

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO
CÂMPUS CAMPINAS

THALES AUGUSTO PALETTI POMARI

**ESTUDO COMPARATIVO ENTRE ARQUITETURAS DE DEEP LEARNING
PARA DETECÇÃO DE FALSIFICAÇÃO EM IMAGENS.**

CAMPINAS

2019

THALES AUGUSTO PALETTI POMARI

**ESTUDO COMPARATIVO ENTRE ARQUITETURAS DE DEEP LEARNING
PARA DETECÇÃO DE FALSIFICAÇÃO EM IMAGENS.**

Trabalho de Conclusão de Curso apresentado como exigência parcial para obtenção do diploma do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia Câmpus Campinas.

Orientador: Tiago José de Carvalho

Coorientadora:

CAMPINAS

2019

Ficha catalográfica

Instituto Federal de São Paulo – Câmpus Campinas

Biblioteca

Danielle Sarmento – CRB 8/8669

P784e

Pomari, Thales Augusto Paletti

Estudo comparativo entre arquiteturas de Deep Learning para detecção de falsificação em imagens / Thales Augusto Paletti Pomari. - Campinas, SP: [s.n.], 2019.

45f. : il.

Orientador: Tiago José de Carvalho.

Trabalho de Conclusão de Curso (graduação) – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo Câmpus Campinas. Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, 2019.

1. Máquina - Aprendizado. 2. Classificação. 3. Falsificação. 4. Redes neurais convolucionais. 5. Aprendizado profundo. 6. Iluminação I. Instituto Federal de Educação, Ciência e Tecnologia de São Paulo Câmpus Campinas. Curso de Tecnologia em Análise e Desenvolvimento de Sistemas. II. Título.

Thales Augusto Paletti Pomari

**ESTUDO COMPARATIVO ENTRE ARQUITETURAS DE DEEP LEARNING
PARA DETECÇÃO DE FALSIFICAÇÃO EM IMAGENS**

Trabalho de Conclusão de Curso apresentado
como exigência parcial para obtenção do di-
ploma do Curso de Tecnologia em Análise
e Desenvolvimento de Sistemas do Instituto
Federal de Educação, Ciência e Tecnologia
Câmpus Campinas.

Aprovado pela banca examinadora em: 17 de maio de 2019

BANCA EXAMINADORA



Prof. Dr. Tiago José de Carvalho (orientador)
IFSP Câmpus Campinas

Prof. Dr. Ricardo Barz Sovat
IFSP Câmpus Campinas

Me. Rafael Soares Padilha
UNICAMP

“Dedico este trabalho principalmente aos meus professores, família e amigos, os quais sempre me deram apoio durante todas as dificuldades enfrentadas nesta jornada. Este, que serviu de alicerce não apenas para este trabalho, mas também para o meu futuro.”

RESUMO

Com o grande crescimento no compartilhamento de imagens online, a quantidade de imagens de cunho duvidoso que circulam pelas redes é incalculável. Tendo isso em mente, é necessário algum meio que possa garantir a veracidade de uma foto. Existe uma série de métodos na literatura que são capazes de estimar a chance de uma imagem ser fruto de uma falsificação ou não. Movido por esta ideia central, este trabalho avalia o impacto em um método proposto para detecção das falsificações em imagens de cada rede neural convolucional que já foi o estado da arte. Esta análise foi feita por meio de testes em diferentes bases de dados públicas com variações dos seus espaços de cores, visando um possível melhor desempenho. Espera-se que ao final do trabalho seja possível analisar individualmente cada rede juntamente com um caso de uso adequado.

Palavras-chave: Aprendizado de máquina. Classificação. Falsificação. Redes Neurais Convolucionais. Aprendizado Profundo. Iluminação.

ABSTRACT

With the great growth in online image sharing, the number of dubious images circulating across networks is incalculable. Having this in mind, some means are necessary to guarantee the veracity of a photo. There is a number of approaches in the literature that are capable of calculating whether an image is the result of falsification or not. Moved by this central idea, this work evaluates the impact of each convolutional neural network that has already been the state of art of object detection and recognition in a proposed method for the detection of falsification in images. This analysis will be carried out by means of tests in different public data bases, which will allow for a better performance. It is expected that at the end of the work it will be possible to analyze each network individually with an appropriate use case.

Keywords: Machine learning. Classification. Splicing. Convolutional Neural Networks. Machine Learning. Illumination.

LISTA DE FIGURAS

Figura 1 – Foto em que o Trotsky foi apagado	11
Figura 2 – Exemplo de imagem do tipo composição (<i>Splicing</i>).	12
Figura 3 – Primeiro registro de imagem digital.	15
Figura 4 – Rede proposta por LeCun et al.	19
Figura 5 – Tabela com as redes propostas por Simonyan e Zisserman.	21
Figura 6 – Ideia principal sobre bloco residual desprezando a função de ativação e o bias.	22
Figura 7 – Módulo com redução de dimensão da arquitetura Inception.	23
Figura 8 – SVM com duas classes.	25
Figura 9 – Exemplo do <i>Kernel Trick</i>	26
Figura 10 – Visão geral do método.	28
Figura 11 – Exemplo de imagem <i>splicing</i> que compõe a base de dados DSO.	31
Figura 12 – Exemplo de imagem <i>splicing</i> que compõe a base de dados DSI.	32
Figura 13 – Exemplo de imagem <i>splicing</i> que compõe a base de dados Columbia.	32
Figura 14 – Curva ROC e Matriz de Confusão da VGG19 com a base DSO IIC.	34
Figura 15 – Curva ROC e Matriz de Confusão da ResNet50 com a base DSI IIC.	34
Figura 16 – Curva ROC e Matriz de Confusão da ResNet50 com a base Columbia IIC.	35
Figura 17 – Curva ROC e Matriz de Confusão da ResNet50 com a base DSO GGE.	36
Figura 18 – Curva ROC e Matriz de Confusão da ResNet50 com a base DSI GGE.	36
Figura 19 – Curva ROC e Matriz de Confusão da ResNet50 com a base Columbia GGE.	37
Figura 20 – Curva ROC e Matriz de Confusão da InceptionV3 com a base DSO RGB.	38
Figura 21 – Curva ROC e Matriz de Confusão da VGG19 com a base DSI RGB.	38
Figura 22 – Curva ROC e Matriz de Confusão da ResNet50 com a base Columbia RGB.	39

LISTA DE TABELAS

Tabela 1 – Modelos disponíveis no Keras.	20
Tabela 2 – Exemplo da estrutura de uma matriz de confusão.	27
Tabela 3 – Resultados utilizando o mapa IIC.	33
Tabela 4 – Resultados utilizando o mapa GGE.	35
Tabela 5 – Resultados utilizando o mapa RGB.	37
Tabela 6 – Resultados gerais da acurácia das redes.	39
Tabela 7 – Resultados gerais da acurácia por base de dados.	40

SUMÁRIO

1	INTRODUÇÃO	11
1.1	JUSTIFICATIVA	13
1.2	OBJETIVO GERAL	14
1.3	OBJETIVOS ESPECÍFICOS	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	IMAGENS DIGITAIS	15
2.2	FALSIFICAÇÃO DE IMAGENS	16
2.3	PROPRIEDADES DE ILUMINAÇÃO EM IMAGENS	16
2.4	REDES NEURAIS	17
2.5	REDES CONVOLUCIONAIS	18
2.6	ARQUITETURAS	20
2.6.1	VGG	21
2.6.2	RESNET	22
2.6.3	INCEPTION	22
2.7	APRENDIZADO POR TRANSFERÊNCIA	24
2.8	CLASSIFICADORES	24
2.8.1	SVM	25
2.9	MATRIZ DE CONFUSÃO	26
2.10	CURVAS ROC	27
3	METODOLOGIA	28
3.1	EXTRAÇÃO DOS MAPAS ILUMINANTES	28
3.2	PRÉ-PROCESSAMENTO DA BASE DE DADOS	29
3.3	EXTRAÇÃO DE CARACTERÍSTICAS DAS IMAGENS	29
3.4	CLASSIFICAÇÃO DAS CARACTERÍSTICAS EXTRAÍDAS	30
4	EXPERIMENTOS E DISCUSSÃO	31
4.1	BASES DE DADOS	31
4.2	AMBIENTE COMPUTACIONAL	32
4.3	CONFIGURAÇÕES E PARÂMETROS DOS MODELOS	33
4.4	EXPERIMENTOS	33
5	CONCLUSÃO E TRABALHOS FUTUROS	41
5.1	CONCLUSÃO	41

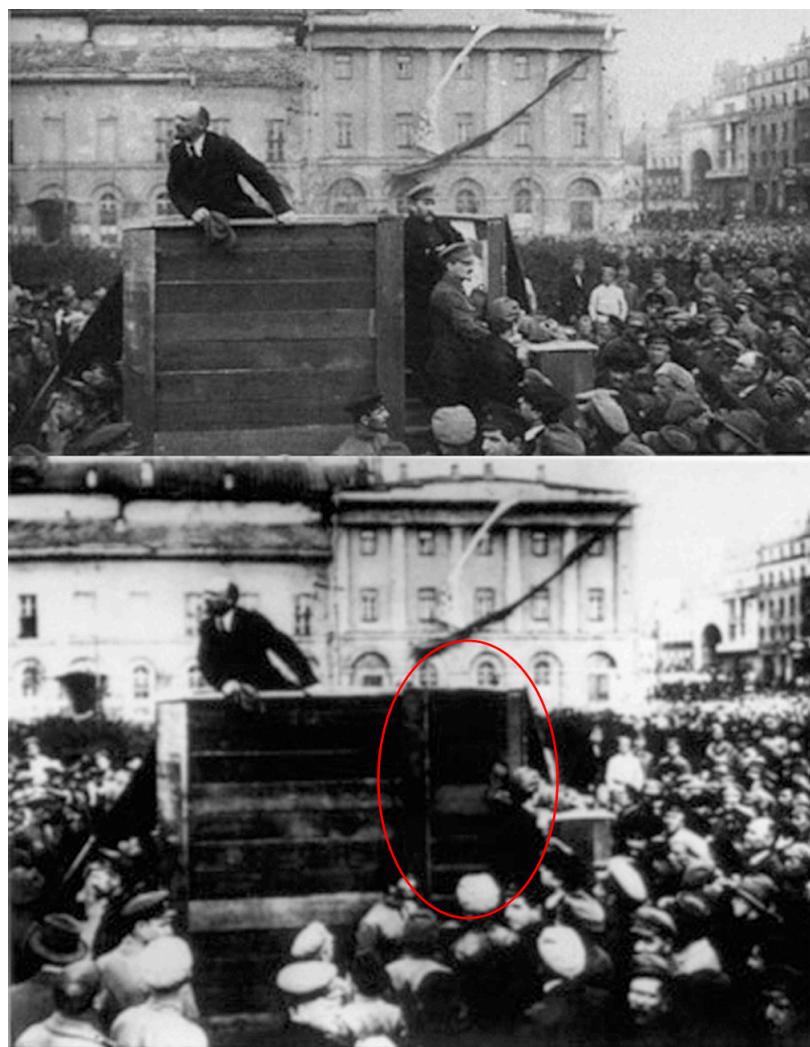
5.2	TRABALHOS FUTUROS	42
-----	-----------------------------	----

REFERÊNCIAS	43
------------------------------	-----------

1 INTRODUÇÃO

O primeiro uso de imagens como provas em processos judiciais data do século XIX, durante a Comuna de Paris. Desde então, este tipo de prática se tornou comum em tribunais ao redor do mundo. Com o passar do tempo e a popularização das fotografias analógicas, vieram também as primeiras falsificações de imagens. Tal ação ficou bem popular durante a ascensão do Partido Comunista na União Soviética, com a subida de Stalin ao poder. Após se desentender com Trotsky em uma disputa interna do partido, Stalin adulterou todas as suas fotos ao lado de Trotsky, apagando o segundo das imagens, como é exemplificado na Figura 1.

Figura 1 – Foto em que o Trotsky foi apagado



Fonte: (GOLDSTEIN, 2018)

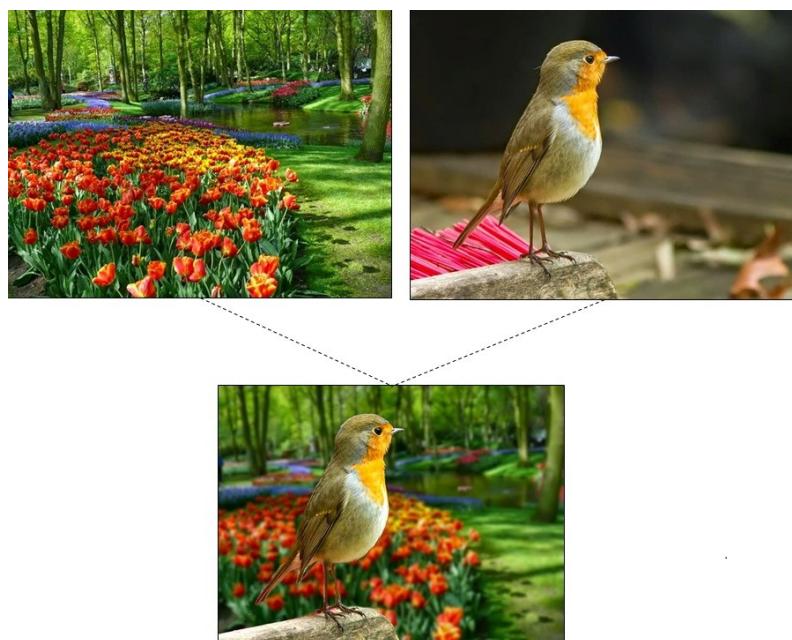
Quando comparamos o volume de falsificações atual com o volume de falsificações das décadas de 40 e 50, por exemplo, podemos perceber que a quantidade de composições cresce juntamente com o volume de momentos registrados. Com o avanço da tecnologia e o surgimento das fotografias digitais, surgiram também diversos softwares que permitem realizar alterações nestas de maneira quase imperceptível. Por isso, é de extrema necessidade a criação de métodos

que ajudem a garantir a veracidade das imagens, para que, quando usadas como provas, órgãos como a Polícia Federal e o Poder Judiciário possa-se ter total certeza de sua autenticidade, de forma a basear suas decisões.

Atualmente existem diversos métodos para realizar a análise de uma imagem quanto à sua autenticidade (DEVAGIRI; CHEDDAD, 2017; ZHANG *et. al*, 2016; FARID, 1999; NG; CHANG, 2004; CARVALHO *et. al*, 2013). Apesar de representarem um grande avanço para a área forense, todos esses métodos possuem algum tipo de restrição. Um dos tipos de restrição mais comuns é a restrição ao tipo de formato da imagem. Como mostrado nos trabalhos de Murali et al. (2013) e Zhang, Li e Wang (2013), estes métodos funcionam apenas em imagens do tipo JPEG e detectam se uma imagem apresenta algum traço de adulteração maliciosa, também chamada de falsificação.

Este trabalho por sua vez, possui um foco no estudo comparativo entre variações das seguintes arquiteturas de *deep learning*: Residual Networks (ResNets), VGGs e Inception, aplicadas ao problema de detecção de falsificações em imagens do tipo *splicing*, que consiste da falsificação pela composição de duas ou mais imagens, como mostrado na Figura 2.

Figura 2 – Exemplo de imagem do tipo composição (*Splicing*).



Fonte: Autor

Como mostrado em trabalhos da literatura (CARVALHO *et. al*, 2013; CARVALHO *et. al*, 2016), imagens criadas através do processo de composição geralmente apresentam inconsistências em suas características de iluminação. Isso porque as partes utilizadas para compor a imagem dificilmente são capturadas sob as mesmas condições de iluminação, dado que possivelmente são capturadas em ocasiões e horários diferentes, sob diferentes fontes de luz.

Adicionalmente, as redes neurais profundas (LECUN; BENGIO; HINTON, 2015) surgi-

ram como uma das grandes promessas na área de processamento de imagens, visão computacional e reconhecimento de padrões, principalmente no campo de reconhecimento de objetos em imagens (SIMONYAN; ZISSERMAN, 2014; SZEGEDY *et. al.*, 2015; HE *et. al.*, 2016).

Desta forma, podemos formular uma interessante questões científica “Quais as diferenças entre diferentes arquiteturas de uma rede neural profunda, desenvolvida com o objetivo de detectar objetos em imagens, para um problema que envolve inconsistências no padrão de iluminação das imagens?”

Adicionalmente, pretendemos investigar também como espaços transformados (CARVALHO *et. al.*, 2013; CARVALHO *et. al.*, 2016) e a constância de cores baseada na borda (WEIJER; GEVERS; GIJSENIJ, 2007) se comportam em conjunto com as redes neurais convolucionais. Uma vez convertida a um espaço de iluminação específico, nossa proposta consiste em utilizar diferentes arquiteturas de redes convolucionais profundas (do inglês *deep convolutional neural networks* - CNN) (JI *et. al.*, 2013; MATSUGU *et. al.*, 2003; KARPATHY; FEI-FEI, 2015), aliadas ao conceito de transferência de conhecimento (do inglês, *transfer learning*) (BENGIO, 2012), (DENG; YU *et. al.*, 2014) para caracterizar tais imagens e permitir sua classificação quanto à autenticidade. As principais contribuições alcançadas com este trabalho são: (1) definir diferentes algoritmos classificadores em conjunto com as arquiteturas de *deep learning*; (2) avaliar diferentes arquiteturas de CNNs na tarefa de detecção de imagens de composição; (3) avaliar a influência de espaços transformados de iluminação quando utilizando CNNs, (4) analisar e comparar desempenho das arquiteturas propostas;

O restante do texto deste trabalho é dividido da seguinte forma: ainda nesta na seção serão apresentados a Justificativa (1.1) e os Objetivos Gerais (1.2) e Específicos (1.3). Na Seção 2 apresentamos os fundamentos teóricos seguidos pela descrição da proposta metodológica contida na Seção 3. Na Seção 4 apresentamos os resultados obtidos. Em sequência são apresentadas algumas conclusões na Seção 5 e, por fim, é listada toda a bibliografia utilizada no trabalho.

1.1 JUSTIFICATIVA

A análise do desempenho de redes neurais convolucionais disponíveis de forma pública juntamente com a definição de suas diferenças e funcionamento é de extrema importância, pois determinadas redes possuem certas particularidades técnicas como tamanho de entrada da imagem, ou o modo como as próprias imagens são processadas por cada bloco de convolução. Dado esse contexto, torna-se importante a realização de uma análise deste tipo de abordagem voltada ao contexto de falsificação de imagens, uma vertente importante da área forense que está se desenvolvendo cada vez mais.

1.2 OBJETIVO GERAL

O objetivo deste trabalho é realizar um estudo comparativo entre variações das seguintes arquiteturas de *deep learning*: Res-Net50, VGG16, VGG19 e Inception, aplicadas ao problema de detecção de falsificações em imagens do tipo *splicing*. Também será feita uma análise dos ganhos proporcionados pela mudança dos espaços de cores da imagem.

1.3 OBJETIVOS ESPECÍFICOS

- Definir diferentes algoritmos classificadores em conjunto com as arquiteturas de *deep learning*;
- Definir algoritmos para realizar a conversão dos espaços de cores das imagens.
- Selecionar e preparar bases de dados de imagens para serem convertidas e utilizadas nos testes das arquiteturas previamente definidas;
- Realizar os experimentos de detecção de falsificações do tipo *splicing*;
- Analisar e comparar o desempenho das arquiteturas selecionadas;
- Analisar e comparar o desempenho entre os diferentes espaços de cores.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar uma revisão bibliográfica que fundamenta este trabalho, apresentando os principais conceitos abordados, como: uma introdução a respeito de imagens, descrevendo o surgimento das imagens digitais, falsificação em imagens e suas propriedades de iluminação; aprendizado de máquina, dando uma explicação geral sobre o conceito base e aprofundando mais nas arquiteturas utilizadas para processar as imagens; e, por fim, abordaremos os conceitos por trás dos métodos de análise que foram usados neste trabalho.

2.1 IMAGENS DIGITAIS

Apesar do primeiro registro de uma fotografia digital ter sido obtido através de um escâner, datado de 1957¹ feito por Russell Kirsch no NIST² (National Institute of Standards and Technology), a primeira máquina fotográfica digital ainda seria concebida quase duas décadas mais tarde.

Figura 3 – Primeiro registro de imagem digital.



Fonte: Russell A. Kirsch [Public domain].

A ideia utilizada por Russell consistia em usar o conceito de *pixel*³ para representar o menor elemento de uma imagem. Cada *pixel*, naquele caso, representava o nível de cinza no ponto correspondente da imagem analógica.

Por volta de 1969, dois pesquisadores dos Laboratórios Bell desenvolveram o CCD (Charge-Coupled-Device), que era um sensor que permitia a conversão da luz em sinais elétricos (BOYLE; SMITH, 1970), sendo o conceito chave utilizado por Steve Sasson, pesquisador da Eastman Kodak e inventor da primeira câmera digital em 1975. Sua invenção era capaz de registrar uma imagem preta e branca e armazená-la de forma digital através de fitas magnéticas, podendo ser vista em um televisor.

¹ <https://www.nist.gov/node/774341>

² Chamado na época de NBS - National Bureau of Standards.

³ O termo *pixel* não possui um registro exato de seu surgimento.

Em 1976 foi patenteado o conceito do Filtro de Bayer, que revolucionou a fotografia digital, permitindo a criação de sensores capazes de registrar imagens coloridas. Tal ideia se baseia na aplicação de padrões das cores primárias (vermelho, verde e azul) ao sensor (BAYER, 1976).

O Padrão de Bayer ainda é uma das bases da imagem colorida que temos hoje, utilizando uma combinação de intensidade das cores primárias para representar a cor de um pixel. A partir disso temos três valores diferentes que juntos representam uma informação, justificando a ideia do uso de redes neurais convolucionais (que será explicado mais afundo na Seção 2.5) para realizar a análise de imagens digitais.

2.2 FALSIFICAÇÃO DE IMAGENS

Como dito por Silva e Rocha (2011) em seu trabalho, atualmente existem tecnologias que permitem, com muita facilidade, realizar falsificações em documentos digitais, até para usuários que não possuem muita experiência. O autor argumenta que, por causa da evolução das tecnologias, a captura de dados em forma de vídeos ou fotos digitais se tornou tão cotidiana, que o uso das mesmas se tornou algo quase natural para as pessoas.

Em virtude deste fato, uma área de pesquisa chamada de Análise Forense de Documentos (AFD) é responsável por tentar detectar com mais clareza as possíveis adulterações em uma imagem. Farid (2008) explica em seu trabalho que este segmento possui como um dos principais objetivos propor métodos para a detecção de adulterações em imagens. Tais métodos podem analisar diversas propriedades da imagem, tais como *pixels*, dispositivo gerador, além de outros.

2.3 PROPRIEDADES DE ILUMINAÇÃO EM IMAGENS

No trabalho de Carvalho et al. (2013), tem-se que a conversão de imagens para outros mapas de cores, ou mapas de iluminantes, possui um ganho no desempenho da acurácia de acordo com a base de dados. Porém, para definir o que são mapas de iluminantes é necessário primeiro ter o conceito do que são fontes de luz. Uma fonte de luz é um corpo que emite energia em cada comprimento de onda no espectro visível em quantidades distintas. A cor de um objeto é algo subjetivo, podendo variar de acordo com a fisiologia e psicologia de quem enxerga e a física que envolve a reflexão da luz em determinado objeto, desta forma sendo difícil a criação de um padrão universal de cores.

Logo, para definir a cor de um objeto, foi criado um sistema chamado CIE (Comissão internacional de Iluminação) em que há um padrão com dez graus diferentes representando a sensibilidade do olho humano em relação a mistura das três cores primárias (vermelho, verde e azul). Então, para nos referirmos à relação da reflexão da onda emitida por uma fonte de

luz com o objeto podemos usar a Distribuição Espectral de Potência Radiante (*Spectral Power Distribuiton* - SPD), que define a concentração, em função do comprimento de onda, de qualquer quantidade de luz emitida (LOPES, 2009).

Compreendendo as definições de cor, fonte de luz e a SPD, os mapas de iluminância se caracterizam pelos números que definem a curva do SPD de uma fonte de luz. Estes mapas, de forma geral, são usados para a medição de cor de um determinado objeto e suas variações geométricas sob um determinado tipo de luz.

Na literatura existem duas classes principais de algoritmos para estimar mapas de iluminância: os que são baseados na física e os que são baseados na estatística. O baseado na física se resume em formulações teóricas de como a luz interage com o objeto. Já os que são baseados em estatísticas dependem da relação estimada de cada pixel da imagem medindo a variação de cor desprezando a influência da luz sobre o objeto refletor.

Tendo em vista esta diferença entre os métodos, decidimos usar neste trabalho uma técnica de cada classe. Na parte dos mapas baseados na física o escolhido foi o *inverse-intensity chromaticity space* (IIC), proposto por Riess e Angelopoulou (2010). Na área dos mapas baseados na estatística escolhemos o *generalized grayworld estimates algorithm* (GGE), que criado e descrito por Weijer, Gevers e Gijsenij (2010);

2.4 REDES NEURAIS

O conceito inicial de simular matematicamente o que um cérebro humano faz começou a ter atenção em 1943, com a publicação “A Logical Calculus Of The Ideas Immanet In Nervous Activity” do neurofisiologista McCulloch e o matemático Walter Pitts, que descreve um modelo matemático simples de um neurônio humano. Em 1958, Rosemblatt descreveu em seu livro “Principles of Neurodynamics” um modelo destinado ao reconhecimento de padrões chamado “Perceptrons”, que apresentava uma organização em camadas de entrada e saída, na qual cada uma era formada por um conjunto de neurônios e suas conexões continham pesos que eram alterados para simular as sinapses.

Rumelhart, Hinton e Williams (1986) descrevem um novo método de aprendizado denominado de retropropagação (backpropagation), que possui uma camada de entrada, uma camada de saída e uma camada oculta. Este procedimento consiste no reajuste repetido dos pesos para que a diferença entre a saída real da rede e a saída esperada seja minimizada. Ainda no mesmo ano, Rumelhart e McClelland editaram o livro “Parallel Distributed Processing”, em que a ideia de um modelo de processamento paralelamente distribuído é adotado. Conforme Haykin (2007), “Estes modelos assumem que o processamento de informação acontece através da interação de um grande número de neurônios, com cada neurônio enviando sinais excitadores e inibitórios para outros neurônios da rede”.

Como uma evolução dos Perceptrons, as redes Multilayer Perceptron, ou MLP, são formadas por neurônios organizados em uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída, em que sua quantidade de neurônios varia de acordo com o domínio do problema. Segundo Braga, Carvalho e Ludemir (2000) estas redes podem ser úteis em problemas de classificação não lineares.

2.5 REDES CONVOLUCIONAIS

Segundo Schmidhuber (2015) as CNN's (*Convolutional Neural Networks*) são uma vertente do aprendizado de máquina profundo (*Deep Learning*). Vertente que pode ser vista como uma variação das MLP's, explicadas na seção anterior, propostas para trabalhar com dados bidimensionais, como imagens. A primeira aparição relevante de CNN's acontece com LeCun et al. (1998), em que o autor descreve uma rede neural com várias camadas treinadas com backpropagation, denominada LeNet5. A rede possuía o objetivo de reconhecer dígitos escritos à mão, conseguindo uma acurácia de 99.2% na base de dados MNIST [The MNIST database of handwritten digits]⁴.

O diferencial desta arquitetura é a camada convolucional, usada para aplicar filtros (ou *kernels*) na imagem de entrada para detectar algumas características, da mais simples à mais complexa. Esses *kernels* representam o tamanho do campo de visão da camada de convolução. Normalmente, o modelo base é constituído por uma camada de convolução que aplica um ou mais filtros sobre uma imagem e o resultado dessas convoluções é aplicado a uma função de ativação não linear. A função de ativação é responsável por determinar se a informação que passou pelo filtro foi relevante para o processo de classificação ou não. Essa camada gera um mapa de características, que é usado como entrada para a próxima camada.

A ideia base para este tipo de rede é o fato da análise da imagem ser feita agrupando os *pixels* de entrada. Segundo Haykin (2009), de modo geral, a arquitetura de uma rede neural convolucional consiste em três partes principais: a extração de características, o mapeamento das características e a subamostragem (*downsampling*).

Durante a extração de características, as informações são analisadas de forma grupal. Em uma imagem, por exemplo, os *pixels* são analisados junto com seus vizinhos, havendo uma busca por grupos que representam algo que possa ser usado para identificar a qual classe final a imagem pertence. Caso uma rede esteja buscando por pessoas em imagens serão analisados grupos que representem informações a qual sua complexidade vai aumentando conforme a profundidade das camadas, por exemplo, as primeiras camadas procuram informações de bordas e mudanças de cores, depois procuram por bocas, narizes, orelhas, até que ao final é buscado uma face completa.

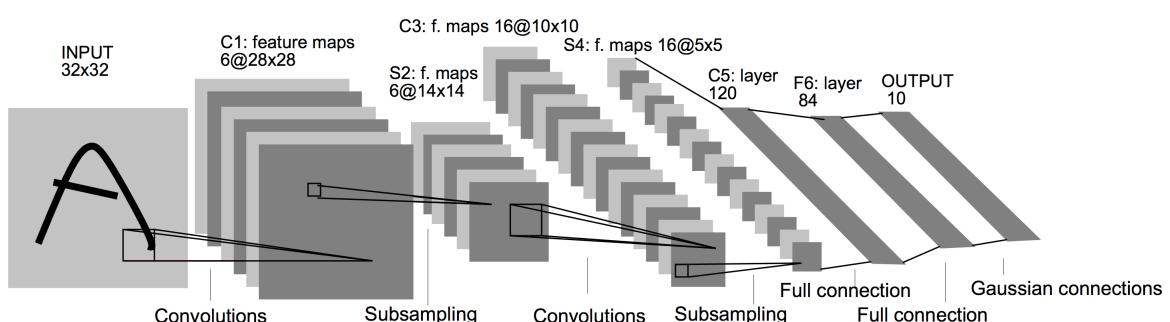
⁴ <http://yann.lecun.com/exdb/mnist/>

O mapeamento das características é responsável pelo filtro (*kernel*) que corresponde aos pesos de cada ligação entre os neurônios. Este peso torna a rede capaz de tratar variações e ruídos na imagem, conseguindo determinar de forma mais complexa se o pedaço analisado é relevante ou não para sua predição.

O *downsampling* é responsável por realizar uma condensação das informações extraídas pelos filtros anteriores a fim de obter uma maior eficiência computacional, como se fosse uma espécie de diminuição na resolução da imagem. Além disso, estas técnicas ajudam na prevenção do *overfitting*, que é caracterizado quando uma rede se torna muito específica para determinada base de dados. Além disto há abordagens como o *pooling* do mapa de características ao final de uma camada convolucional, que é responsável por aplicar uma redução por região no mapa gerado para tornar a amostra mais abstrata, e o *dropout*, que pode ser definido pela eliminação de determinadas conexões durante a interação dos mapas de características entre as camadas a fim de generalizar as informações obtidas (SRIVASTAVA *et. al.*, 2014). Estas abordagens não são de uso obrigatório, porém, além do ganho de processamento, elas aumentam a efetividade da rede.

Por fim, após o grupo das camadas convolucionais, está presente a camada totalmente conectada, que faz a função de classificação dos dados. Diferente da camada convolucional, normalmente todos os neurônios são conectados juntos e são usados para receber dados vetorizados como entrada. Seu objetivo principal é preparar os dados para que eles passem por uma função de ativação que calcula o valor de probabilidade de cada uma das classes existentes de ser a classe correta da imagem de entrada.

Figura 4 – Rede proposta por LeCun et al.



Fonte: (LECUN *et. al.*, 1998)

A Figura 4 descreve a LeNet5 proposta em (LECUN *et. al.*, 1998). Possui cinco camadas diferentes, sendo duas de convolução, duas de subamostragem e uma totalmente conectada. Nesta arquitetura a imagem de entrada possui o tamanho de 32×32 pixels, sendo reduzida até 5×5 antes de entrar nas camadas totalmente conectadas. Após isso, as informações são dispostas em um vetor para serem processadas pelo classificador, que irá realizar uma predição sobre a informação inicial.

2.6 ARQUITETURAS

As redes utilizadas neste trabalho foram selecionadas através da seleção dos cinco melhores resultados após um teste preliminar com todas as arquiteturas de *deep learning* disponíveis na biblioteca de rede neural para python Keras⁵. A Tabela 1 apresenta todas as arquiteturas disponíveis para o problema de classificação de imagens, juntamente com sua quantidade de parâmetros.

Tabela 1 – Modelos disponíveis no Keras.

Modelo	Parâmetros
Xception	22,910,480
VGG16	138,357,544
VGG19	143,667,240
ResNet50	25,636,712
ResNet101	44,707,176
ResNet152	60,419,944
ResNet101V2	44,675,560
ResNet152V2	60,380,648
ResNeXt50	25,097,128
ResNeXt101	44,315,560
InceptionV3	23,851,784
InceptionResNetv2	55,873,736
MobileNet	4,253,864
MobileNetv2	3,538,984
DenseNet121	8,062,504
DenseNet169	14,307,880
DenseNet201	20,242,984
NASNetMobile	5,326,716
NASNetLarge	88,949,818

Fonte: Autor

As arquiteturas que obtiveram um maior desempenho de acurácia foram a VGG16 e VGG19, que variam de um modelo criado por Simonyan e Zisserman (2014), pesquisadores da Visual Geometry Group da Universidade de Oxford. O grupo obteve os melhores resultados no *Image Net Large Scale Visual Recognition Challenge* (ILSVRC⁶) em 2014, ficando em primeiro lugar na categoria de localização de objetos em imagens e segundo lugar na categoria de classificação de conteúdo com esta abordagem. A ResNet50, que foi descrita por pesquisadores da Microsoft Research (HE *et. al*, 2016), foi vencedora do ILSVRC em 2015 nas categorias de classificação de imagens, localização e detecção de objetos. Por fim, as arquiteturas InceptionV3 e InceptionResNetV2 são variações do modelo descrito por pesquisadores da Google (SZEGEDY

⁵ <https://keras.io/>

⁶ <http://www.image-net.org/challenges/LSVRC/>

et. al, 2015) ficando em segundo lugar no ILSVRC14 nas tarefas de classificação e localização em imagens.

Nas próximas seções as principais ideias que diferenciam as arquiteturas utilizadas serão descritas.

2.6.1 VGG

A VGG16 é compostas por seis blocos, sendo os cinco primeiros convolucionais, e o sexto um bloco totalmente conectado. Na VGG16, o primeiro bloco possui duas camadas com sessenta e quatro filtros 3×3 , e após a última camada de convolução é acoplado uma camada de *max pooling* 2×2 . O segundo bloco convolucional segue o padrão do primeiro, porém a diferença esta no número de filtros, que são duplicados. O terceiro, quarto e o quinto bloco possuem a mesma quantidade de camadas convolucionais com filtros 3×3 e cada bloco é seguido por um *max pool* com *kernel* igual a 2×2 . O bloco classificador possui três camadas totalmente conectadas, sendo que duas delas possuem 4096 nós e a última é uma camada de saída com 1000 nós que utilizam uma função de ativação da Softmax.

Figura 5 – Tabela com as redes propostas por Simonyan e Zisserman.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Fonte: (SIMONYAN; ZISSERMAN, 2014)

A diferença entre a VGG16 e VGG19 é a adição de uma camada convolucional no quarto,

quinto e sexto bloco, totalizando três camadas a mais. A Figura 5 descreve todas as variações apresentadas pelos autores em seu trabalho, mas apenas serão avaliados as variações D e E.

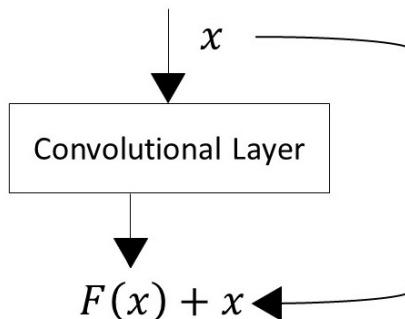
2.6.2 RESNET

Redes neurais residuais (ResNets) são rede neurais convolucionais profundas, desenvolvidas por pesquisadores da Microsoft. O principal aspecto que particulariza esta variação é o conceito de bloco residual. Em geral, este tipo de arquitetura usa atalhos entre as camadas, adicionando os valores de entrada na função ReLu (NAIR; HINTON, 2010). A Equação 2.1, definida por He et. al(2016), explica como este atalho funciona.

$$y = F(x, Wi) + x. \quad (2.1)$$

Tendo em mente que a dimensão de x e F são iguais, He et. al(2016) explicam que o y representa a saída e o x representa a entrada do bloco de atalho. $F(x, Wi)$ é a função que define o mapa residual que foi gerado pelo boco. Se as dimensões forem diferentes, uma projeção linear é usada em x para igualar com a dimensão de F . A saída y passa pela função ReLU para ser normalizada para o próximo bloco. A Figura 6 demonstra a ideia do bloco residual.

Figura 6 – Ideia principal sobre bloco residual desprezando a função de ativação e o bias.

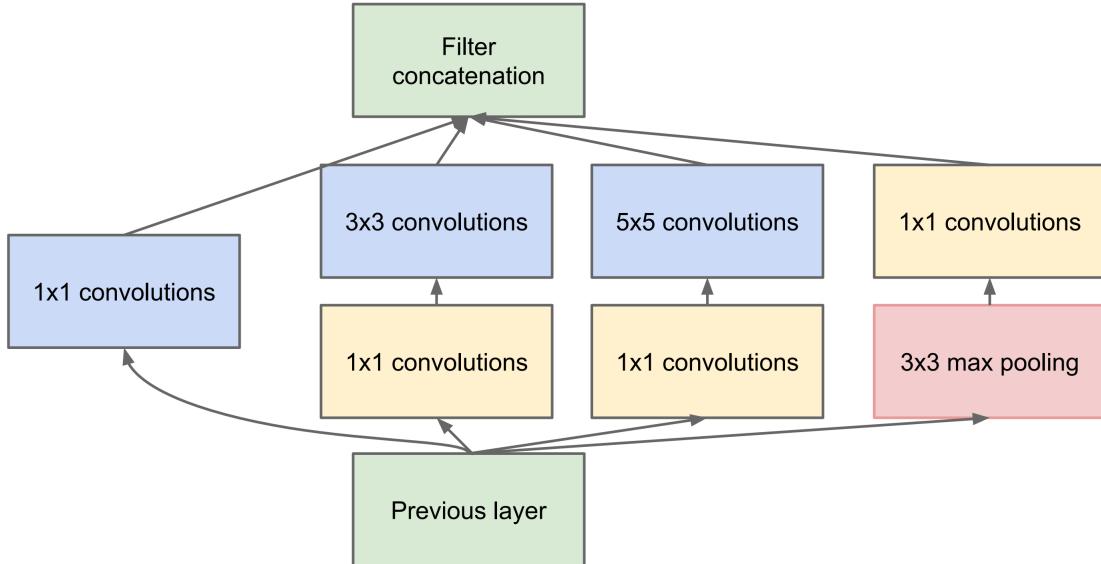


Fonte: Autor

2.6.3 INCEPTION

A principal ideia dos módulos usados nesta arquitetura é usar variações de tamanho de filtros juntos com uma camada de *pooling* e concatenar a saída de todos os filtros no final, como descrito na Figura 7. Diferente das demais, nesta abordagem cada módulo pode ser comparado com uma mini rede neural convolucional sem o classificador. Outro ponto chave da Inception é o uso de camadas 1×1 para reduzir o tamanho do mapa de características de entrada tornando convoluções maiores mais fáceis de lidar, computacionalmente falando.

Figura 7 – Módulo com redução de dimensão da arquitetura Inception.



Fonte: (SZEGEDY *et. al*, 2015)

Para tornar este conceito mais palpável, pode-se pensar na situação em que temos um tamanho de entrada de $28 \times 28 \times 192$ seguido por uma camada convolucional, chamada Camada Convolucional A, a qual possui 32 filtros 5×5 . Sabendo que cada filtro diminui todos os canais da imagem de entrada em apenas um vetor, a saída desta camada seria $28 \times 28 \times 32$. O mapa de saída precisa ter o mesmo tamanho de entrada, pois nenhum tipo de *downsampling* foi realizado. O número de operações matemáticas necessários para realizar a convolução do filtro nos dados de entrada por camada convolucional é denotado pela Equação 2.2.

$$M = (Wi \times Hi \times F) \times (Sc^2 \times Ci) \quad (2.2)$$

O Wi representa a largura do vetor de entrada e Hi representa sua altura, F é o número de filtros, Sc é o tamanho do filtro e Ci é o número de canais do vetor de entrada. Deste modo, obtém-se $M \approx 120,000,000$.

Adicionando uma camada de convolução (Camada Convolucional B) que possui 16 filtros 1×1 antes da camada convolucional proposta inicialmente, e também ajustando a entrada da Camada Convolucional A para se igualar aos filtros da Camada Convolucional B recém adicionada, nós teremos uma entrada com tamanho de $28 \times 28 \times 192$, uma Camada Convolucional B contendo 16 filtros $1 \times 1 \times 192$, uma Camada Convolucional A contendo 32 filtros de $5 \times 5 \times 16$ e um mapa de características com tamanho de $28 \times 28 \times 32$.

Inicialmente, a convolução da Camada B na entrada resultará em uma saída $28 \times 28 \times 16$, servindo como entrada da Camada A que resultará em um novo mapa $28 \times 28 \times 32$. A saída final desta nova rede possui o mesmo tamanho da primeira versão proposta, sem a Camada B.

Contudo, o cálculo do número de operações matemáticas realizadas será feito do mesmo modo, porém somando as operações da Camada A e B. Para a Camada A, temos $Ma \approx 10,000,000$, enquanto para a B temos $Mb \approx 2,400,000$, totalizando um valor de aproximadamente 12,4 milhões de cálculos, número quase $10\times$ menor em relação ao primeiro cenário. Isso demonstra a efetividade do uso de filtros 1×1 nos blocos de convolução.

2.7 APRENDIZADO POR TRANSFERÊNCIA

A técnica de aprendizado por representação (*transfer learning*) consiste na utilização dos pesos das conexões de uma rede que foi treinada para um problema *A* em outro problema *B*, de forma que a rede não precise realizar todo o processo de aprendizado novamente para extrair as características dos dados (PAN; YANG *et. al*, 2010). O ajuste dos pesos de uma rede requer um alto poder computacional, uma grande quantidade de tempo e um conjunto de dados com milhares de imagens, que apesar de ser o melhor cenário, não necessariamente ocorre sempre.

Apesar desta técnica poder ser aplicada para qualquer tipo de problema independente de seu domínio, existem casos em que a utilização pode apresentar resultados mais relevantes comparados a outros. Como apresentado por Bengio, Courville e Vincent (2013), quando os dados dos problemas são do mesmo tipo as chances de obter um melhor resultado são maiores. Por exemplo, usar os pesos de um problema voltado ao reconhecimento de fala provavelmente não trará as mesmas características que seriam eficazes para detecção de pessoas em imagens. Assim como a semelhança do domínio dos dados, uma rede treinada para detectar pessoas em imagens não terá um desempenho eficaz o suficiente para detectar aviões, por exemplo. Outro ponto relevante levantado é a quantidade de amostras utilizadas para treino: quanto maior o número de amostras, melhor o desempenho (BENGIO; COURVILLE; VINCENT, 2013).

2.8 CLASSIFICADORES

O classificador de uma rede tem a função de atribuir uma classe para cada amostra processada a partir de cada vetor de característica extraído. Esta etapa pode ser explicada como um agrupamento por semelhanças que a própria rede aprende baseada nos dados utilizados. A classificação pode ser dividida em duas abordagens diferentes: o aprendizado não supervisionado e o supervisionado.

No não supervisionado, os dados não são rotulados e a tarefa principal do classificador é buscar características que possam separar os dados em n classes de acordo com o problema (BARLOW, 1989). Este tipo de aprendizado é mais empregado quando o resultado que os dados devem apresentar não são claros. Isto pode ser exemplificado quando temos uma empresa com 500 projetos, por exemplo, e devemos agrupar estes projetos de acordo com características como

valores estimados, tempo de execução, número de requisitos, ou alguma outra relação mais genérica.

Já no paradigma supervisionado nós temos como entrada dados rotulados, isso quer dizer que o resultado dos dados utilizados para treinar o classificador é conhecido previamente e a tarefa de classificação acaba sendo a busca de correlações entre as características dos dados por classe (CARUANA; NICULESCU-MIZIL, 2006). Ainda existe uma subdivisão dentro desta abordagem, que podemos classificar em regressão e classificação. Em problemas de regressão, utilizamos os dados de entrada para prever uma saída com dados contínuos, já no de classificação deve-se gerar saídas com dados discretos.

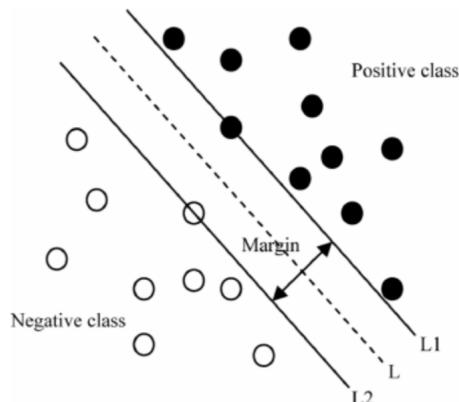
O problema abordado neste trabalho é um problema supervisionado de classificação, pois além de passar as classes durante o treinamento do classificador, precisamos prever se há ou não falsificação nas imagens. Para este problema se encaixar no caso de regressão, poderíamos tentar estimar a quantidade de falsificações contidas nas imagens, por exemplo.

As redes que serão utilizadas neste trabalho foram descritas com classificadores próprios, que são uma camada totalmente conectada na qual um neurônio apenas será ativado no final, representando sua classe segundo a rede. Contudo, a partir de testes prévios entre diferentes classificadores optamos por usar um classificador diferente dos padrões de cada rede. A partir de testes prévios, temos a Máquina de Vetores de Suporte (do inglês Support Vector Machine, ou SVM), que será explicada na Seção 2.8.1, como classificador para este problema.

2.8.1 SVM

Em sua essência, este algoritmo procura o hiperplano com a maior margem possível entre os dados criando uma espécie de fronteira das classes (SUYKENS; VANDEWALLE, 1999). A categorização de novas amostras se dá pela posição que essa amostra se encontra em relação ao hiperplano. A Figura 8 explica de forma visual o conceito básico desta abordagem.

Figura 8 – SVM com duas classes.



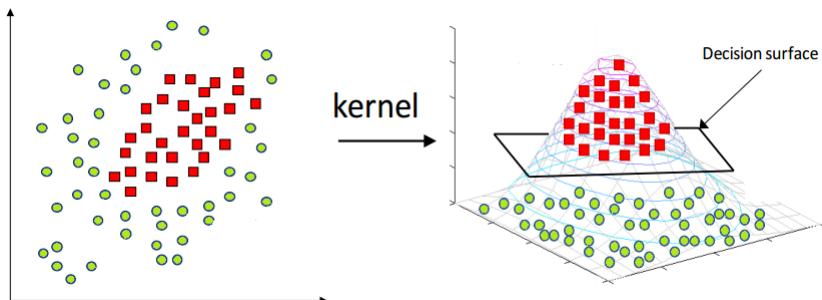
Fonte: (LI *et. al*, 2013)

Caso uma nova amostra estivesse entre a linha L e a L_2 , ela seria classificada como pertencente a classe Negativa. A probabilidade dela pertencer a cada classe pode ser calculada a partir da distância entre a amostra e a margem L em relação à margem da classe predita.

Contudo, nem sempre os dados utilizados são simples de serem separados como na Figura 8. Por este motivo, existem algumas otimizações que podem ser adotadas para adaptar este tipo de classificador ao problema. Técnicas como a aplicação de álgebra linear no espaço (chamada de *Kernel Trick*) (JAKKULA, 2006), variação do parâmetro de regularização que permite uma certa tolerância de erro em relação às amostras para separação do hiperplano (SCHÖLKOPF et. al, 2002), ou, em alguns casos, o parâmetro chamado de *Gamma* que indica até qual distância as amostras conhecidas podem influenciar amostras novas (SCHÖLKOPF et. al, 2002).

O *Kernel Trick* é uma técnica que pode ser utilizada quando os dados não são linearmente separáveis em um espaço bidimensional. Uma fórmula pode ser aplicada nos dados dando mais dimensões ao espaço, como mostrado na Figura 9. Dentre as várias fórmulas existentes como linear, não-linear, polinomial, *radial basis function (RBF)*, e *sigmoid*. (Hussain et. al, 2011), adotaremos a RBF devido ao seu uso padrão na biblioteca utilizada.

Figura 9 – Exemplo do *Kernel Trick*.



Fonte: Rashmi Jain.

2.9 MATRIZ DE CONFUSÃO

A matriz de confusão é uma forma de análise do resultado de classificações binárias. No caso deste trabalho, classificação corretas das imagens (verdadeiros positivos e verdadeiros negativos) pelas classificações erradas (falsos positivos e falsos negativos). Neste tipo de visualização a diagonal principal contém as taxas de acertos por classe, enquanto na outra diagonal se localizam as taxas de erros. Com a matriz é possível também calcular a confiabilidade positiva ou negativa da rede, além da precisão total, sensitividade e especificidade da rede, que são dados essenciais para a análise de desempenho (MONARD; BARANAUSKAS, 2003).

A Tabela 2 representa o modelo que será adotado neste trabalho. Cada linha representa a classe real da imagem, enquanto a coluna representa a predição realizada pelo classificador. Nas matrizes que representarão os resultados, a informação contida nos campos "Classificação

"Correta" e "Classificação Incorreta" correspondem a porcentagem cuja soma representa a acurácia da rede.

Tabela 2 – Exemplo da estrutura de uma matriz de confusão.

	Verdadeira	Splicing
Verdadeira	Classificação Correta	Classificação Incorreta
Splicing	Classificação Incorreta	Classificação Correta

Fonte: Autor

2.10 CURVAS ROC

Curvas ROC (*Receiver Operating Characteristic*) podem ser definidas como a representação direta entre a sensibilidade e a especificidade de um teste, onde a sensibilidade representa a proporção de verdadeiros positivos enquanto a especificidade representa a proporção de verdadeiros negativos. Neste caso, levando em conta que as amostras positivas representam as imagens que apresentam falsificações, o eixo y irá representar a sensibilidade mostrando a taxa de quantas amostras falsificadas a rede classificou corretamente, enquanto o eixo x irá mostrar a quantidade de imagens verdadeiras que a rede classificou erroneamente através da subtração $\text{especificidade} - 1$. Cada ponto do plano representa a relação entre sensibilidade e especificidade de acordo com um limite, comumente chamado de *threshold*. (FAWCETT, 2006)

O cálculo da sensibilidade é feito através da taxa de verdadeiros positivos (VP) em relação a taxa de falso negativo (FN), como demonstrado na Equação 2.3. Já a especificidade é obtida através da relação entre o número de falsos positivos (FP) com os verdadeiros positivos (VP), demonstrando a quantidade de amostras falsas classificadas como verdadeiras, representado pela Equação 2.4.

$$\text{Sensibilidade} = \frac{VP}{VP - FN} \quad (2.3)$$

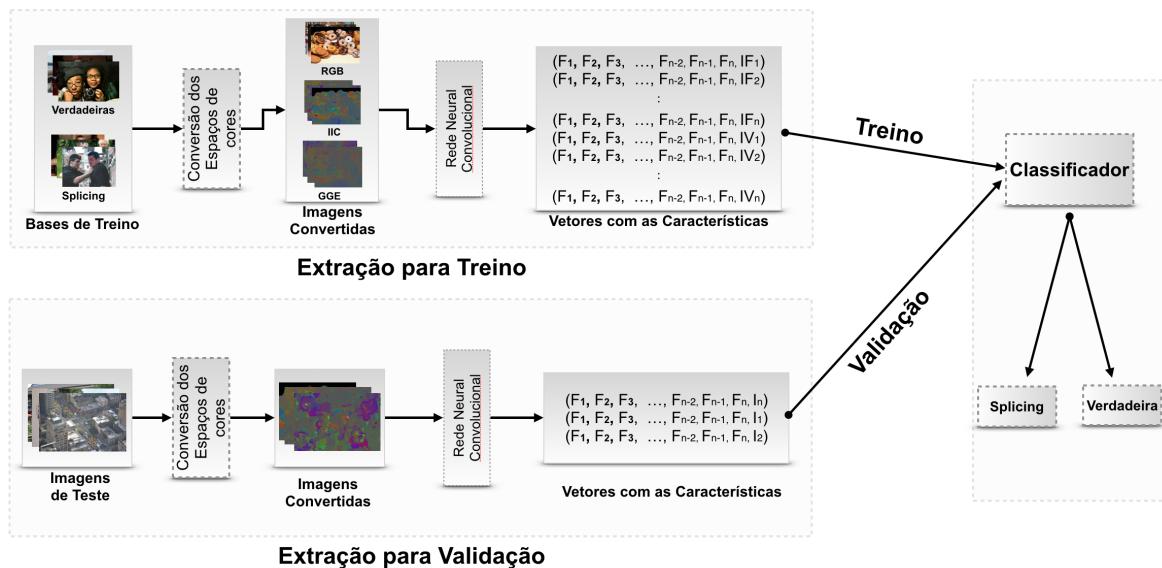
$$\text{Especificidade} = \frac{FP}{FP - VN} \quad (2.4)$$

As curvas ROC podem ser vistas como um meio de facilitar a decisão de qual limite de erro aplicado ao problema traz o melhor desempenho. Ao mesmo tempo, a área sob a curva (AUC, *Area Under Curve*) também pode ser uma métrica relevante na escolha de qual algoritmo obteve uma melhor performance, em que quanto maior a área melhor o desempenho geral do algoritmo (CRISTIANO, 2017).

3 METODOLOGIA

Durante todo o trabalho desenvolvido a mesma metodologia foi utilizada em todos os testes. A única diferença foi apenas a rede neural convolucional utilizada para realizar a extração das características das imagens: todos os parâmetros do classificador foram mantidos constantes entre os testes. A metodologia proposta está descrita de forma visual na Figura 10 e pode ser dividida entre (1) conversão dos mapas de cores, (2) pré-processamento dos dados, (3) extração das características e (4) classificação das características extraídas.

Figura 10 – Visão geral do método.



Fonte: Autor

3.1 EXTRAÇÃO DOS MAPAS ILUMINANTES

Como primeiro passo do método fazemos a conversão de todas as imagens que serão utilizadas durante o experimento. Utilizamos a imagem em sua dimensão original, para que não haja perda de informação, para aplicar os algorítimos dos dois espaços de cores utilizados.

Para realizar a conversão para o espaço IIC (inverse-intensity chromaticity), utilizaremos o algoritmo descrito por Riess e Angelopoulou (2010) com os parâmetros descrito por Carvalho et. al (2016). O algoritmo consiste em segmentar a imagem em grupos de *pixels* que possuem cores semelhantes, chamados de *superpixels*, e calculam a intensidade e a cromaticidade de cada canal de cor dos *superpixels* de acordo com as influências geométricas presentes na imagem, como a superfície de um objeto por exemplo. Ao final um mapa é montado utilizando a combinação das informações obtidas em cada um dos três canais diferentes dos *superpixels* analisados.

O Generalized Grayworld Estimates (GGE), descrito por (WEIJER; GEVERS; GIJSENIJ, 2007), procura estimar a cor da fonte de luz a partir da cromaticidade dos *pixels* presentes na

imagem. Esta abordagem se baseia na ideia de que sob uma fonte de luz branca, não há variação cromática na cor média dos *pixels*. Caso a imagem seja fruto de uma composição, há a hipótese de que esse espaço de cor possa ressaltar variações entre artefatos que inicialmente não faziam parte da mesma captura.

3.2 PRÉ-PROCESSAMENTO DA BASE DE DADOS

Nesta etapa do método ocorre uma adaptação da imagem para que ela possa servir corretamente como dados de entrada no formato que cada rede espera. As imagens são carregadas e redimensionadas para as dimensões de entrada padrão de cada rede, que devem ser matrizes de $224 \times 224 \times 3$ para a ResNet e VGG, e $299 \times 299 \times 3$ para a Inception.

A biblioteca Keras possui funções de pré-processamentos para imagens¹, em que ocorre o carregamento e o redimensionamento da mesma. Logo após, utilizamos outra função que realiza a separação dos *pixels* da imagem por canal para corresponder as matrizes de entrada esperadas de cada rede.

Com a imagem representada por uma matriz $L1 \times L2 \times C$, em que $L1$ e $L2$ representam a altura e largura representada pelos valores dos *pixels*, e C sendo a quantidade de canais da imagem. Neste caso $C = 3$, devido ao RGB, estes valores devem ser adicionados ao conjunto de amostras que será usado pela rede. Esse conjunto recebe o nome de *batch* e é representado por uma matriz $S \times L1 \times L2 \times C$, em que o S corresponde ao número de amostras carregadas que serão usadas.

Contudo, algumas arquiteturas não utilizam o valor padrão dos *pixels* variando de 0 a 255, podendo ser entre 0 e 1 ou entre -1 e 1. Logo, utilizamos uma função do Keras que normaliza os valores de acordo com as arquiteturas e adiciona a imagem à matriz de *batches*. Após todo esse processo ocorrer em todas as imagens, elas já se encontram preparadas para a extração de características.

3.3 EXTRAÇÃO DE CARACTERÍSTICAS DAS IMAGENS

Nesta etapa do método irá ocorrer a alimentação da rede neural convolucional, resultando na criação de vetores de características representando todas as informações que a rede definiu como relevantes para o processo de classificação perante cada imagem processada.

Através da utilização da técnica de *Transfer Learning*, explicada na Seção 2.7 e em que não há necessidade de treinamento e ajuste dos pesos da rede, este processo se resumirá em processar as imagens extraíndo seus vetores de características, sem realizar o treinamento do modelo extrator.

¹ <https://keras.io/preprocessing/image/>

O conceito base desta etapa pode ser explicado com a rede aprendendo uma função matemática baseada em dados A e aplicando essa função para dados B . O aprendizado de todas as redes ocorreu utilizando uma base de dados chamada de ImageNet (RUSSAKOVSKY *et. al.*, 2015), em que existem aproximadamente 14.2 milhões de imagens. Os pesos utilizados foram obtidos durante o ImageNet Challenge, que é uma competição de processamento de imagem em que times participantes enviam os resultados de suas arquiteturas voltadas para a solução dos problemas de detecção e localização de objeto e classificação da cena contida.

3.4 CLASSIFICAÇÃO DAS CARACTERÍSTICAS EXTRAÍDAS

Após o término do processo de extração de características o vetor resultante será dividido em dois novos, um de treino e outro de teste. Então, será instanciado um modelo do classificador, que receberá inicialmente o vetor de treino e as classes que cada vetor representa para que possa ocorrer o treino do modelo.

Assim que o treino for completo, o classificador receberá como entrada o vetor de características com as informações que foram separados para a validação do treino. Como saída desta etapa teremos um vetor com as previsões das amostras feitas pelo classificador. Tais previsões serão comparadas com as classes originais das imagens respectivas e esse resultado será utilizado para calcular o desempenho do classificador utilizando as métricas descritas na Seção 2.9.

4 EXPERIMENTOS E DISCUSSÃO

Este capítulo apresenta informações sobre as bases utilizadas (Seção 4.1), assim como o ambiente computacional em que os testes foram feitos (Seção 4.2) e a descrição dos parâmetros utilizados nas configurações das redes neurais convolucionais (Seção 4.3). Também estão descritos todos os resultados obtidos durante as baterias de testes realizadas juntamente com uma análise dos mesmos (Seção 4.4), com destaque para os melhores por espaço de cor.

4.1 BASES DE DADOS

Três bases de dados públicas comuns na literatura foram selecionadas para realizar a comparação entre as arquiteturas. Cada uma das três possui um foco de falsificação de composição diferente.

A base DSO, criada e descrita por (CARVALHO *et. al.*, 2013), é composta por 200 imagens tanto com iluminação natural quanto artificial com o intuito de simularem montagens do mundo real, a fim de enganar o observador, como pode ser observado na Figura 11. A segunda base de dados foi a DS1 (CARVALHO *et. al.*, 2013), que é composta por 50 imagens que representam imagens mais voltadas as que são circuladas na mídia, como em capas de revistas ou reportagens, exemplificado na Figura 12. Por último, a base Columbia (HSU; CHANG, 2006), fornecida por DVMM Laboratory of Columbia University¹, consiste em 363 imagens em que as falsificações não se preocupam com a semântica da imagem, representada pela Figura 13.

As bases DSO e DS1 estão divididas igualmente entre verdadeiras e falsas, enquanto a Columbia possui 183 imagens autenticas e 180 imagens falsificadas.

Figura 11 – Exemplo de imagem *splicing* que compõe a base de dados DSO.



Fonte: (CARVALHO *et. al.*, 2013)

¹ <http://www.ee.columbia.edu/ln/dvmm/downloads/authsplcuncmp/>

Figura 12 – Exemplo de imagem *splicing* que compõe a base de dados DSI.



Fonte: (CARVALHO *et. al*, 2013)

Figura 13 – Exemplo de imagem *splicing* que compõe a base de dados Columbia.



Fonte: (HSU; CHANG, 2006)

4.2 AMBIENTE COMPUTACIONAL

A máquina utilizada para criação e execução dos *scripts* utiliza um processador Intel® Xeon® E5-2620, com 100 *gigabytes* de memória RAM e uma placa gráfica Nvidia GeForce GTX Titan X.

A respeito do código, o método proposto foi implementado utilizando a linguagem de programação Python 3.5, utilizando as bibliotecas Keras 2.0.3², e TensorFlow 1.0.1³ para criação e execução das redes neurais convolucionais extratoras de características. Também foi usado a biblioteca NumPy 1.14.0 para armazenar e tratar os vetores de características extraídos. Também foi utilizado o Scikit-Learn 0.19.1 para criação do classificador. O Matplotlib 2.1.2 foi utilizado para gerar gráficos dos resultados, como a matriz de confusão e as curvas ROC.

² <https://keras.io>

³ <https://www.tensorflow.org>

4.3 CONFIGURAÇÕES E PARÂMETROS DOS MODELOS

Durante os testes utilizando *transfer learning*, todas as redes foram compiladas usando seus respectivos pesos obtidos durante o ImageNet Challenge, adquiridos diretamente via parâmetro pela biblioteca Keras. O tipo de *pooling* selecionado foi o *average* como padrão para todos os modelos. Como neste caso todas as redes foram carregadas sem a camada totalmente conectada para classificação, no lugar usamos o SVM SVC⁴ com sua configuração padrão, em que o *kernel* utilizado é o Radial basis function (RBF), que é o padrão, e a com o parâmetro *probability* = *True*, para que a probabilidade de cada amostra pertencer a cada classe seja armazenada. A *seed* dos números randômicos da biblioteca NumPy foi estabelecida em 1, o tamanho de entrada para imagem foi de 224,224 para a VGG16, VGG19 e ResNet50, enquanto para InceptionV3 e a InceptionResNetV2 foi de 299,299.

Em todos os experimentos foi utilizado o K-fold⁵ para realizar a validação cruzada do classificador visando seu poder de generalização, devido a quantidade de amostras. Cada base de dados foi repartida em 5 subconjuntos distintos, para que parte deles sejam utilizados para treino e outra parte validação do treino, de forma que cada amostra seja utilizada ao menos uma vez para validar o treino do classificador sem estar no conjunto de testes (KOHAVI *et. al*, 1995).

4.4 EXPERIMENTOS

Para a realização dos experimentos seguimos a metodologia apresentada na Figura 3, variando as redes convolucionais utilizadas para extração de características. A seguir são apresentados os resultados obtidos, a matriz de confusão junto com a curva ROC para cada rede que obteve o melhor desempenho de acordo com a base de dados e o espaço de cor utilizado. As Tabelas 3, 4 e 5 referem-se aos espaços de cores IIC, GGE e RGB, respectivamente. Cada tabela apresenta a porcentagem de acerto das redes em cada uma das três bases de dados utilizadas.

Tabela 3 – Resultados utilizando o mapa IIC.

IIC			
Rede	DSO	DSI	Columbia
ResNet50	91%	90%	82,3%
VGG16	91,5%	86%	77,9%
VGG19	94%	84%	78,5%
InceptionV3	87%	84%	63,1%
InceptionResNetV2	93%	86%	51,8%

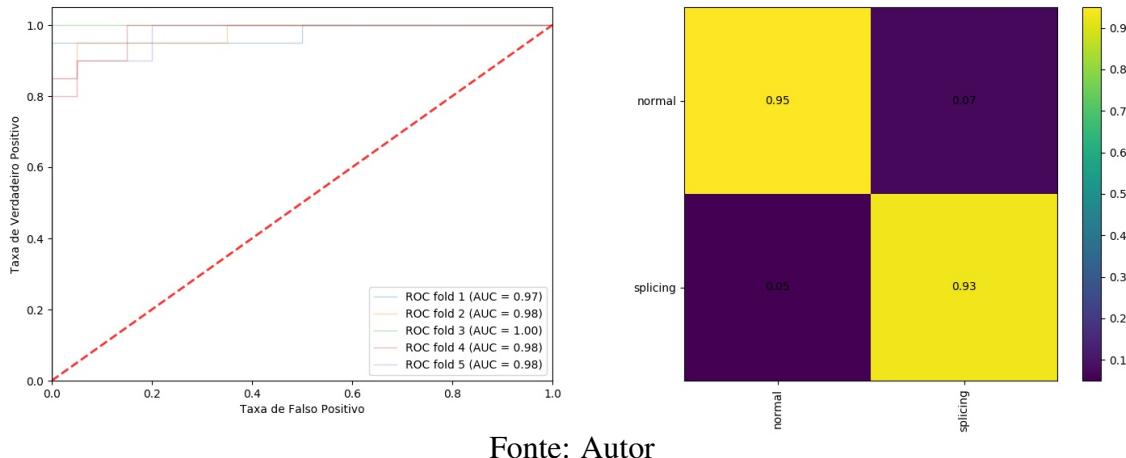
Fonte: Autor

⁴ <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

⁵ https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html

Como pode ser observado na Tabela 3, o melhor resultado obtido na base de dados DSO foi de 94% utilizando o modelo VGG19, enquanto para o DSI e o Columbia a ResNet50 alcançou respectivamente 90% e 82,3% de acurácia.

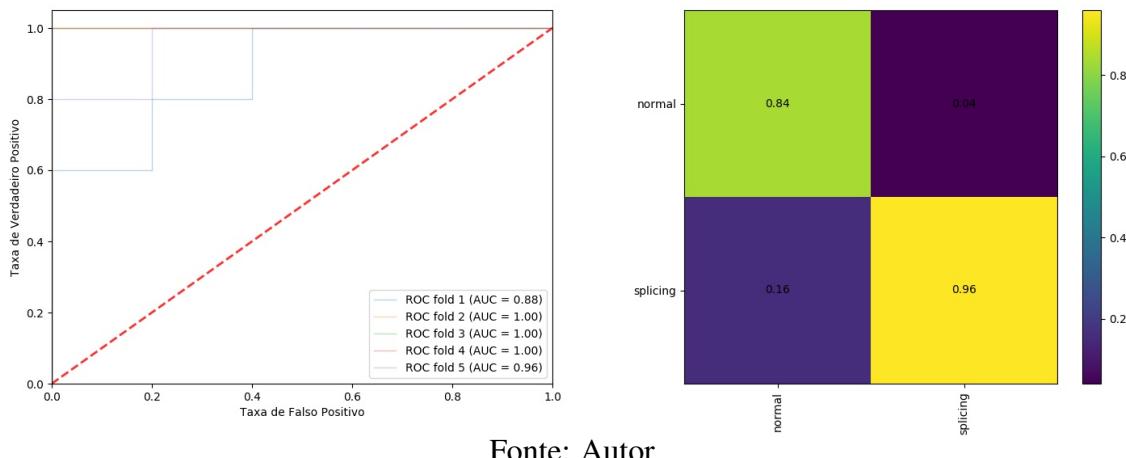
Figura 14 – Curva ROC e Matriz de Confusão da VGG19 com a base DSO IIC.



Fonte: Autor

A Figura 14 apresenta respectivamente a curva ROC e a matriz de confusão gerados durante o teste do modelo VGG19 utilizando a base de dados DSO convertida para o espaço de cor IIC. Nota-se um desempenho levemente maior no acerto da classificação das imagens normais em relação as imagens falsas. A diferença de área entre a pior curva em relação a melhor é pequena em relação aos outros resultados entre as outras bases convertidas para o mesmo espaço de cor, sendo equivalente a 3%.

Figura 15 – Curva ROC e Matriz de Confusão da ResNet50 com a base DSI IIC.

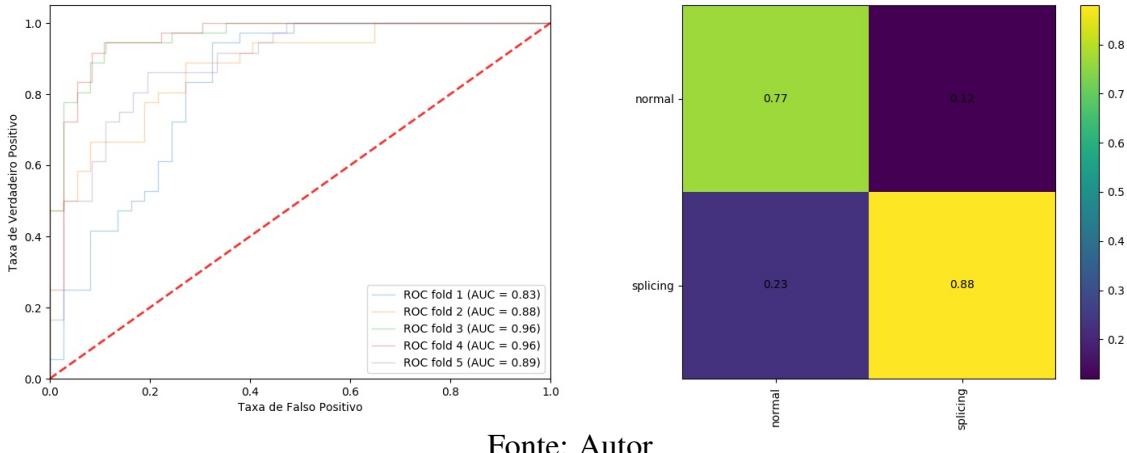


Fonte: Autor

O número de amostras falsas classificadas como verdadeiras apresentado na matriz de confusão da Figura 15 gerados durante os testes da ResNet50 com a base DSI IIC foi o menor

comparado aos resultados das redes com melhor desempenho nas outras duas bases. Contudo, a diferença entre a área do menor conjunto da validação cruzada com a maior foi de 12%, enquanto na ResNet50 com a base Columbia IIC foi de 13%.

Figura 16 – Curva ROC e Matriz de Confusão da ResNet50 com a base Columbia IIC.



Fonte: Autor

A curva ROC da Figura 16 torna clara a maior discrepância da acurácia entre as pastas da validação cruzada. Este resultado pode ser justificado pela maior quantidade de amostras contidas na base de dados Columbia em relação às outras duas. Analisando a matriz de confusão também é possível notar uma maior diferença entre a taxa de acerto das amostras falsas e verdadeiras quando comparado às matrizes das outras redes com melhor desempenho no espaço de cor IIC.

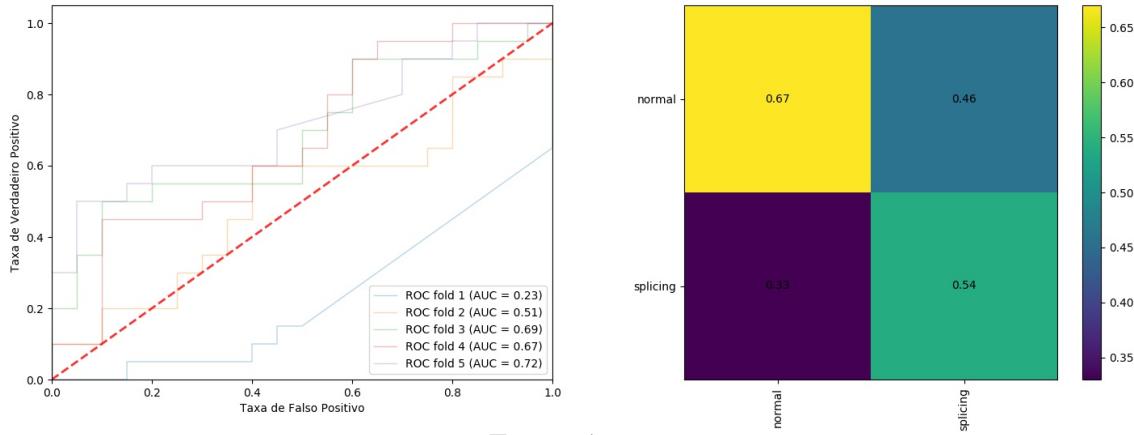
Tabela 4 – Resultados utilizando o mapa GGE.

GGE			
Rede	DSO	DSI	Columbia
ResNet50	60,5%	68%	81,5%
VGG16	59,5%	60%	77,7%
VGG19	59%	62%	81%
InceptionV3	53,5%	60%	73%
InceptionResNetV2	45,5%	44%	60%

Fonte: Autor

A Tabela 4 apresenta todos os resultados obtidos com todos os modelos utilizados em todas as bases de dados no espaço de cor GGE. Nota-se que a ResNet50 obteve o melhor resultado em todas as bases, tendo a maior variação entre melhor e pior resultado na base DSI.

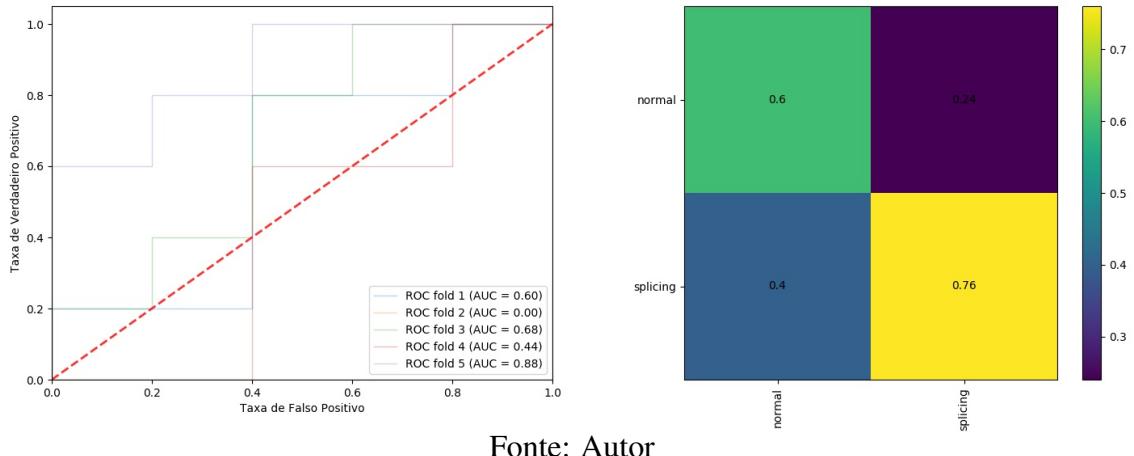
Figura 17 – Curva ROC e Matriz de Confusão da ResNet50 com a base DSO GGE.



Fonte: Autor

Como visto na Figura 17, o pior desempenho entre as melhores redes dos testes em todas as bases e em todos os espaços de cores foi o da ResNet50 na base DSO GGE, que obteve uma média de acurácia de 60,5%. Neste caso, a rede obteve um melhor desempenho nas classificações das amostras normais com 67%, enquanto errou aproximadamente a metade das imagens que continham falsificações do tipo *splicing*.

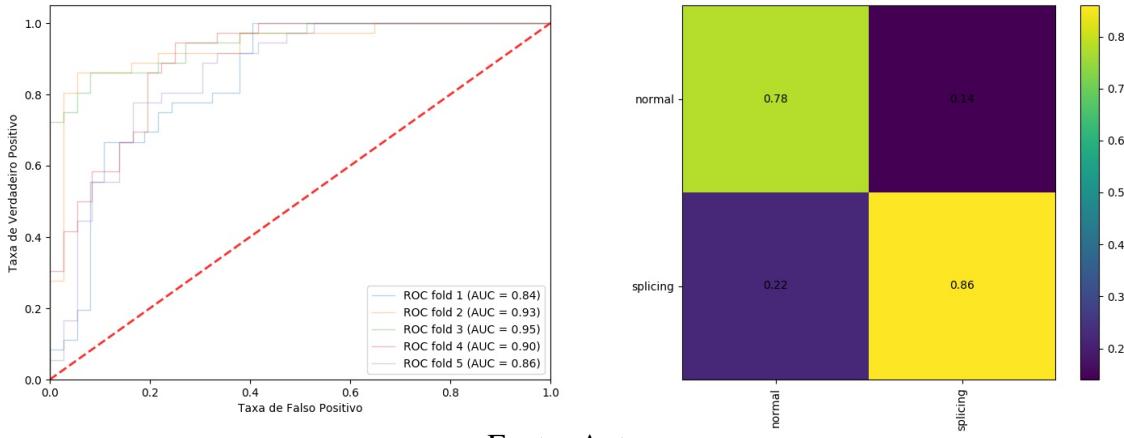
Figura 18 – Curva ROC e Matriz de Confusão da ResNet50 com a base DSI GGE.



Fonte: Autor

Levando em conta que a rede com a melhor acurácia foi a ResNet50 para todas as bases no espaço de cor GGE, na base de dados DSI o desempenho apresentado foi um pouco superior em relação a base DSO, porém relativamente distante da base Columbia. A base de dados DSI apresentou o maior erro em classificações errôneas de imagens *splicing*, com uma taxa de 40%.

Figura 19 – Curva ROC e Matriz de Confusão da ResNet50 com a base Columbia GGE.



Fonte: Autor

Com o melhor resultado, a ResNet50 na base Columbia GGE obteve uma acurácia 81,5%, sendo 16,6% superior em relação à base DSI e 25,8% em relação à base DSO. O desempenho de classificação de imagens do tipo *splicing* corretamente foi 8% maior comparado as normais.

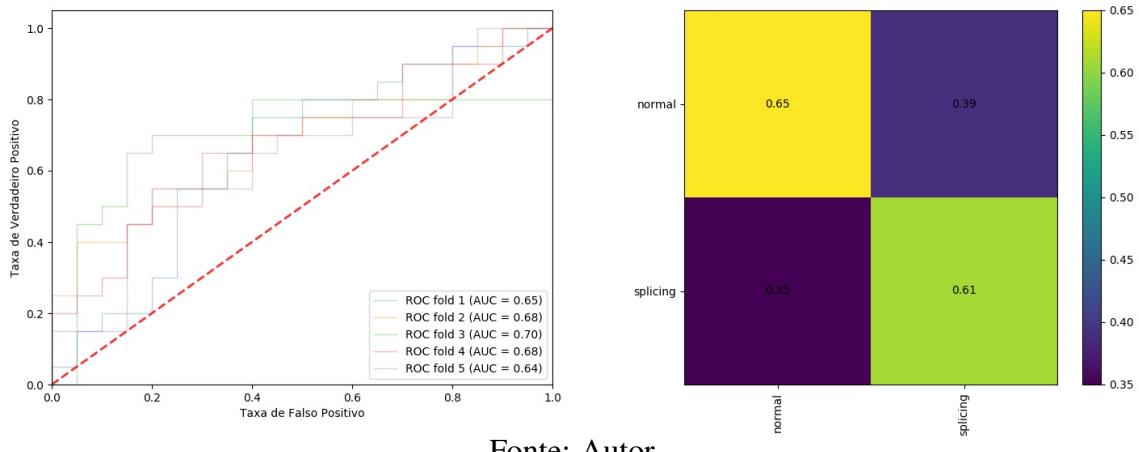
Tabela 5 – Resultados utilizando o mapa RGB.

RGB				
Rede	DSO	DSI	Columbia	
ResNet50	58,5%	72%	84,6%	
VGG16	58,5%	72%	82%	
VGG19	53,5%	74%	82%	
InceptionV3	63%	42%	63%	
InceptionResNetV2	56%	56%	70%	

Fonte: Autor

A Tabela 5 apresenta os resultados obtidos utilizando-se as imagens no espaço de cor RGB, ou seja, sem a extração dos mapas iluminantes. Como nos testes utilizando o espaço GGE, a base Columbia obteve o melhor desempenho, seguido da base DSI e por fim a DSO. A ResNet50 continuou obtendo o melhor resultado na base Columbia, como visto nos outros espaços, enquanto a rede com o melhor resultado na DSI foi a VGG19 e na base DSO sendo a InceptionV3.

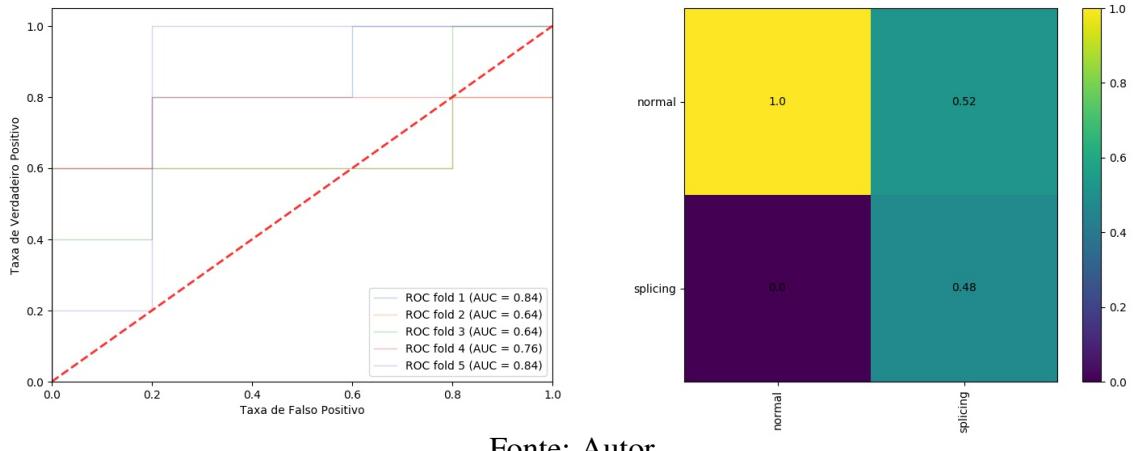
Figura 20 – Curva ROC e Matriz de Confusão da InceptionV3 com a base DSO RGB.



Fonte: Autor

Como pode ser observado na Figura 20, a rede obteve um desempenho muito parecido na classificação correta das imagens normais comparado à classificação correta das imagens falsas. Como pode ser observado na curva ROC seus resultados nas 5 pastas diferentes estão localizados na parte superior porém bem próximos a linha pontilhada vermelha, significando a rede obteve um desempenho levemente melhor do que uma amostra acertada para uma errada.

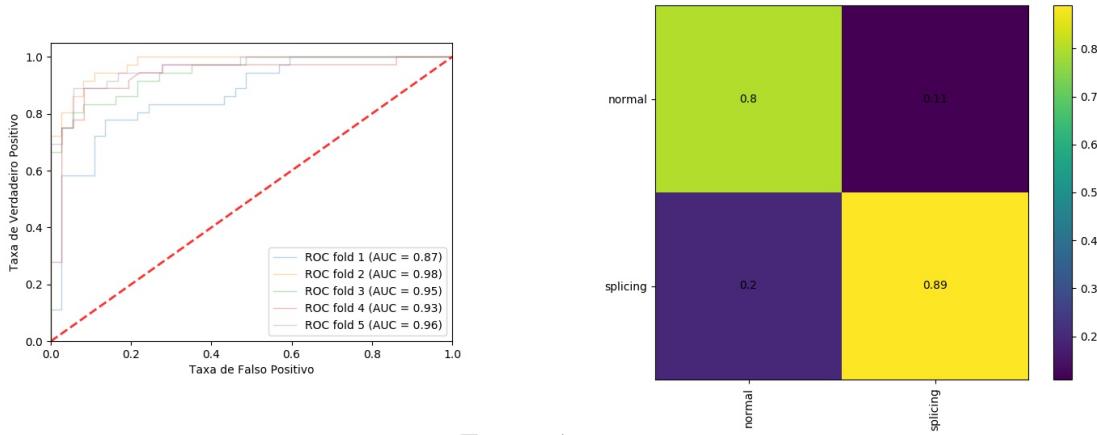
Figura 21 – Curva ROC e Matriz de Confusão da VGG19 com a base DSI RGB.



Fonte: Autor

Na Figura 21 notamos um resultado bem diverso dos outros testes, em que a VGG19 classificou corretamente todas as amostras verdadeiras da base DSI no espaço de cor RGB, porém 52% das amostras *splicing* foram classificadas de forma errada. Neste caso ocorreu a maior disparidade de acertos em relação às classes dentre os melhores resultados analisados.

Figura 22 – Curva ROC e Matriz de Confusão da ResNet50 com a base Columbia RGB.



Fonte: Autor

Por último, a Figura 22 apresenta uma acurácia consistente obtida pelo uso da ResNet50 na base de dados Columbia RGB. Com uma variação de 9% entre as pastas da validação cruzada, a rede obteve um desempenho de 84,6% de acurácia, acertando 80% das amostras normais e 89% nas amostra do tipo *splicing*.

A Tabela 6 contém uma análise geral dos resultados obtidos em que a segunda coluna representa a quantidade de bases em que cada rede obteve o melhor desempenho em relação as outras e a terceira coluna contém uma média aritmética simples de todos os resultados obtidos em todas as variações das bases.

Tabela 6 – Resultados gerais da acurácia das redes.

Modelo	Melhores Resultados	ACC Média
ResNet50	6	76,4%
VGG19	2	74,2%
InceptionV3	1	65,4%
VGG 16	0	73,9%
InceptionResNetv2	0	62,4%

Fonte: Autor

Das nove melhores acuráncias obtidas por base de dados em cada espaço de cor, seis foram alcançadas através do uso do modelo ResNet50 como extrator de características, enquanto duas foram obtidas através da VGG19 e uma pela InceptionV3. Contudo, podemos ver que, mesmo a VGG16 não tendo obtido a melhor acurácia em nenhum dos testes, sua diferença entre a ResNet50 foi de apenas 2,5% e de apenas 0,3% em relação a VGG19, sendo o terceiro modelo com melhor acurácia média.

A Tabela 7 apresenta uma média aritmética simples do desempenho de todas as redes de acordo com cada base de dados separada pelos espaços de cores.

Tabela 7 – Resultados gerais da acurácia por base de dados.

Base	IIC	GGE	RGB
DSO	91,3%	55,6%	57,9%
DSI	86%	58,8%	63,2%
Columbia	70,7%	75,6%	76,3%

Fonte: Autor

Podemos observar que a nas bases DSO e DSI o desempenho do espaço de cor IIC foi relativamente superior aos demais, obtendo porém o pior resultado entre os três na base de dados Columbia. De forma geral, o espaço de cor IIC obteve uma média de 82,6% de acurácia, enquanto o GGE obteve 63,3% e o RGB 65,8%. Ao analisar a média dos três espaços de cores por base de dados, a base DSO obteve uma média de 68,2% , a base DSI obteve uma média de 69,3% e a base Columbia obteve uma média de 75,2%.

5 CONCLUSÃO E TRABALHOS FUTUROS

Após a realização de todos os testes foram observadas determinadas situações que podem sustentar possíveis afirmações em relação ao problema abordado neste trabalho. Esta seção contém as reflexões obtidas durante o trabalho e possíveis linhas que podem ser seguidas para complementar a pesquisa feita até o momento.

5.1 CONCLUSÃO

Foram avaliados 3 tipos diferentes de espaços de cor, dois próprios para realçar características de iluminação e o espaço de cores RGB, bem como diferentes arquiteturas de CNNs para a extração de características. Ao aplicar metodologia proposta a três bases de dados públicas diferentes, a base de dados com o melhor desempenho geral foi a Columbia, seguida da DSI e por fim a DSO. Este resultado pode ser justificado principalmente pela quantidade de amostras por base, reforçando a ideia de que quanto mais amostras melhor é o treinamento.

Também foi possível identificar que, para o problema de falsificação de composição do tipo splicing, o uso do espaço de cores IIC resulta em uma melhora nos resultados em que a semântica das imagens segue o mundo real, como nas bases DSO e DSI. Apesar da base de dados Columbia ter tido uma acurácia média superior às outras duas, o espaço de cor que obteve o melhor resultado foi o RGB, seguido do GGE e depois o IIC, podendo concluir que neste tipo de dado, em que a falsificação não necessariamente faz sentido ao olhar humano, o realce de características de iluminância não possui uma grande eficácia, mesmo a acurácia média entre os três espaços tendo apresentado uma amplitude de apenas 5,6%.

É válido ressaltar o desempenho muito superior do modelo ResNet50 aplicado para este tipo de problema, que obteve três vezes mais melhores resultados em relação à segunda melhor rede e seis vezes mais em relação à terceira. Mesmo a diferença entre a acurácia média entre ResNet50 e a segunda melhor colocada (VGG19) tendo sido de apenas 2,2%, que pode ser facilmente resolvido com a aplicação de técnicas de melhoramento, a ResNet50 possui quase seis vezes menos hiperparâmetros para serem ajustados e processados, tornando seu custo computacional bem mais baixo.

Analizando a relação entre a acurácia e o poder computacional exigido pelo modelo a ResNet50 demonstrou ter o melhor custo-benefício, pois possui apenas 7,1% de parâmetros a mais em relação a rede com menor número e obteve o melhor resultado geral.

Desta forma, podemos concluir que a escolha de uma rede para um determinado problema pode não estar atrelada somente a acurácia da sua base, mas também ao perfil dos dados utilizados e ao poder computacional disponível para uso.

5.2 TRABALHOS FUTUROS

Como um próximo passo para deixar este trabalho mais completo, pretende-se testar variações da máquina de vetor de suporte usada para classificar as imagens neste trabalho. Também pretende-se aplicar e analisar a quantidade de ganho possível ao aplicarem-se técnicas de melhoramento na rede como o retreino das camadas mais profundas das redes responsáveis por procurar por características mais específicas ao problema abordado.

O uso de combinações de redes e características, como usar duas redes convolucionais distintas para extrair características de profundidade e de iluminação por exemplo, também é um objetivo futuro.

REFERÊNCIAS

- BARLOW, Horace B. Unsupervised learning. **Neural computation**, MIT Press, v. 1, n. 3, p. 295–311, 1989.
- BAYER, Bryce E. **Color imaging array**. [S.l.]: Google Patents, jul. 20 1976. US Patent 3,971,065.
- BENGIO, Yoshua. Deep learning of representations for unsupervised and transfer learning. In: **Proceedings of ICML Workshop on Unsupervised and Transfer Learning**. [S.l.: s.n.], 2012. p. 17–36.
- BENGIO, Yoshua; COURVILLE, Aaron; VINCENT, Pascal. Representation learning: A review and new perspectives. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 35, n. 8, p. 1798–1828, 2013.
- BOYLE, Willard S; SMITH, George E. Charge coupled semiconductor devices. **Bell System Technical Journal**, Wiley Online Library, v. 49, n. 4, p. 587–593, 1970.
- BRAGA, A de P; CARVALHO, APLF; LUDELMIR, Teresa Bernarda. **Redes neurais artificiais: teoria e aplicações**. [S.l.]: Livros Técnicos e Científicos Rio de Janeiro, 2000.
- CARUANA, Rich; NICULESCU-MIZIL, Alexandru. An empirical comparison of supervised learning algorithms. In: ACM. **Proceedings of the 23rd international conference on Machine learning**. [S.l.], 2006. p. 161–168.
- CARVALHO, T. *et. al.* Illuminant-based transformed spaces for image forensics. **IEEE Transactions on Information Forensics and Security**, v. 11, n. 4, p. 720–733, April 2016. ISSN 1556-6013.
- CARVALHO, T. J. d. *et. al.* Exposing digital image forgeries by illumination color classification. **IEEE Transactions on Information Forensics and Security**, v. 8, n. 7, p. 1182–1194, July 2013. ISSN 1556-6013.
- CRISTIANO, Mariana Vitória de Menezes Bordalo. **Sensibilidade e especificidade na curva roc: um caso de estudo**. Tese (Doutorado), 2017.
- DENG, Li; YU, Dong *et. al.* Deep learning: methods and applications. **Foundations and Trends® in Signal Processing**, Now Publishers, Inc., v. 7, n. 3–4, p. 197–387, 2014.
- DEVAGIRI, Vishnu Manasa; CHEDDAD, Abbas. Splicing forgery detection and the impact of image resolution. In: **IEEE. Electronics, Computers and Artificial Intelligence (ECAI), 2017 9th International Conference on**. [S.l.], 2017. p. 1–6.
- FARID, Hany. Detecting digital forgeries using bispectral analysis. 1999.
- FARID, Hany. Digital image forensics. **Scientific American**, JSTOR, v. 298, n. 6, p. 66–71, 2008.
- FAWCETT, Tom. An introduction to roc analysis. **Pattern recognition letters**, Elsevier, v. 27, n. 8, p. 861–874, 2006.
- GOLDSTEIN, Grigory. **Lenin's Speech**. 2018. Disponível em: <https://commons.wikimedia.org/wiki/File:Lenin%27s_speech.jpg>. Acesso em: 13 Out. 2018.

- HAYKIN, Simon. **Redes neurais: princípios e prática.** [S.l.]: Bookman Editora, 2007. 61 p.
- HAYKIN, Simon S. **Neural networks and learning machines.** [S.l.]: Pearson Upper Saddle River, 2009.
- HE, Kaiming *et. al.* Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition.** [S.l.: s.n.], 2016. p. 770–778.
- HSU, Y.-F.; CHANG, S.-F. Detecting image splicing using geometry invariants and camera characteristics consistency. In: **International Conference on Multimedia and Expo.** [S.l.: s.n.], 2006.
- Hussain, M. *et. al.* A comparison of svm kernel functions for breast cancer detection. In: **2011 Eighth International Conference Computer Graphics, Imaging and Visualization.** [S.l.: s.n.], 2011. p. 145–150.
- JAKKULA, Vikramaditya. Tutorial on support vector machine (svm). **School of EECS, Washington State University**, v. 37, 2006.
- JI, Shuiwang *et. al.* 3d convolutional neural networks for human action recognition. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 35, n. 1, p. 221–231, 2013.
- KARPATHY, Andrej; FEI-FEI, Li. Deep visual-semantic alignments for generating image descriptions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition.** [S.l.: s.n.], 2015. p. 3128–3137.
- KOHAVI, Ron *et. al.* A study of cross-validation and bootstrap for accuracy estimation and model selection. In: MONTREAL, CANADA. **Ijcai**. [S.l.], 1995. v. 14, n. 2, p. 1137–1145.
- LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015.
- LECUN, Yann *et. al.* Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, IEEE, v. 86, n. 11, p. 2278–2324, 1998.
- LI, Dawei *et. al.* Integrating a statistical background-foreground extraction algorithm and svm classifier for pedestrian detection and tracking. **Integrated Computer-Aided Engineering**, v. 20, p. 201–216, 07 2013.
- LOPES, Lincoln da Cunha. Controle metrológico da cor aplicado à estamparia digital de materiais têxteis. 2009. 142 f. **Dissertação (Mestrado em Metrologia)–Pontifício Universicole Cofólico, Rio Ge Joneiro**, 2009.
- MATSUGU, Masakazu *et. al.* Subject independent facial expression recognition with robust face detection using a convolutional neural network. **Neural Networks**, Elsevier, v. 16, n. 5-6, p. 555–559, 2003.
- MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre aprendizado de máquina. **Sistemas Inteligentes-Fundamentos e Aplicações**, v. 1, n. 1, p. 32, 2003.
- MURALI, S *et. al.* Comparision and analysis of photo image forgery detection techniques. **arXiv preprint arXiv:1302.3119**, 2013.

- NAIR, Vinod; HINTON, Geoffrey E. Rectified linear units improve restricted boltzmann machines. In: **Proceedings of the 27th International Conference on International Conference on Machine Learning**. USA: Omnipress, 2010. (ICML'10), p. 807–814. ISBN 978-1-60558-907-7. Disponível em: <<http://dl.acm.org/citation.cfm?id=3104322.3104425>>.
- NG, T-T; CHANG, S-F. A model for image splicing. In: **IEEE. 2004 International Conference on Image Processing, 2004. ICIP'04**. [S.l.], 2004. v. 2, p. 1169–1172.
- PAN, Sinno Jialin; YANG, Qiang *et. al.* A survey on transfer learning. **IEEE Transactions on knowledge and data engineering**, Institute of Electrical and Electronics Engineers, Inc., 345 E. 47 th St. NY NY 10017-2394 USA, v. 22, n. 10, p. 1345–1359, 2010.
- RIESS, Christian; ANGELOPOULOU, Elli. Scene illumination as an indicator of image manipulation. In: SPRINGER. **International Workshop on Information Hiding**. [S.l.], 2010. p. 66–80.
- RUMELHART, David E; HINTON, Geoffrey E; WILLIAMS, Ronald J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group, v. 323, n. 6088, p. 533, 1986.
- RUSSAKOVSKY, Olga *et. al.* ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, v. 115, n. 3, p. 211–252, 2015.
- SCHMIDHUBER, Jürgen. Deep learning in neural networks: An overview. **Neural networks**, Elsevier, v. 61, p. 85–117, 2015.
- SCHÖLKOPF, Bernhard *et. al.* **Learning with kernels: support vector machines, regularization, optimization, and beyond**. [S.l.]: MIT press, 2002.
- SILVA, Ewerton Almeida; ROCHA, Anderson. Análise forense de documentos digitais: além da visão humana. **Saúde, Ética & Justiça**, v. 16, n. 1, p. 9–17, 2011.
- SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.
- SRIVASTAVA, Nitish *et. al.* Dropout: A simple way to prevent neural networks from overfitting. v. 15, p. 1929–1958, 06 2014.
- SUYKENS, Johan AK; VANDEWALLE, Joos. Least squares support vector machine classifiers. **Neural processing letters**, Springer, v. 9, n. 3, p. 293–300, 1999.
- SZEGEDY, Christian *et. al.* Going deeper with convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 1–9.
- WEIJER, Joost Van De; GEVERS, Theo; GIJSENIJ, Arjan. Edge-based color constancy. **IEEE Transactions on image processing**, IEEE, v. 16, n. 9, p. 2207–2214, 2007.
- ZHANG, Ying *et. al.* Image region forgery detection: A deep learning approach. In: **SG-CRC**. [S.l.: s.n.], 2016. p. 1–11.
- ZHANG, Yu-jin; LI, Sheng-hong; WANG, Shi-lin. Detecting shifted double jpeg compression tampering utilizing both intra-block and inter-block correlations. **Journal of Shanghai Jiaotong University (Science)**, v. 18, n. 1, p. 7–16, Feb 2013. ISSN 1995-8188. Disponível em: <<https://doi.org/10.1007/s12204-013-1362-9>>.