aesthetic quality assessment method for natural images using basic rules of photography. *Proceedings - International Conference on Image Processing, ICIP*, p. 887–891, 2015.

[9] TANG, Y. Deep learning using linear support vector machines. *CoRR*, abs/1306.0239, 2013. Disponível em: <http://arxiv.org/abs/1306.0239>.

[10] LIU, T.-J.; LIN, W.; KUO, C.-C. J. Image quality assessment using multi-method fusion. *IEEE transactions on image processing*, v. 22, p. 1793–1807, 2013.

[11] KE, Y.; TANG, X.; JING, F. The design of high-level features for photo quality assessment. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 419–426, 2006.

[12] WU, O.; HU, W.; GAO, J. Learning to predict the perceived visual quality of photos. *Proceedings of the IEEE International Conference on Computer Vision*, p. 225–232, 2011.

[13] BHATTACHARYA, S.; SUKTHANKAR, R.; SHAH, M. Automated aesthetic analysis of photographic images. *Proceedings of the international conference on Multimedia MM '10*, p. 271–280, 2010.

[14] LI, C.; LOUI, A. C.; CHEN, T. Towards aesthetics: a photo quality assessment and photo selection system. *Proceedings of the international conference on Multimedia*, p. 10 – 13, 2010.

[15] GAO, Z.; WANG, S.; JI, Q. Multiple aesthetic attribute assessment by exploiting relations among aesthetic attributes. *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval - ICMR '15*, p. 575–578, 2015.

[16] MARCHESOTTI, L.; PERRONNIN, F.; MURRAY, N. Discovering beautiful attributes for aesthetic image analysis. *International Journal of Computer Vision*, v. 113, p. 246–266, 2015.

[17] ZHANG, L. et al. Fusion of multichannel local and global structural cues for photo aesthetics evaluation. *IEEE Transactions on Image Processing*, v. 23, p. 1419–1429, 2014.

[18] DONG, Z. et al. Photo quality assessment with dcnn that understands image well. *International Conference on Multimedia Modeling*, 2015.

[19] TIAN, X. et al. Query-dependent aesthetic model with deep learning for photo quality assessment. *IEEE Transactions on Multimedia*, v. 17, p. 1–1, 11 2015.

[20] WANG, W. et al. A multi-scene deep learning model for image aesthetic evaluation. *Signal Processing: Image Communication*, v. 47, 05 2016.

[21] JIN, X. et al. Efficient deep aesthetic image classification using connected local and global features. *arXiv:1610.02256v2 [cs.CV]*, 2017.

[22] KHAN, S. S.; DAVID, D. V. Evaluating visual aesthetics in photographic portraiture. *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, p. 55–62, 2012.

[23] LUO, Y.; TANG, X. Photo and video quality evaluation: Focusing on the subject. *Computer Vision ECCV 2008*, p. 1 – 14, 2008.

[24] DHAR, S.; ORDONEZ, V.; STONY, T. L. B. High level describable attributes for predicting aesthetics and interestingness. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 1657–1664, 2011.

[25] LO, K.-Y.; LIU, K.-H.; CHEN, C.-S. Assessment of photo aesthetics with efficiency. *International Conference on Pattern Recognition (ICPR), 2012*, p. 2186–2189, 2012.

[26] NISHIYAMA, M. et al. Aesthetic quality classification of photographs based on color harmony. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 33–40, 2011.

[27] WONG, L.-K.; LOW, K.-L. Saliency-enhanced image aesthetics class prediction. *Proceedings - International Conference on Image Processing, ICIP*, p. 997–1000, 2009.

[28] LUO, W.; WANG, X.; TANG, X. Content-based photo quality assessment. *IEEE Transactions on Multimedia*, v. 15, p. 1930–1943, 2013.

[29] LI, C. et al. Aesthetic quality assessment of consumer photos with faces. *Proceedings - International Conference on Image Processing, ICIP*, p. 3221–3224, 2010.

[30] JIANG, W.; LOUI, A. C.; CEROSALETTI, C. D. Automatic aesthetic values assessment in photographic images. *IEEE International Conference on Multimedia and Expo (ICME), 2010*, p. 920 – 925, 2010.

[31] AYDIN, T. O.; SMOLIC, A.; GROSS, M. Automated aesthetic analysis of photographic images. *IEEE Transactions on Visualization and Computer Graphics*, v. 21, p. 31–42, 2015.

[32] MURRAY, N.; MARCHESOTTI, L.; PERRONNIN, F. Learning to rank images using semantic and aesthetic labels. *British Machine Vision Conference*, p. 1–10, 2012.

[33] MARCHESOTTI, L.; PERRONNIN, F. Learning beautiful (and ugly) attributes. *British Machine Vision Conference*, p. 1–11, 2013.

[34] LU, X. et al. Deep multi-patch aggregation network for image style, aesthetics, and quality estimation. *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), ICCV '15*, p. 990–998, 2015.

[35] MA, S.; LIU, J.; CHEN, C. W. A-lamp: Adaptive layout-aware multi-patch deep convolutional neural network for photo aesthetic assessment. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*, p. 722–731, 2017.

[36] TALEBI, H.; MILANFAR, P. Nima: Neural image assessment. *IEEE Transactions on Image Processing*, p. 3998 – 4011, 2018.

[37] LU, X. et al. Rapid: Rating pictorial aesthetics using deep learning. *Proceedings of the ACM International Conference on Multimedia - MM '14*, p. 457–466, 2014.

[38] KAO, Y.; HE, R.; HUANG, K. Visual aesthetic quality assessment with multi-task deep learning. 2016.

[39] WANG, Z. et al. Brain-inspired deep networks for image aesthetics assessment. *arXiv preprint*, 2016.

[40] MURRAY, N.; MARCHESOTTI, L.; PERRONNIN, F. Ava: A large-scale database for aesthetic visual analysis. *CVPR 2012*, p. 2408–2415, 2012.

[41] ISOLA, P. et al. Understanding the intrinsic memorability of images. *NIPS*, p. 2429–2437, 2011.

[42] SURAN, S.; K., S. Aesthetic quality assessment of photographic images: A literature survey. *International Journal of Computer Applications*, v. 132, p. 11–15, 2015.

[43] KUHN, M.; JOHNSON, K. *Applied Predictive Modeling*. [S.l.]: Springer, 2013.

[44] BOTTOU, L.; BOUSQUET, O. The tradeoffs of large scale learning. *NIPS*, 2007.

[45] HSU, C.-W.; LIN, C.-J. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, v. 13, 2002.

[46] AN Introduction to Statistical Learning with Applications in R. [S.l.]: Springer, 2015.

[47] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. Http://www.deeplearningbook.org.

[48] DONAHUE, J. et al. Decaf: A deep convolutional activation feature for generic visual recognition. In: XING, E. P.; JEBARA, T. (Ed.). *Proceedings of the 31st International Conference on Machine Learning*. Bejing, China: PMLR, 2014. (Proceedings of Machine Learning Research, 1), p. 647–655. Disponível em: <http://proceedings.mlr.press/v32/donahue14.html>.

[49] HOWARD, A. G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. Disponível em: <http://arxiv.org/abs/1704.04861>.

[50] HE, K. et al. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Disponível em: <http://arxiv.org/abs/1512.03385>.

[51] KINGMA, D. P.; BA immy L. Adam: A method for stochastic optimization. *ICLR*, 2015.

[52] TIELEMAN, T.; HINTON, G. *Lecture 6.5 - RMSProp, COURSERA: Neural Networks for Machine Learning*. [S.l.], 2012.

[53] DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 2011.

[54] KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012.

[55] SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. Disponível em: <http://arxiv.org/abs/1409.1556>.

[56] SZEGEDY, C. et al. Rethinking the inception architecture for computer vision.

[57] ZOPH, B. et al. Learning transferable architectures for scalable image recognition. *CoRR*, abs/1707.07012, 2017. Disponível em: <http://arxiv.org/abs/1707.07012>.

[58] SANDLER, M. et al. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018. Disponível em: <http://arxiv.org/abs/1801.04381>.

[59] SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. Disponível em: <http://arxiv.org/abs/1512.00567>.

[60] SZEGEDY, C.; IOFFE, S.; VANHOUCKE, V. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016. Disponível em: <http://arxiv.org/abs/1602.07261>.

[61] SUPPORT Vector Machines and other kernel-based learning methods. [S.l.]: Cambridge University Press, 2005.

[62] BURGES, C. J. C. A tutorial on support vector machines for patternrecognition. *Data Mining and Knowledge Discovery 2*, 1998.

[63] JAKKULA, V. Tutorial on support vector machine (svm). 2006.

[64] ROSASCO, L. et al. Are loss functions all the same? *Neural Computation*, 2004.

[65] MOORE, R. C.; DENERO, J. L1 and l2 regularization for multiclass hinge loss model. *Proc. Symp. on Machine Learning in Speech and Language Processing*, 2011.

[66] TORREY, L.; SHAVLIK, J. Transfer learning. In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. [S.l.]: IGI Global, 2010. p. 242–264.

[67] WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D. A survey of transfer learning. *Journal of Big Data*, v. 3, n. 1, p. 9, May 2016. ISSN 2196-1115. Disponível em: <https://doi.org/10.1186/s40537-016-0043-6>.

[68] PAN, S. J.; YANG, Q. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 22, n. 10, p. 1345–1359, out. 2010. ISSN 1041-4347. Disponível em: <http://dx.doi.org/10.1109/TKDE.2009.191>.

[69] DATTA, R. et al. Studying aesthetics in photographic images using a computational approach. *ECCV*, p. 7–13, 2006.

[70] BALDI, P.; SADOWSKI, P. J. Understanding dropout. Curran Associates, Inc., p. 2814–2822, 2013. Disponível em: <http://papers.nips.cc/paper/4878-understanding-dropout.pdf>.

[71] SMITH, L. N. No more pesky learning rate guessing games. *CoRR*, abs/1506.01186, 2015. Disponível em: <http://arxiv.org/abs/1506.01186>.

[72] WANG, Z. et al. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, v. 13, p. 600–612, 2004.